5. Fuzzy Logic

- Introduced by Lotfi Zadeh, UC Berkeley
- Process data allowed **partial set membership** instead of “crisp membership”
- Deals with noisy, imprecise,vague, ambiguous data
- **Higher reliability**
- “People also do not require precise numerical input”
- *These slides are based on “Fuzzy Logic Tutorial” by Encoder Newsletter of the Seattle Robotics Society*

Fuzzy Logic

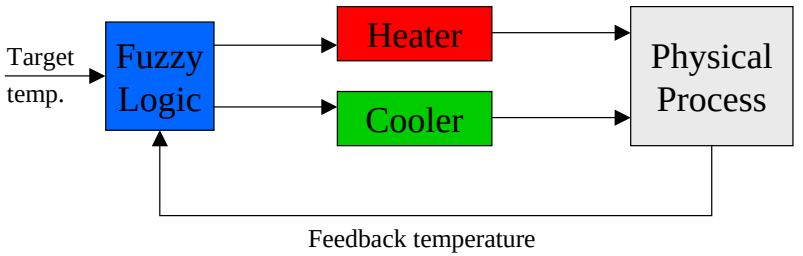
- Empirical model
- Simple rule-based approach: IF x THEN y
  Here no numbers but fuzzy set memberships, e.g.:
  IF (temp is too_cool) AND (time is progressed) THEN (cool fast)
- Imprecise, but descriptive
- Mimic human control logic
- **Robust**, requires little tuning
- Can model **non-linear systems** that are mathematically very difficult (or impossible) to model

5.1 Fuzzy Logic Guide

- Consider inputs, outputs, system control, failure modes
- Determine input/output relationships
- Use minimum number of input variables, e.g.
  - **error** (control difference)
  - **change-of-error** (error derivative)
- Set up system as a number of IF-THEN rules
- Create membership functions, giving meaning to input/output terms
- Create pre-/post-processing terms
- Test system, evaluate, tune, and test again

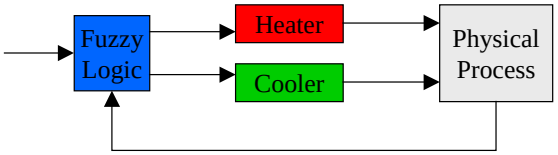
Fuzzy Logic Example

Fuzzy logic temperature control of a physical process



source: Encoder, Seattle Robotics Soc.

Fuzzy Logic Example



- Target input: temperature
- Error feedback: negative, zero, positive
- Error-dot feedback: negative, zero, positive
- Fuzzy logic output: heat, cool, or “do nothing”

Linguistic Variables

Introduction of linguistic variables:

- Input 1: “**error**”: positive (P), zero (Z), negative (N)
- Input 2: “**error-dot**”: positive (P), zero (Z), negative (N)
- Output: heat (H), do_nothing (–), cool (C)

Error = target – feedback
(P = too cold, Z = just right, N = too hot)

ErrorDot = d Error / dt
(P = getting colder, Z = no change, N = getting hotter)

Output H = activate heater, “–” = do nothing, C = activate cooler

5.2 Linguistic Rules

Set up linguistic rules:

IF-THEN (antecedent and consequent blocks)

Example:

IF too_hot AND getting_hotter THEN cool

target-feedback=N d(target-feedback)/dt=N output = C

Linguistic Rules

1. IF too_hot AND getting_colder THEN ??
2. IF just_right AND getting_colder THEN ??
3. IF too_cold AND getting_colder THEN ??

4. IF too_hot AND no_change THEN ??
5. IF just_right AND no_change THEN ??
6. IF too_cold AND no_change THEN ??

7. IF too_hot AND getting_hotter THEN ??
8. IF just_right AND getting_hotter THEN ??
9. IF too_ cold AND getting_hotter THEN ??

Linguistic Rules

1. IF too_hot AND getting_colder THEN cool
2. IF just_right AND getting_colder THEN heat
3. IF too_cold AND getting_colder THEN heat
4. IF too_hot AND no_change THEN cool
5. IF just_right AND no_change THEN do_nothing
6. IF too_cold AND no_change THEN heat
7. IF too_hot AND getting_hotter THEN cool
8. IF just_right AND getting_hotter THEN cool
9. IF too_ cold AND getting_hotter THEN heat

Rule Matrix

Build rule matrix from linguistic rules

	too cold	**just right**	**too hot**
getting colder	cool	heat	heat
no change	cool	do nothing	heat
getting hotter	cool	cool	heat

source: Encoder, Seattle Robotics Soc.

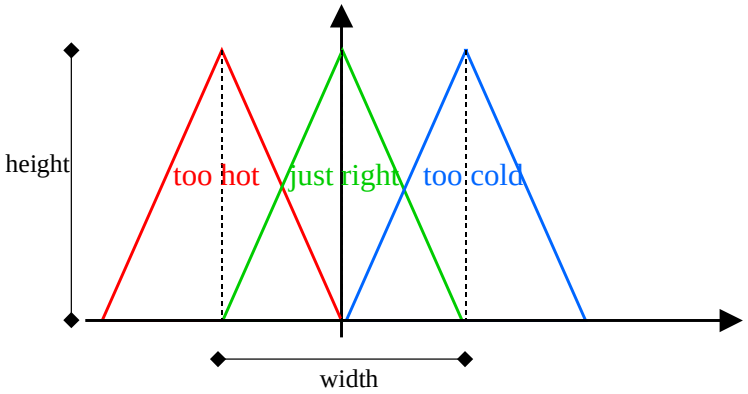
5.3 Membership Functions

- Membership function is a graphical representation of "magnitude of participation" of each input
- Associates weighting with each input
- Defines overlaps between inputs
- Determines output response
- **Fuzzification:** Translating input values to fuzzy set memberships
- **Defuzzification:** Translating fuzzy output back to crisp system output (e.g. a number)

Membership Functions

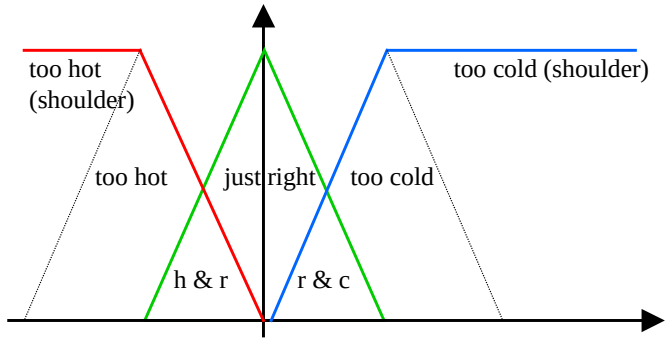
- **Shape** of membership function is usually triangle (but could also be trapezoidal or other)
- **Height** usually normalized to 1 (so in total: [0..1])
- **Width** of the base of function depends on number of functions
- **Shouldering** is used to lock height in an outer function (leftmost and rightmost functions evaluate to 1 to the outsides)
- **Center** points of functions evaluate to 1
- **Overlap** is usually 50% at base of function

Membership Functions



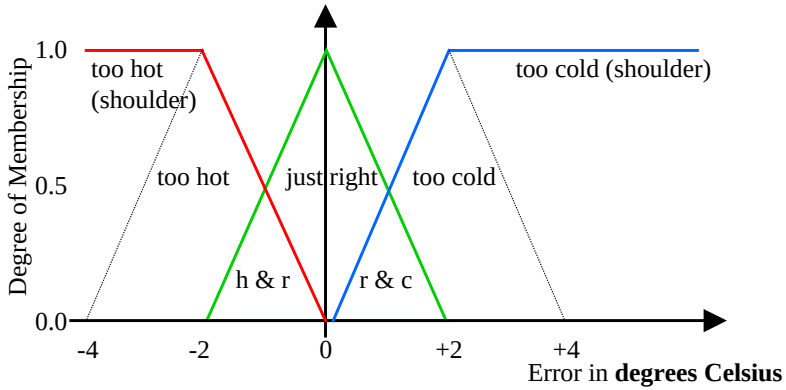
Membership Functions

- Add shoulders



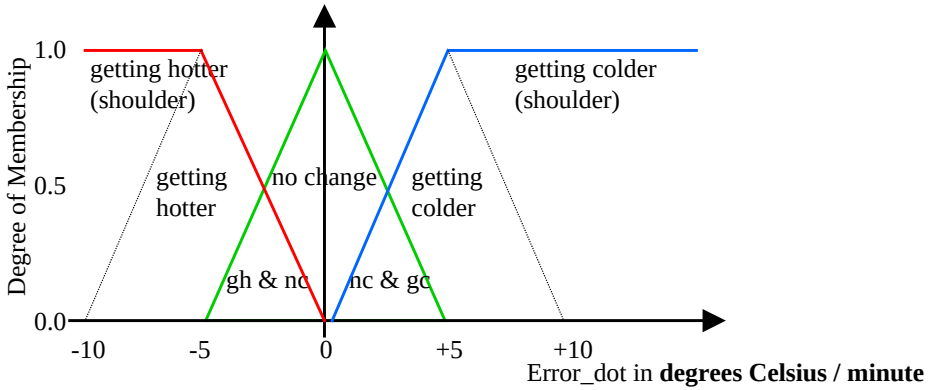
Membership Functions

- Add values, here: for **error** (temperature difference in deg. Celsius)

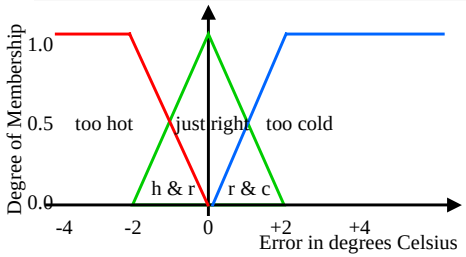


Membership Functions

- Add values, here for **error_dot** (temperature change in deg. Celsius / minute)



Membership Functions

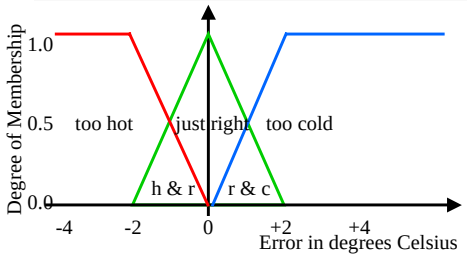


Example

- Error = –1

⇒ Membership(too_hot) = 0.5
⇒ Membership(just_right) = 0.5
⇒ Membership(too_cold) = 0.0

Membership Functions



Example

- Error = +5

⇒ Membership(too_hot) = 0.0
⇒ Membership(just_right) = 0.0
⇒ Membership(too_cold) = 1.0

Membership Functions

Note

- Sum of all membership values for a certain input value is always 1.0
- Different value range for different input variables (e.g. error and error_dot)
- Membership values (antecedents) are evaluated for the produced conclusion (consequent)

5.4 Inferences

- Deviation from Boolean logic (true, false or 1, 0)
- Now we have continuous membership values [0..1]
- Build **logical sums** using membership values in combination with rule matrix
- Use **minimum** of all antecedent values (AND-part) for output

Inferences

Example:

error = –1

error_dot = –2.5

Membership error:

too_hot = 0.5, just_right = 0.5, too_cold = 0.0

Membership error_dot

getting_hotter = 0.5, no_change = 0.5, getting_colder = 0.0

Inferences

- too_hot = **0.5**, just_right = **0.5**, too_cold = 0.0
- getting_hotter = **0.5**, no_change = **0.5**, getting_colder = 0.0

1. IF too_hot AND getting_colder THEN cool
2. IF just_right AND getting_ colder THEN heat
3. IF too_cold AND getting_ colder THEN heat
4. IF too_hot AND no_change THEN cool
5. IF just_right AND no_change THEN do_nothing
6. IF too_cold AND no_change THEN heat
7. IF too_hot AND getting_hotter THEN cool
8. IF just_right AND getting_hotter THEN cool
9. IF too_ cold AND getting_hotter THEN heat

Inferences

- too_hot = **0.5**, just_right = **0.5**, too_cold = 0.0
- getting_hotter = **0.5**, no_change = **0.5**, getting_colder = 0.0

use minimum

1. IF **too_hot** AND getting_colder THEN cool 0.5 & 0.0 = 0.0
2. IF **just_right** AND getting_ colder THEN heat 0.5 & 0.0 = 0.0
3. IF too_cold AND getting_ colder THEN heat 0.0 & 0.0 = 0.0
4. IF **too_hot** AND **no_change** THEN cool 0.5 & 0.5 = 0.5
5. IF **just_right** AND **no_change** THEN do_nothing 0.5 & 0.5 = 0.5
6. IF too_cold AND **no_change** THEN heat 0.0 & 0.5 = 0.0
7. IF **too_hot** AND **getting_hotter** THEN cool 0.5 & 0.5 = 0.5
8. IF **just_right** AND **getting_hotter** THEN cool 0.5 & 0.5 = 0.5
9. IF too_ cold AND **getting_hotter** THEN heat 0.0 & 0.5 = 0.0

Inferences

- too_hot = **0.5**, just_right = **0.5**, too_cold = 0.0
- getting_hotter = 0.0, no_change = **0.5**, getting_colder = **0.5**

use minimum

1. IF **too_hot** AND getting_colder THEN cool
2. IF **just_right** AND getting_ colder THEN heat
3. IF too_cold AND getting_ colder THEN heat
4. IF **too_hot** AND **no_change** THEN cool 0.5 & 0.5 = 0.5
5. IF **just_right** AND **no_change** THEN do_nothing 0.5 & 0.5 = 0.5
6. IF too_cold AND **no_change** THEN heat
7. IF **too_hot** AND **getting_hotter** THEN cool 0.5 & 0.5 = 0.5
8. IF **just_right** AND **getting_hotter** THEN cool 0.5 & 0.5 = 0.5
9. IF too_ cold AND **getting_hotter** THEN heat

5.5 Output Combination

- Find out “firing strength” of each rule
- Combine logical outputs for each rule must be combined before defuzzification
- Combining rule outputs
  - MAX-MIN
  - MAX-DOT (MAX-PRODUCT)
  - AVERAGING
  - ROOT-SUM-SQUARE

Output Combination

MAX-MIN

- Test magnitude of all rules – select highest one
  Use horizontal coordinate of “fuzzy centroid” as output
  Note: This method does not combine individual results!

MAX-DOT

- Scale each member function to fit under its peak value, take the horizontal coordinate of the fuzzy centroid of the composite area as output.
  This shrinks all member functions to their peaks equal the magnitude of the respective function.
  Note: This method prodruces a smooth, continuous output combining all active rules

Output Combination

AVERAGING

- Each function is clipped at the average value and the fuzzy centroid of the composite area is computer
  Note: This method does not give increased weight if multiple rules generate the same output member!

ROOT-SUM-SQUARE

- Combination of several approaches:
  Scale functions to respective magnitudes with root of sum of squares, compute fuzzy centroid of composite area.
  Note: This method gives a good weighted influence to all firing rules.

Output Combination

1.	IF too_hot AND getting_colder THEN **cool**	**0**
2.	IF just_right AND getting_ colder THEN heat	0
3.	IF too_cold AND getting_ colder THEN heat	0
4.	IF too_hot AND no_change THEN **cool**	**0.5**
5.	IF just_right AND no_change THEN do_nothing	0.5
6.	IF too_cold AND no_change THEN heat	0
7.	IF too_hot AND getting_hotter THEN **cool**	**0,5**
8.	IF just_right AND getting_hotter THEN **cool**	**0.5**
9.	IF too_ cold AND getting_hotter THEN heat	0

Output membership functions:

“**cool**” $= \sqrt{(rule1^2 + rule4^2 + rule7^2 + rule8^2)}$

$= \sqrt{(0 + 0.5^2 + 0.5^2 + 0.5^2)}$

$= \sqrt{0.75}$

$= \mathbf{0.866}$

Output Combination

1. IF too_hot AND getting_colder THEN cool 0
2. IF just_right AND getting_ colder THEN heat 0
3. IF too_cold AND getting_ colder THEN heat 0

4. IF too_hot AND no_change THEN cool 0.5
5. IF just_right AND no_change THEN **do_nothing** **0.5**
6. IF too_cold AND no_change THEN heat 0

7. IF too_hot AND getting_hotter THEN cool 0,5
8. IF just_right AND getting_hotter THEN cool 0.5
9. IF too_ cold AND getting_hotter THEN heat 0

Output membership functions:

"**do_nothing**" = $\sqrt{(\text{rule}5^2)}$
= $\sqrt{(0.5^2)}$
= **0.5**

Output Combination

1. IF too_hot AND getting_colder THEN cool 0
2. IF just_right AND getting_ colder THEN **heat** **0**
3. IF too_cold AND getting_ colder THEN **heat** **0**

4. IF too_hot AND no_change THEN cool 0.5
5. IF just_right AND no_change THEN do_nothing 0.5
6. IF too_cold AND no_change THEN **heat** **0**

7. IF too_hot AND getting_hotter THEN cool 0,5
8. IF just_right AND getting_hotter THEN cool 0.5
9. IF too_ cold AND getting_hotter THEN **heat** **0**

Output membership functions:

"**heat**" = $\sqrt{(\text{rule}2^2 + \text{rule}3^2 + \text{rule}6^2 + \text{rule}9^2)}$
= $\sqrt{(0 + 0 + 0 + 0)}$
= **0.0**

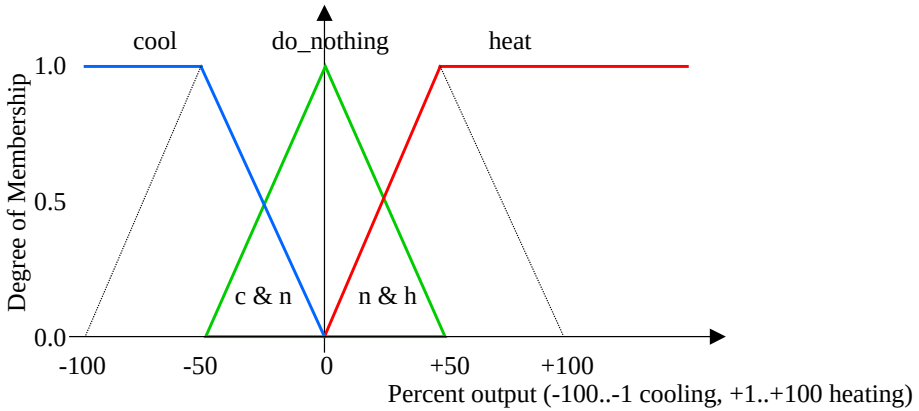
5.6 Defuzzification

- Defuzzification of combined results of inference process
- Result will be a crisp (numeric) output
- Calculation of "fuzzy centroid":

1. Multiply the weighted strength of each output member function by the respective member function center points
2. Add these values
3. Divide area by the sum of the weighted member function strength

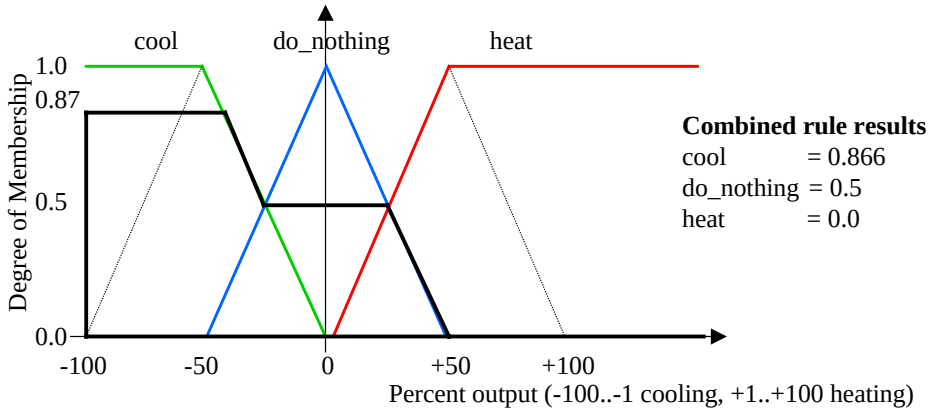
Defuzzification

- Output Membership Function (action to be taken)



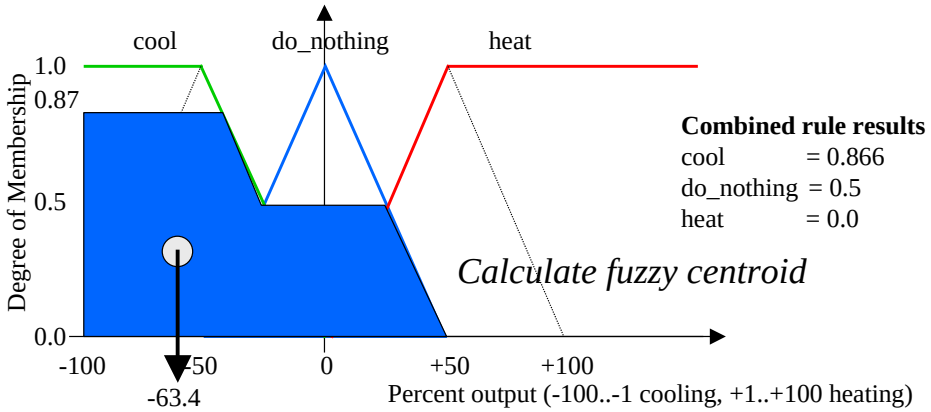
Defuzzification

- Output Membership Function (action to be taken)



Defuzzification

- Output Membership Function (action to be taken)



Defuzzification

- Input:
  - error = –1
  - error_dot = –2.5
- Output:
  - activate cooling at 63.4%

Defuzzification

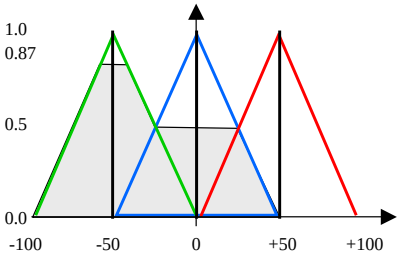
- How to calculate centroid
  - graphically: intuitively
  - numerically: see formula below

$$Output = \frac{\sum_{i=1}^{N} (center_i \cdot strength_i)}{\sum_{i=1}^{N} strength_i}$$

"Moment"

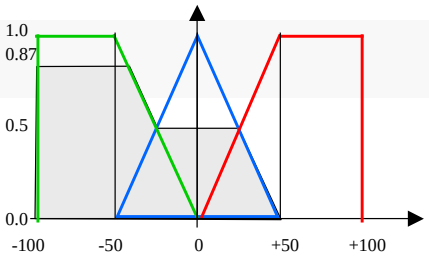
N is number of output members (here: 3)

Defuzzification

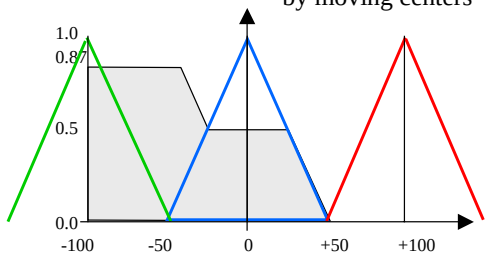


without shoulders: simple formula

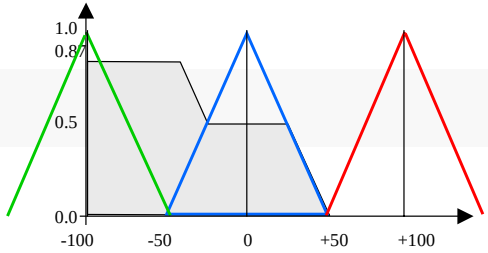
$$Output = \frac{\sum_{i=1}^{N} (center_i \cdot strength_i)}{\sum_{i=1}^{N} strength_i}$$



with shoulders: adjust formula by moving centers



Defuzzification



$$Output = \frac{center_{cool} \cdot strength_{cool} + center_{nothing} \cdot strength_{nothing} + center_{heat} \cdot strength_{heat}}{strength_{cool} + strength_{nothing} + strength_{heat}}$$

$$Output = \frac{-100 \cdot 0.866 + 0 \cdot 0.5 + 100 \cdot 0}{0.866 + 0.5 + 0.0} = -63.4$$

Result: Activate cooling at 63.4%

5.7 Tuning

Tuning of a fuzzy system can be accomplished by:

- Changing rule antecedents
- Changing rule conclusions
- Changing centers of the input/output membership functions
- Adding additional membership functions
  e.g. negative, low_neg, zero, low_pos, positive
  → additional rules

5.8 Applications

Fuzzy control can be applied to a number of control problems for example:

- Motor control
- Driving
- Balancing
- Walking
- …