ECE478

Homework 3: Application of Fuzzy Logic for Robot Navigation

by

Tristan Shores

Introduction

Boolean logic applied to robotic navigation permits only two potential outcomes for each robot decision. In contrast, fuzzy logic allows unlimited outcomes for each robot decision.

For example, a fuzzy logic rule might state: "**turn-right** if **front-proximity** is **close** and **left-proximity** is **close** and **right-proximity** is **far**", where:

- **front-proximity**, **left-proximity**, and **right-proximity** are sensor readings.
- **close** and **far** are input membership functions that flexibly translate a specific proximity (crisp value) in a defined sensor reading range to a decimal value in the range of 0 to 1 (degree of membership).
- **turn-right** is an output membership function that flexibly translates a decimal value in the range of 0 to 1 (degree of membership) into a clockwise rotation in the range of 0 - 90° (crisp value).
- AND is a fuzzy logic operator that typically asserts the minimum of the operands.

If the evaluation of all conditions is satisfied to any extent, then the robot will execute a clockwise turn in the range of 0 to 90°, with the exact rotation dependent on the degree to which conditions are satisfied. In this way, a single fuzzy logic rule can generate smooth robot navigation.

An example of a fuzzy logic system used to solve a robot navigational decision (degree of robot rotation) is given in Fig. 1.

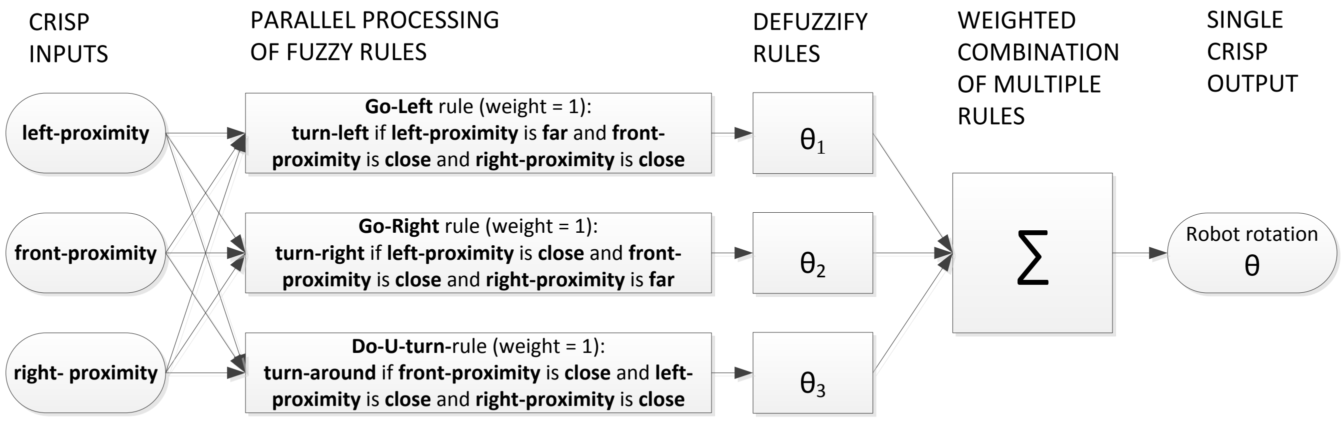
CRISP INPUTS
PARALLEL PROCESSING OF FUZZY RULES
DEFUZZIFY RULES
WEIGHTED COMBINATION OF MULTIPLE RULES
SINGLE CRISP OUTPUT
left-proximity
front-proximity
right- proximity
Go-Left rule (weight = 1): turn-left if left-proximity is far and front-proximity is close and right-proximity is close
Go-Right rule (weight = 1): turn-right if left-proximity is close and front-proximity is close and right-proximity is far
Do-U-turn-rule (weight = 1): turn-around if front-proximity is close and left-proximity is close and right-proximity is close
θ_1
θ_2
θ_3
Σ
Robot rotation θ

Figure 1. Triple input, triple rule, single output fuzzy system.

Crisp Inputs

A crisp value is a real non-scaled value, such as a robot's unobstructed front distance in feet. The system shown in Fig. 1 has three crisp inputs that provide the unobstructed distance on a robot's left, front, and right sides.

Parallel Processing of Fuzzy Rules

A fuzzy rule consists of one or more conditions that when combined comprise the rule's predicate which determines the rule's consequence. For example, the predicate of the rule **Go-Right** is "if **front-proximity** is **close** and **left-proximity** is **close** and **right-proximity** is **far**", and the consequence of the rule is **turn-right**. Each condition independently evaluates to a degree of membership μ in a set specified by that condition. For example, "**left-proximity** is **far**" evaluates to **left-proximity**'s degree of membership μ_{far} in the set **far**.

μ is in the range of 0 to 1, where a μ of 0 indicates that a condition was not satisfied to any extent, a μ of 1 indicates full satisfaction of a condition, and a fractional μ indicates the partial degree to which a condition was satisfied.

In the example shown in Fig. 1, each rule has three conditions that individually need to be evaluated. For example, the **Go-Left** rule requires evaluation of "**left-proximity** is **far**", "**front-proximity** is **close**", and "**right-proximity** is **far**". The terms **close** and **far** represent fuzzy sets to which **left-proximity**, **front-proximity**, and **right-proximity** may belong to some degree. The exact extent to which **left-proximity**, **front-proximity**, and **right-proximity** belong to the **close** and **far** sets is evaluated using the input membership function for each set.

The set **close** interprets an input parameter (representing proximity) according to the input membership function shown in Figure 2. For example, for a proximity of 8ft, the function outputs a μ_{close} value of 0.2. This indicates that this proximity value is a marginal member of the set.

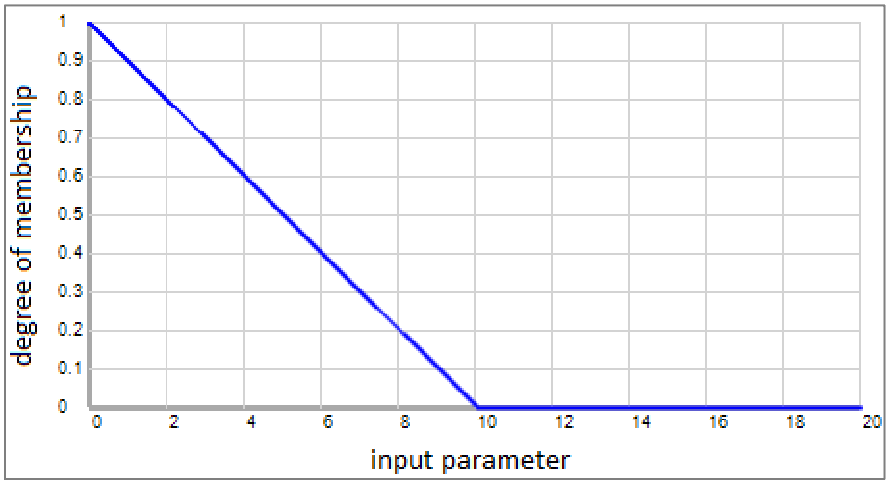

Figure 2. Input membership function for the **close** set.

The set **far** interprets an input parameter (representing proximity) according to the input membership function shown in Figure 3. For example, for a proximity of 8ft, the function outputs a μ_{far} value of 0.8. This indicates that this proximity value is a strong member of the set.

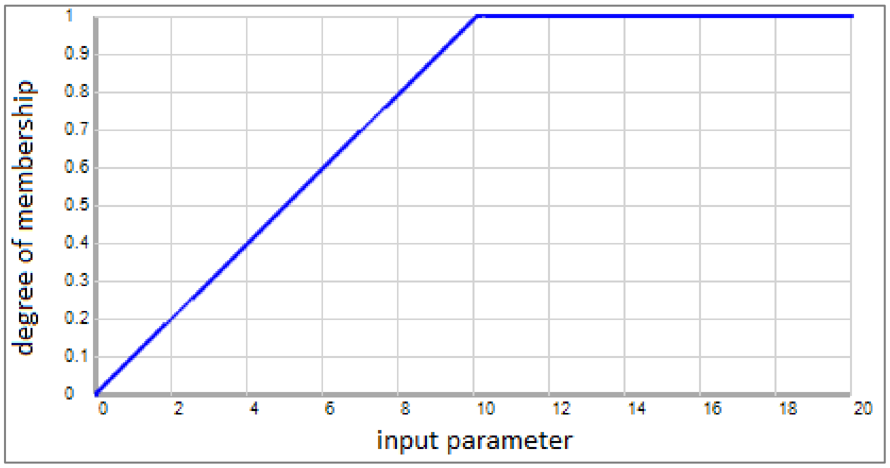

Figure 3. Input membership function for the **far** set.

It is significant that proximity values in the non-inclusive range 0 to 10ft have partial membership in both the **close** and **far** fuzzy sets, as will be discussed later.

Fuzzy logic operators determine the way in which multiple conditions will combine to yield an overall degree of membership for the rule's predicate. This degree of membership applies to the set represented by the rule's consequence. For example, the predicate of the **Go-Right** rule evaluates to a degree of membership $\mu_{predicate}$ in the **turn-right** set.

Common fuzzy logic operators are AND, OR and, NOT, where:

- AND typically asserts the minimum of the operands.
- OR typically asserts the maximum of the operands.
- NOT inverts the associated operand (i.e. evaluates to 1 minus the operand value).

The use of fuzzy logic operators to determine $\mu_{predicate}$ for each of the **Go-Left**, **Go-Right**, and **Do-U-Turn** rules is shown in Figs. 4A, 4B, and 4C respectively. The three rules are processed in parallel to yield 3 separate $\mu_{predicate}$ values.

Crisp input names	Crisp input values	membership in **close** set μ_{close}	membership in **far** set μ_{far}	left-proximity is far AND front-proximity is close AND right-proximity is close	$\mu_{predicate}$
left-proximity	8	0.2	0.8	min (0.8, 0.8, 0.5)	0.5
front-proximity	2	0.8	0.2		
right-proximity	5	0.5	0.5		

Figure 4A. Application of the fuzzy logic AND operator to the **Go-Left** rule.

Crisp input names	Crisp input values	membership in **close** set μ_{close}	membership in **far** set μ_{far}	left-proximity is close AND front-proximity is close AND right-proximity is far	$\mu_{predicate}$
left-proximity	8	0.2	0.8	min (0.2, 0.8, 0.5)	0.2
front-proximity	2	0.8	0.2		
right-proximity	5	0.5	0.5		

Figure 4B. Application of the fuzzy logic AND operator to the **Go-Right** rule.

Crisp input names	Crisp input values	membership in **close** set μ_{close}	membership in **far** set μ_{far}	left-proximity is close AND front-proximity is close AND right-proximity is close	$\mu_{predicate}$
left-proximity	8	0.2	0.8	min (0.2, 0.8, 0.5)	0.2
front-proximity	2	0.8	0.2		
right-proximity	5	0.5	0.5		

Figure 4C. Application of the fuzzy logic AND operator to the **Do-U-Turn** rule.

Defuzzify Rules

In the example shown of Fig. 1, the **Go-Left** rule is defuzzified to yield a crisp value that is determined by the degree of membership of its $\mu_{predicate}$ in the **turn-left** set. Similarly, the **Go-Right** rule is defuzzified to yield a crisp value that is determined by the degree of membership of its $\mu_{predicate}$ in the **turn-right** set. Finally, the **Do-U-Turn** rule is defuzzified to yield a crisp value that is determined by the degree of membership of its $\mu_{predicate}$ in the **turn-around** set. The crisp value for a given $\mu_{predicate}$ is determined using the output membership function for the set to which the degree of membership applies.

The output membership function shown in Figure 5, specifies a crisp value for each degree of membership $\mu_{predicate}$ in the **turn-left** set. For example, given a $\mu_{predicate}$ of 0.5, the function outputs a crisp value of -45°. This crisp value is a middle member of the set.

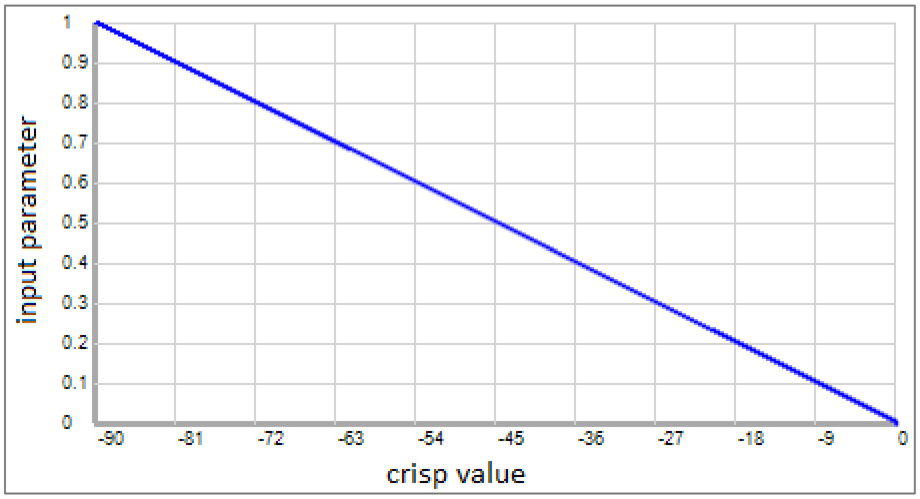

Figure 5. Output membership function for the **turn-left** set.

The output membership function shown in Figure 6, specifies a crisp value for each degree of membership $\mu_{predicate}$ in the **turn-right** set. For example, given a $\mu_{predicate}$ of 0.2, the function outputs a crisp value of +18°. This $\mu_{predicate}$ value is a marginal member of the set.

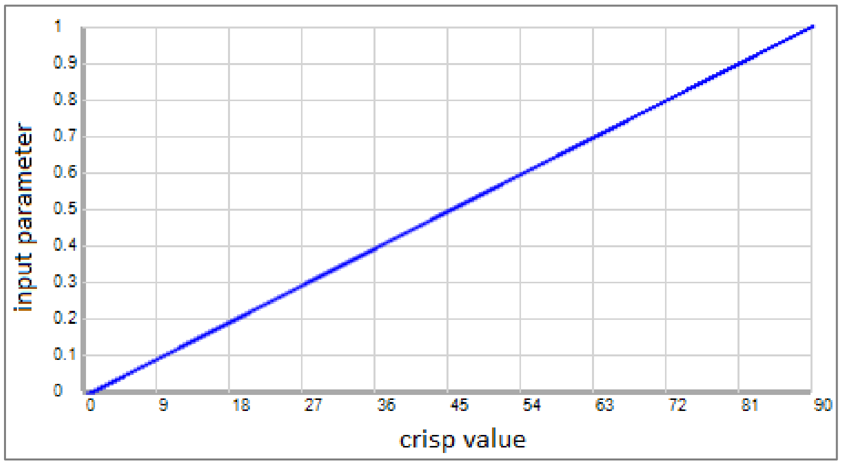

Figure 6. Output membership function for the **turn-right** set.

The output membership function shown in Figure 7, specifies a crisp value for each degree of membership $\mu_{predicate}$ in the **turn-around** set. For example, given a $\mu_{predicate}$ of 0.2, the function outputs a crisp value of 0°. This $\mu_{predicate}$ value is a non-member of the set.

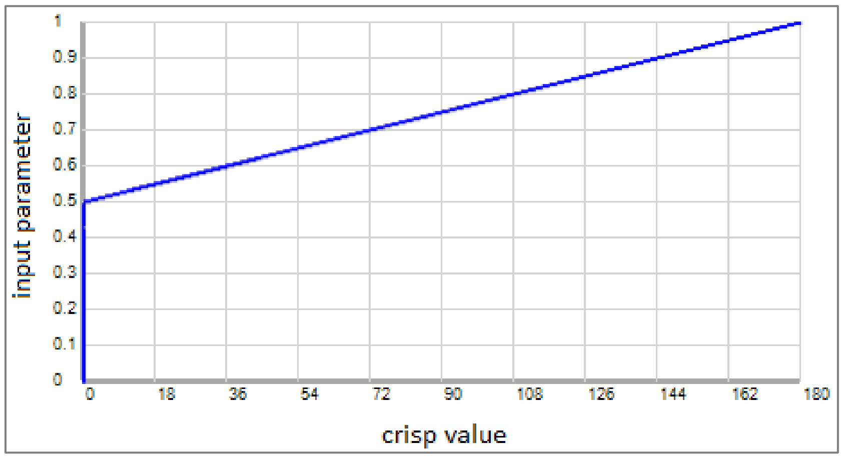

Figure 7. Output membership function for the **turn-around** set.

In the case of a single rule fuzzy system the resulting crisp value found from the output membership function would be the single crisp output of the fuzzy system. However, in the example shown of Fig. 1, the three rules are processed in parallel and need to be combined in order to generate the single crisp output of the fuzzy system.

Weighted Combination of Multiple Fuzzy Rules

In the example shown of Fig. 1, each fuzzy rule weighting is unity and therefore a multiplication factor is not applied to the crisp value of any rule.

The combination of multiple crisp values can occur in various ways. In the case of the example shown in Fig. 1, a straight forward summation is used, as shown in Figure 8.

Crisp input names	Crisp input values	Rule name	μ of the predicate	crisp value	∑	robot rotation
leftProximity	8	Go Left Rule	0.5	-45°	-27°	turn 27° to the left
frontProximity	2	Go Right Rule	0.2	+18°		
rightProximity	5	Do U-Turn Rule	0.2	0°		

Figure 8. Combination of three rules to generate a single crisp output.

Another method for combination of fuzzy rules truncates each output membership function plot at the $\mu_{predicate}$ for that plot, aggregates the truncated plot areas, and then calculates the centroid of the aggregated area. The crisp value of the centroid (x-coordinate) is then used as the single crisp output.

Discussion

The parallel processing of rules potentially allows multiple rules to influence the output crisp value, depending on each rule's crisp evaluation and weighting. This helps avoid sharp switching between opposing rules which could result in non-smooth crisp output transitions.

However, to harness the benefits of parallel rule processing, the fuzzy system designer must ensure overlapping transition domains within the membership functions of diametrically opposed sets. For example, proximity values in the non-inclusive range 0 to 10ft have partial membership in both the **close** and **far** sets. This is shown in Fig. 9, which is a plot of superimposed input membership functions for the **close** and **far** sets.

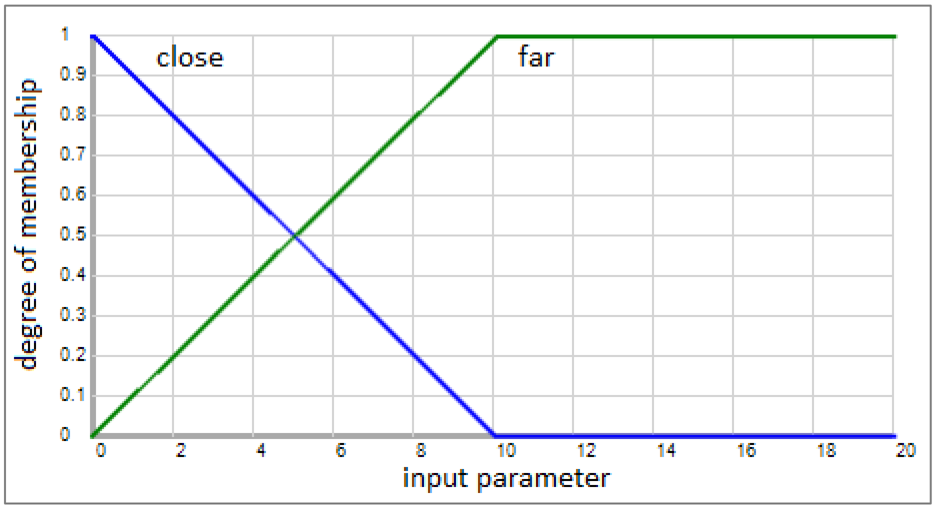

Figure 9. Overlapping transition domains for the input membership functions for the **close** and **far** sets.

Note that the input membership function for the **far** set is equivalent to the input membership function of a NOT **close** set. Thus, the rule "**turn-right** if **front-proximity** is **close** and **left-proximity** is **close** and **right-proximity** is **far**" could equivalently have been phrased "**turn-right** if **front-proximity** is **close** and **left-proximity** is **close** and **right-proximity** is NOT **close**".

Conclusion

The application of fuzzy logic to robot navigation permits vastly more complex and fine grained robot actuation than could normally be achieved through the use of Boolean logic. The challenge with fuzzy logic is to derive a meaningful set of rules that meets the intended goal of robot navigation in a particular environment. Rule generation is sufficiently complex and time consuming to warrant development via a neural network or genetic algorithm. Ideally, rule generation would occur in a simulated environment, where time and mechanical issues are less likely to impede rule development.

References

Braunl, T. (2008). Embedded Robotics. Berlin, Heidelberg: Springer-Verlag.

Fuzzy Control System. (n.d.). In Wikipedia. Retrieved November 25, 2012, from http://en.wikipedia.org/wiki/Fuzzy_control_system

Fuzzy Inference Process. (n.d.) In MathWorks. Retrieved November 25, 2012, from http://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html#FP346

Li W. (1994). Fuzzy-Logic-Based Reactive Behavior Control of an Autonomous Mobile System in Unknown Environments. Engng Applic. Artif. Intell. Vol. 7, No. 5, pp. 521-531

Luger, G. (2009). *Artificial Intelligence.* Boston, MA: Pearson.

Russel, S. and Norvig, P. (2010). *Artificial Intelligence : a modern approach.* Upper Saddle River, NJ: Pearson.

Wolfer, J. and George, C. (2006). *Fuzzy Logic Control For Robot Maze Traversal: An Undergraduate Case Study.* World Congress on Computer Science, Engineering and Technology Education.