ECE 478-578
Intelligent Robotics I

PhD. Husnu Melih Erdogan – Electrical & Computer Engineering

herdogan@pdx.edu **Teaching Assistant**

Portland State University

Introduction to ROS Part – 6
&
Fuzzy Logic with Scikit-Fuzzy

ROS

Scikit Fuzzy

S2DR

Portland State
UNIVERSITY

Course Structure

- Part 1 - Overview
  - What is ROS?
  - Introduction to ROS
  - ROS architecture, philosophy, history
  - How to install ROS?
  - Examples
  - Installation
  - ROS Master
  - ROS Nodes
  - ROS Topic
  - ROS Messages
  - Console Commands
  - ROS Packages
  - ROS Launch-files
  - Catkin Workspace and Build System
  - Turtlesim
- Part 2 - Basics
  - ROS File System
  - ROS Package
  - How to create a package?
  - How to build a package?
  - Creating a Publisher Node
  - Creating a Subscriber Node
  - Assignment 3
- Part 3 - Debug
  - ROS Launch File
  - How to use ROS .bagfiles?
  - ROS Parameters
  - ROS Namespace
- Part 4 - Speech
  - ROS Services
  - Speech Recognition
  - Speech Synthesis
  - Google Dialogflow
- Part 5 - Speech
  - Amazon Polly
  - ROS Actions
  - Assignment 4

    (Optional)
- Part 6 - Fuzzy
  - 2D Multi-Robot Simulator
  - Fuzzy Logic
  - Assignment 5
- Part 7 - Network
  - ROS Messages
  - Rviz
  - ROS Networking
  - ROS and RaspberryPi

Portland State UNIVERSITY

2D Robot Simulation STDR

STDR

- Simple Two Dimensional Robot Simulator (STDR Simulator) is a 2-D multi-robot Unix simulator.
- STDR Simulator's goal is not to be the most realistic simulator, or the one with the most functionalities. The intention is to make a single robot's, or a swarm's simulation as simple as possible, by minimizing the needed actions the researcher has to perform to start his/hers experiment.
- In addition, STDR can function with or without a graphical environment, which allows for experiments to take place even using ssh connections

S2DR

Portland State UNIVERSITY

STDR

- STDR Simulator is created in way that makes it totally ROS compliant.
- Every robot and sensor emits a **ROS transformation (tf)** and all the measurements are published in ROS topics.
- In that way, STDR uses all ROS advantages, aiming at easy usage with the world's most state-of-the-art robotic framework.
- STDR can work together with **ROS Rviz**

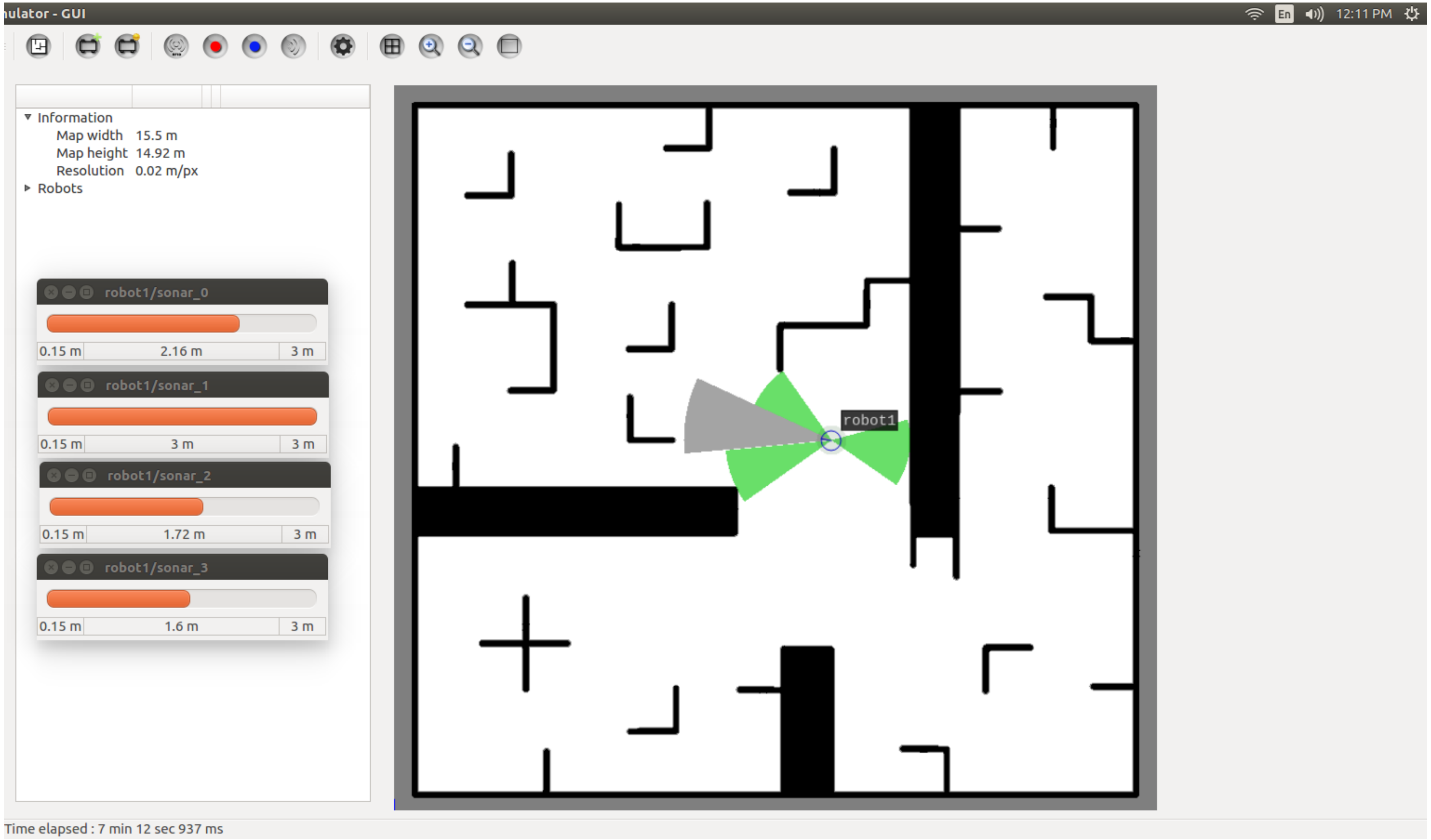
STDR

- Install STDR from source
  - **cd catkin_ws/src**
  - **git clone https://github.com/stdr-simulator-ros-pkg/stdr_simulator.git**
  - **cd .. rosdep install --from-paths src --ignore-src --rosdistro $ROS_DISTRO**
  - **catkin_make**
  - **source devel/setup.bash**
- Start a new server
  - **roslaunch stdr_launchers server_no_map.launch**
- load a map
  - **cd catkin_ws/src/stdr_simulator/stdr_resources**
  - **rosrun stdr_server load_map maps/sparse_obstacles.yaml**
  - **cd catkin_ws/src/stdr_simulator/stdr_resources**
  - **rosrun stdr_server load_map maps/map1.yaml**
- start guiroslaunch
  - **stdr_gui stdr_gui.launch**

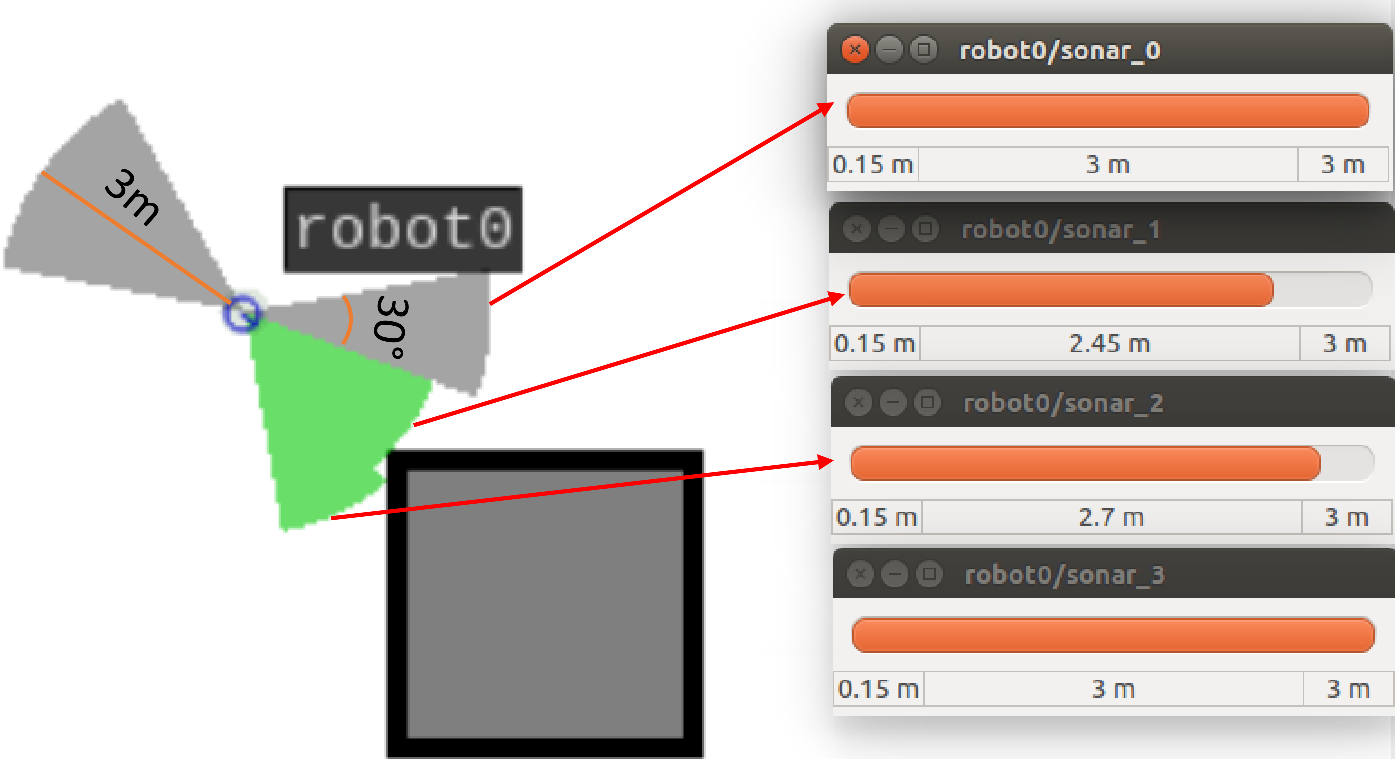
STDR

- load a robot
  - **cd catkin_ws/src/stdr_simulator/stdr_resources**
  - **rosrun stdr_robot robot_handler add resources/robots/pandora_robot.yaml 9 7 1.57**
  - **rosrun stdr_robot robot_handler add resources/robots/robot.xml 5 5 1.2**
- modify robots
  - **rosrun stdr_robot robot_handler replace robot0 2 2 0**
  - **rosrun stdr_robot robot_handler delete robot0**
- test
  - **rostopic pub -1 /robot1/cmd_vel geometry_msgs/Twist -- '[0.0, 0.0, 0.0]' '[0.0, 0.0, 0.0]’**
- test with keyboard
  - **rosrun teleop_twist_keyboard teleop_twist_keyboard.py cmd_vel:=robot0/cmd_vel**

STDR – How to use it?

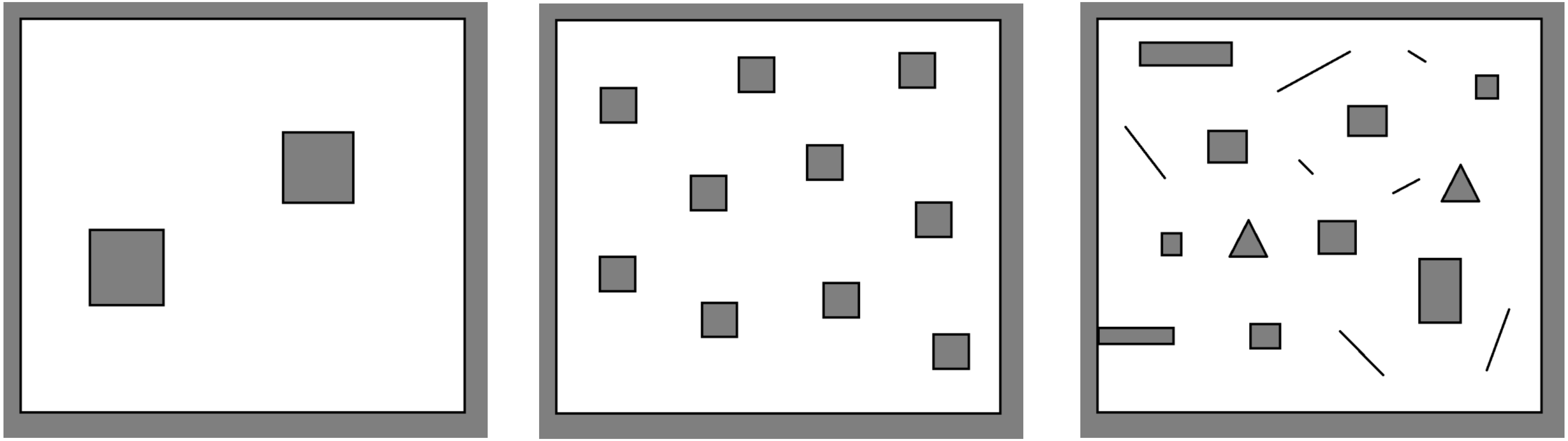


STDR - Robot



Portland State UNIVERSITY

STDR - Maps

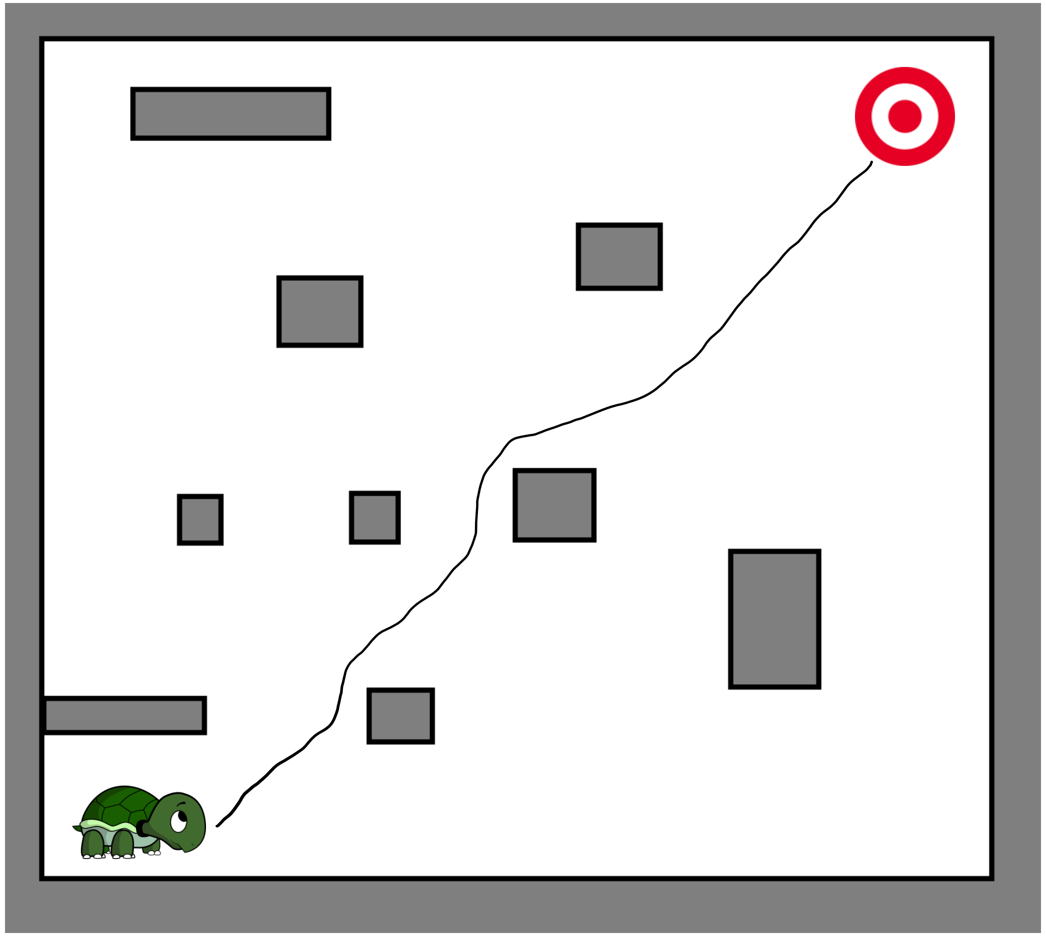


Fuzzy Logic Review

Portland State University

Fuzzy Logic for Obstacle Avoidance

- ROS + stdr + fuzzy logic + scikit-fuzzy



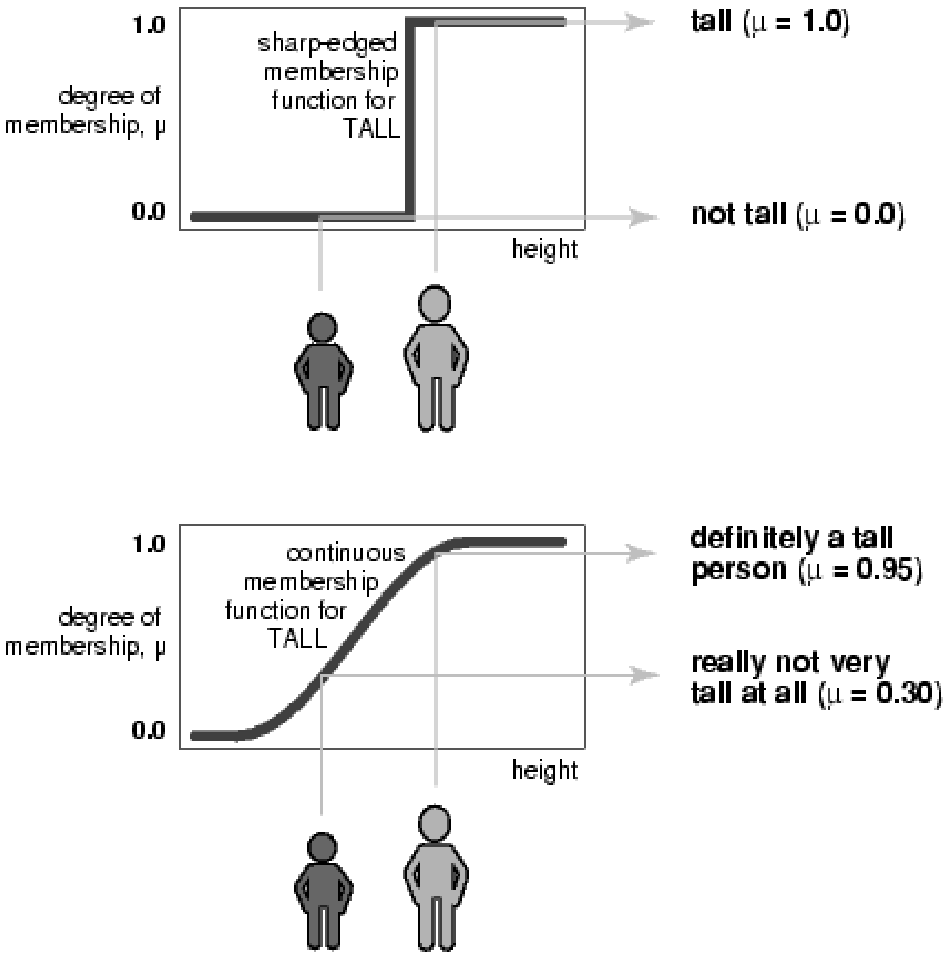
Fuzzy Logic Review

- Fuzzy logic is an extension of Boolean Logic
- Fuzzy logic is based on the theory of fuzzy sets.
- It is a generalization of the classical set theory.
- It introduces the notion of degree in the verification of a condition
- It enables a condition to be in a state other than true or false,
- Fuzzy logic provides a very valuable flexibility for reasoning
- It is possible to take into account inaccuracies and uncertainties

Fuzzy Logic Review - Examples:

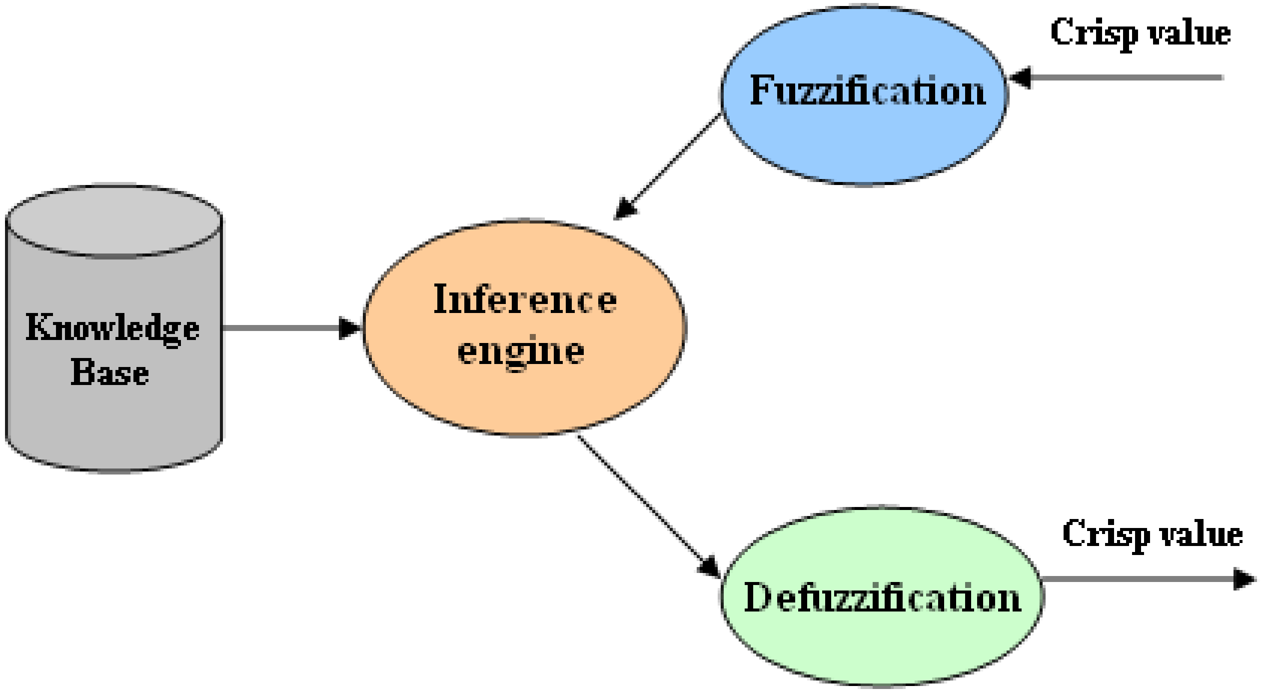
- If service is poor or food is bad, then tip is cheap
- If service is good, then tip is average
- If service is excellent or food is delicious, then tip is generous
- If you drive fast, then you can arrive your destination in about 2 hours.

Fuzzy Logic Review - Example



Fuzzy Logic Review

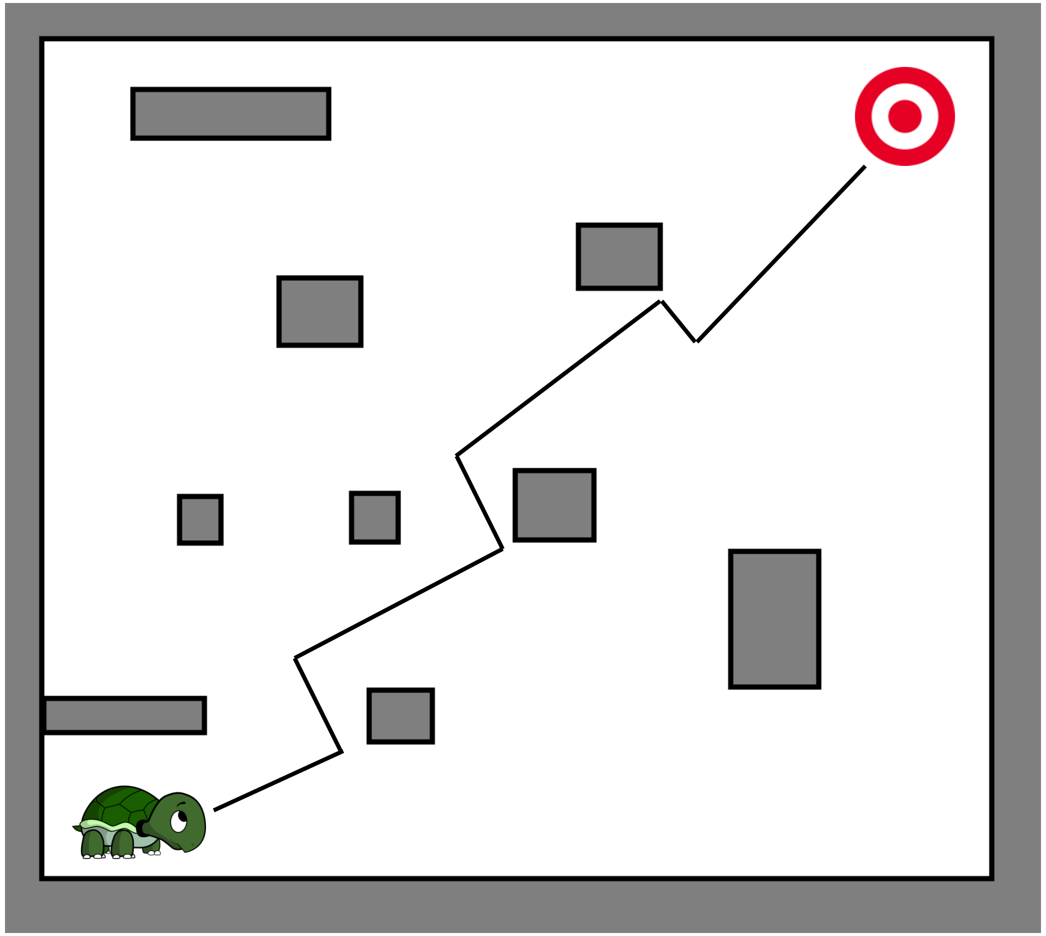
Give crisp input calculate the crisp output



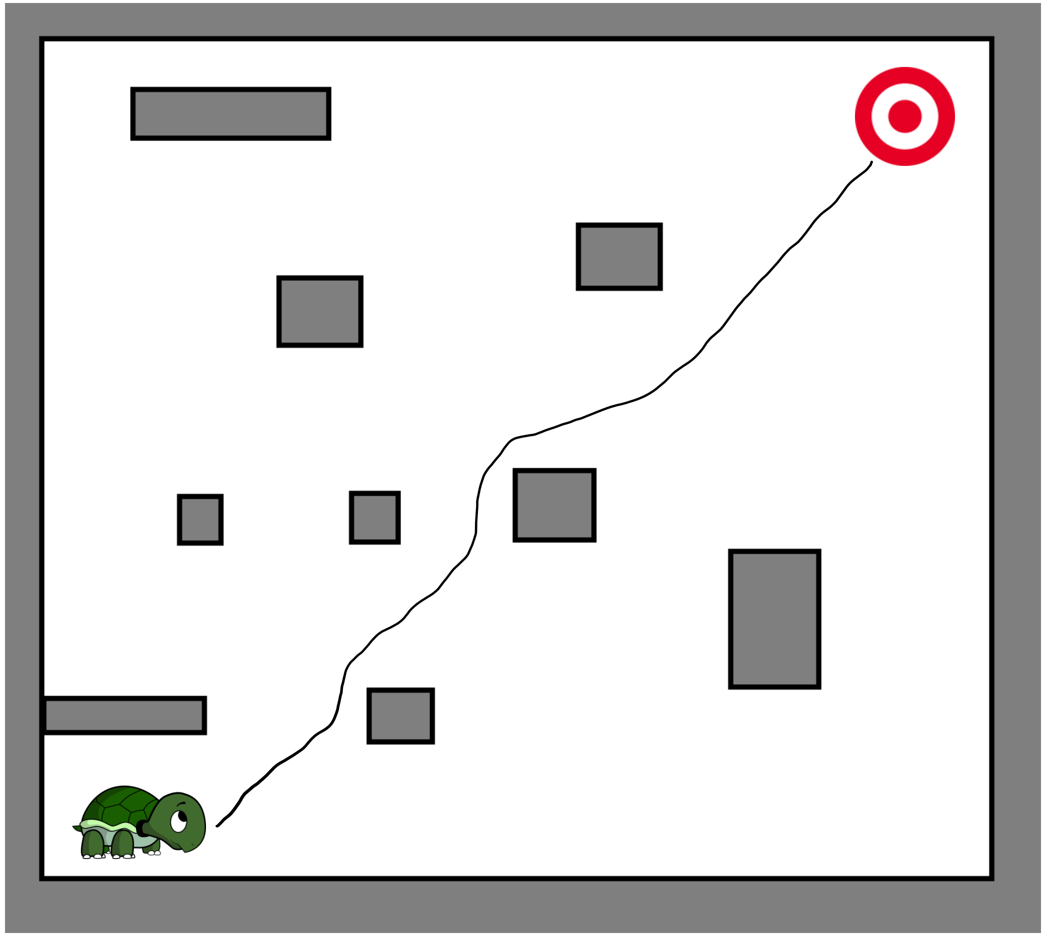
Fuzzy Logic Review

- **Fuzzification module:** transforms the system inputs, which are crisp numbers, into fuzzy sets. This is done by applying a fuzzification function.
- **Knowledge base:** stores IF-THEN rules provided by experts.
- **Inference engine:** simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules.
- **Defuzzification module:** transforms the fuzzy set obtained by the inference engine into a crisp value.

Fuzzy Logic for Obstacle Avoidance

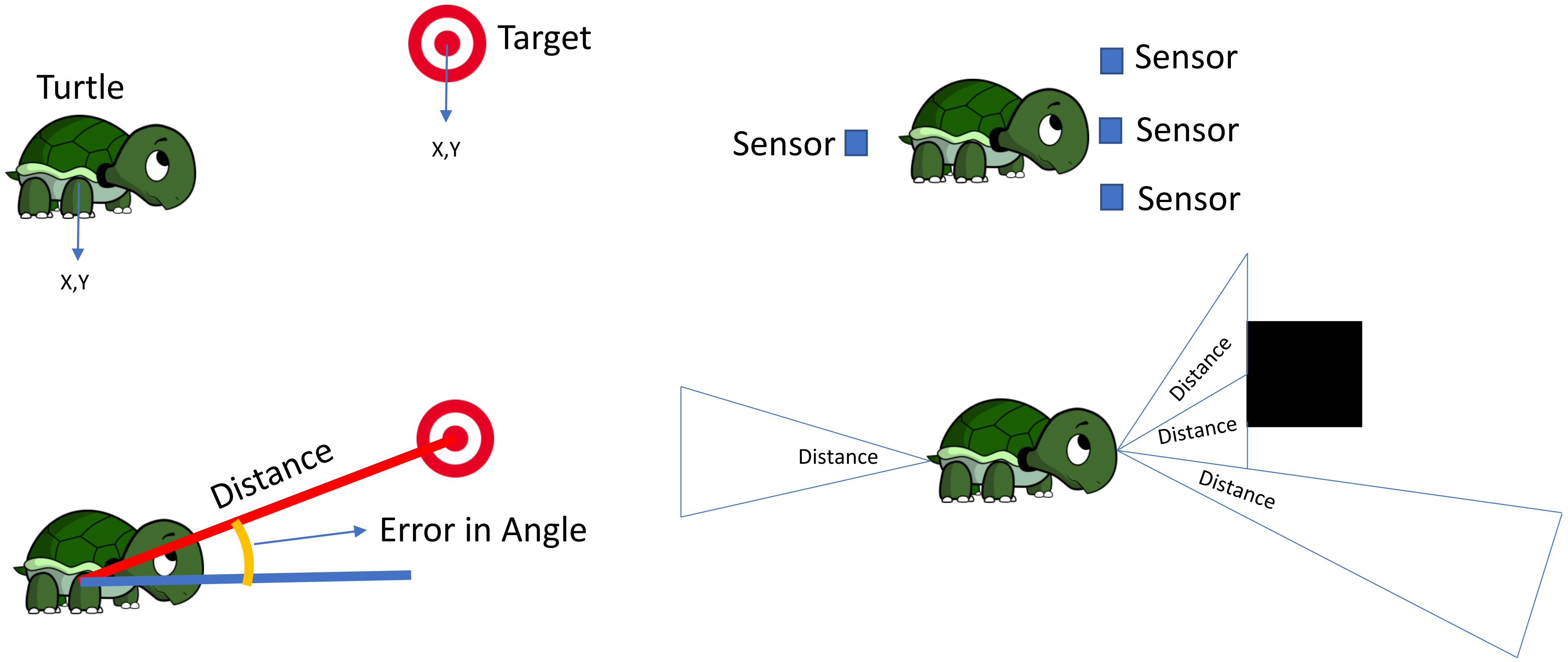


Fuzzy Logic for Obstacle Avoidance



Crisp Inputs

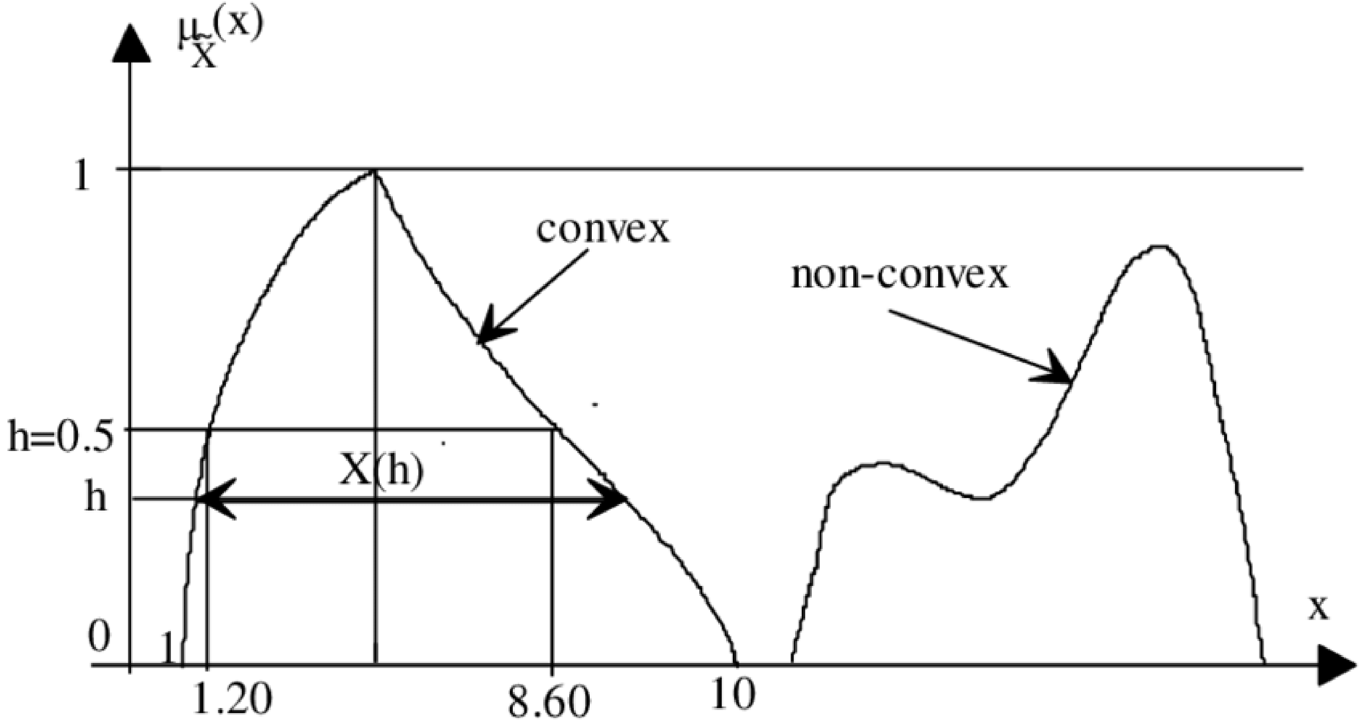
- Sensor Values, Destination Distance, Destination Angle



Portland State University

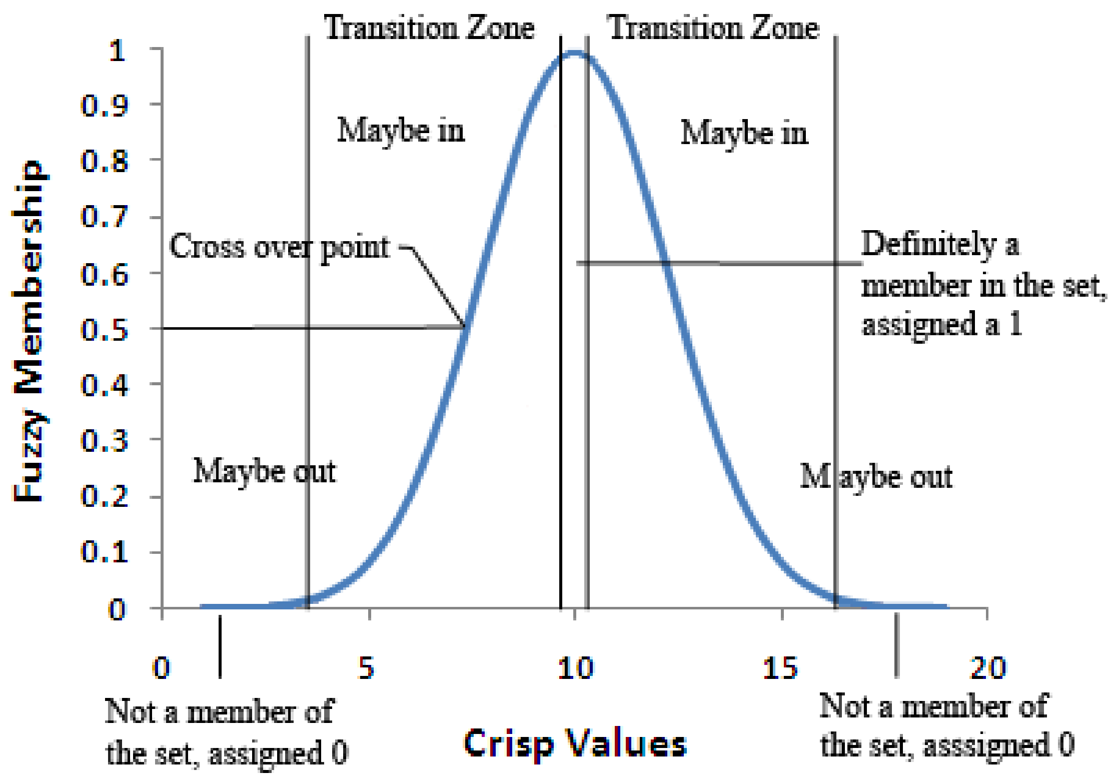
Fuzzy Number/Fuzzy Set

- Fuzzy number is simply a fuzzy set defined on the real numbers.
- Fuzzy numbers must be both normal and convex

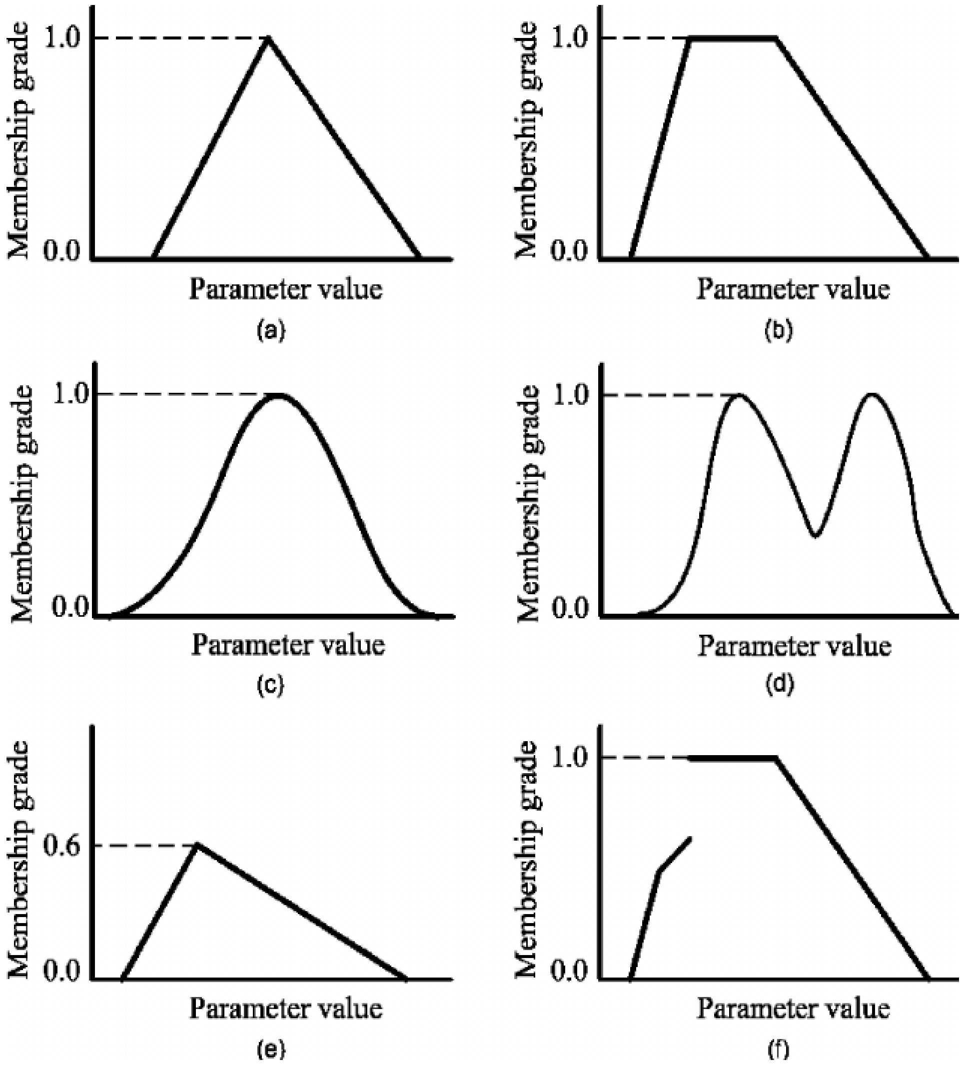


Membership Functions

- A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1.
- Degree of membership for each fuzzy number (set).



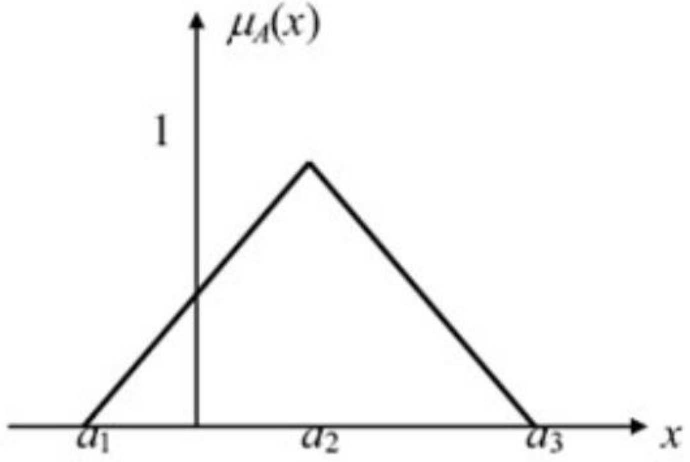
Fuzzy Number Types



Triangular Fuzzy Number

- Fuzzy number represented with three points: $A = (a_1, a_2, a_3)$

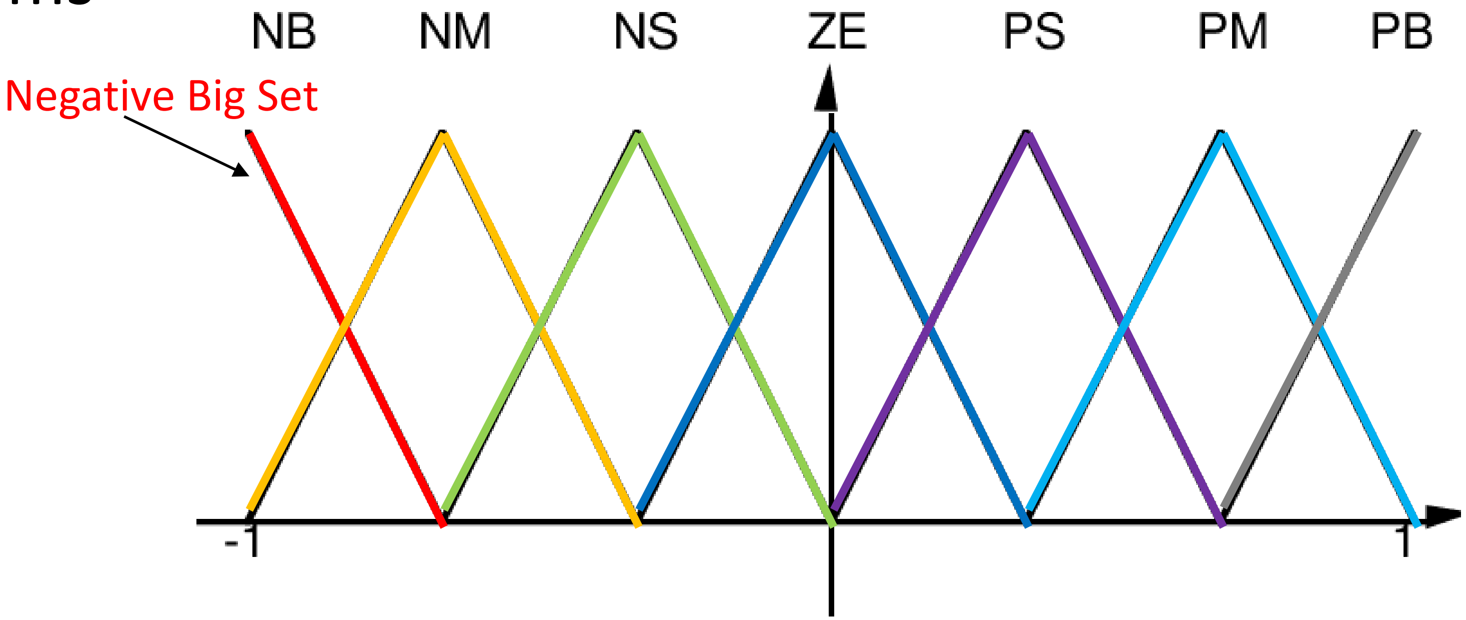
$$\mu_{(A)}(x) = \begin{cases} 0, & x < a_1 \\ \dfrac{x - a_1}{a_2 - a_1}, & a_1 \le x \le a_2 \\ \dfrac{a_3 - x}{a_3 - a_2}, & a_2 \le x \le a_3 \\ 0, & x > a_3 \end{cases}$$



Portland State University

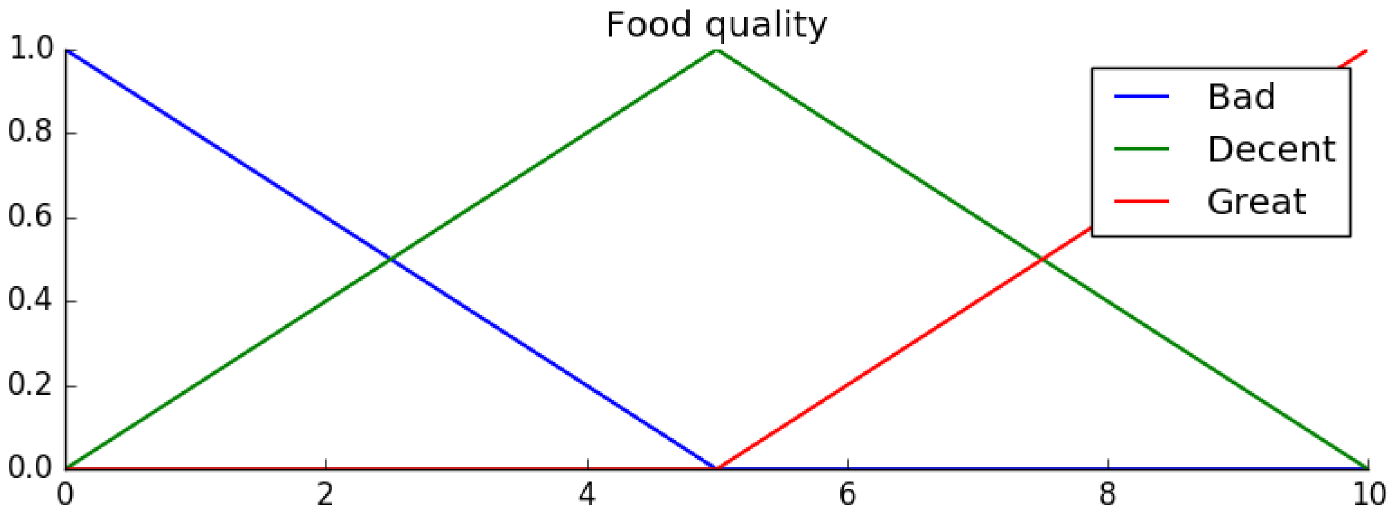
Linguistic Variables

- Fuzzy control was used to be called linguistic control.
- When we define the sets we use linguistic variables
- Set of Fuzzy Linguistic Terms
  - NB: Negative Big
  - NM: Negative Medium
  - NS: Negative Small
  - ZO: Zero or Near Zero
  - PS: Positive Small
  - PM: Positive Medium
  - PB: Positive Big



- There must be a relation between the base variable space and the term set.

Fuzzy Relation For a Linguistic Variable



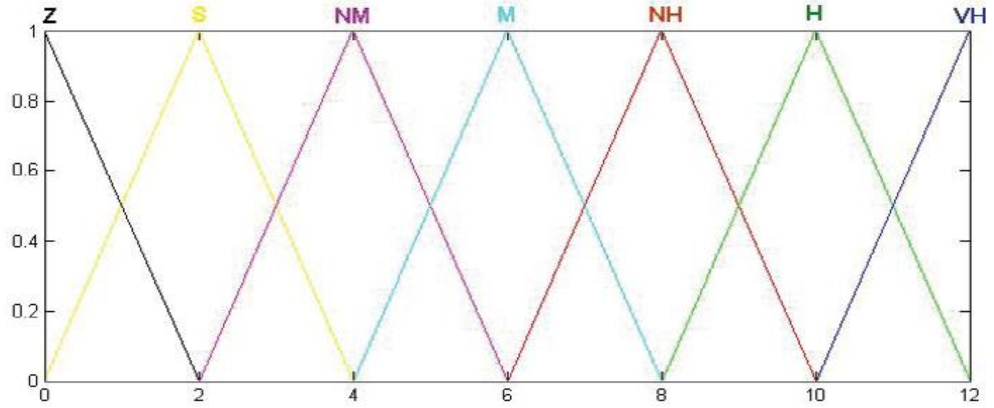
Fuzzy Relation For a Linguistic Variable

- When define the fuzzy sets of linguistic variables, the goal is not to exhaustively define the linguistic variables.
- Instead, we only define a few fuzzy subsets that will be useful later in definition of the rules that we apply it.
- For example we didn’t define subset "average" for the quality of the food. Indeed, this subset will not be useful in our rules.
- Similarly, it is also the reason why (for example) 30 is a higher tip than 25.
- We have not created of fuzzy set "very high" because we do not need it in our rules.

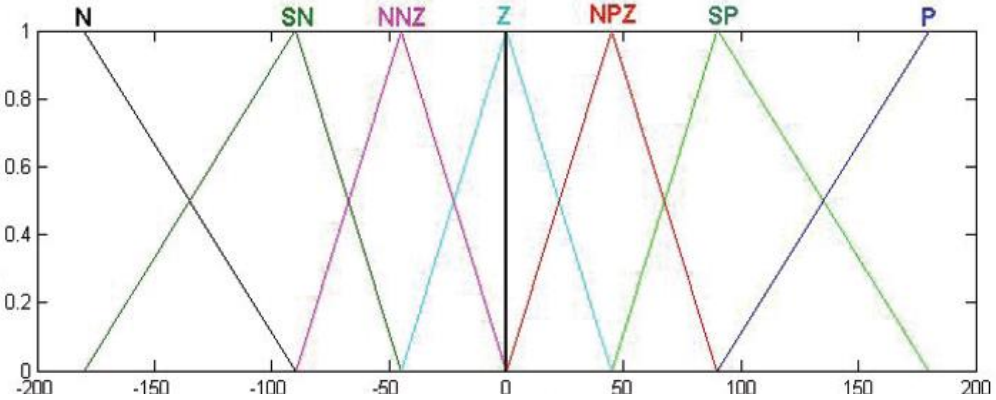
Portland State UNIVERSITY

Fuzzy Input Set

- Obstacle Avoidance Example



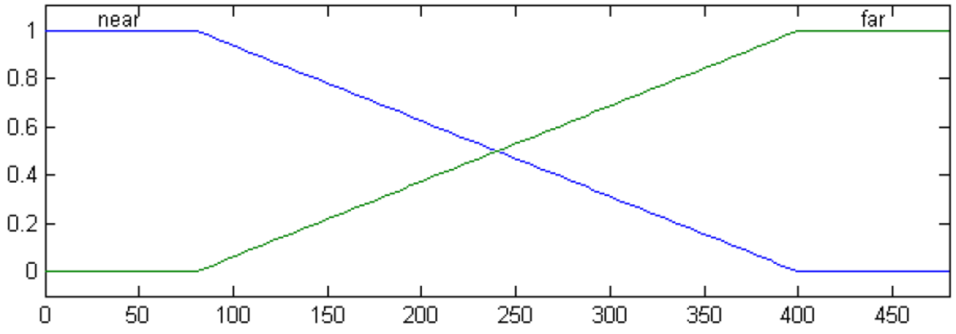
Membership Functions for Distance



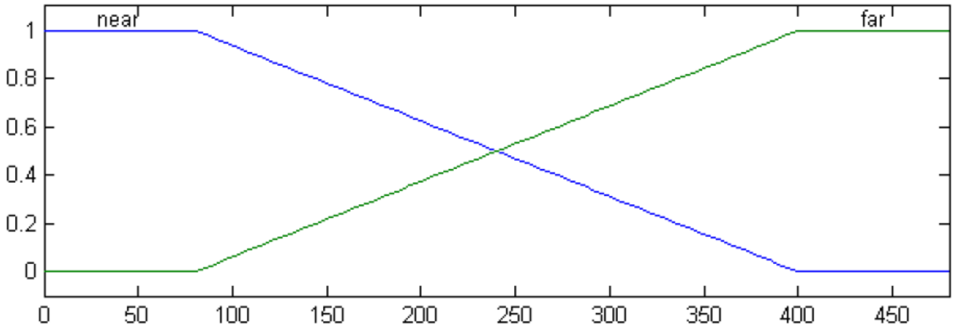
Membership Functions for Error in Angle

Fuzzy Input Sets

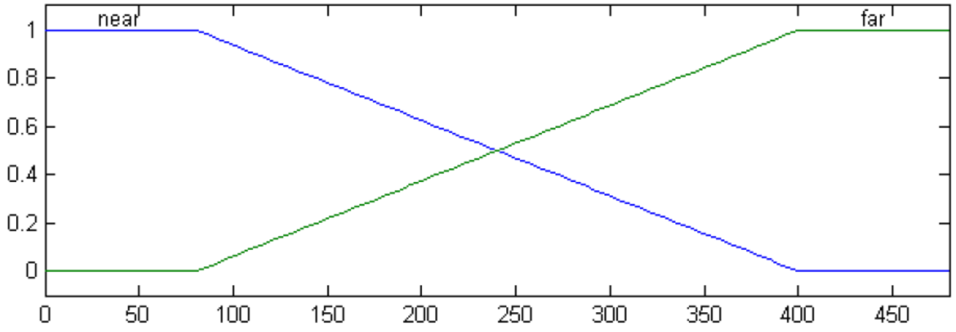
- Obstacle Avoidance Example



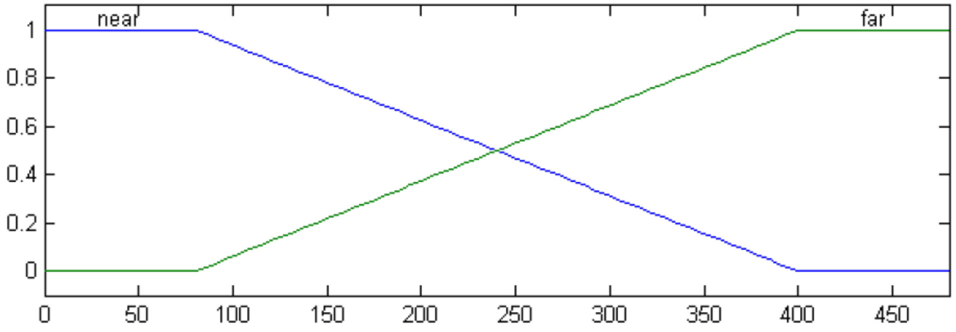
Membership Functions for Front Left Sensor



Membership Functions for Front Right Sensor



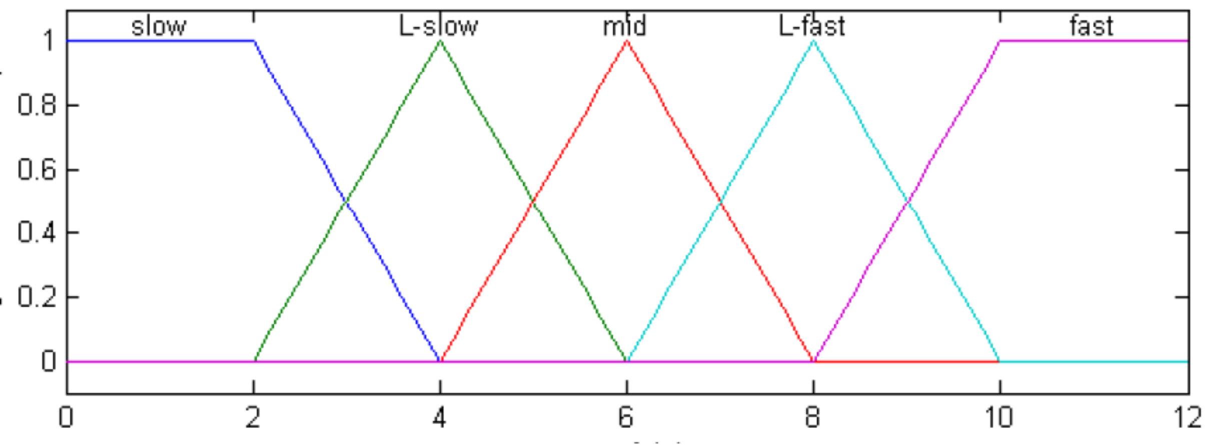
Membership Functions for Front Middle Sensor



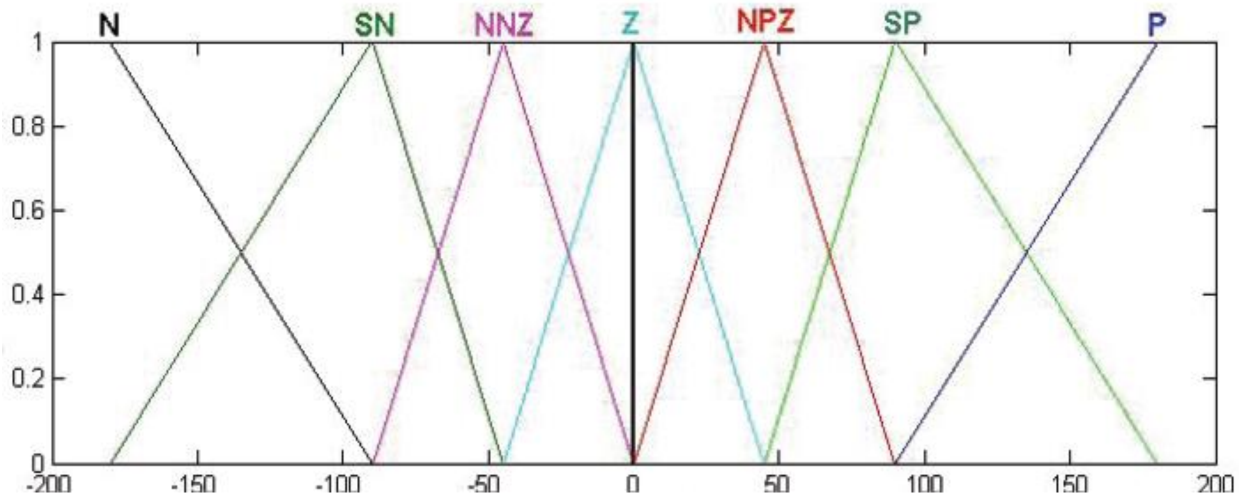
Membership Functions for Back Sensor

Fuzzy Output Sets

- Obstacle Avoidance



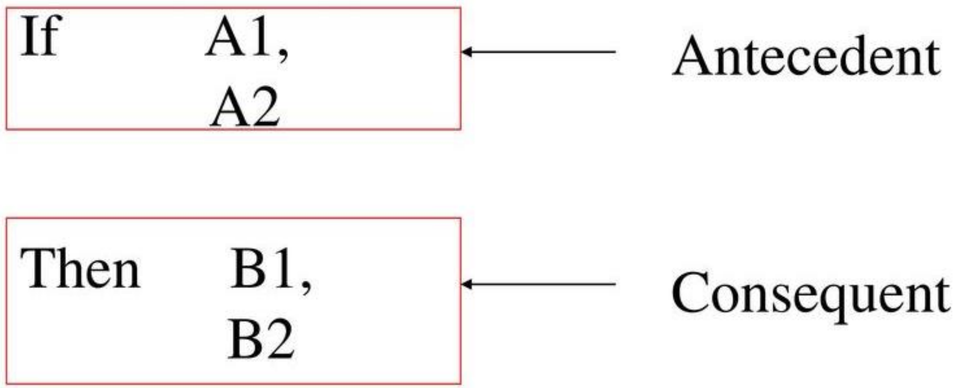
Membership Functions for Velocity



Membership Functions for Error in Angle

Fuzzy Rules – Knowledge Base

- Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic.
- If-then rule statements are used to formulate the conditional statements that comprise fuzzy logic.
- if *x1* is *A* and x2 is A2 then *y1* is *B1 and y2 is B2*



Fuzzy Rules - Knowledge Base

- Food Example - Written Format

If the service is bad or the food is awful	then the tip is low
If the service is good	then the tip is average
If the service is excellent or the food is delicious	then the tip is high

Fuzzy Rules - Knowledge Base

- Obstacle Avoidance Example - Table Format

Input			Output	
LD	FD	RD	RV	LV
N	N	N	NH	NH
N	N	M	N	NH
N	N	F	N	NH
N	M	N	NH	NH
N	M	M	N	NH
N	M	F	N	NH
N	F	N	NH	NH
N	F	M	N	NH
N	F	F	N	NH
M	N	N	NH	N
M	N	M	NH	NH
M	N	F	VHP	P
M	M	N	P	VHP
M	M	M	VHP	P
M	M	F	VHP	P
M	F	N	NH	N
M	F	M	VHP	P
M	F	F	VHP	P
F	N	N	NH	N
F	N	M	P	VHP
F	N	F	NH	NH
F	M	N	NH	N
F	M	M	P	VHP
F	M	F	VHP	P
F	F	N	NH	N
F	F	M	P	VHP
F	F	F	HP	HP

Fuzzy Operators

Name	Intersection AND: $\mu_{A\cap B}(x)$	Union OU: $\mu_{A\cup B}(x)$	Complement NOT: $\mu_{\bar{A}(x)}$
Zadeh Operators MIN/MAX	$min\,(\mu_A(x), \mu_B(x))$	$max\,(\mu_A(x), \mu_B(x))$	$1 - \mu_A(x)$

- Union: Let μ_A and μ_B be membership functions that define the fuzzy sets A and B, respectively, on the universe X. The union of fuzzy sets A and B is a fuzzy set defined by the membership function:

$$\mu_{AUB}(x) = Max(\mu_A(x),\mu_B(x))$$

- Intersection: Let μ_A and μ_B be membership functions that define the fuzzy sets A y B, respectively, on the universe X. The intersection of fuzzy sets A and B is a fuzzy set defined by the membership function:

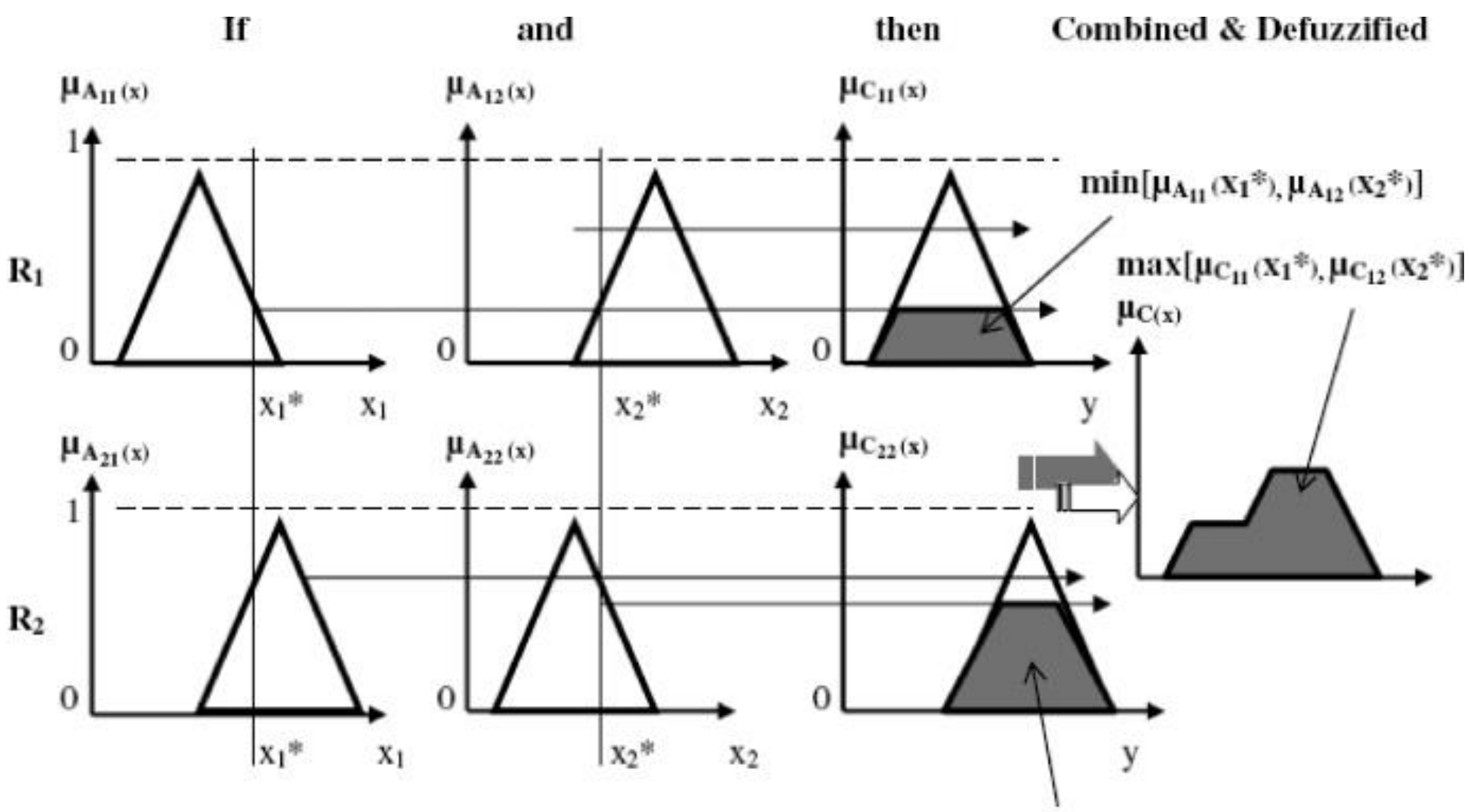
$$\mu_{A\cap B}(x) = Min(\mu_A(x),\mu_B(x))$$

- Complement: Let μ_A be a membership function that defines the fuzzy set A, on the universe X. The complement of A is a fuzzy set defined by the membership function:

$$\mu_{A^c}(x) = 1 - \mu_A(x)$$

Portland State UNIVERSITY

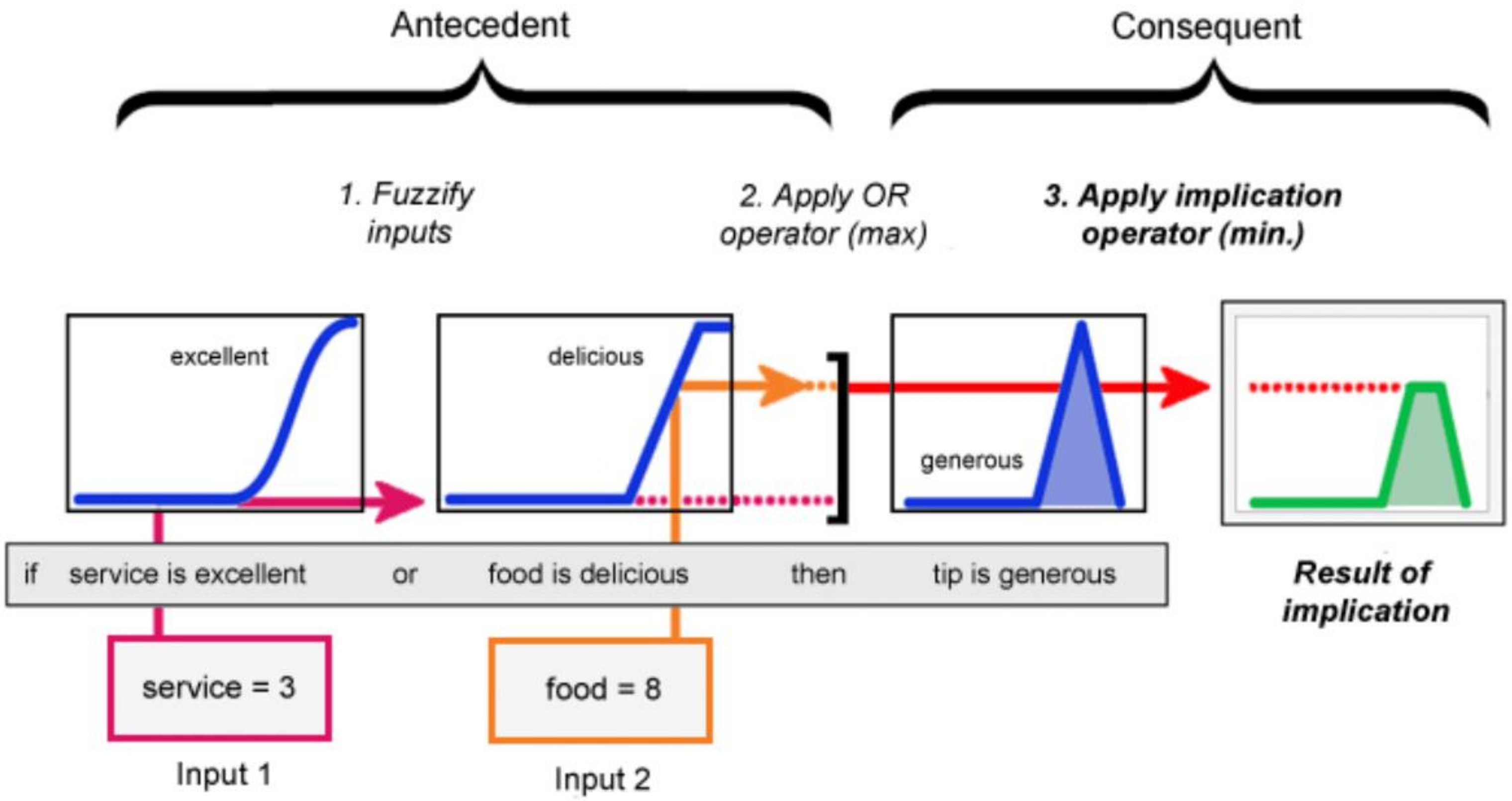
Fuzzy Reasoning and Mamdani Method



Mamdani Fuzzy Logic Reasoning Process

Fuzzy Inference – Fuzzy Reasoning

- Food Example



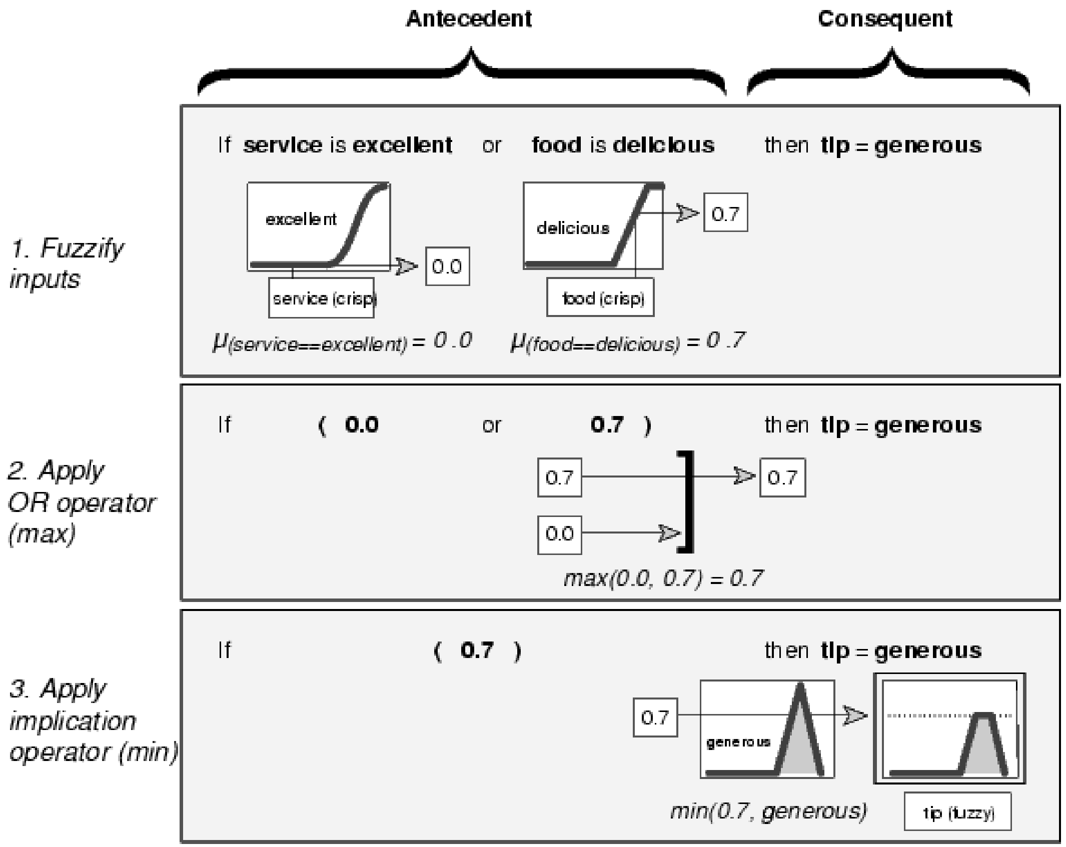
Fuzzy Inference – Fuzzy Reasoning

1- The consequent specifies a fuzzy set be assigned to the output.

2- The implication function then modifies that fuzzy set to the degree specified by the antecedent.

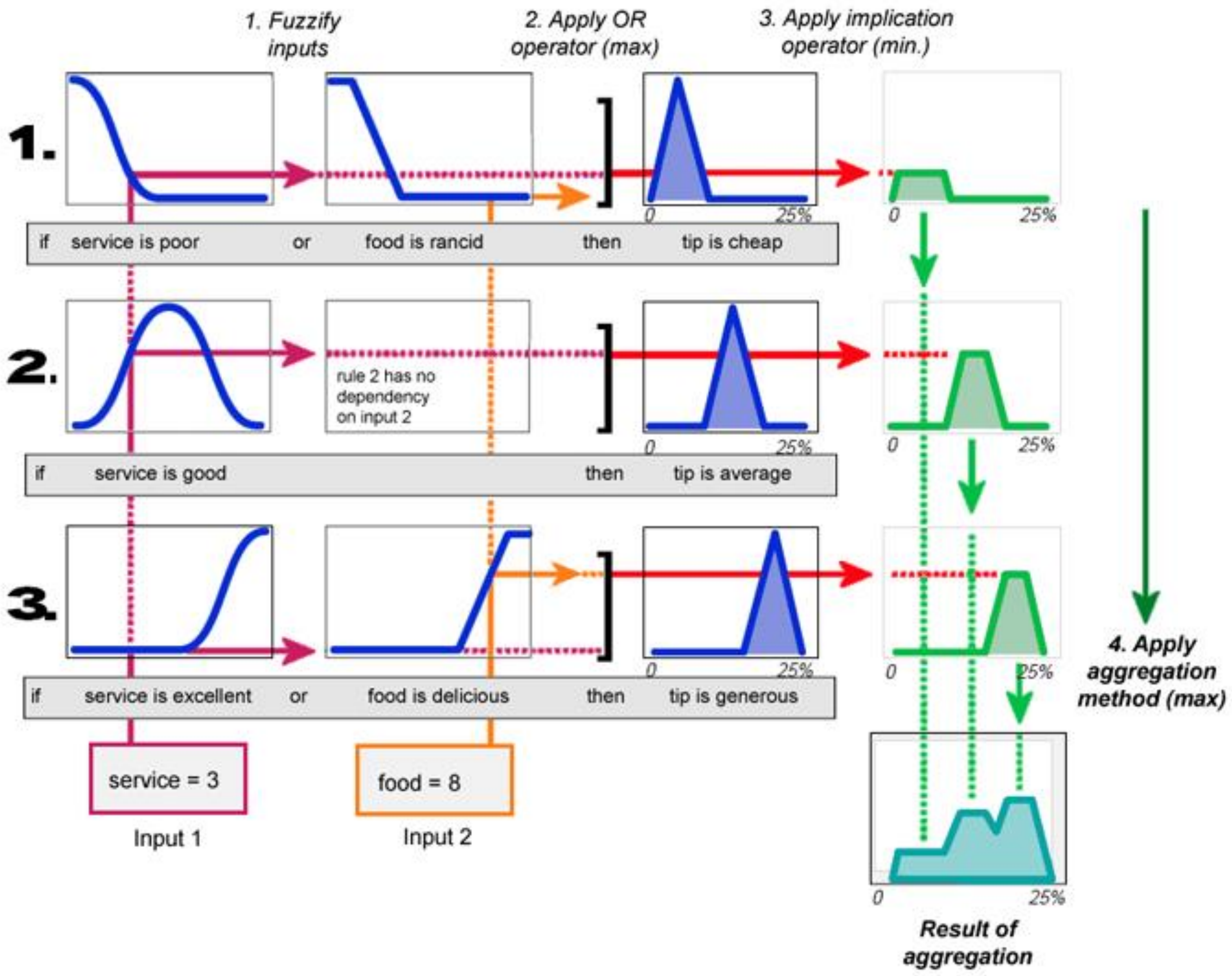
3-The most common ways to modify the output fuzzy set are truncation using the min function (where the fuzzy set is "chopped off" as shown)

- Another ways is using scaling using the prod function (where the output fuzzy set is "squashed"). Both are supported by the Fuzzy Logic Toolbox, but we use truncation for the examples in this section.



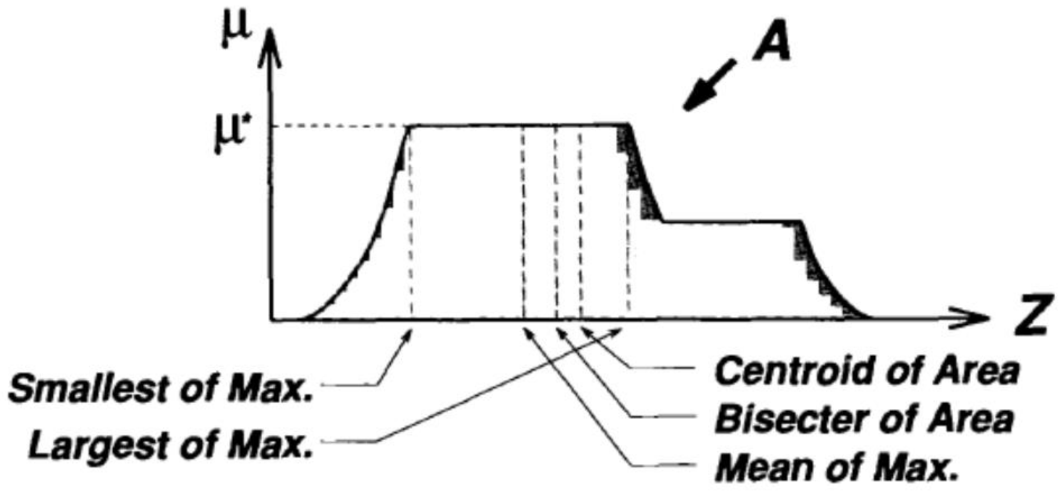
Fuzzy Inference – Fuzzy Reasoning

- Food Example



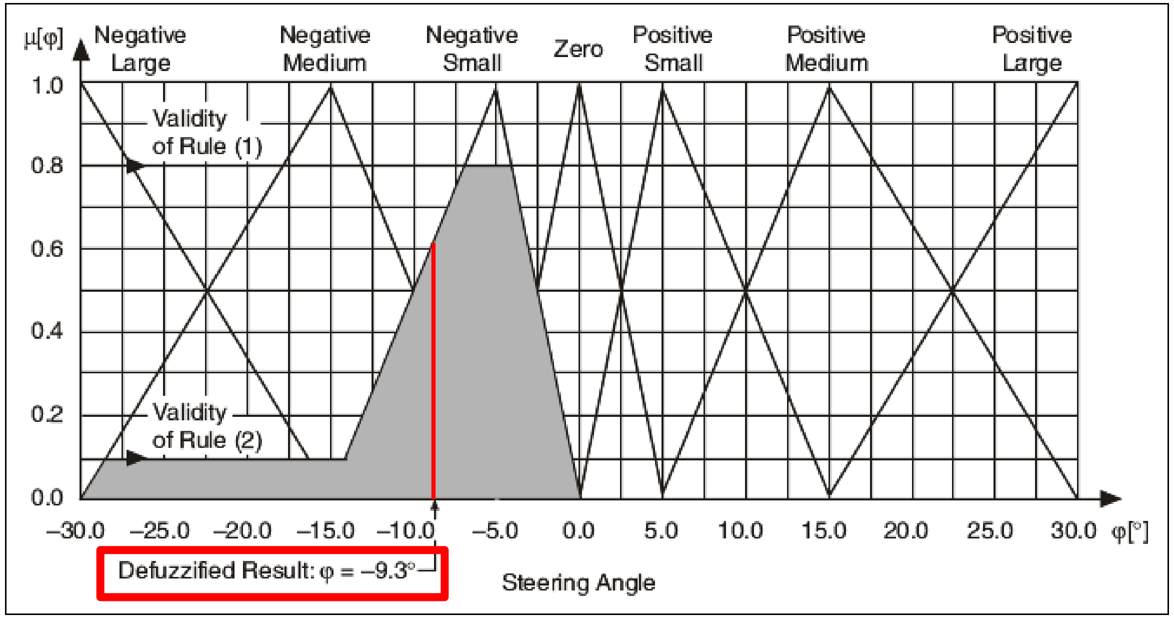
Defuzzification

- Defuzzification refers to the way a crisp value is extracted from a fuzzy set as a representative value.
- In general, there are five methods for defuzzifying a fuzzy set A of a universe of discourse Z.



Defuzzification – Centroid of Area

- In the Center of Area (CoA) defuzzification method, also called the Center of Gravity (CoG) methodthe fuzzy controller first calculates the area under the scaled membership functions and within the range of the output variable.
- The fuzzy logic controller then uses the equation to calculate the geometric center of this area.

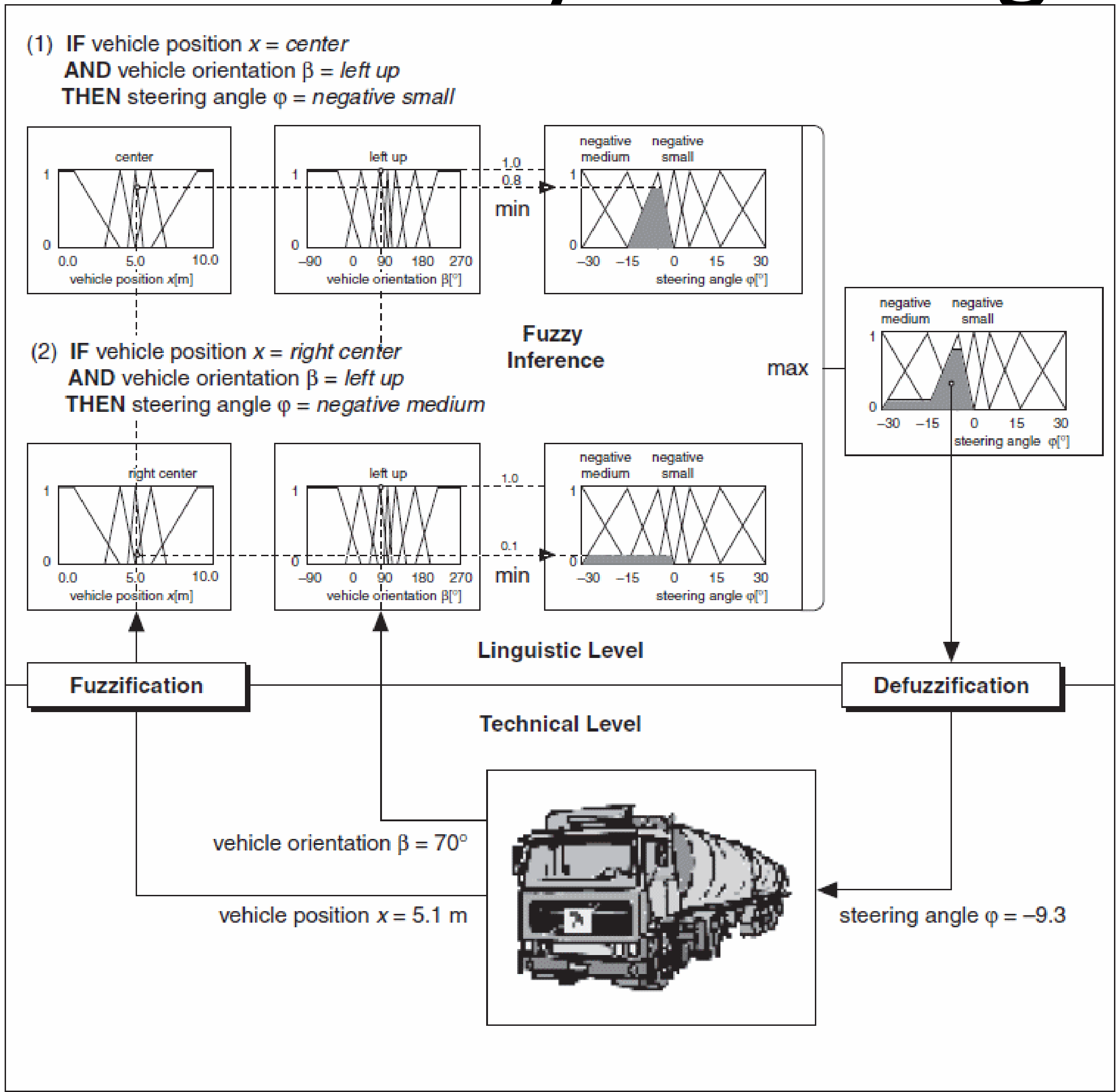


$$CoA = \frac{\int_{x_{min}}^{x_{max}} f(x) * x \, dx}{\int_{x_{min}}^{x_{max}} f(x) dx}$$

- CoA is the center of area
- x is the value of the linguistic variable
- x_{min} and x_{max} represent the range of the linguistic variable.

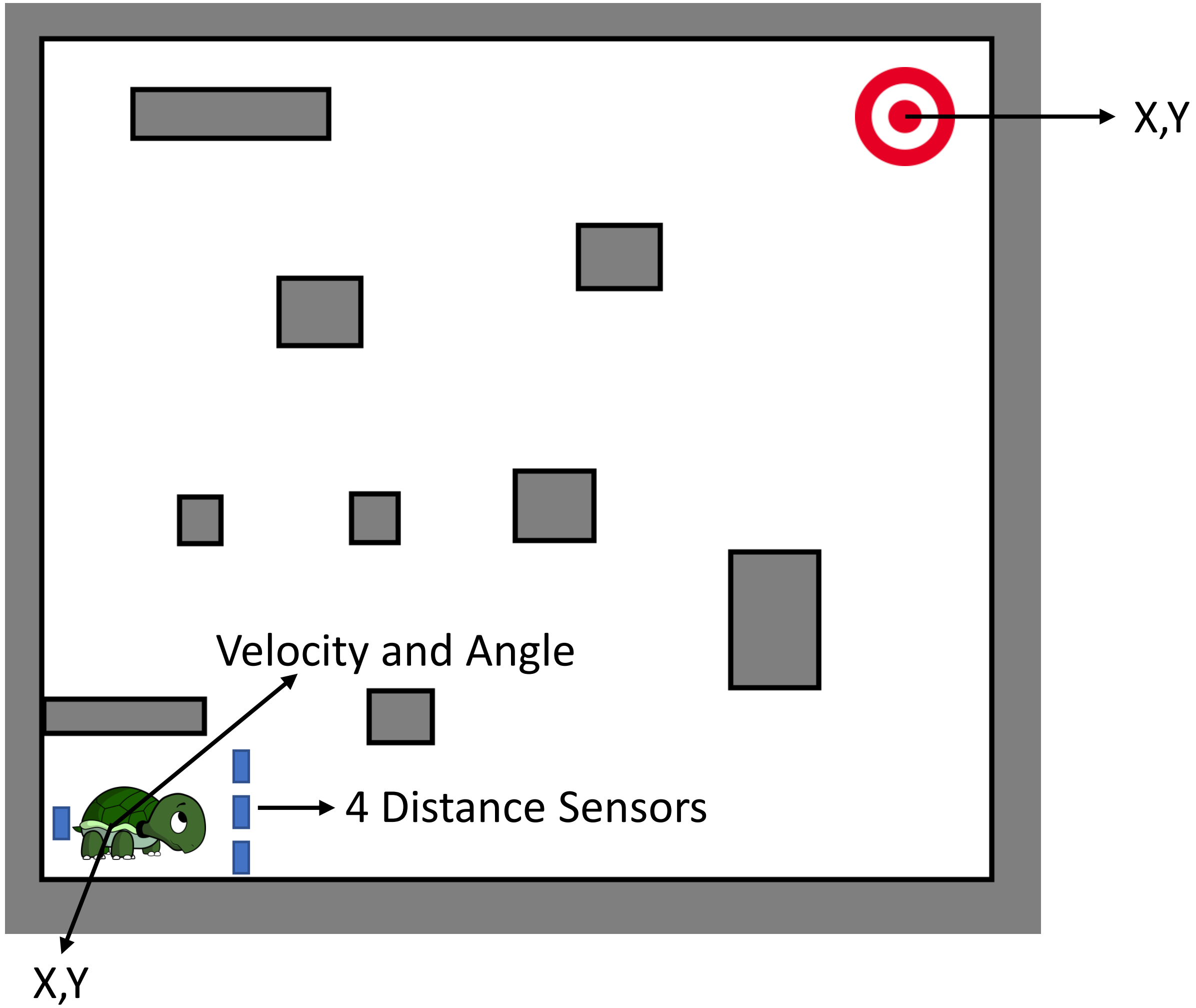
Fuzzy Inference – Fuzzy Reasoning

- Car Example



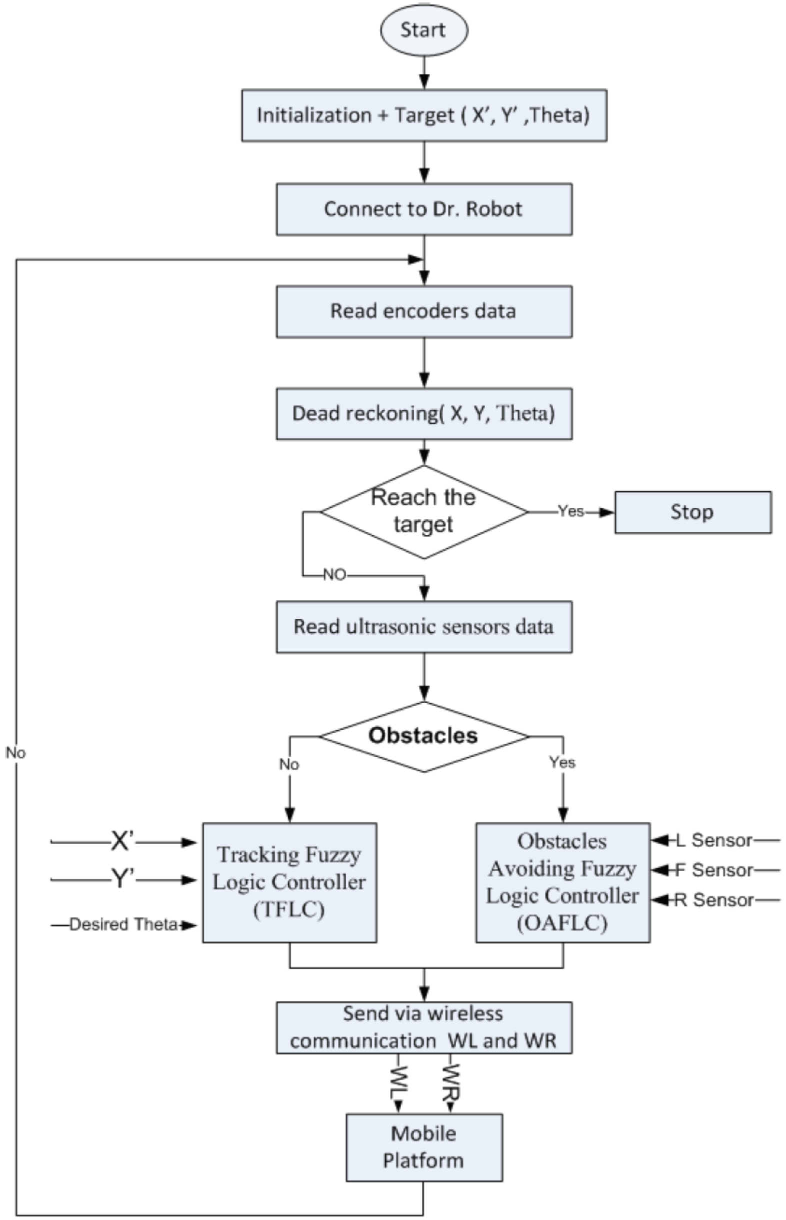
Fuzzy Logic for Obstacle Avoidance

- Obstacle Avoidance Example



Fuzzy Logic for Obstacle Avoidance

Obstacle Avoidance Example

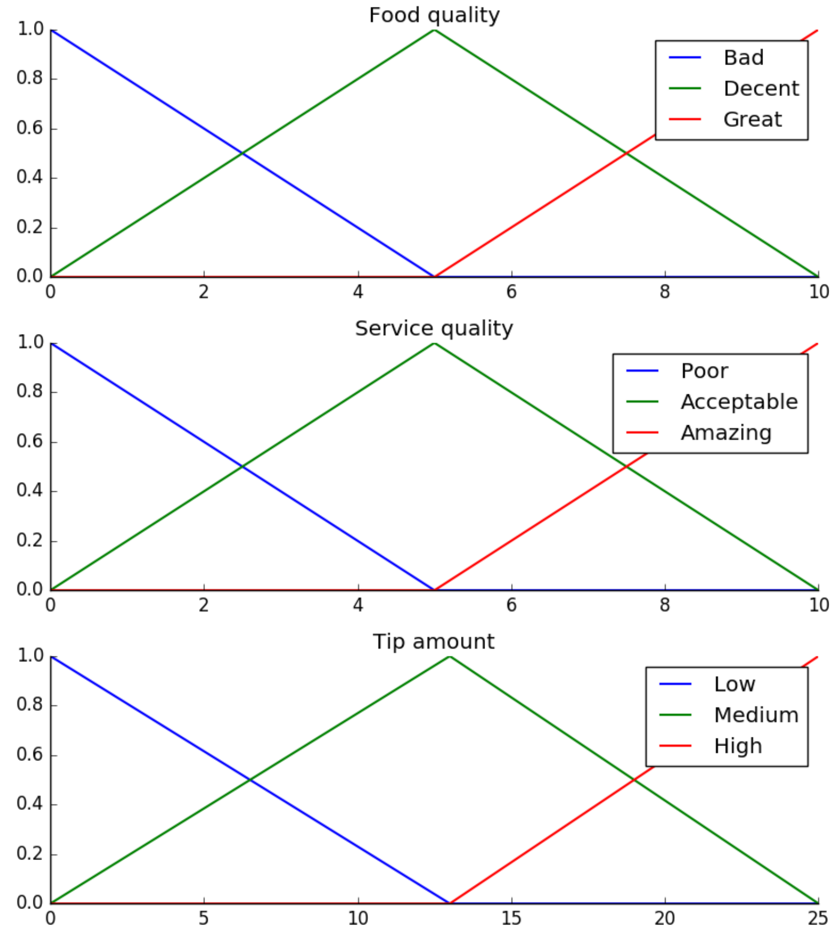


Scikit-Fuzzy

- Scikit-Fuzzy is a collection of fuzzy logic algorithms intended for use in the SciPy Stack, written in the Python computing language.
- You can easily use create a fuzzy controller in a few lines of code
- There are ready to use methods to create fuzzy sets, membership functions, and fuzzy rules
- It comes with many different fuzzy logic options
- https://pythonhosted.org/scikit-fuzzy/api/api.html

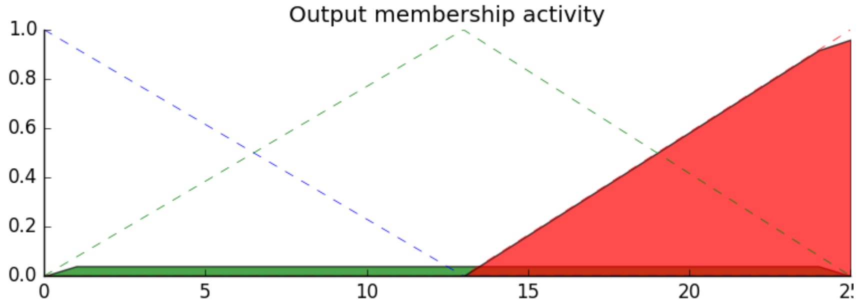
Scikit-Fuzzy – Method 1

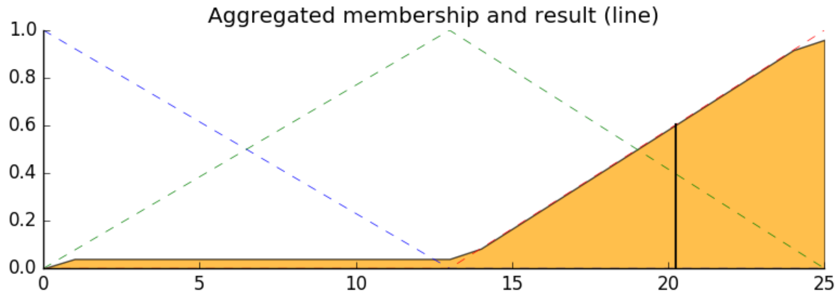
```
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt

# Generate universe variables - Quality  [0, 10] - Tip [0, 25] - units of percentage points
x_qual = np.arange(0, 11, 1)
x_serv = np.arange(0, 11, 1)
x_tip  = np.arange(0, 26, 1)

# Generate fuzzy membership functions Input
qual_lo = fuzz.trimf(x_qual, [0, 0, 5])
qual_md = fuzz.trimf(x_qual, [0, 5, 10])
qual_hi = fuzz.trimf(x_qual, [5, 10, 10])
serv_lo = fuzz.trimf(x_serv, [0, 0, 5])
serv_md = fuzz.trimf(x_serv, [0, 5, 10])
serv_hi = fuzz.trimf(x_serv, [5, 10, 10])

# Generate fuzzy membership functions Output
tip_lo = fuzz.trimf(x_tip, [0, 0, 13])
tip_md = fuzz.trimf(x_tip, [0, 13, 25])
tip_hi = fuzz.trimf(x_tip, [13, 25, 25])

# Activation of our fuzzy membership functions at these values.
qual_level_lo = fuzz.interp_membership(x_qual, qual_lo, 6.5)
qual_level_md = fuzz.interp_membership(x_qual, qual_md, 6.5)
qual_level_hi = fuzz.interp_membership(x_qual, qual_hi, 6.5)

serv_level_lo = fuzz.interp_membership(x_serv, serv_lo, 9.8)
serv_level_md = fuzz.interp_membership(x_serv, serv_md, 9.8)
serv_level_hi = fuzz.interp_membership(x_serv, serv_hi, 9.8)
```



Scikit-Fuzzy – Method 1

```
# Create rules and apply them.
# Rule 1 : bad food OR service = low tip.
active_rule1 = np.fmax(qual_level_lo, serv_level_lo)

# Apply this by clipping the top off the corresponding output membership function with `np.fmin`
tip_activation_lo = np.fmin(active_rule1, tip_lo)

# For rule 2
# Acceptable service = medium tipping
# Apply this by clipping the top off the corresponding output membership function with `np.fmin`
tip_activation_md = np.fmin(serv_level_md, tip_md)

# For rule 3
# High service OR high food = high tipping
active_rule3 = np.fmax(qual_level_hi, serv_level_hi)

# Apply this by clipping the top off the corresponding output membership function with `np.fmin`
tip_activation_hi = np.fmin(active_rule3, tip_hi)

# Aggregate all three output membership functions together
aggregated = np.fmax(tip_activation_lo, np.fmax(tip_activation_md, tip_activation_hi))

# Calculate defuzzified result
tip = fuzz.defuzz(x_tip, aggregated, 'centroid')

print tip

input("Press Enter to continue...")
```





Portland State UNIVERSITY

Scikit-Fuzzy – Method 2

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Antecedent/Consequent objects hold universe variables and membership functions
quality = ctrl.Antecedent(np.arange(0, 11, 1), 'quality')
service = ctrl.Antecedent(np.arange(0, 11, 1), 'service')
tip = ctrl.Consequent(np.arange(0, 26, 1), 'tip')

# Auto-membership function population is possible with .automf(3, 5, or 7)
quality.automf(5)
service.automf(5)

# Custom membership functions
tip['low'] = fuzz.trimf(tip.universe, [0, 0, 13])
tip['medium'] = fuzz.trimf(tip.universe, [0, 13, 25])
tip['high'] = fuzz.trimf(tip.universe, [13, 25, 25])

# View membership functions

rule1 = ctrl.Rule(quality['poor'] | service['poor'], tip['low'])
rule2 = ctrl.Rule(service['average'], tip['medium'])
rule3 = ctrl.Rule(service['good'] | quality['good'], tip['high'])

rule1.view()
#rule2.view()
#rule3.view()

tipping_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
tipping = ctrl.ControlSystemSimulation(tipping_ctrl)

# Pass inputs to the ControlSystem using Antecedent labels with Pythonic API
# Note: if you like passing many inputs all at once, use .inputs(dict_of_data)
tipping.input['quality'] = 6.5
tipping.input['service'] = 9.8

# Calculate the result
tipping.compute()

print tipping.output['tip']
tip.view(sim=tipping)

input("Press Enter to continue...")
```

Reference

- http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/main_en.htm
- http://researchhubs.com/post/engineering/fuzzy-system/mamdani-fuzzy-model.html
- https://slideplayer.com/slide/4598308/
- https://edoras.sdsu.edu/doc/matlab/toolbox/fuzzy/fuzzytu5.html
- https://www.physik.uzh.ch/local/teaching/SPI301/LV-2015-Help/lvpidmain.chm/center_of_area.html

Portland State UNIVERSITY