G54ARS
Autonomous Robotic Systems
Lecture 4

Fuzzy Logic Control

Last week – PID Control and DGC

- Control
  - The problem
  - Open-Loop Control
- PID Control
  - PID principles
  - PID parameter effects
  - PID tuning
  - Live example – PID DC motor control
  - PID implementation
- Video - The DARPA Grand Challenge

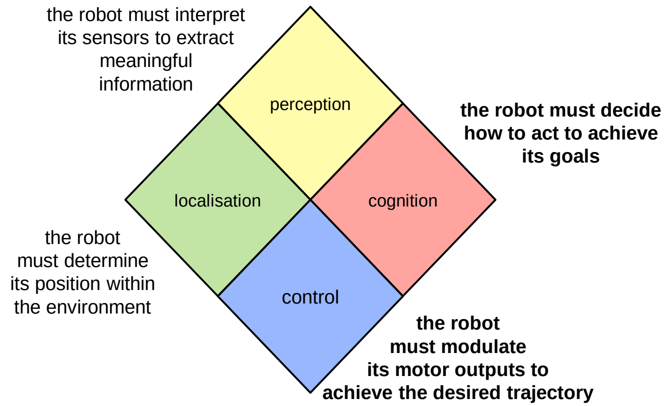
Kahoot!

https://kahoot.it/

This week…

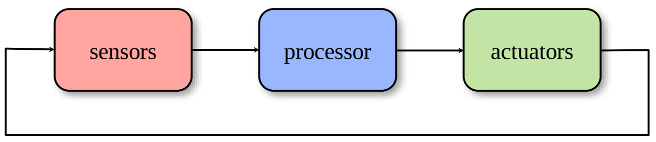
- Fuzzy Logic Control
  - Fuzzy Logic Control Principles
  - Origins and History
  - Fuzzy Sets and FLC components
  - FLC control examples
  - FLC design and tuning
  - FLC implementation notes

Autonomous Mobile Robots - Navigation



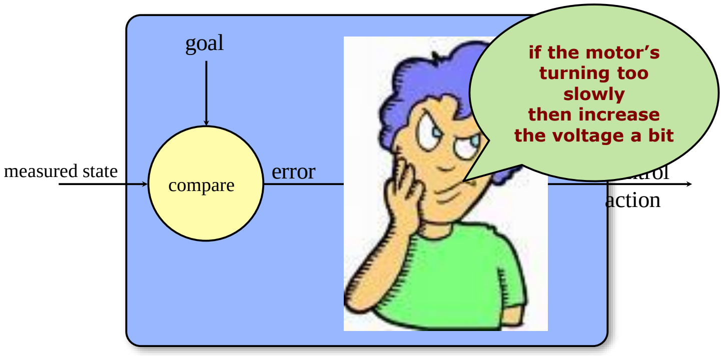
Principles of Fuzzy Logic Control

The Sense-Think-Act Cycle



- Repeat
  - sense the current state
  - reduce difference between current state and goal state
- Until
  - current state = goal state

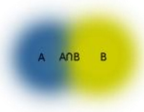
The Fuzzy Approach

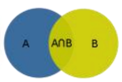


Fuzzy Control

- Fuzzy control is a methodology built on the framework of ‘fuzzy logic’ and ‘fuzzy sets’
- Fuzzy Logic is an extension of classical logic:
  - ‘conventional’, formal logic
    - false (F) and true (T)
  - Boolean logic
    - zero (0) and one (1)
  - fuzzy logic
    - ‘completely false’, ‘partially true’, ‘true-ish’, etc. (any value between 0 and 1)
- Fuzzy Sets support partial membership

Fuzzy Sets:
Things can be part of one or more sets to a degree (e.g. $\mu_A(x) = 0.8$)





Crisp Sets:
Something is either part or not part of a specific set (e.g. $\mu_A(x) = 0$ OR $\mu_A(x) = 1$)

Fuzzy Control – Some History

– Fuzzy Logic introduced by Lotfi Zadeh in 1965 through introduction of Fuzzy Sets.
  - Zadeh, L.A. (1965). "Fuzzy sets". *Information and Control* 8 (3): 338–353.

– First applications in control appeared quickly:
  - 1975: Cement Kiln in Denmark
  - 1985: Sendai Railway in Japan (braking, acceleration)

– Traditionally, Fuzzy Logic based applications have been embraced in Asia – less so in the West.

– However, today there are a vast number of applications in control and beyond.

VW DSG gearbox

Panasonic FL rice cooker

Samsung FL washing machine

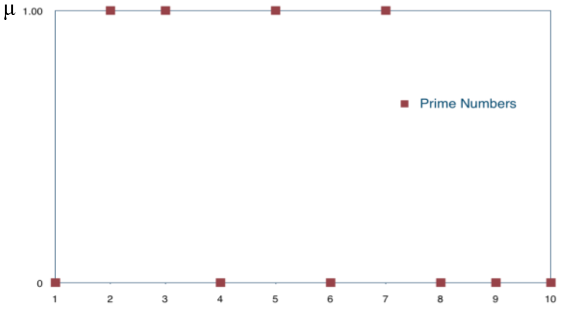
CNC milling

…and of course - robots

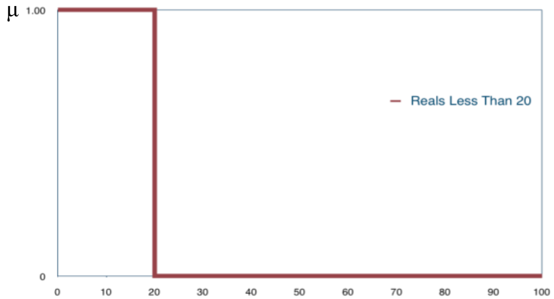
One step back - What is Fuzzy Logic?

A brief primer on fuzzy sets.

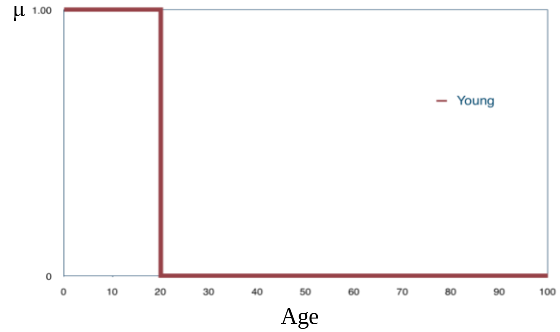
Crisp Sets



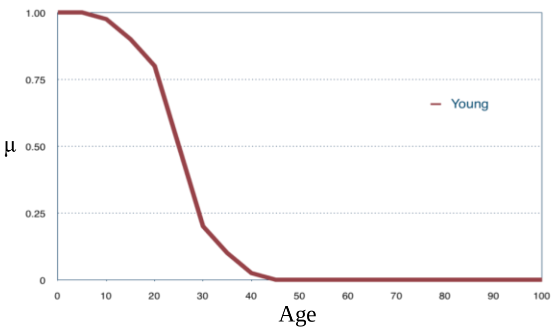
Crisp Sets



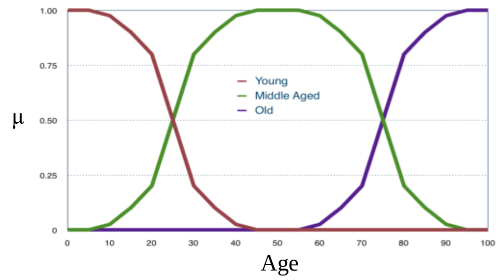
Crisp Sets?



Fuzzy Sets



Linguistic Variables



A fuzzy set provides a relationship between an element and the element's grade of membership in that set

i.e. the *degree of truth* of the notion that the element belongs to the corresponding set

Fuzzy Logic Operations

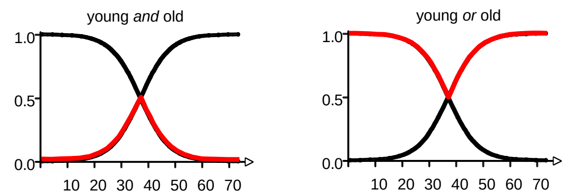
- If A and B are fuzzy sets with membership functions μ_A and μ_B, defined over a base variable x

not A = $1 - \mu_A(x)$, $\forall x$

A ∩ B = $\min(\mu_A(x), \mu_B(x))$, $\forall x$ *(intersection - AND)*

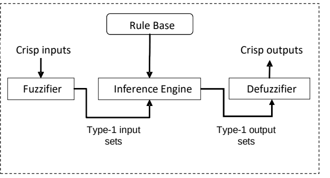
A ∪ B = $\max(\mu_A(x), \mu_B(x))$, $\forall x$ *(union - OR)*

Note: more generally, ∩ and ∪ are t-Norms, resp. t-Conorms; we use min and max.



Fuzzy Inference for FLCs

- Fuzzy Inference is the process underlying fuzzy logic controllers (FLCs).
- The inference process relates the input of an FLC to its output based on logical **rules**.
- There are different types of inference in FLCs, the most common being Mamdani and Takagi-Sugeno-Kang (TSK).
- Diagram of a Mamdani FLC:



Fuzzy Inference

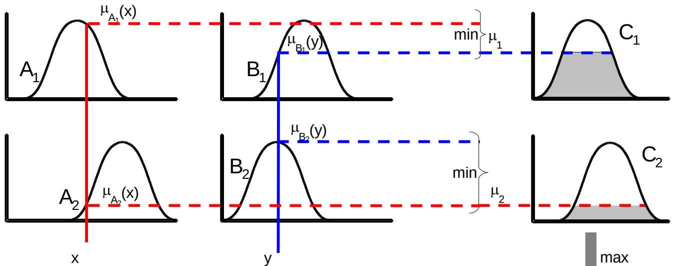
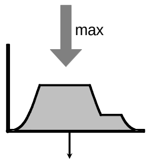


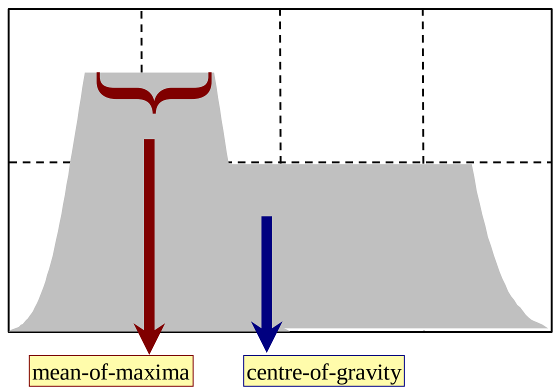
illustration of Mamdani style fuzzy inference in a two rule system:

IF x is A_1 AND y is B_1 THEN C_1
IF x is A_2 AND y is B_2 THEN C_2



defuzzify solution

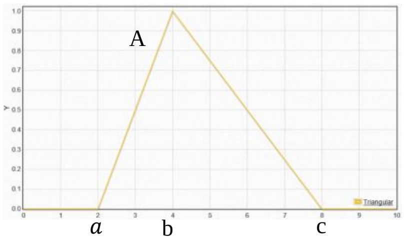
Defuzzification



Extracting Degree of Membership

- Triangular Membership Function
- Three parameters (a, b, c).

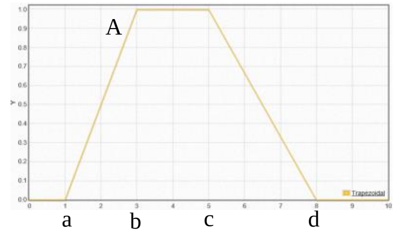
$$\mu_A(x) = \begin{cases} 0, & x \le a \\ 0, & x \ge c \\ \dfrac{x-a}{b-a}, & x \epsilon [a,b] \\ \dfrac{c-x}{c-b}, & x \epsilon [b,c] \end{cases}$$



Extracting Degree of Membership ctd.

- Trapezoidal Membership Function
- Four parameters (a,b,c,d)

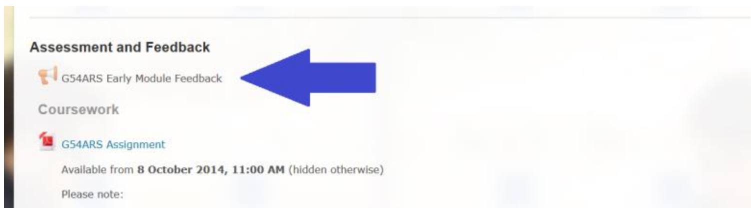
$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, x \epsilon [a,b] \\ 1, x \epsilon [b,c] \\ \dfrac{d-x}{d-c}, x \epsilon [c,d] \\ 0, otherwise \end{cases}$$



- Of course, many other membership functions exist, e.g., Gaussian, Bell, etc.

Early Module Feedback

- School process
- Purpose: Gather student feedback to allow module adaptation <u>during</u> the year
- Access via the G54ARS Moodle page

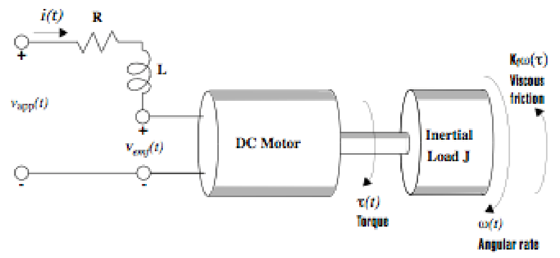


Fuzzy Control Examples

Example 1
DC Motor Control
(similar objective to PID in Lecture 3)

DC Motor

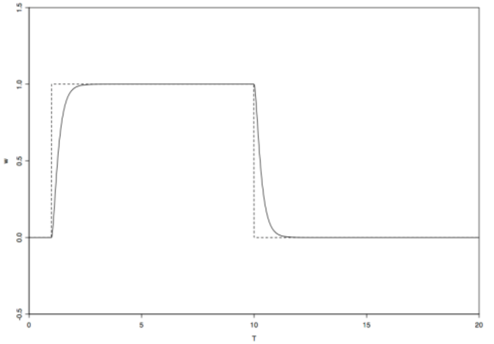
Input: voltage $v_{app}(t)$



Output: angular velocity $\omega(t)$

Open-Loop Control

$v_{app}(t) = 26.681\text{V}$

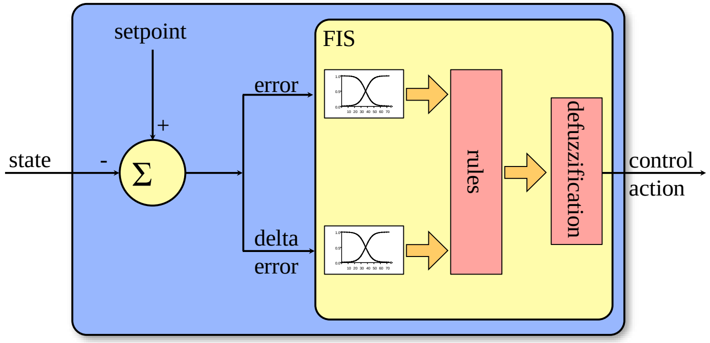


$\omega(t)$ reaches approx. 1 rotation per second

Fuzzy Architecture for DC motor control

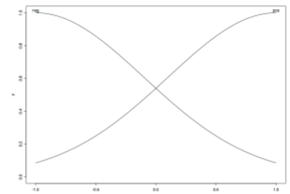
- A fuzzy inference system (FIS) takes one or more control parameters as inputs and produces the control action as output:
  - Inputs
    - error
    - delta error (rate of change of error)
  - output
    - control action : change in output (delta output)

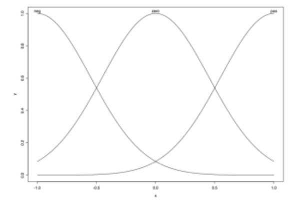
Fuzzy Control Diagram



Four-Rule System

- Each input:
  error = setpoint - state
  delta error = e[t-1]-e[t]
  has two membership functions:
  - neg(-ative) & pos(-itive)
- Output has three membership functions:
  - neg, zero & pos
- Four rules are used





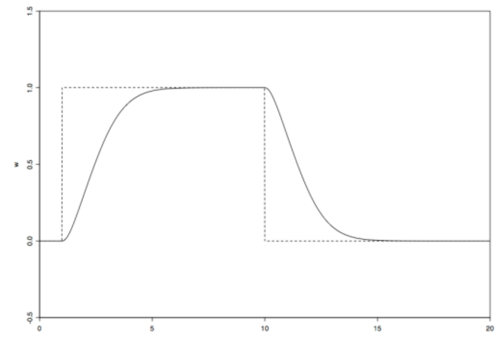
Four Rules

1. If *error* is Neg	and *delta_error* is Neg	then *output* is Neg
2. If *error* is Neg	and *delta_error* is Pos	then *output* is Zero
3. If *error* is Pos	and *delta_error* is Neg	then *output* is Zero
4. If *error* is Pos	and *delta_error* is Pos	then *output* is Pos

		delta_error	
		Neg	Pos
error	Neg	Neg	Zero
	Pos	Zero	Pos

<u>Basic Example:</u>
setpoint = 10rps
state[t-1] = 6rps
error[t-1] = 4
state[t]= 7rps
error[t]=3 ($\mu_{\mathbf{e\_pos}}$=1)
delta_error[t] = 1($\mu_{\mathbf{de\_pos}}$=1)
→output is **Pos** (i.e. faster)

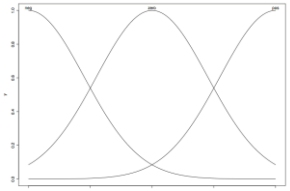
Motor Control based on 4 rule FLC:

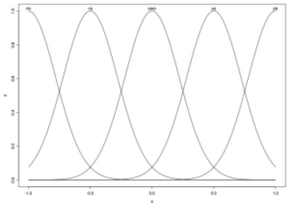


Not bad – but not great!

Nine-Rule System

- Each input
  error
  delta error
  now has three MFs
  – neg., zero & pos.
- Output now has five MFs
  – neg. big, neg. small
  – zero
  – pos. big, pos. small
- Nine rules are needed





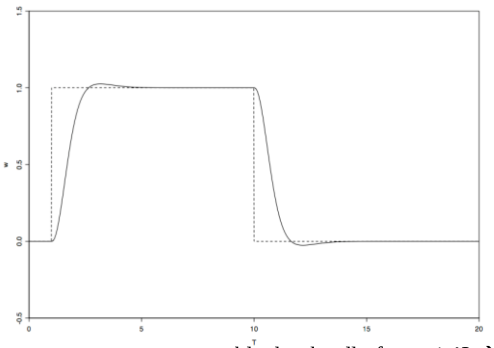
Nine Rules

1. If *error* is Neg and *delta_error* is Neg then *output* is NB
2. If *error* is Neg and *delta_error* is Zero then *output* is NS
3. If *error* is Neg and *delta_error* is Pos then *output* is Zero
4. If *error* is Zero and *delta_error* is Neg then *output* is NS
5. If *error* is Zero and *delta_error* is Zero then *output* is Zero
6. If *error* is Zero and *delta_error* is Pos then *output* is PS
7. If *error* is Pos and *delta_error* is Neg then *output* is Zero
8. If *error* is Pos and *delta_error* is Zero then *output* is PS
9. If *error* is Pos and *delta_error* is Pos then *output* is PB

Nine-Rule Table

		delta_error		
		Neg	Zero	Pos
error	Neg	NB	NS	Zero
	Zero	NS	Zero	PS
	Pos	Zero	PS	PB

Motor Control based on 9 rule FLC:



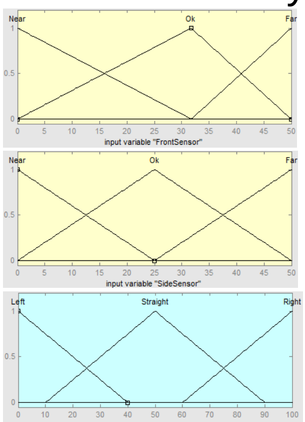
acceptable, but hardly fantastic!? → Tuning!

Example 2
Robot Control
(Left Hand Wall Following Behaviour)

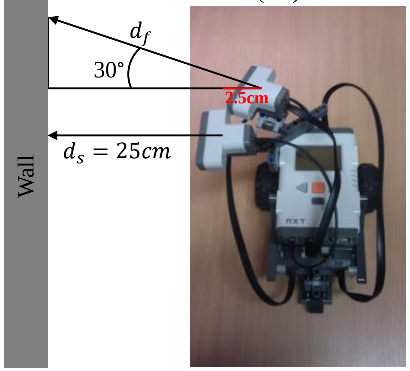
Problem: Left Hand Wall Following

- Desired distance to wall:
  - 25cm
- 2 inputs:
  - front sonar sensor facing the wall at an angle
  - side sonar sensor directly facing the wall
- 1 output:
  - Steering from 0-100 where 0 is hard left and 100 is hard right
- Note, design your own FLCs:
  - Matlab Fuzzy Logic Toolbox (run Matlab and type "fuzzy")
  - JuzzyOnline: http://juzzyonline.wagnerweb.net

Fuzzy Set Design



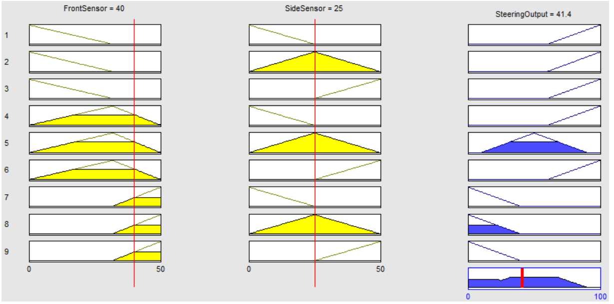
$$d_f = \frac{25 + 2.5}{\cos(30°)} \cong 31.75cm$$



Design Rule Base:

Rule Number	Front Sensor	Side Sensor	Steering Output
1	Near	Near	
2	Near	Ok	
3	Near	Far	
4	Ok	Near	
5	Ok	Ok	
6	Ok	Far	
7	Far	Near	
8	Far	Ok	
9	Far	Far	

Example Output for **a** rule base



FLC Tuning & Automatic Design

FLC Design Choices

- There are very many design choices in any fuzzy inference system, including:
  – variables
    - choice & number of input variables
    - number & shape of MFs in input & output variables
  – rules
    - number & form of rules
  – inferencing mechanism
    - Mamdani, TSK (Tagaki-Sugeno-Kang)
    - alternative operators (t-Norm, t-Conorm)
  – defuzzification method
    - centre-of-gravity
    - mean-of-maxima

Tuning & Design Methodologies

- “Expert-based” design & tuning
  – Design MFs based on experience
  – Manual rule-base design

- Automated design & tuning methods
  – Data driven methods, e.g.:
    - Adaptive neuro fuzzy inference system (ANFIS)
    - Wang-Mendel automatic rule creation (L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man, and Cyber- netics 22:6 (1992) 1414-1427.)
    - *Note: an existing (potentially human control) solution is required for data-driven methods in order to generate the data.*
  – Goal driven methods, e.g.:
    - evolutionary fuzzy systems
    - simulated annealing
    - *Note: a fitness function or similar is required.*

FLC Capabilities

FLCs can be hard to design appropriately but they offer the potential for "interpretable systems" and the inclusion of expert knowledge/intuition.

Fuzzy Logic and FLCs are not magic, they deterministically map inputs to outputs

Different types of FLCs exist, we have looked at type-1 FLCs

FLC Implementation Notes

- Implementing a type-1 FLC is straightforward in C
  - Operations are based on simple math, mainly *min* and *max*.
  - MFs are easily implemented using structures in C
- Also many libraries exist, e.g.:
  - in Matlab (as seen)
  - In Java and other languages: http://lucidresearch.org/software

Summary

- Fuzzy Logic Control
  - Fuzzy Logic Control Principles
  - Origins and History
  - Fuzzy Sets and FLC components
  - FLC control examples
  - FLC design and tuning
  - FLC implementation notes