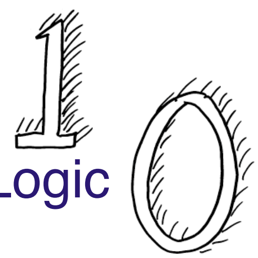
Fuzzy Logic



"Soft Computing"

- Neural networks
- Fuzzy logic
- Neuro-Fuzzy control
- Genetic algorithms

Reference

Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence

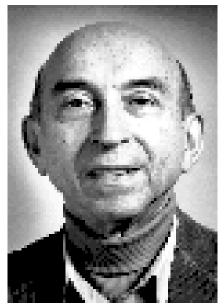
Jyh-Shing Roger Jang, et al, Prentice-Hall, 1996

Fuzzy Logic History

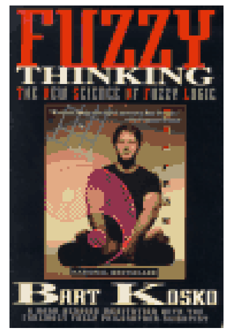
- 1937, Max Black: Vagueness, an exercise in logical analysis, Phil. of Science, **4**, 427-455
- 19xx, Jan Lukasiewicz
- 1967, Lotfi Zadeh, UCB: Fuzzy Sets, in Information and Control J.
- 1974, E.H. Mamdani, Control Systems
- 1980's-90's: Bart Kosko, USC

Fuzzy Founders/Followers

Jan Lukasiewicz
(1878-1956)



Lotfi Zadeh
(1921-)



Bart Kosko

Prof. Zadeh:

As the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.

Fuzzy Jounrals

- IEEE Trans. on Fuzzy Systems
- International J. of Approximate Reasoning
- Intelligent and Fuzzy Systems
- Journal of Cybernetics

Fuzzy Logic Applications

- Subway ride smoothness control
- Camcorder auto-focus and jiggle control
- Braking systems
- Saturn automobile transmission
- Copier quality control
- Rice-cooker temperature control

More Applications of Fuzzy Logic

- Automatic control of dam gates for hydroelectric-powerplants (Tokyo Electric Power)
- Camera aiming for the telecast of sporting events (Omron)
- Cruise-control for automobiles (Nissan, Subaru)
- Positioning of wafer-steppers in the production of semiconductors (Canon)
- Prediction system for early recognition of earthquakes (Inst. of Seismology Bureau of Metrology, Japan)
- Controlling of subway systems in order to improve driving comfort, precision of halting and power economy (Hitachi)

Air-Conditioning System (Mitsubishi)

Problem description:
  Industrial air-conditioning system shall be able to react flexibly to changing ambient conditions

Realization:
    50 rules
    6 linguistic variables
    Resolution: 8 bit
    Input variables: room temperature, wall temperature and temporal evaluation of these signals

Development:
    4 days to create the prototype
    20 days for testing and integration
    80 days for optimization with real test objects
    Implementation as pure software solution on standard microcontroller

Results:
    Reduction of starting processes down to 40 percent of the standard solution
    Sustaining of the temperature even with interference factors (like open window, etc.) substantially improved
    Fewer sensors required, Established energy saving by testing: 24 percent

Fuzzy Silver Bullet?

- Fuzzy logic may not provide any new mechanisms that weren't there before.
- It provides a **viewpoint**, that helps expedite problem solving.
- Analogy: Object-Oriented Programming didn't create any new computable functions.

Fuzzy Set Basics

- Classical ("crisp") sets:
  - Membership in a set is all or nothing
  - Characteristic function c_S: Universe $\rightarrow$ {0, 1}
  - $c_S(x) = 1$ iff $x \in S$
- Fuzzy sets:
  - Membership in a set is a degree
    - membership function c_S: Universe $\rightarrow$ [0, 1]

range of real numbers

Linguistic Characterizations of Degree of Membership

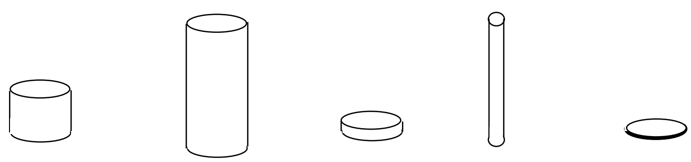
- Consider the set of “hot” days in Claremont in 2003.
- Was Oct. 29 “hot”? It might have been called one of:
  - “very hot”
  - “sort of hot”
  - “not hot”
- The answer depends on the observer, time, etc.

Sounds similar to probability, but isn’t

- Probability deals with uncertainty or likelihood of occurrence.
- Fuzzy logic deals with ambiguity, vagueness of description.

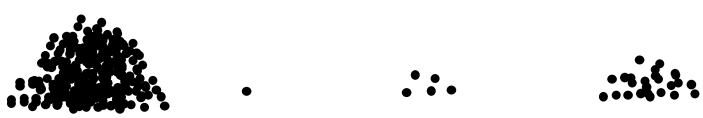
Fuzzy Membership

Are these disks, cylinders, or rods?

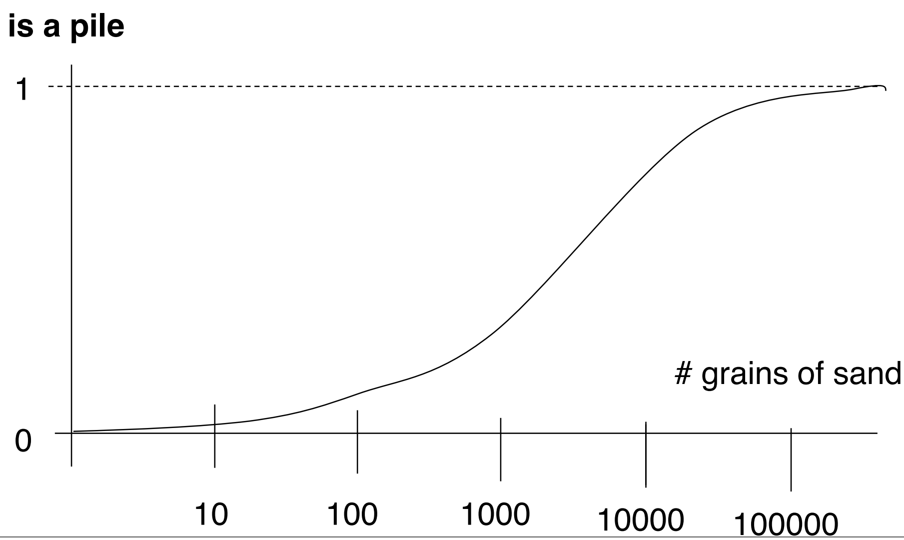


Fuzzy Membership

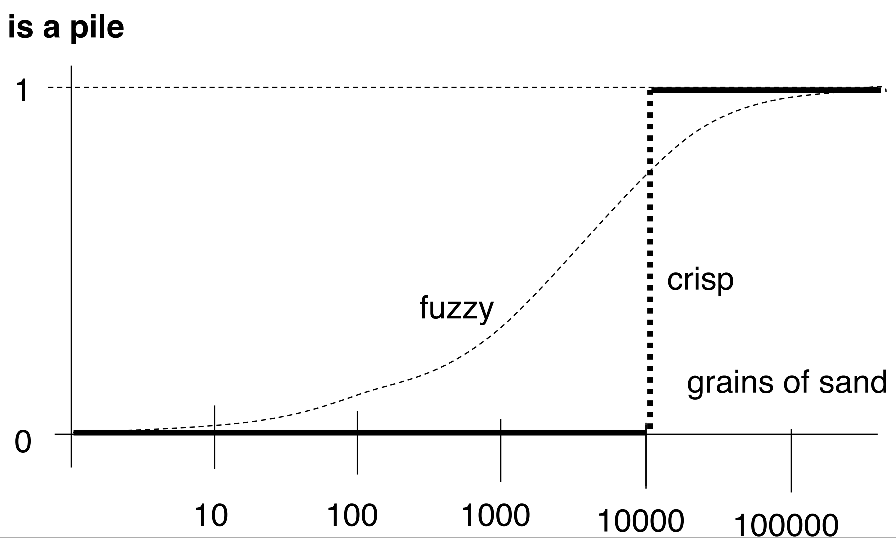
Which of these is a pile of sand?



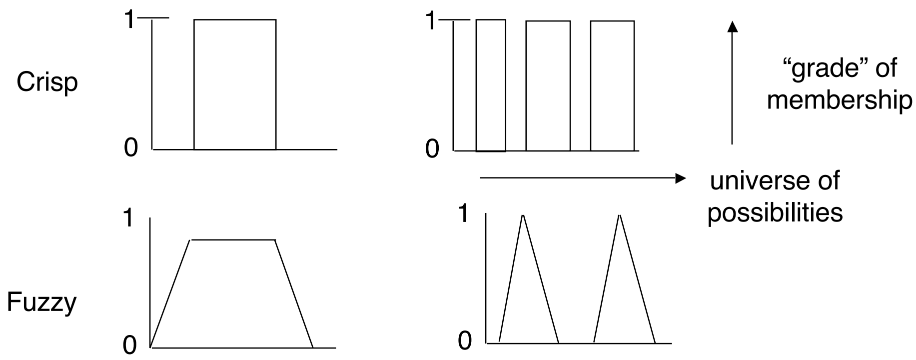
Membership function plots



Crisp vs. Fuzzy Membership Functions

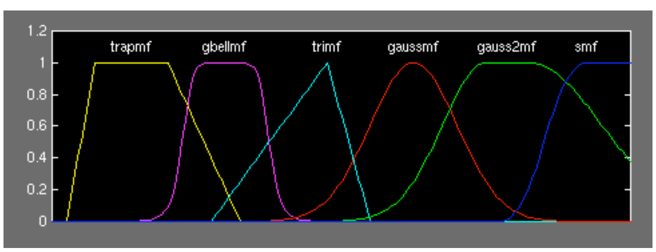


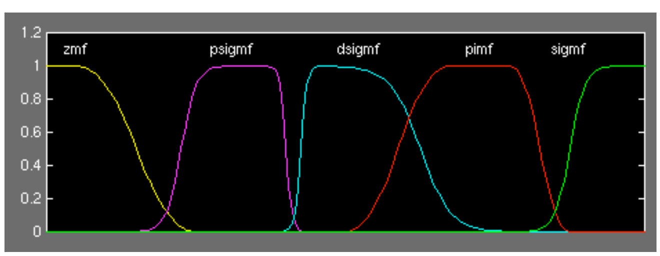
More Crisp vs. Fuzzy Membership Functions



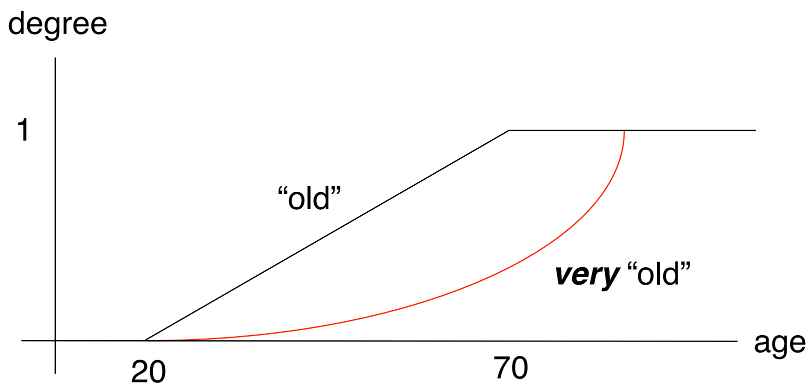
Note: Universe can be continuous or discrete, ordered or unordered.

Matlab's Builtin Membership Functions

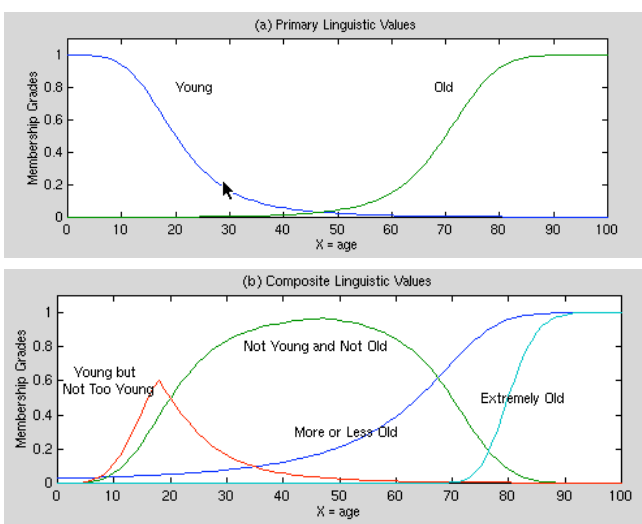




Linguistic Modifiers

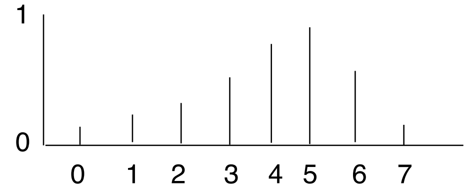


Example: Composites of Young and Old

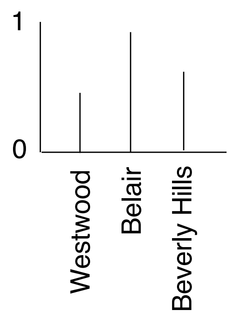


Discrete Universe Examples

Ideal number of courses to take
(ordered universe)

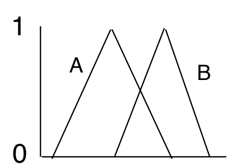


Desirable place to live
(unordered universe)



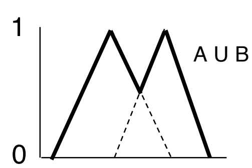
Fuzzy-Set Operations

expressed using membership functions



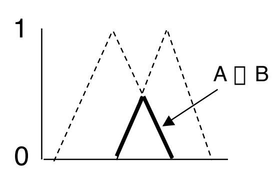
Fuzzy OR (union)

$c_{A \cup B}(x) = \max(c_A(x), c_B(x))$



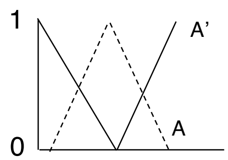
Fuzzy AND (intersection)

$c_{A \cap B}(x) = \min(c_A(x), c_B(x))$



Fuzzy Complement

$c_{A'}(x) = 1 - c_A(x)$.



Which Set-Theoretic Rules Hold?

A□B = B□A

A□B = B□A

(A□B) □ C = A □ (B□C)

(A□B) □ C = A □ (B□C)

(A□B) □ C = (A□C) □ (B□C)

(A□B) □ C = (A□C) □ (B□C)

(A□B)' = A'□B'

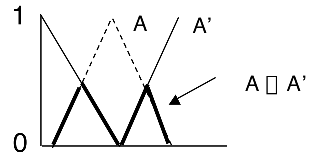
(A□B)' = A'□B'

etc.

Fuzzy Anomaly?

The intersection of a set with its complement is not necessarily empty.

$c_{A'}(x) = 1 - c_A(x)$.



Fuzzy Implication

There is no single standard. A variety of versions exists:

Larsen:	$x \rightarrow y = xy$
Lukasiewicz:	$x \rightarrow y = \min(1, 1-x+y)$
Mamdani:	$x \rightarrow y = \min(x, y)$
Standard strict:	$x \rightarrow y = x \leq y \ ?\ 1 : 0$
Goedel:	$x \rightarrow y = x \leq y \ ?\ 1 : y$
Gaines:	$x \rightarrow y = x \leq y \ ?\ 1 : y/x$
Kleene-Dienes:	$x \rightarrow y = \max(1-x, y)$
Kleene-Dienes-Luk:	$x \rightarrow y = 1-x+xy$

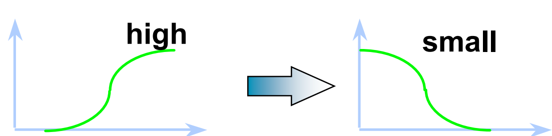
Linguistic Rules

- In Fuzzy Logic, rules are expressed **qualitatively and linguistically**, rather than quantitatively.
- The result is qualitatively understandable, yet can be **interpreted quantitatively** when desired.
- The interpretive framework can be **adjusted to suit**.

Fuzzy If-Then Rules

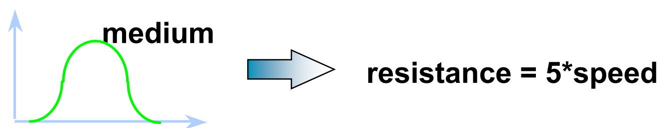
- **Mamdani style**

  If pressure is high then volume is small

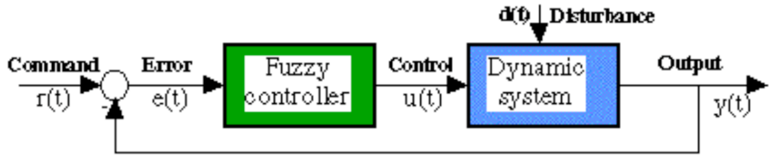


- **Sugeno style (uses *equations* on rhs)**

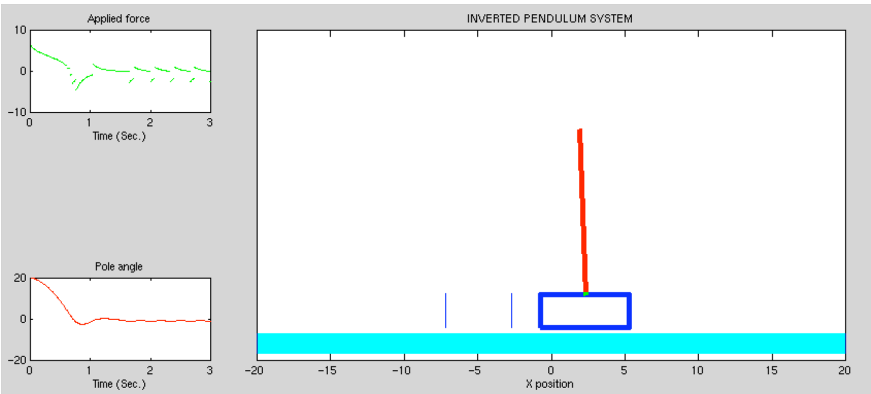
  If speed is medium then resistance = 5*speed



Mamdani Control Model (next several slides)



Pole-on-Cart (Inverted Pendulum) Example

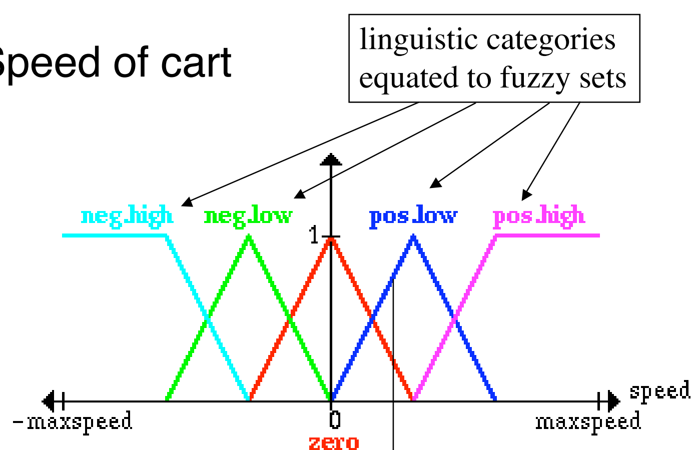


Problem

- Determine a fuzzy rule-base adequate to specify the pole-cart controller.

Pole-on-Cart Example

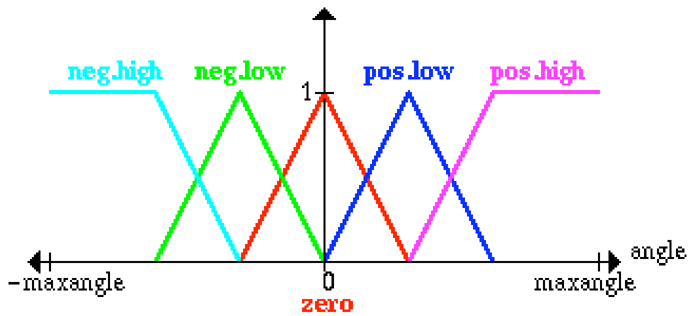
- Speed of cart



The speed can have non-zero membership in more than 1 category.

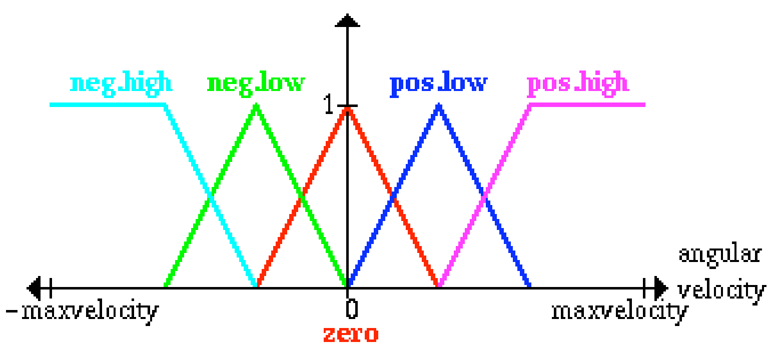
Pole-on-Cart Balancing Example

- Angle of pole



Pole-on-Cart Balancing Example

- Angular velocity of pole



Fuzzy Rule Base
(Kosko: FAM- Fuzzy Associative Memory)

Example of a fuzzy-logic rule represented in this table:
"**If** the angular velocity is pos. low **and** the angle is zero, **then** set the speed at low."

angle

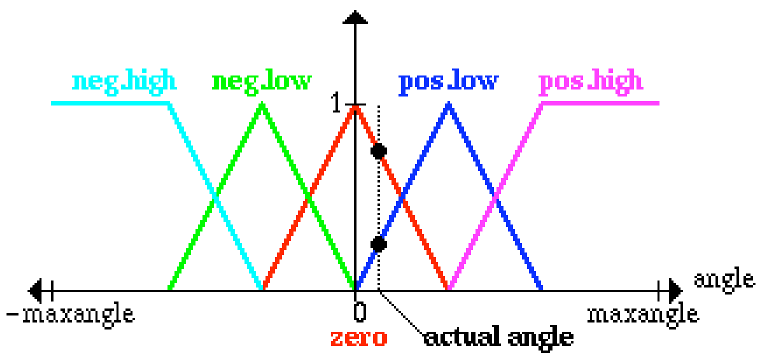
angular velocity	**neg. high**	**neg. low**	**zero**	**pos. low**	**pos. high**
neg. high			neg. high		
neg. low			neg. low		
zero	neg. high	neg. low	zero	pos. low	pos. high
pos. low		zero	low		
pos. hi		high			

control speed
as a function of angle and angular velocity

Inference in a
(Mamdani-style) Fuzzy System

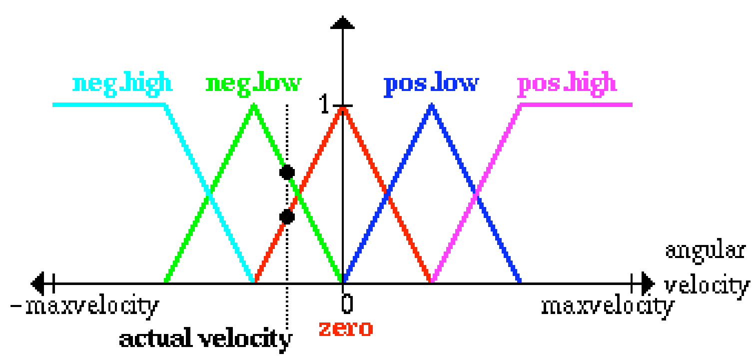
- Start with quantitative input data
- **Fuzzify** the data
- Derive conclusion based on fuzzy data
- **De-fuzzify** the conclusion to get quantitative output

Fuzzification



In this case, the actual *angle* is a **mixture** of zero and pos. low.

Fuzzification



Here the actual *angular velocity* is a **mixture** of zero and neg. low.

Multiple Applicable Rules:

- angle is a **mixture** of zero and pos. low.
- angular velocity is a **mixture** of zero and neg. low

angular velocity \ angle	**neg. high**	**neg. low**	**zero**	**pos. low**	**pos. high**
neg. high			neg. high		
neg. low			**neg. low**		
zero	neg. high	neg. low	**zero**	**pos. low**	pos. high
pos. low		zero	low		
pos. hi		high			

***Three* entries in the rule base are applicable.**
We must determine *how to combine them*.

Use the diagrams to determine the **degree** to which each rule is applicable.

- Consider the rule

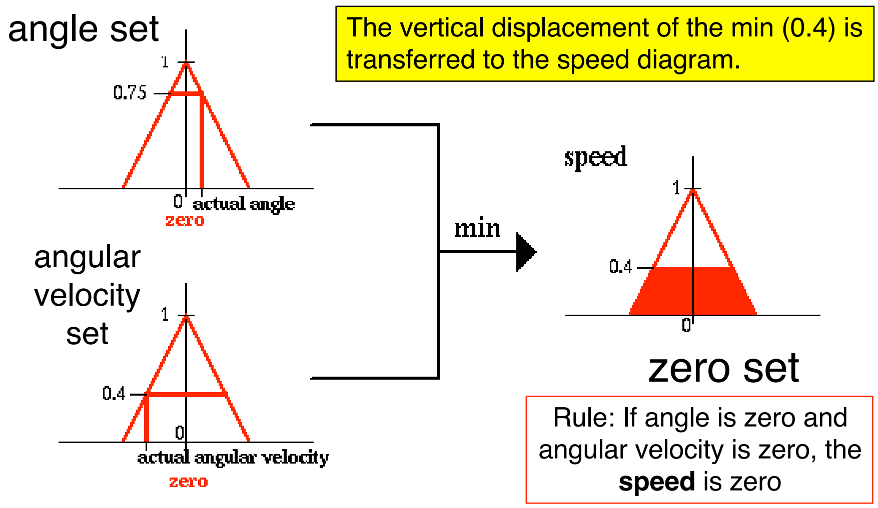
  ”If angle is zero and angular velocity is zero, the **speed** is zero".

  The actual value belongs to the fuzzy set *zero* to a **degree** of 0.75 for "angle" **and** to a **degree** of 0.4 for "angular velocity".

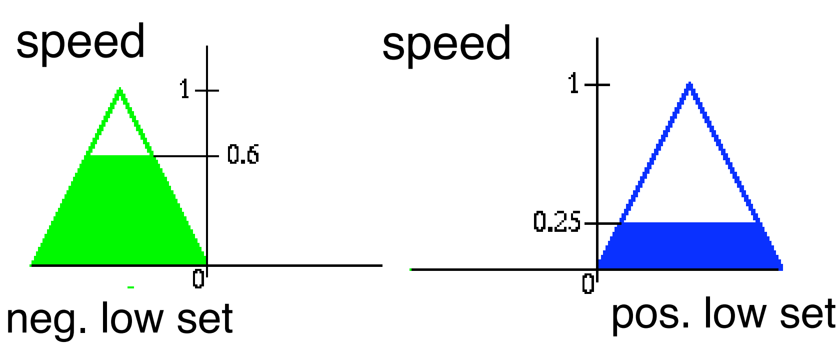
- Since this is an **AND** operation, the **minimum** criterion is used. (For OR, the maximum would be used.)

- The fuzzy set **zero** of the variable "**speed**" is ***cut*** at 0.4 and the patches are shaded up to that area.

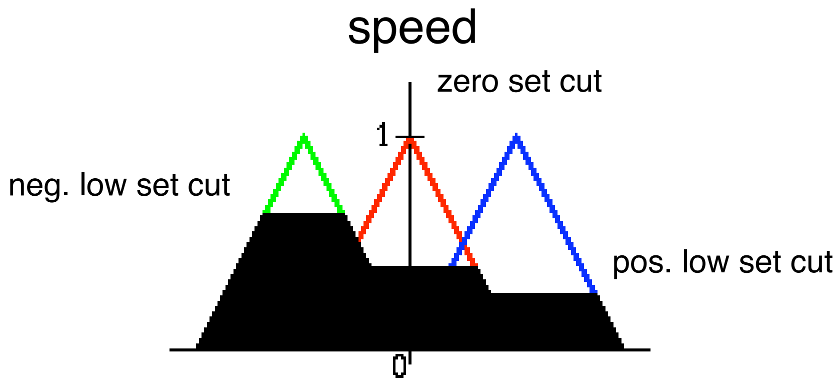
Using min combination for AND
(This is for **one of three speed rules**: zero.)



Similarly, for *each* of the (3) applicable rules we get an inferred speed.

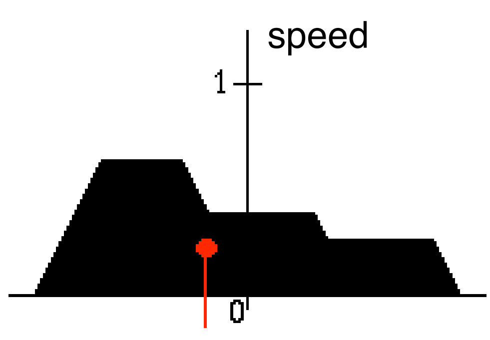


We then combine the results of the three rules to get an **output fuzzy set**.



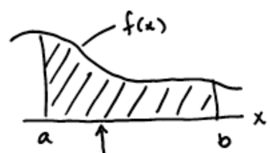
Defuzzification

To actually ***set*** the speed, we need a *number*, not a fuzzy set.
Various rules can be used to get the number.
The most common one is to use the **centroid** of the fuzzy set.



Centroid Review

Centroid (wrt x)



$$\frac{\int_a^b x f(x)\,dx}{\int_a^b f(x)\,dx}$$

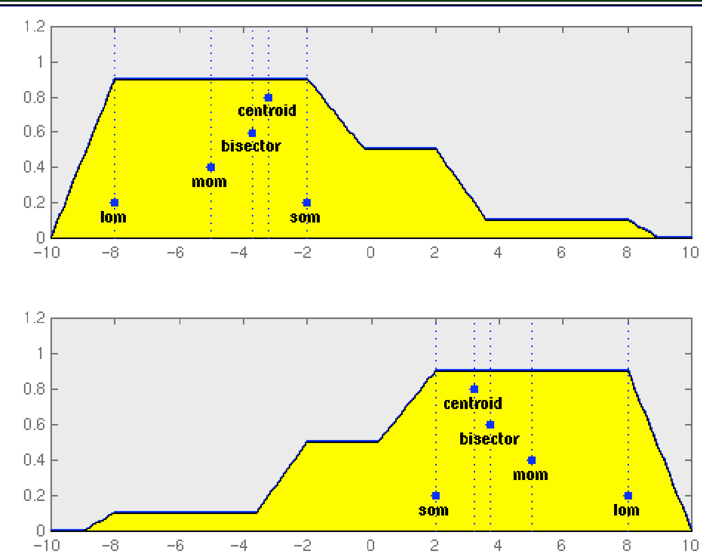
The point at which the areas on either side are equally balanced.

Matlab's Defuzzification Rules

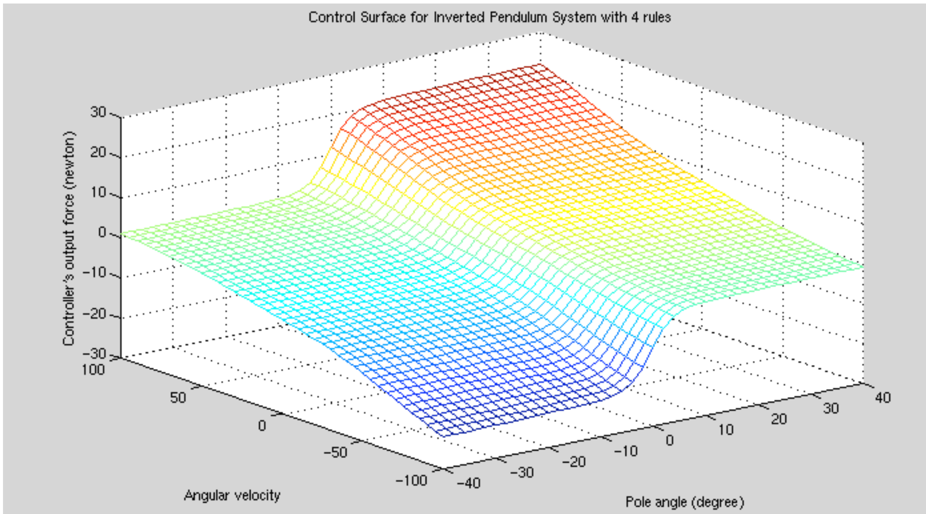
mom = mean of max

lom = largest of max

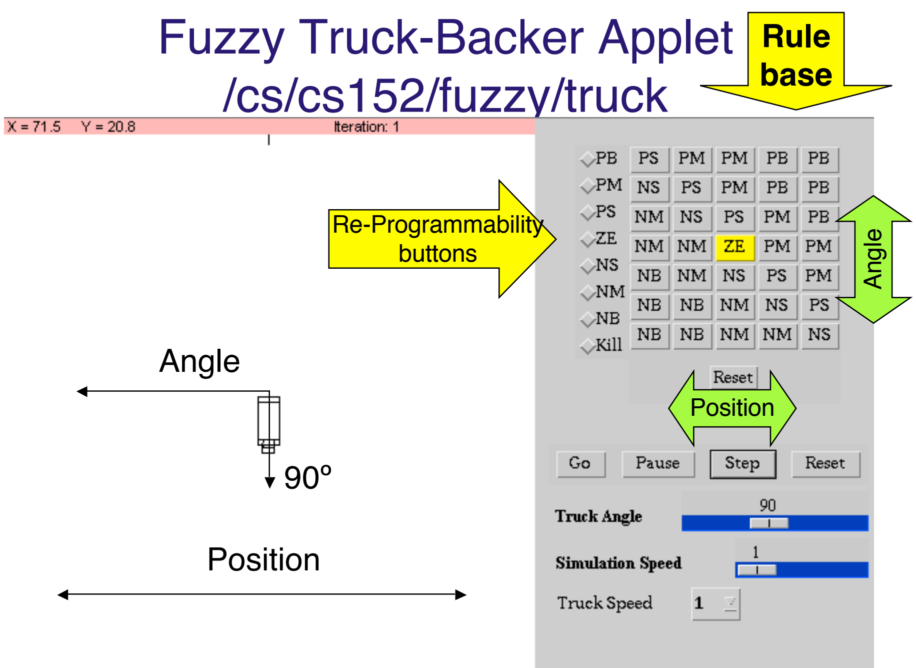
som = smallest of max



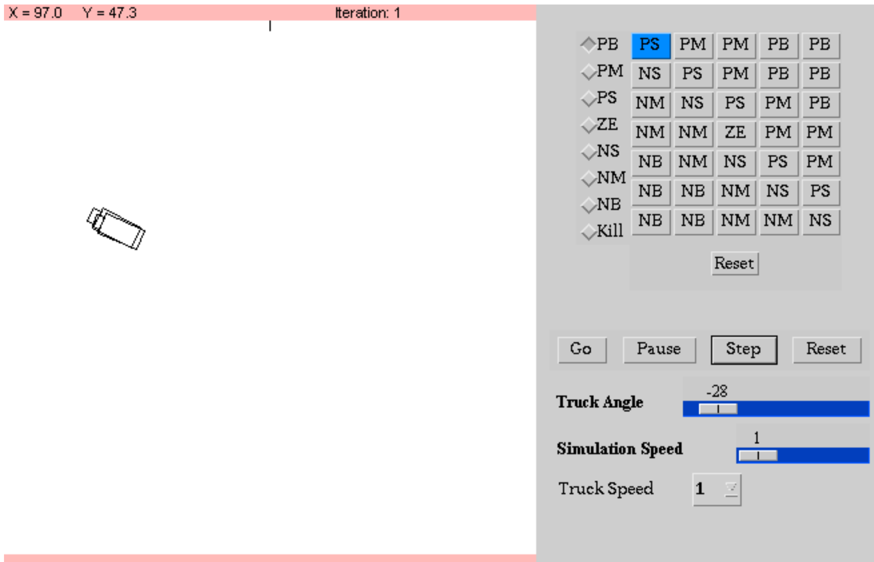
Control Surface for an Inverted Pendulum: Force = f(Angle, AngularVelocity)

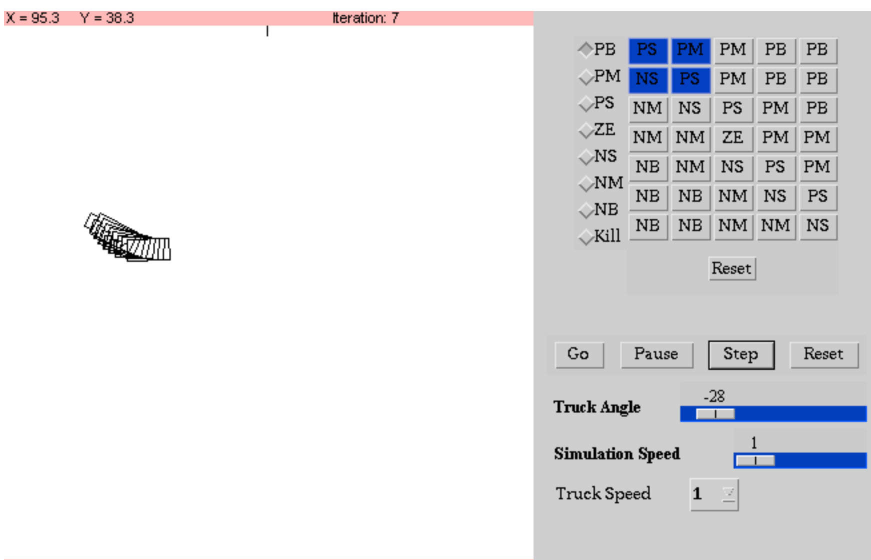


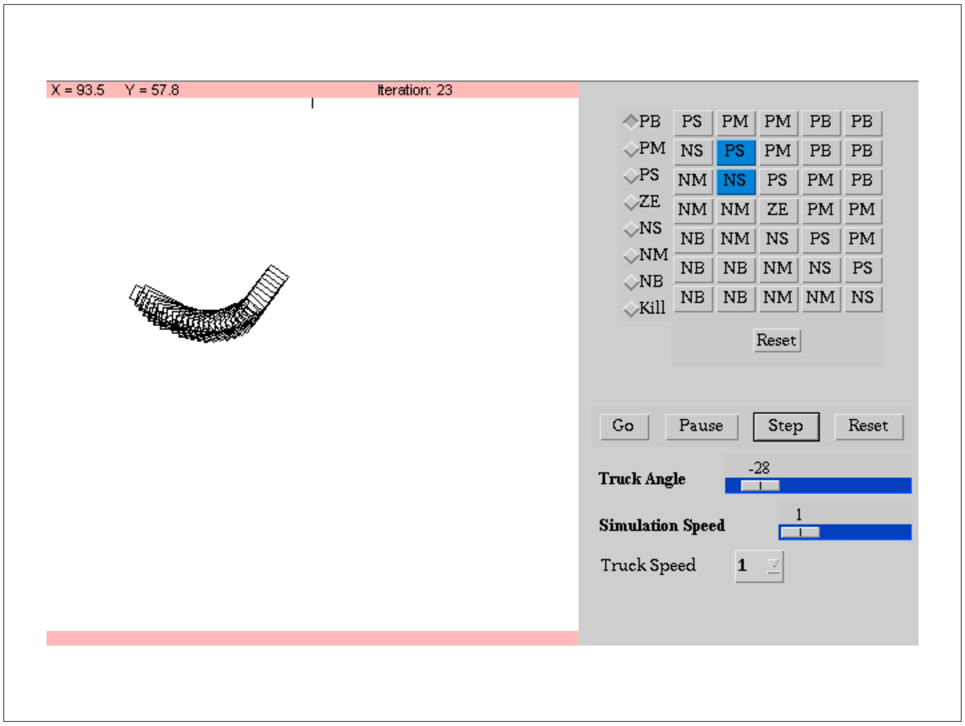
Fuzzy Truck-Backer Applet /cs/cs152/fuzzy/truck

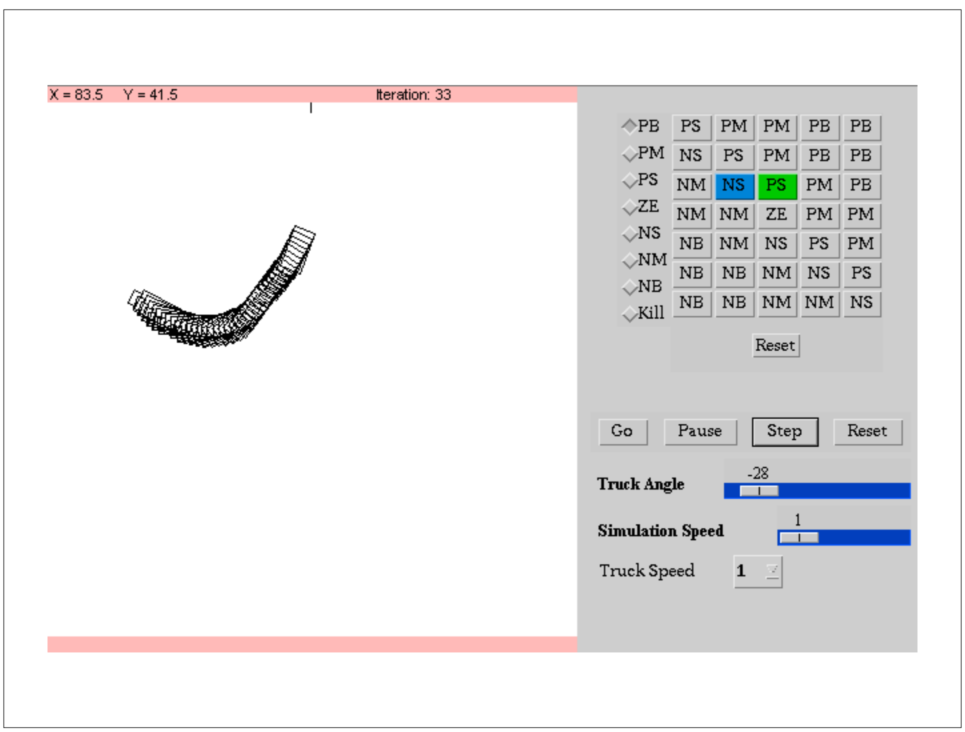


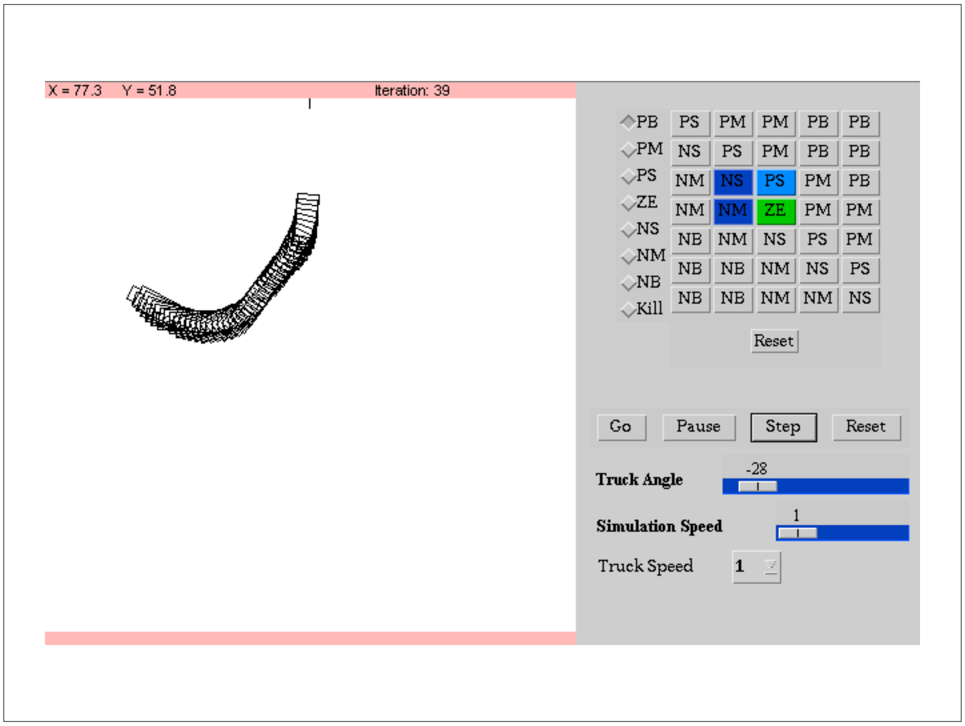
An example simulation

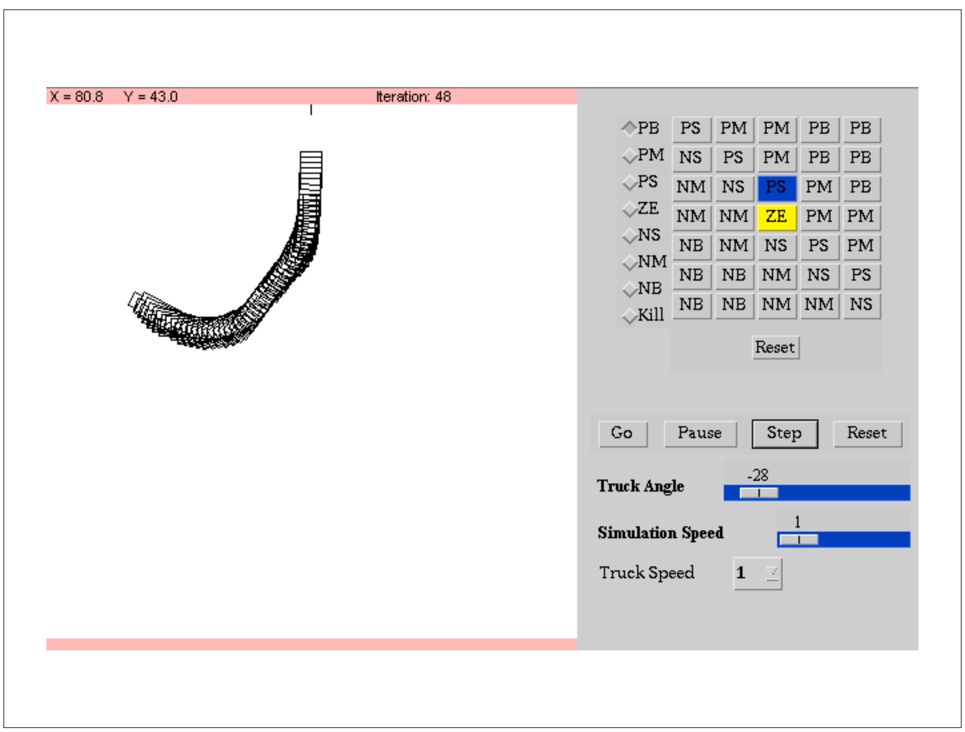


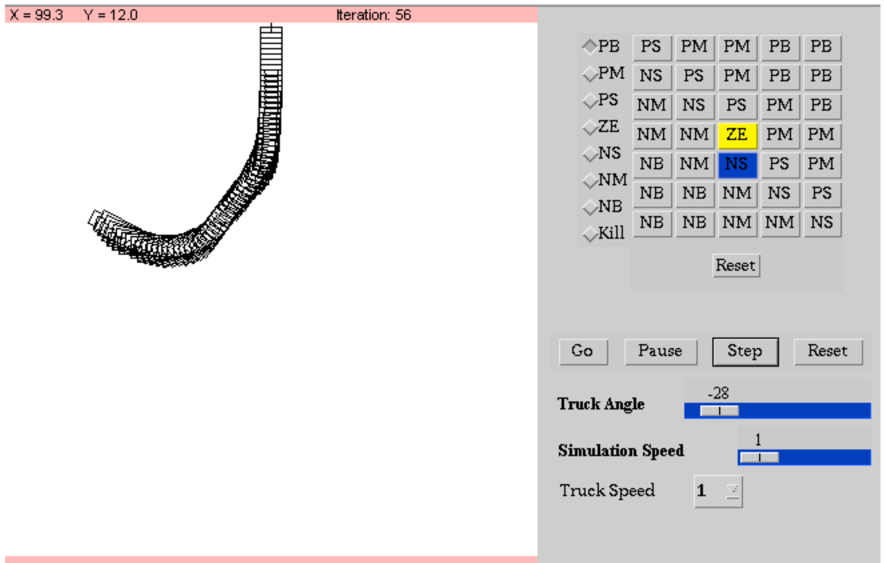



X = 93.5 Y = 57.8
Iteration: 23
PB PS PM PM PB PB
PM NS PS PM PB PB
PS NM NS PS PM PB
ZE NM NM ZE PM PM
NS NB NM NS PS PM
NM NB NB NM NS PS
NB NB NB NM NM NS
Kill
Reset
Go
Pause
Step
Reset
Truck Angle
-28
Simulation Speed
1
Truck Speed
1

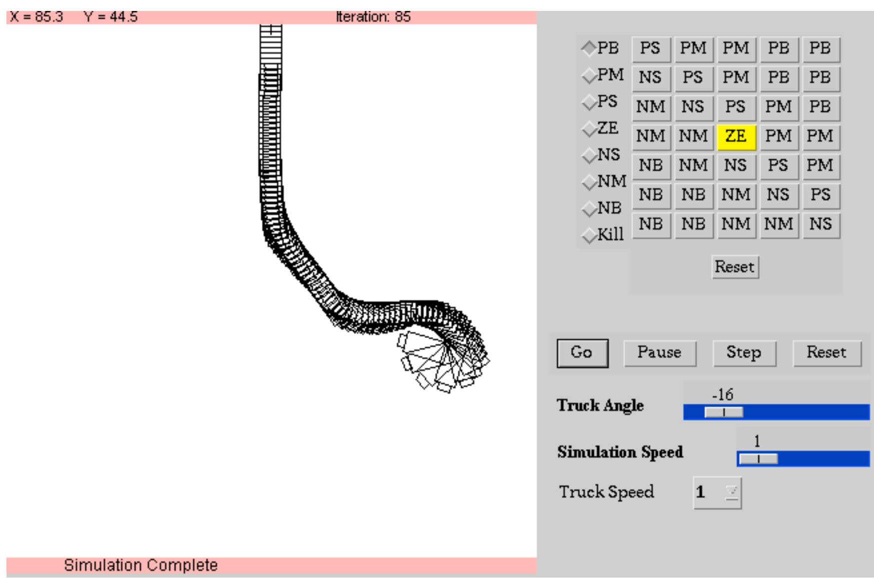

X = 83.5 Y = 41.5
Iteration: 33
PB PS PM PM PB PB
PM NS PS PM PB PB
PS NM NS PS PM PB
ZE NM NM ZE PM PM
NS NB NM NS PS PM
NM NB NB NM NS PS
NB NB NB NM NM NS
Kill
Reset
Go
Pause
Step
Reset
Truck Angle
-28
Simulation Speed
1
Truck Speed
1


X = 77.3 Y = 51.8
Iteration: 39
PB PS PM PM PB PB
PM NS PS PM PB PB
PS NM NS PS PM PB
ZE NM NM ZE PM PM
NS NB NM NS PS PM
NM NB NB NM NS PS
NB NB NB NM NM NS
Kill
Reset
Go
Pause
Step
Reset
Truck Angle
-28
Simulation Speed
1
Truck Speed
1

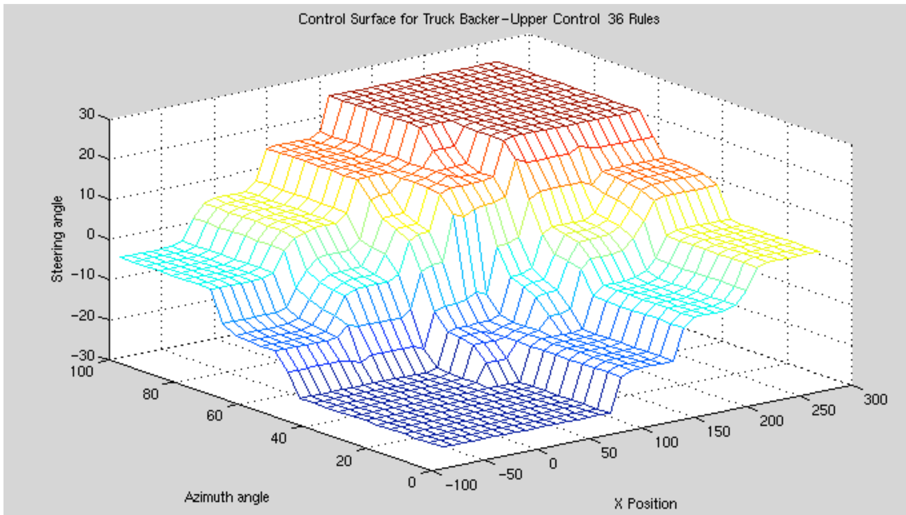

X = 80.8 Y = 43.0
Iteration: 48
PB PS PM PM PB PB
PM NS PS PM PB PB
PS NM NS PS PM PB
ZE NM NM ZE PM PM
NS NB NM NS PS PM
NM NB NB NM NS PS
NB NB NB NM NM NS
Kill
Reset
Go
Pause
Step
Reset
Truck Angle
-28
Simulation Speed
1
Truck Speed
1



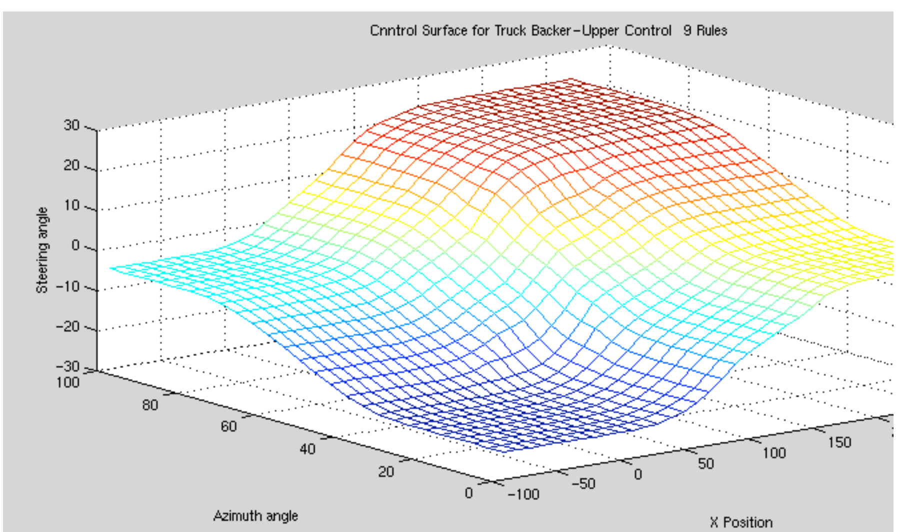
A different example simulation



Control Surface for another Truck-Backer

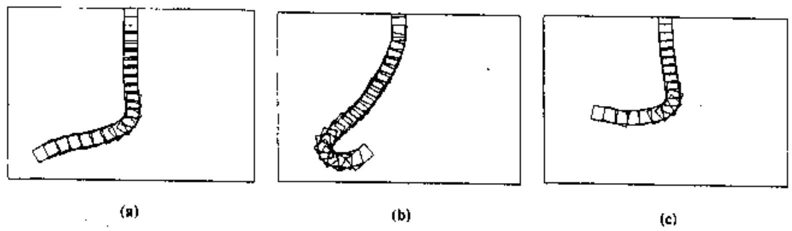


Another Truck Backer using only 9 Rules (/cs/cs152/fuzzy/fismat/fisdemo)

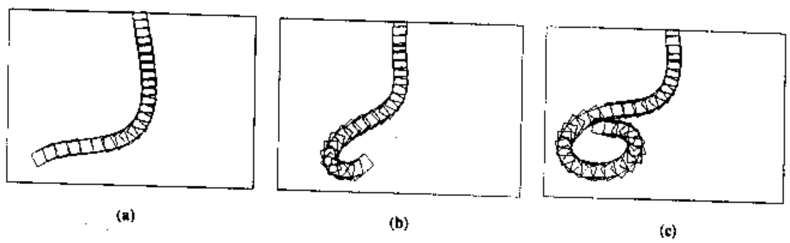


Kong & Kosko compared Fuzzy vs. Neural Controllers

Fuzzy

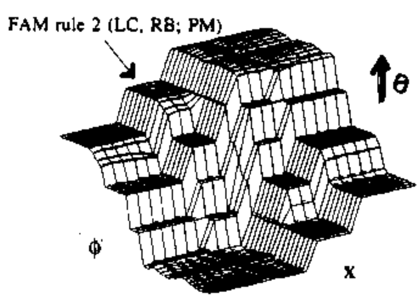


Neural

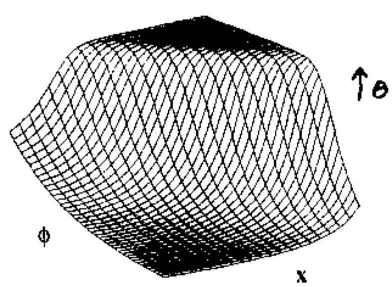


Kong & Kosko Derived Controller Functions

Fuzzy



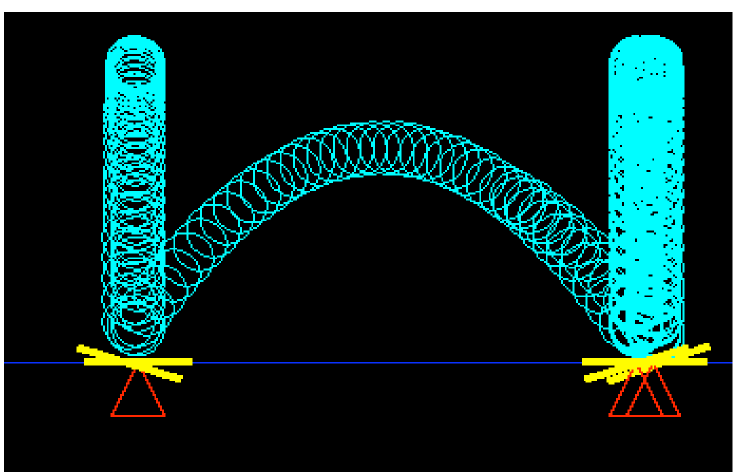
Neural



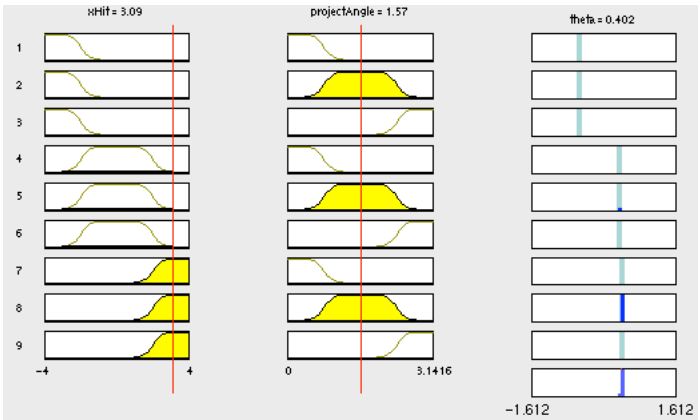
Kong & Kosko Konclusions

- Fuzzy
  - Regular path followed
  - “Trained” by common-sense hand-coded rules
  - Light-weight controller (comparisons and additions only)
- Neural
  - Sometimes followed irregular path
  - Training time-consuming
  - Controller computationally intensive

Matlab Fuzzy Ball-Juggler Demo



Ball-Juggler Rule View



Applications of Fuzzy Logic to Training MLP Neural Networks

- Control of **variable learning rate in backpropagation** (Choi, et al.):
- Let **CE** denote Change of Error.
- Let **CCE** denote Change of CE.
- Fuzzy Rules:
  - If CE is small with no recent sign changes, then increase LR.
  - If CE has recent sign changes, then decrease LR.
  - If CE is small, and CCE is small with no recent sign changes, then increase LR and momentum.

Application of Neural Networks to Fuzzy Logic

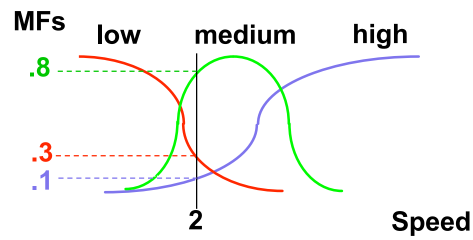
- N. K. Kasabov. Learning Fuzzy Rules through Neural Networks. Proc. of the First New Zealand Intl. Conf. on Artificial Neural Networks and Expert Systems, pages 137--139, 1993.
- ***Learned*** fuzzy rules for **forecasting gas demand**.
- Used backpropagation network with five layers.
- The input layer contains two nodes - temperature and the month and the output layer is the gas demand figure.
- **Fuzzy rules are then extracted from the net**.

Sugeno Control Model

- Extends Mamdani model
- Fuzzy part is still in antecedent of rules, which are used for selection
- Consequent of rules is more complex: some function (e.g. polynomial) of input variables

Fuzzy Inference System Using Sugeno-Style Rules

If speed is low then resistance = 2
If speed is medium then resistance = 4*speed
If speed is high then resistance = 8*speed



Rule 1: w1 = .3; r1 = 2
Rule 2: w2 = .8; r2 = 4*2
Rule 3: w3 = .1; r3 = 8*2

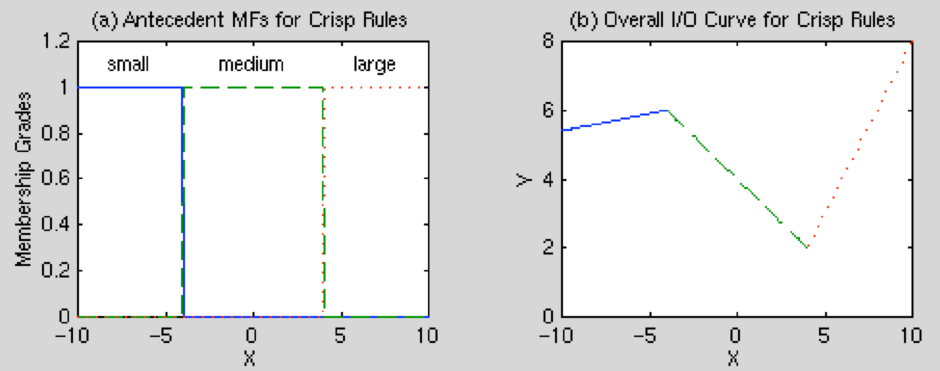
Resistance = Σ(wi*ri) / Σwi
= 7.12

Example: 1-input Sugeno

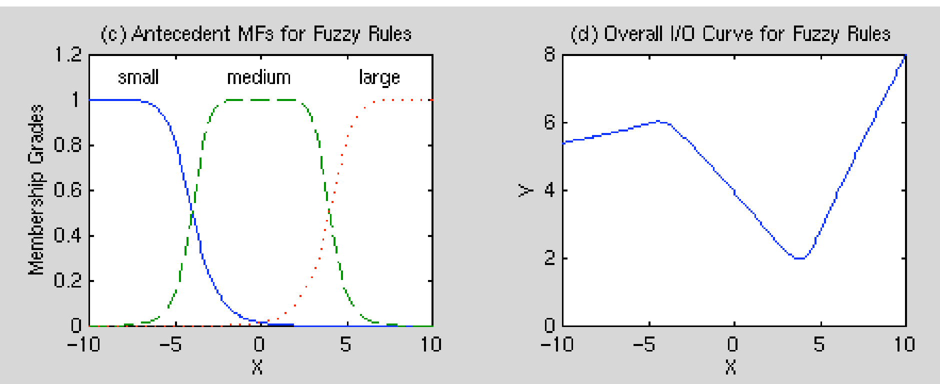
- Rules:
  - If X is small, then Y = 0.1X + 6.4.
  - If X is medium, then Y = -0.5X + 4.
  - If X is large, then Y = X-2.
- The following 2 slides indicate the results of combining these rules using crisp **vs**. fuzzy logic.

demo sug1 (crisp)

(in /cs/cs152/matlab/soft)



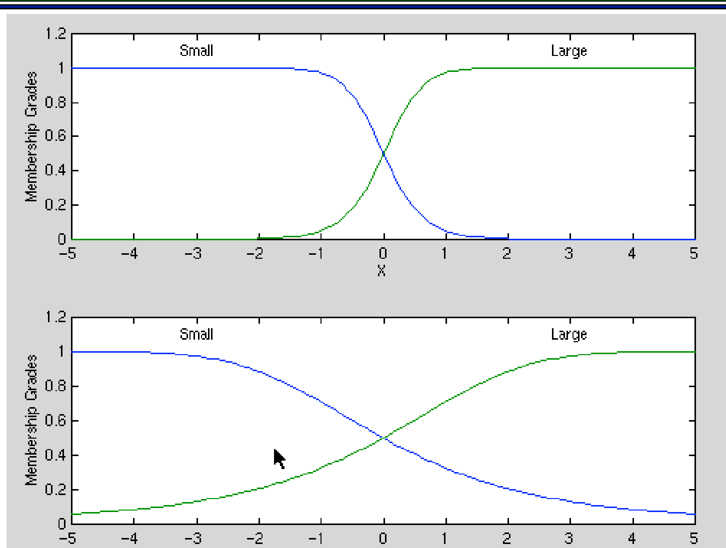
demo sug1 (fuzzy)



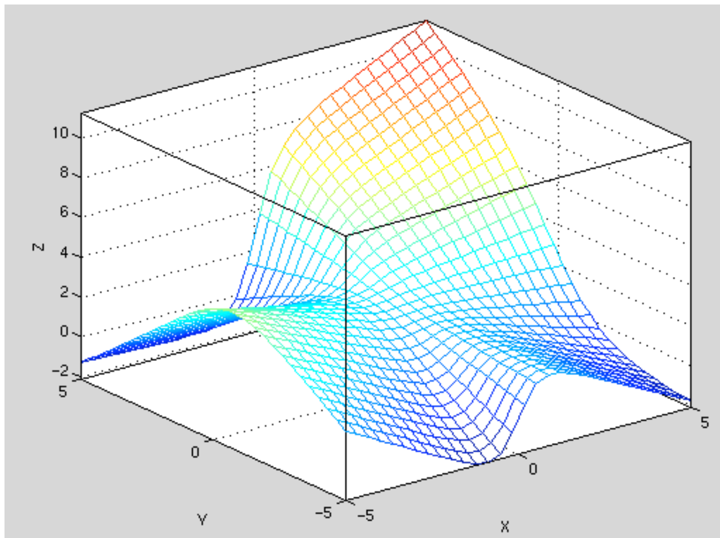
Example: 2-input Sugeno

- Rules:
  - If X is small and Y is small, then Z = -X+Y+1.
  - If X is small and Y is large, then Z = -Y+3.
  - If X is large and Y is small, then Z = -X+3.
  - If X is large and Y is large, then Z = X+Y+2.

MFs for demo sug2 (2-input)



Control surface for sug2



Tsukamoto model

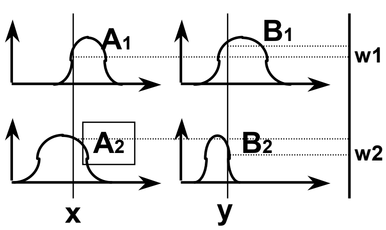
- Aggregate rule outputs by a **weighted average**, rather than by defuzzification.

Example of Hybrids: ANFIS
(Adaptive Neuro-Fuzzy Inference System)

- Developed by J.-S. R. Jang
- Uses Sugeno or Tsukamoto models
- Similar to Radial Basis Function network

ANFIS

- **Fuzzy reasoning**

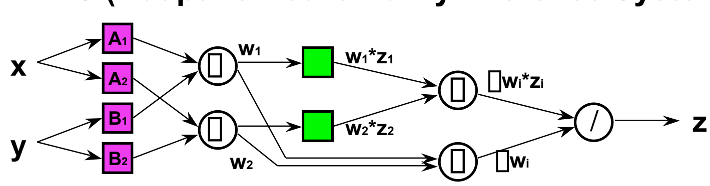


$$z_1 = p_1 * x + q_1 * y + r_1$$

$$z_2 = p_2 * x + q_2 * y + r_2$$

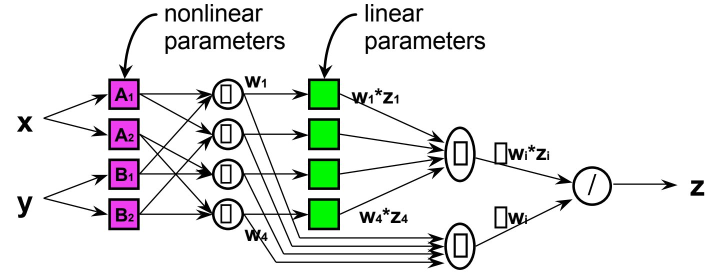
$$z = \frac{w_1 * z_1 + w_2 * z_2}{w_1 + w_2}$$

- **ANFIS (Adaptive Neuro-Fuzzy Inference System)**



ANFIS: Parameter ID

- Hybrid training method



	forward pass	backward pass
MF param. (nonlinear)	fixed	steepest descent
Coef. param. (linear)	least-squares	fixed

Contrary Opinion

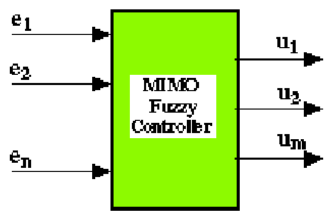
WHY I DESPISE FUZZY FEEDBACK CONTROL, by Michael Athans

http://www.lit.net/ieee/cdc98/debates/cdc98fuzzy-debate/

Asserts that fuzzy control only shown applicable to trivial (SISO) systems, not more complex (MIMO) systems.

Athans' Asserted Shortcomings of Fuzzy Controllers

- Fuzzy rules just generate nonlinear static functions
- Performance specifications "vague" or nonexistent
- Cannot generate "differential equation" controller rules
- Not easy to differentiate noisy sensor signals by finite differencing, as almost always done in fuzzy applications
- No utililization of dynamic (e.g. Kalman) filtering
- I have never seen a multiple-input multiple-output (MIMO) fuzzy control application
- Combinatorial complexity for high-order and multivariable applications



$$u_1 = h_1(e_1, e_2, \ldots, e_n)$$
$$u_2 = h_2(e_1, e_2, \ldots, e_n)$$
$$\ldots\ldots\ldots\ldots$$
$$u_m = h_m(e_1, e_2, \ldots, e_n)$$