

CS 470 Spring 2011

Project 3

Colby Blair

Due April 27th, 2011

Abstract

Fuzzy Controls are a very useful in Artificial Intelligence. They don't resemble intelligence, in that they are fairly static. They don't learn, and only react to inputs in the way originally intended by the programmer. They are, however, very useful. They are best used in environments with set inputs, and the corresponding outputs are imperical values, besides some scaling.

Fuzzy control systems are also much simplier, and respond much better to specific problems. Therefore, their structure can tell us much more about the solutions than more complicated AI solutions, like Artificial Neural Networks. Fuzzy control system also give us great training or teaching algorithms, in which to teach other methods of AI.

Contents

1	Introduction	1
1.1	Fuzzify Inputs	1
1.2	Apply Fuzzy Logic	1
1.3	Defuzzify Fuzzy Logic	2
2	Moon Lander	3
2.1	Fuzzify Inputs	3
2.1.1	Y	3
2.1.2	X	4
2.2	Apply Fuzzy Logic	5
2.2.1	Y	5
2.2.2	X	6
2.3	Defuzzified Fuzzy Logic	6
2.3.1	Y	6
2.3.2	X	7
3	Conclusion	8
4	Results	8
4.1	Y fuzzy values and defuzzified output with scaling (getBurn) .	8
4.2	X fuzzy values and defuzzified output with scaling (getThrust)	10
4.3	Review/Summary	11

List of Figures

1	An example y axis fuzzification	2
2	The moon lander	3
3	Y input fuzzifications	4
4	X input fuzzifications	5
5	Smooth landing with small overcorrection	9
6	Overcorrection with low gravity	9
7	Smooth landing	10
8	Medium wind with overcorrection	11
9	Low wind	12
10	Crash!	12

1 Introduction

Fuzzy control systems are a more imperical way of implementing AI. This project will demonstrate a simple fuzzy control system, while trying to land a moon lander. Fuzzy control systems try to create actions by quantifying inputs. And example for the lander goes: "if high and descending slow, then burn low", where burn is the engines output. A fuzzy system could be made of as many of these rules as one could think of, but a better approach is to create functions that produce the same outputs.

The lander works in environments of varying gravity and wind. To complicate things, the lander has a narrow landing pad to land on. Controls must be fairly tight, but also dynamic to handle many changing environment inputs. The challenge then is that we must quantify the inputs, like 'high'. What is high? Is it either true or false? Probably not. 'High' is probably then a value from 0 to 100, or in our case, 0.0000 to 1.0000. The lander can adjust its burn accordingly, so that more burn is outputed closer to landing.

To get useful outputs from a fuzzy control system, there are 3 steps:

1. fuzzify inputs
2. select fuzzy inputs by applying fuzzy logic
3. defuzzify fuzzy logic output

1.1 Fuzzify Inputs

There has already been a few rules defined now, like "if high and descending slow, then burn low" and "if low and descending medium, then burn medium". This is generalized with some basic fuzzification:

The lander still needs to consider x axis controls, but Figure 1 gives an example of y control that is somewhat similar to the actual findings in this report.

1.2 Apply Fuzzy Logic

In general, fuzzy logic has two operators considered here: *OR* and *AND*:

$$aANDb = \min(a, b)$$

$$aORb = \max(a, b)$$

where

a, b are fuzzified values

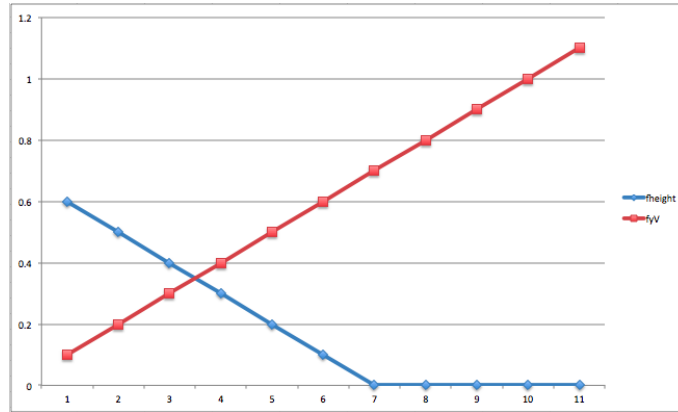


Figure 1: An example y axis fuzzification

1.3 Defuzzify Fuzzy Logic

Defuzzification means to turn our fuzzy logic back into real world values, or useful output. There are several methods to defuzzify our fuzzy logic, including scaling and centroid operations. In this report, the simple method of scaling does fine.

2 Moon Lander

The lander this report uses must be a lander for more than just earth's moon, because random wind gusts exist. Since the x and y forces are completely perpendicular, from physics it is known that the forces are independent. We then have two independent, x and y, controls.



Figure 2: The moon lander

2.1 Fuzzify Inputs

The fuzzifications are actually fairly simple. That said, they did the job well enough to land the lander over 90

2.1.1 Y

The y fuzzy velocity algorithm is as follows:

$$\begin{aligned} f_{v_y}(V_y) &= 0 \text{ if } V_y < V_{y_{max}} \\ &= 1 \text{ if } V_y > V_{y_{max}} + k_1 \\ &= \frac{1-(V_{y_{max}}-V_y)}{V_{y_{max}}} \text{ otherwise} \end{aligned}$$

where

$$k_1 = 3$$

The y fuzzy height algorithm is as follows:

$$f_y(y) = 0 \text{ if } y > y_{cutoff}$$

$$= \frac{(Y_{max}-y)}{Y_{max}} - k_2 \text{ otherwise}$$

where

$$y_{cutoff} = 50$$

$$k_2 = .4$$

The resulting fuzzifications outputs are:

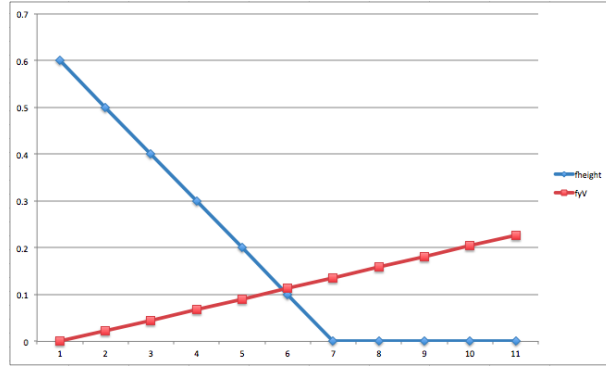


Figure 3: Y input fuzzifications

2.1.2 X

X values were very simple, since they stayed relatively low (0.0 - 2.0). The y fuzzy velocity algorithm is as follows:

$$f_{V_x}(V_x) = \frac{1-(V_{xmax}-V_x)}{V_{xmax}}$$

where

$$V_{xmax} = 5.0$$

The x fuzzy position algorithm is as follows:

$$f_x(x) = \frac{1-(X_{max}-x)}{X_{max}}$$

where

$$V_{xmax} = 2.0$$

The resulting fuzzifications outputs are:

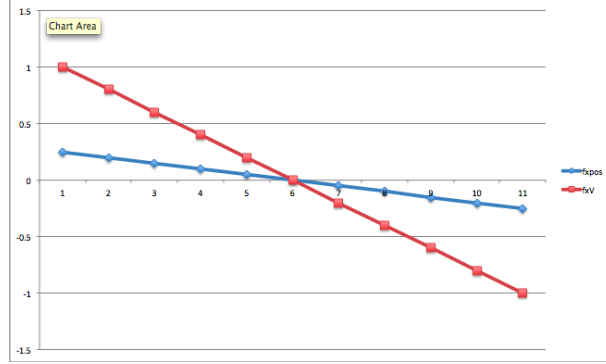


Figure 4: X input fuzzifications

2.2 Apply Fuzzy Logic

For selecting fuzzy y values, the *OR* fuzzy operator was used. For example, this effectively picked $\max(f_{v_y}(V_y), f_y(y))$ from Figure 3. Picking the max values implies that over estimation of action should result. This is common with scenarios like safety, where under estimation can be catastrophic. In the case of the moonlander, this is true.

2.2.1 Y

For selecting fuzzy y values, the *OR* fuzzy operator was used. The resulting values from the figure are:

$f_{v_y}(V_y)$	$f_y(y)$
0.6	0
0.5	0.022727273
0.4	0.045454545
0.3	0.068181818
0.2	0.090909091
0.1	0.113636364
0	0.136363636
0	0.159090909
0	0.181818182
0	0.204545455
0	0.227272727

2.2.2 X

For selecting fuzzy y values, the *OR* fuzzy operator was used on the absolute values, since positive or negative values represented left or right movement. The resulting values from the figure are:

$f_{v_x}(V_x)$	$f_x(x)$
0.25	1
0.2	0.8
0.15	0.6
0.1	0.4
0.05	0.2
0	0
-0.05	-0.2
-0.1	-0.4
-0.15	-0.6
-0.2	-0.8
-0.25	-1

2.3 Defuzzified Fuzzy Logic

2.3.1 Y

For y values, the defuzzified values needed to be relative to gravity. In low gravity, output should be lower than in heavy gravity. Gravity cannot be directly measures, but higher gravity will create higher y velocities, despite our output. This is a loose way of calculating gravity, so y velocity is used to defuzzify the fuzzy logic.

The defuzzification algorithm for y velocity is as follows:

$$df_{V_y}(V_y) = f_{v_y}(V_y) * V_y$$

The defuzzification algorithm for height is as follows:

$$df_y(y) = f_y(y) * V_y * k_3$$

where

$$k_3 = .8$$

2.3.2 X

For x values, the defuzzified values can be scaled similarly to y values by using x velocity.

The defuzzification algorithm for x velocity is as follows:

$$df_{V_x}(V_x) = f_{v_x}(V_x) * V_x * k_4$$

where

$$k_4 = .9$$

The defuzzification algorithm for x position is as follows:

$$df_x(x) = f_x(x) * V_x * k_5$$

where

$$k_5 = 1.8$$

3 Conclusion

In conclusion, a fuzzy control system was extremely useful. It was able to perform very well in what is probably very simplified test cases. But more importantly, its fuzzification and defuzzified output teaches the observer much about the solution of the problem, compared to other AI methods. In this report, none of the original outputs were changed, even though they may have been extreme.

Perhaps the best function the fuzzy controls provide is a teaching/training algorithm for other AI methods, like Artificial Neural Networks. Fuzzy controls worked well here, but may fall apart in more extreme environments. Teaching an ANN from a fuzzy control could create a base for the ANN to start from, and the ANN could then generalize the solution to work in more complicated environments.

4 Results

The selection of mins and maxs can be best seen in some output from actual runs. The results are pretty interesting (all are successful landing unless notified as crash).

4.1 Review/Summary

The system worked very well. It landed over 90% were because of thrust (x output) errors. This could be because of inaccuracies in the x controls, or could be because of the extreme changes of wind during flight.

Notice that over corrections were the only irregularity in y control. This comes from selecting the max fuzzy value, or using the *OR* operator. This defaults on caution over landing too hard. The y control was still able to do land safely, even during extremely high or low gravity, without running out of fuel. Since gravity was measured off of y velocity, over corrections tended to feedback into the system. But this is a limitation from what the lander can observe.

4.2 Y fuzzy values and defuzzified output with scaling (getBurn)

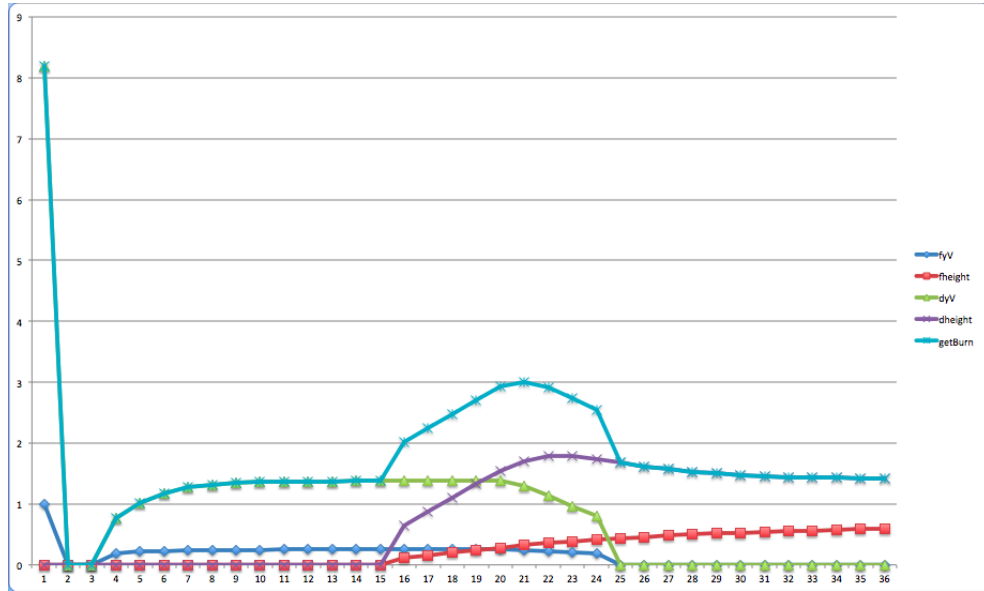


Figure 5: Smooth landing with small overcorrection

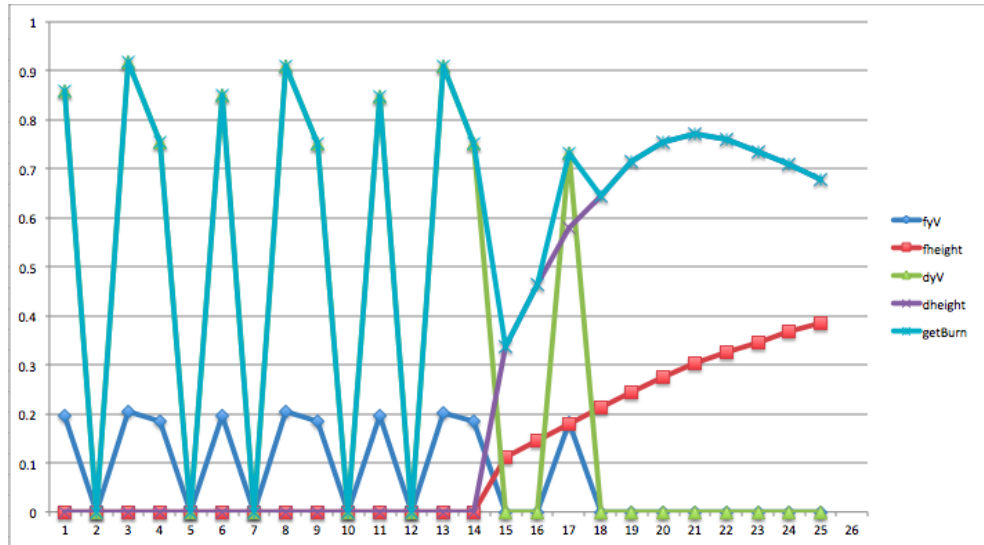


Figure 6: Overcorrection with low gravity

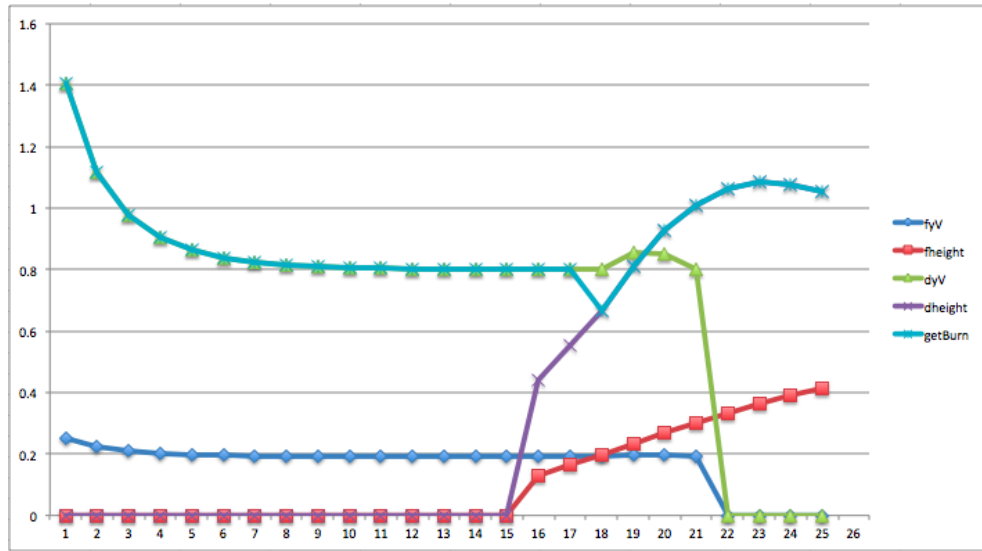


Figure 7: Smooth landing

4.3 X fuzzy values and defuzzified output with scaling (getThrust)

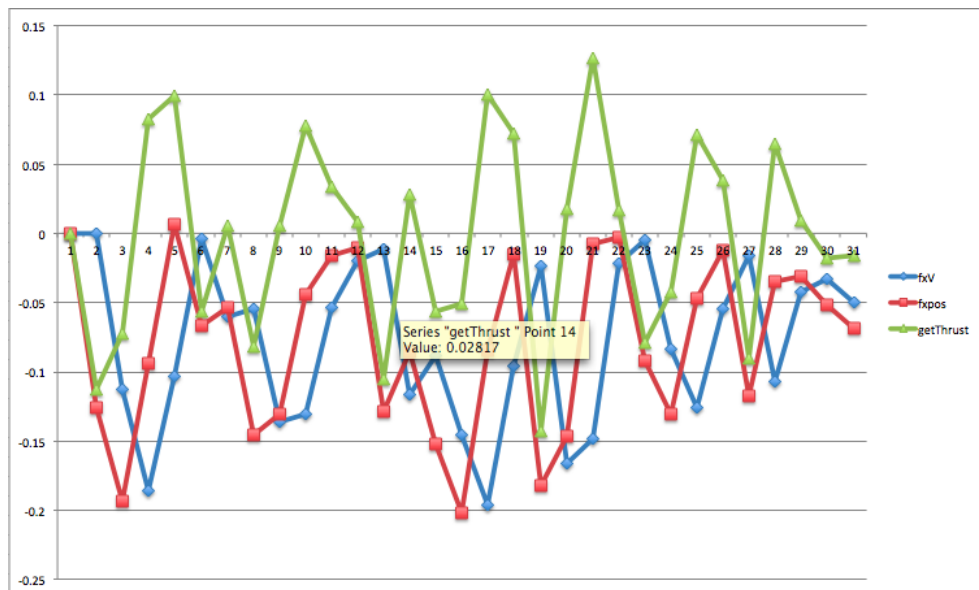


Figure 8: Medium wind with overcorrection

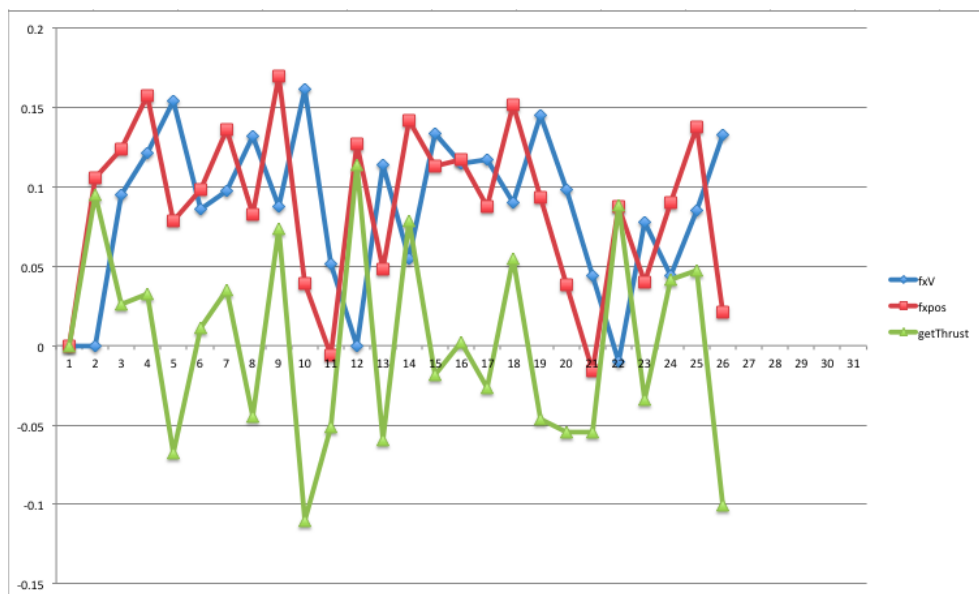


Figure 9: Low wind

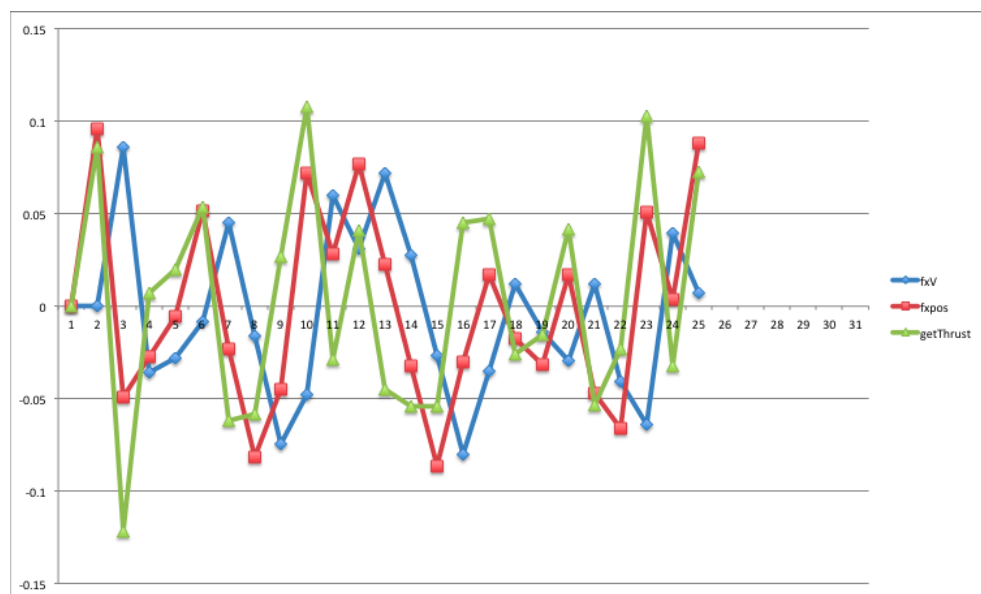


Figure 10: Crash!