Fuzzy Logic

- Today's class:

  No notes, but the following are useful resources:

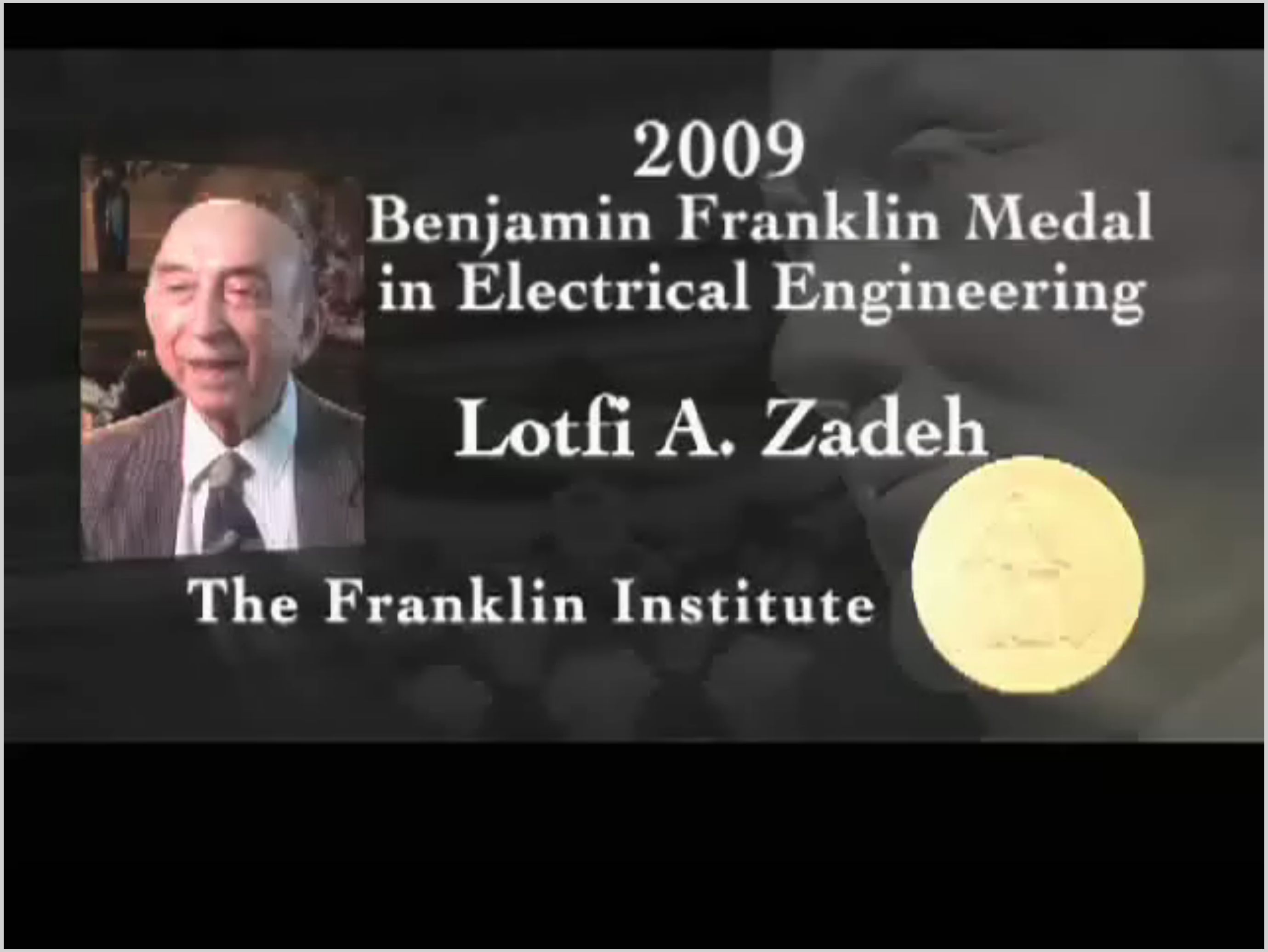
  http://www.seattlerobotics.org/Encoder/mar98/fuz/flindex.html

  http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/fuzzy/part1/faq.html

  http://plato.stanford.edu/entries/logic-fuzzy/

- Objectives:
  - **Broaden FOL treatment in R&N.**
  - **Given a description of a set of behaviors and their fuzzy input and output sets, combine the behaviors using the centroid of area and centroid of largest area methods.**
  - **Be able to evaluate fuzzy rules with conjunctions, disjunctions, and negations**

Lofti Zadeh


2009
Benjamin Franklin Medal
in Electrical Engineering
Lotfi A. Zadeh
The Franklin Institute

Fuzzy Logic

- What if the precise value of the vector output wasn't important?
  - The opposite of "precise" is "fuzzy"
- Turn HARD_RIGHT instead of -90 deg
- Go FAST instead of 0.8*Vmax
- This may be more natural, more linguistic

How do we generate and combine outputs like HARD_RIGHT and SOFT_LEFT?

General Procedure

- Generate fuzzy outputs (Fuzzification)
  – Usually have perception operations that yields a “crisp” number that is then converted to a fuzzy input (fuzzified)
- Apply fuzzy rules (Fuzzy Rules)
  – The control logic has rules to convert the the fuzzy input into a fuzzy output
- Combine fuzzy outputs (Defuzzification)
  – When multiple components produce fuzzy outputs, they have to be converted to a single “crisp” number

Motivating Example: Swerving Robot

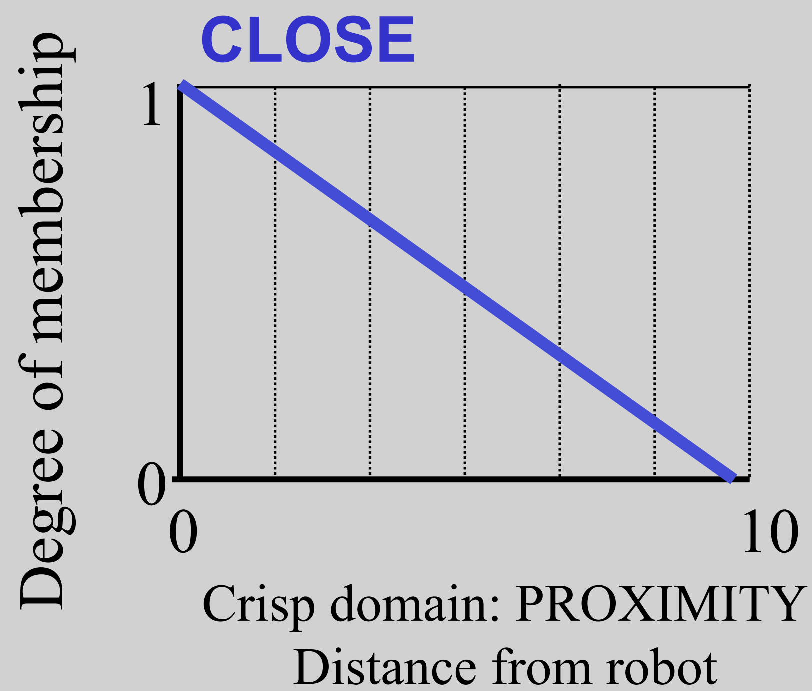
- “Swerve” is a runaway behavior that doesn’t let the robot turn more than 90 degrees from current direction
  - Vdir: [0,90]
  - Closer the robot, the harder the turn to the right
- Robot velocity is same (for simplicity)

Linguistic-based rules

- Swerve 1 (from sensor 1)
  - If an obstacle is close, take a HARD RIGHT

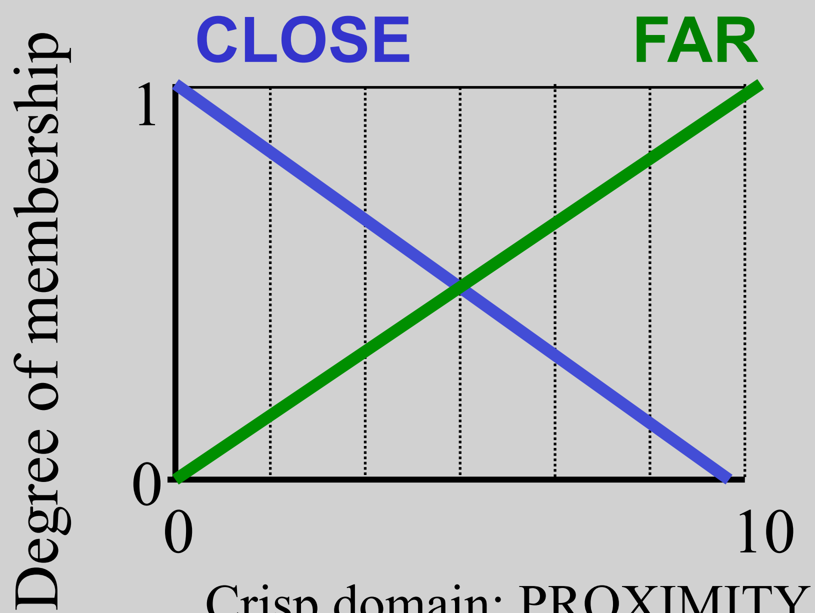
- Swerve 2 (from sensor 2)
  - If an obstacle is far, take a SOFT RIGHT

Fuzzy Sets: Domain



CLOSE is a fuzzy set over the domain

Fuzzy Sets: Multiple Sets

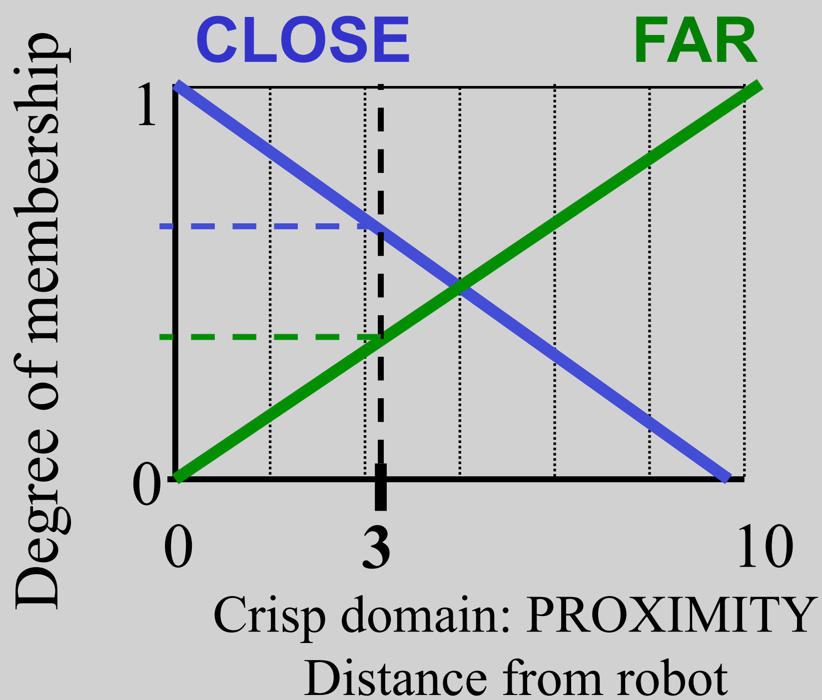


Crisp domain: PROXIMITY
Distance from robot

Notes:

- Fuzzy sets often overlap -- that's seen as a Good Thing
- Set can have different shapes (lines, trapezoids, sigmoids)

Membership Functions

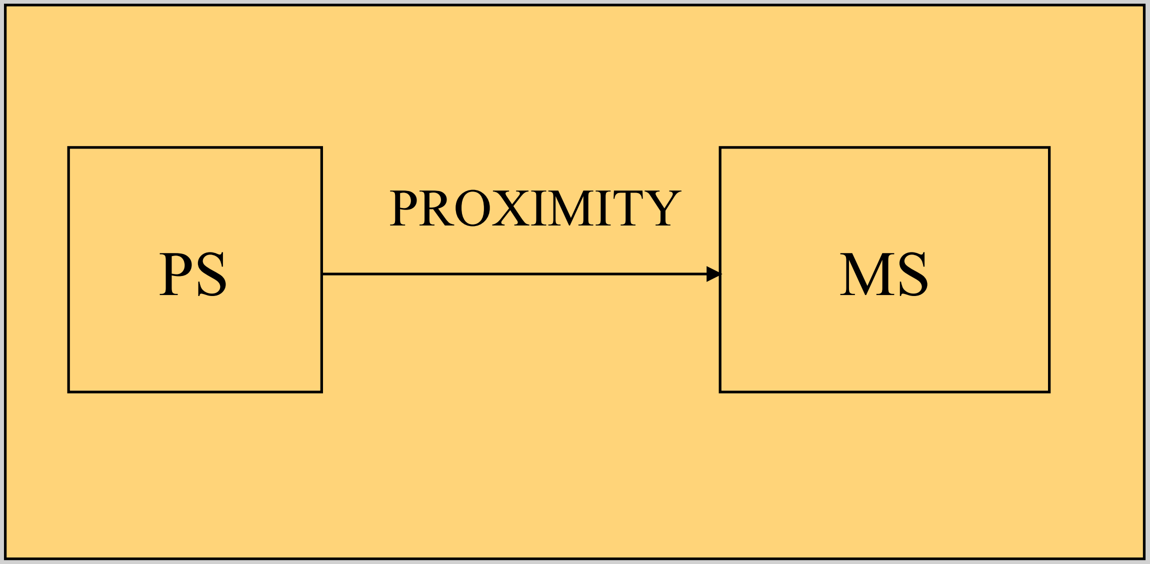


If robot is 3 meters from obstacle, it has a membership in CLOSE of 0.7 and a membership in FAR of 0.3

$$M_{CLOSE}(3)=0.7 \quad M_{FAR}(3)=0.3$$

Back to Swerve

- Perceptual schema is a tuple
  - membership function for the fuzzy variable PROXIMITY
    - $M_{CLOSE}(x)$, $M_{FAR}(X)$



Fuzzy Rules

Motor Schema might be expressed as rule(s):

If PROXIMITY is CLOSE
TURN is HARD_RIGHT } **Fuzzy rule**

If PROXIMITY is FAR
TURN is SOFT_RIGHT } **Fuzzy rule**

Fuzzy Output Variable

Crisp domain: TURN
Distance for robot to
turn to the right

Strength

Motor Schema might be expressed as rule(s):

$M_{CLOSE}(3)=0.7$

If PROXIMITY is CLOSE

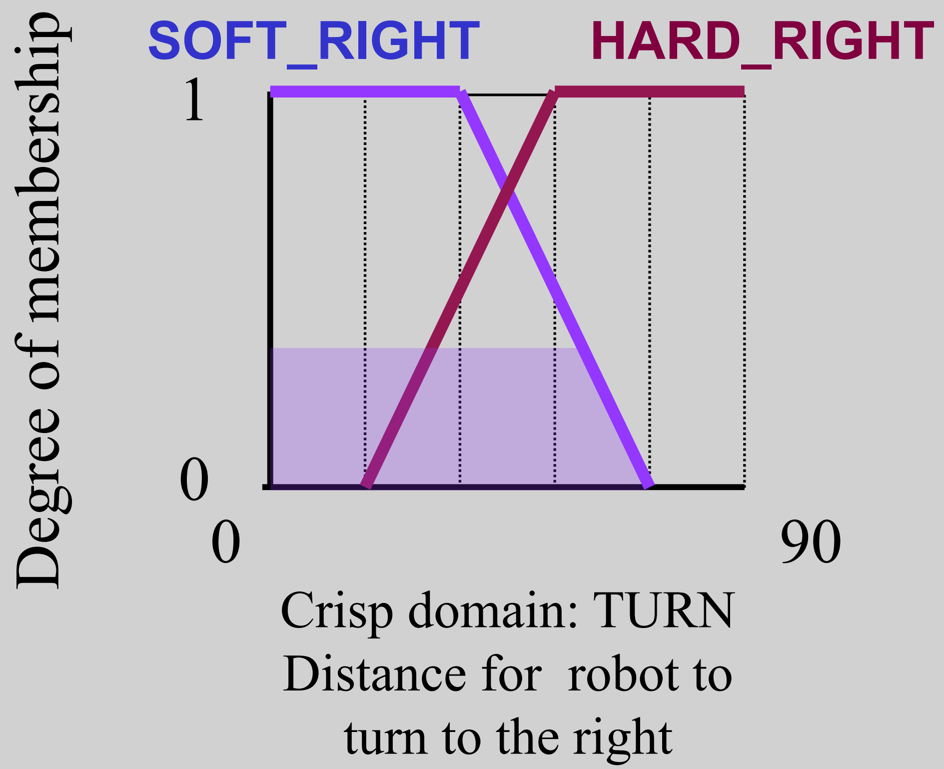
TURN is HARD_RIGHT $M_{HARD\_RIGHT}=0.7$

$M_{FAR}(3)=0.3$

If PROXIMITY is FAR

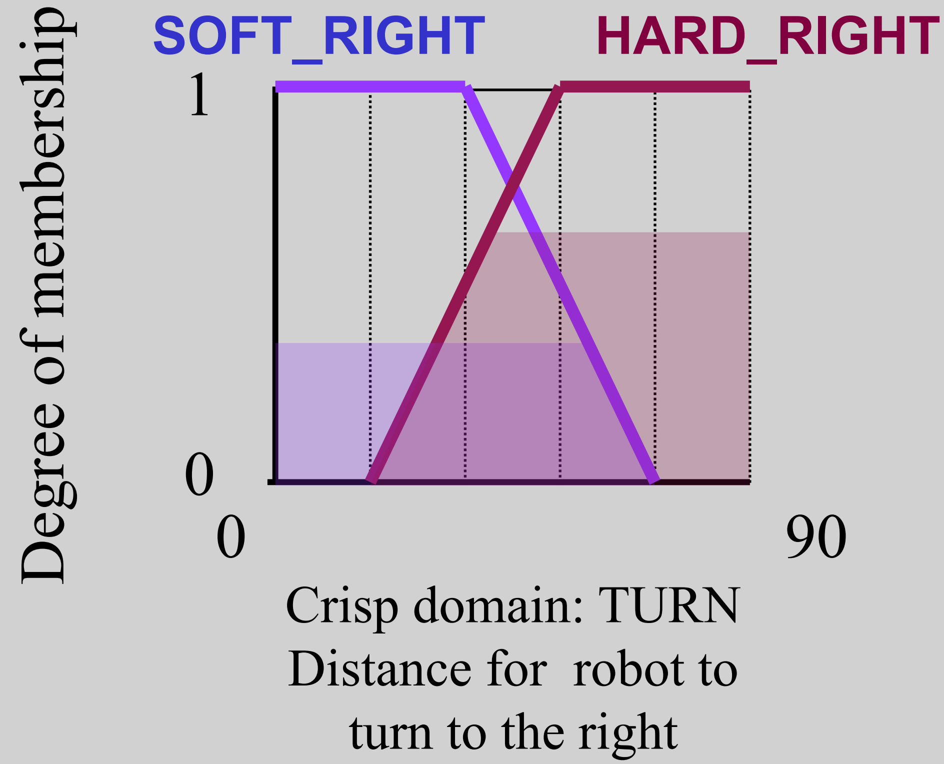
TURN is SOFT_RIGHT $M_{SOFT\_RIGHT}=0.3$

Resulting Membership



$M_{SOFT\_RIGHT}=0.3$

Resulting Membership



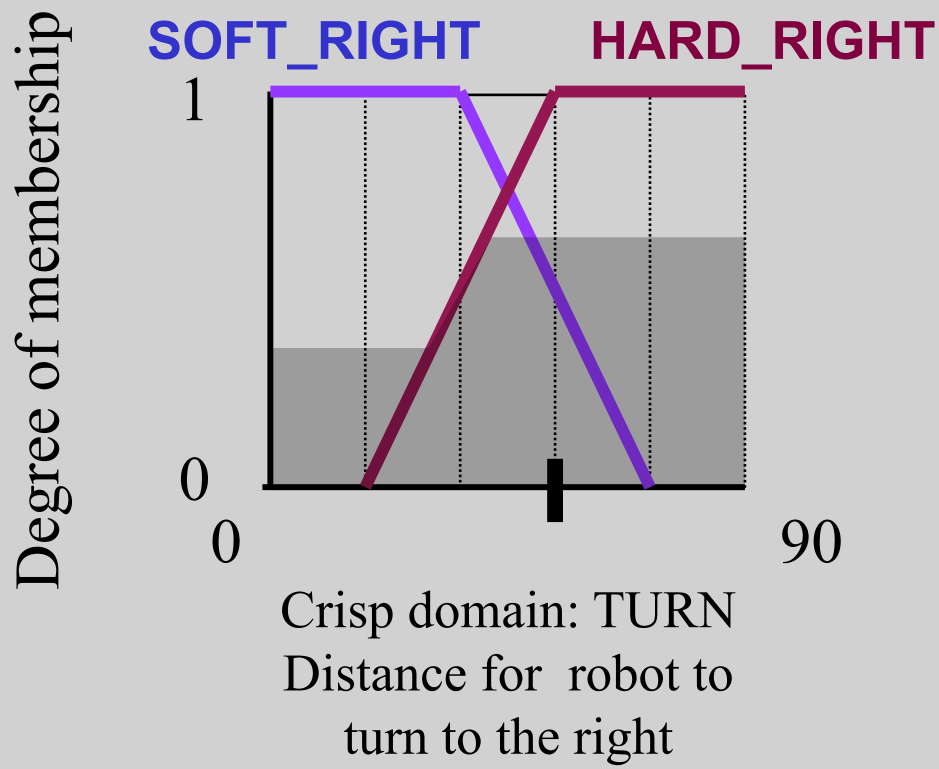
$M_{SOFT\_RIGHT}=0.3$ $M_{HARD\_RIGHT}=0.7$

Defuzzification

- Now we have an output that is a fuzzy variable, but we need to convert it to a crisp value to actually send the motor commands.

- Several alternatives:
  - Take Centroid (along Crisp axis) of Blended Area
  - Take Centroid of Largest Area
  - Weighted Means in area of overlap

Defuzzification: Blended Centroid

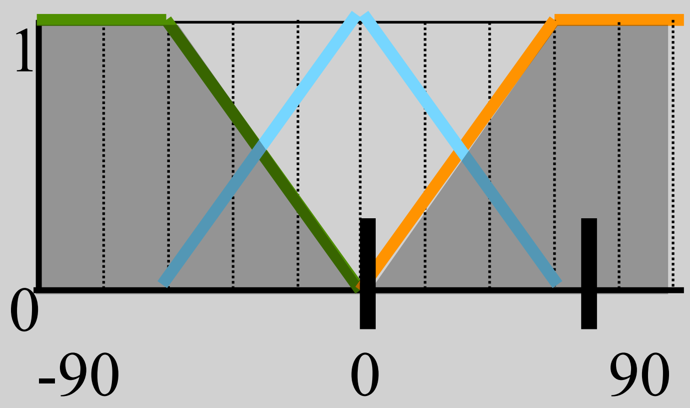


$$C = \frac{\int M_A(X)X}{\int M_A(X)}$$

$$C \approx \frac{174.6}{3.2} = 54.6$$

There can be some problems…

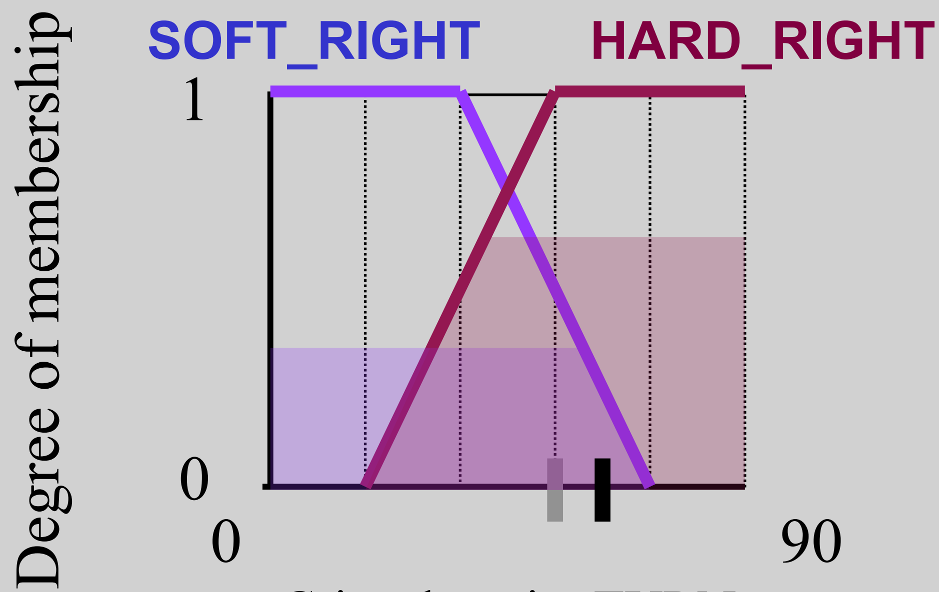
Consider the output of an avoid…



And where would the centroid be?

Consider Centroid of Largest Area (CLA)

Defuzzification: Largest Centroid

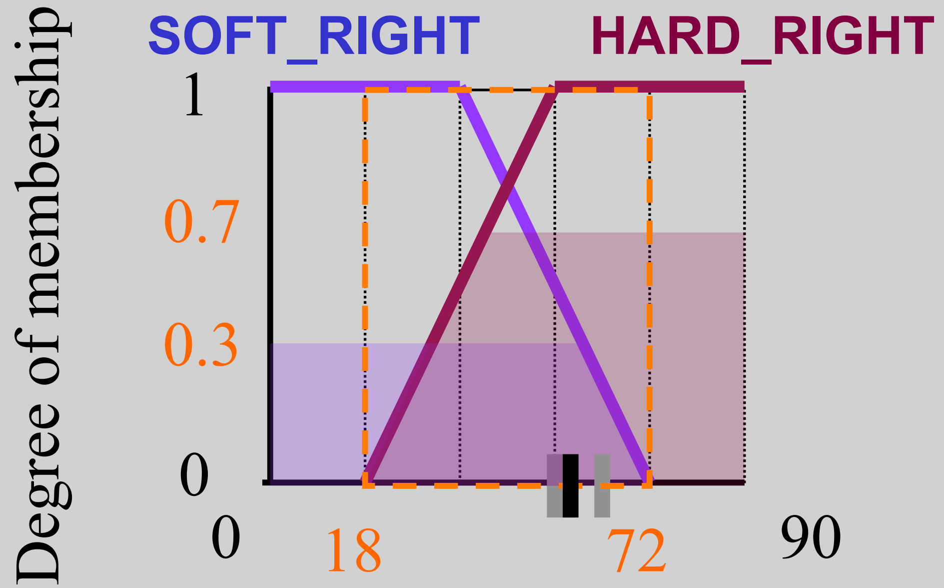


Crisp domain: TURN
Distance for robot to
turn to the right

$$C = \frac{\int M_A(X)X}{\int M_A(X)}$$

$C \approx 61.7$

Defuzzification: Weighted Means

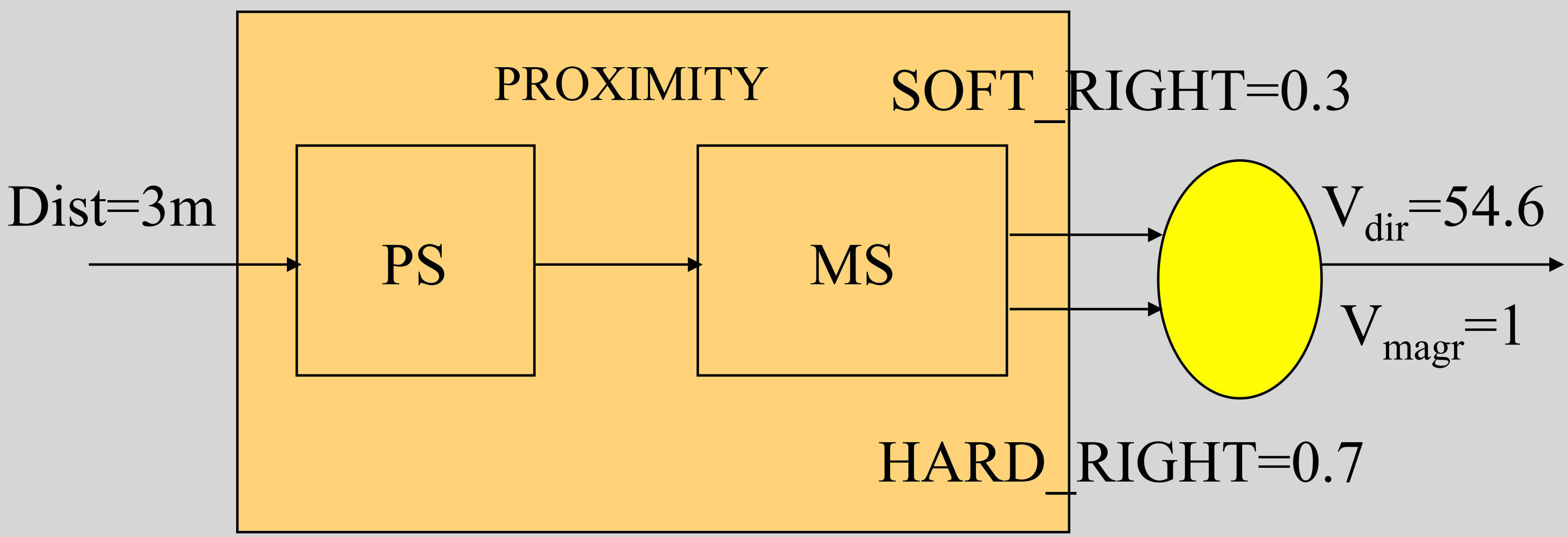


Crisp domain: TURN Distance for robot to turn to the right

$$C = 0.3(18) + 0.7(72) = 55.8$$

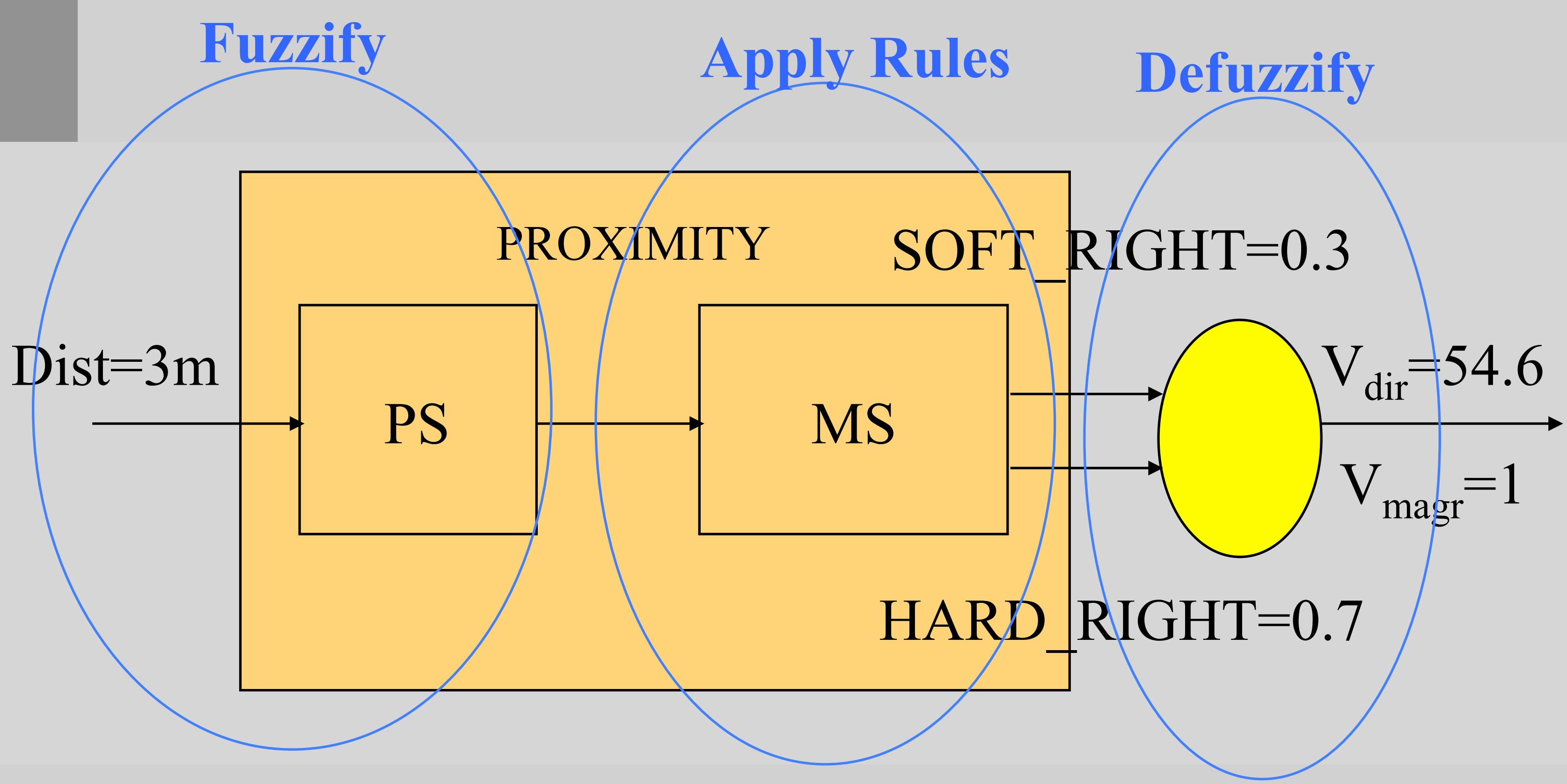
Back to Swerve

- And so we get an answer!



Back to Swerve

- And so we get an answer!



Conjunction, Disjunction, Negation and Hedges

If PROXIMITY is CLOSE AND OPEN is RIGHTSIDE
TURN is HARD_RIGHT

If PROXIMITY is CLOSE AND OPEN is LEFTSIDE
TURN is HARD_LEFT

If PROXIMITY is FAR AND OPEN is RIGHTSIDE
TURN is SOFT_RIGHT

If PROXIMITY is FAR AND OPEN is LEFTSIDE
TURN is SOFT_LEFT

If PROXIMITY is CLOSE AND NOT (OPEN is RIGHTSIDE OR OPEN is LEFTSIDE)
TURN is VERY HARD_RIGHT

Evaluation

- A AND B is minimum(A, B)
  - PROXIMITY is CLOSE= 0.7
  - OPEN IS RIGHTSIDE= 0.3
  - If PROXIMITY is CLOSE AND OPEN is RIGHTSIDE= 0.3

- A OR B is maximum(A, B)
  - PROXIMITY is CLOSE= 0.7
  - OPEN IS RIGHTSIDE= 0.3
  - If PROXIMITY is CLOSE OR OPEN is RIGHTSIDE= 0.7

- NOT A is 1-A
  - OPEN is RIGHTSIDE=0.2
  - NOT OPEN is RIGHTSIDE=1-0.2=0.8

    (1.0 is full membership for our examples)

Example

If PROXIMITY is CLOSE AND NOT (OPEN is RIGHTSIDE OR OPEN is LEFTSIDE)

Where

OPEN is RIGHTSIDE=0.1

OPEN is LEFTSIDE=0.15

PROXIMITY is CLOSE=0.95

Hedges

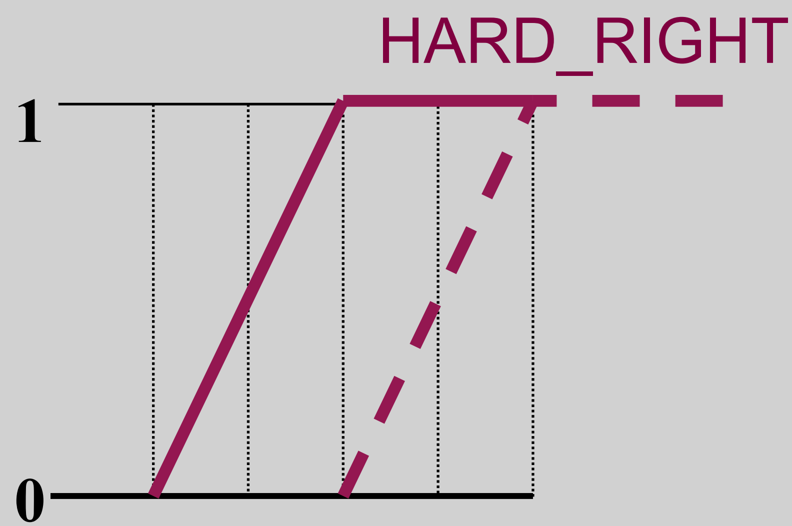
If PROXIMITY is CLOSE AND NOT (OPEN is RIGHTSIDE OR OPEN is LEFTSIDE)

TURN is VERY HARD_RIGHT

OPEN is RIGHTSIDE=0.1

OPEN is LEFTSIDE=0.15

PROXIMITY is CLOSE=0.95



Moves the fuzzy set over

Summary

- Many packages have fuzzy logic: Java, MATLAB

- Fuzzy works by
  - Fuzzification of crisp values
  - Application of rules, each of which has a strength
  - Defuzzification output of rules to produce a crisp output

- Problems include
  - Results may not be what was expected
  - The number and shape of sets impact behavior