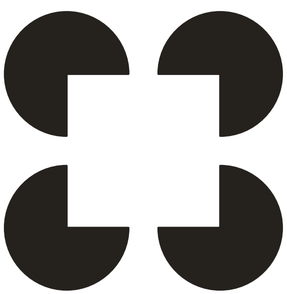
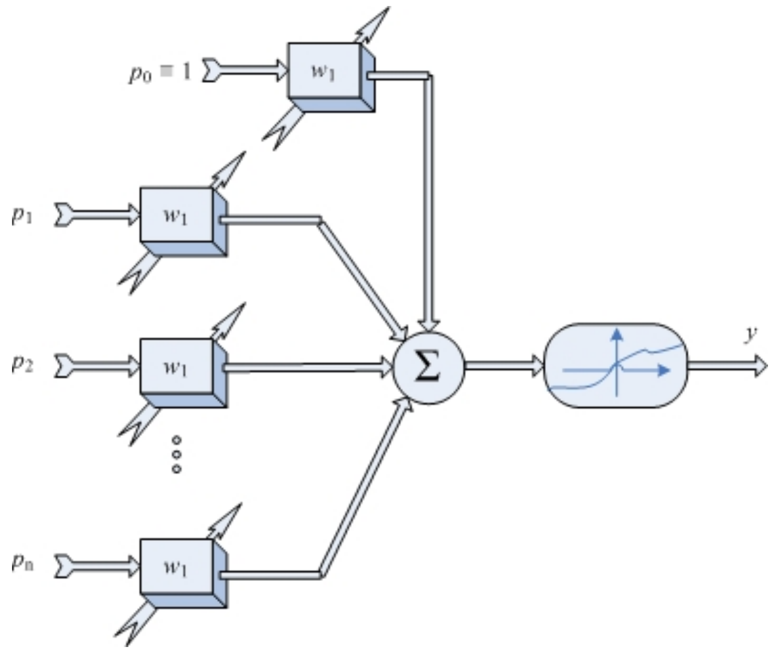
ENE0154 IntelComp – Inteligência Computacional

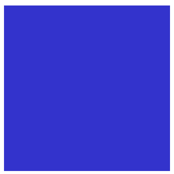
Computational Intelligence

Neural Networks, Deep Machine Learning, Recurrent Networks
Reinforcement Learning and Fuzzy Systems

Prof. Adolfo Bauchspiess







Summary

1 - Introduction – Connectionism, Epistemology, etc.

2 - Artificial Neural Networks – Biological Inspiration, OCR, ADALINE, etc.

3 - Convolutional and Deep Neural Networks – CNN, Cifar10, etc.

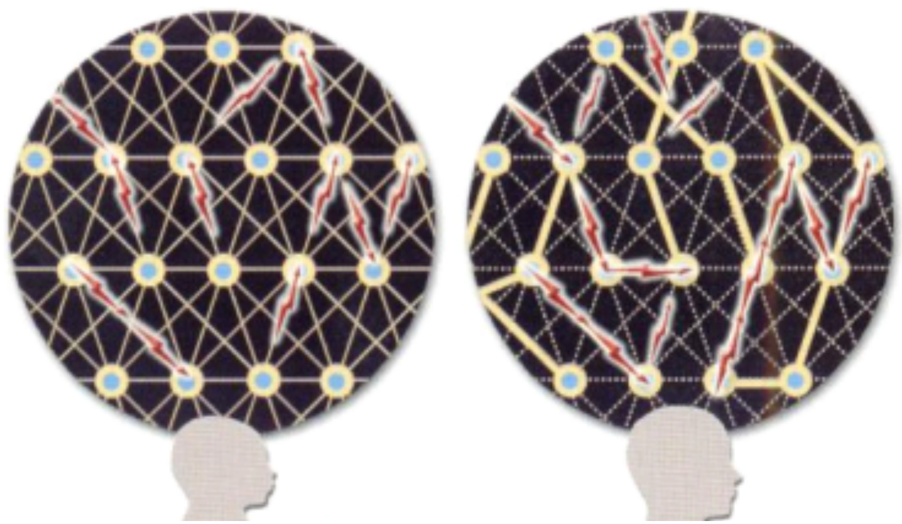
4 - Recurrent Neural Networks – NARMAX

5 - Reinforcement Learning – SARSA, Maze

3 - Fuzzy Logic and Fuzzy Systems – Rule Based Expert Systems, ANFIS

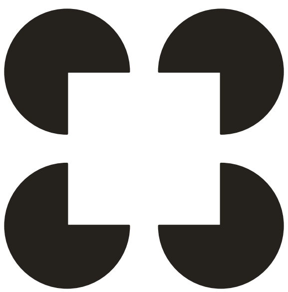
4 - Examples and Applications

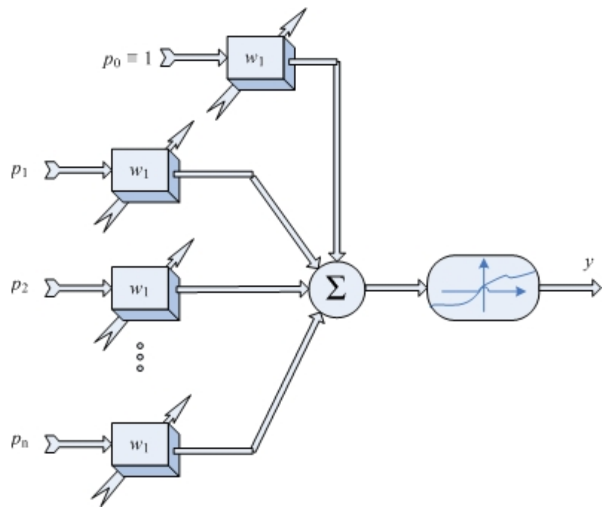
5 - Conclusions

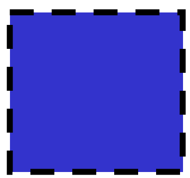


Part 1

Introduction – Connectionist Intelligent Systems







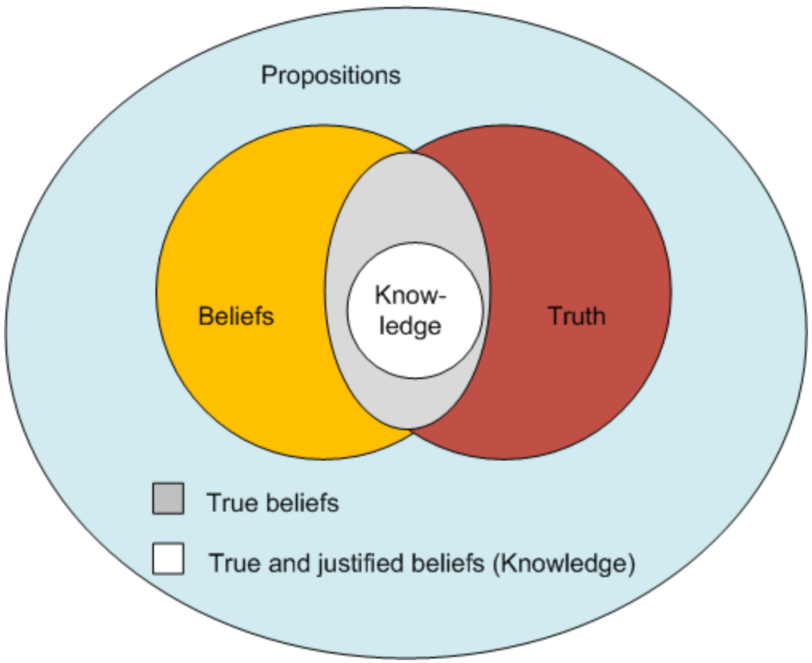
Some publications - Intelligent Systems

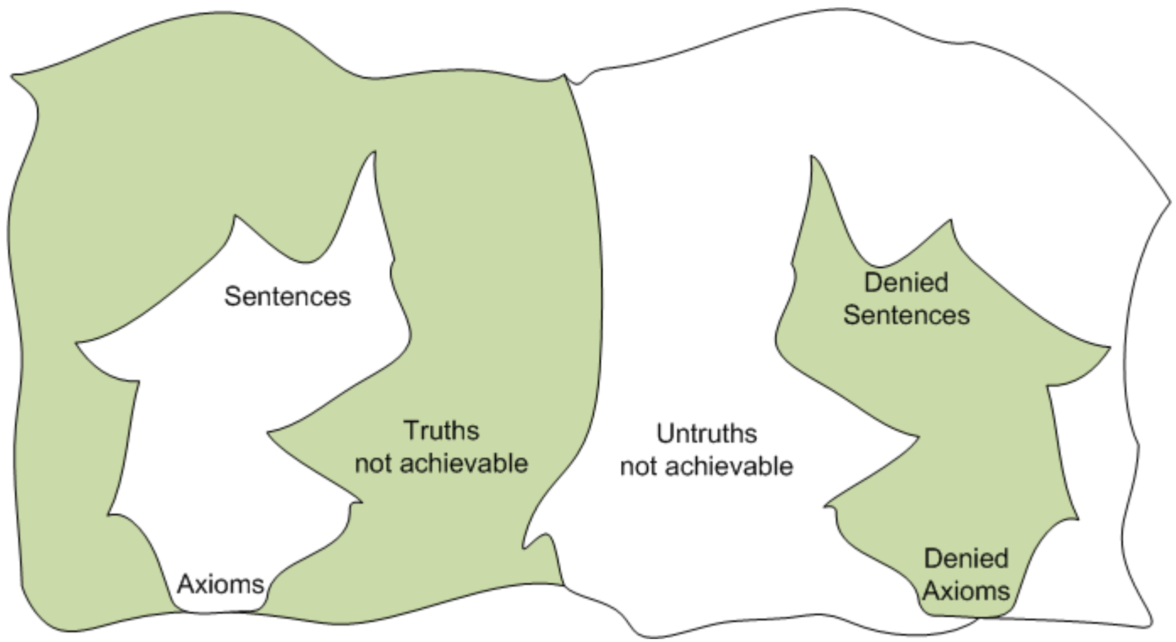
- Internacional Journals
  - Neural Networks, IEEE Transaction son
  - Fuzzy Systems, IEEE Transactions on
  - Intelligent Systems Engineering
  - Intelligent Systems, IEEE
  - Intelligent Transportation Systems, IEEE Transactions on

  .....
- Conferences

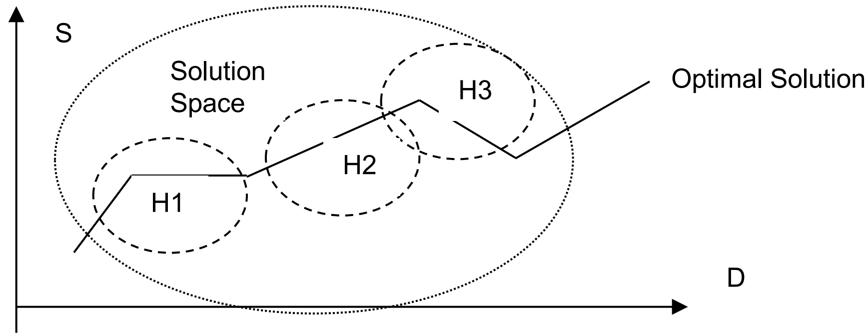
  .....

Epistemology –
"Philosophy of Knowledge"

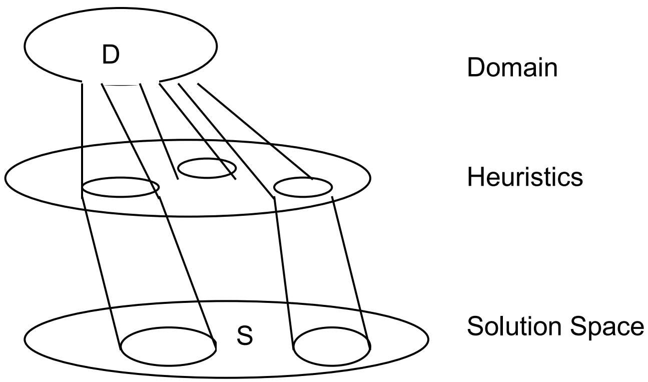




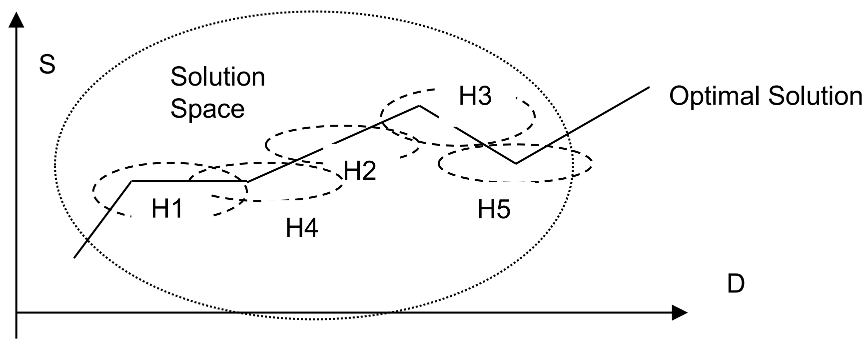
Heuristics



Heuristics give sub-optimal solutions.



A heuristic rule leads from the domain space to the solution space



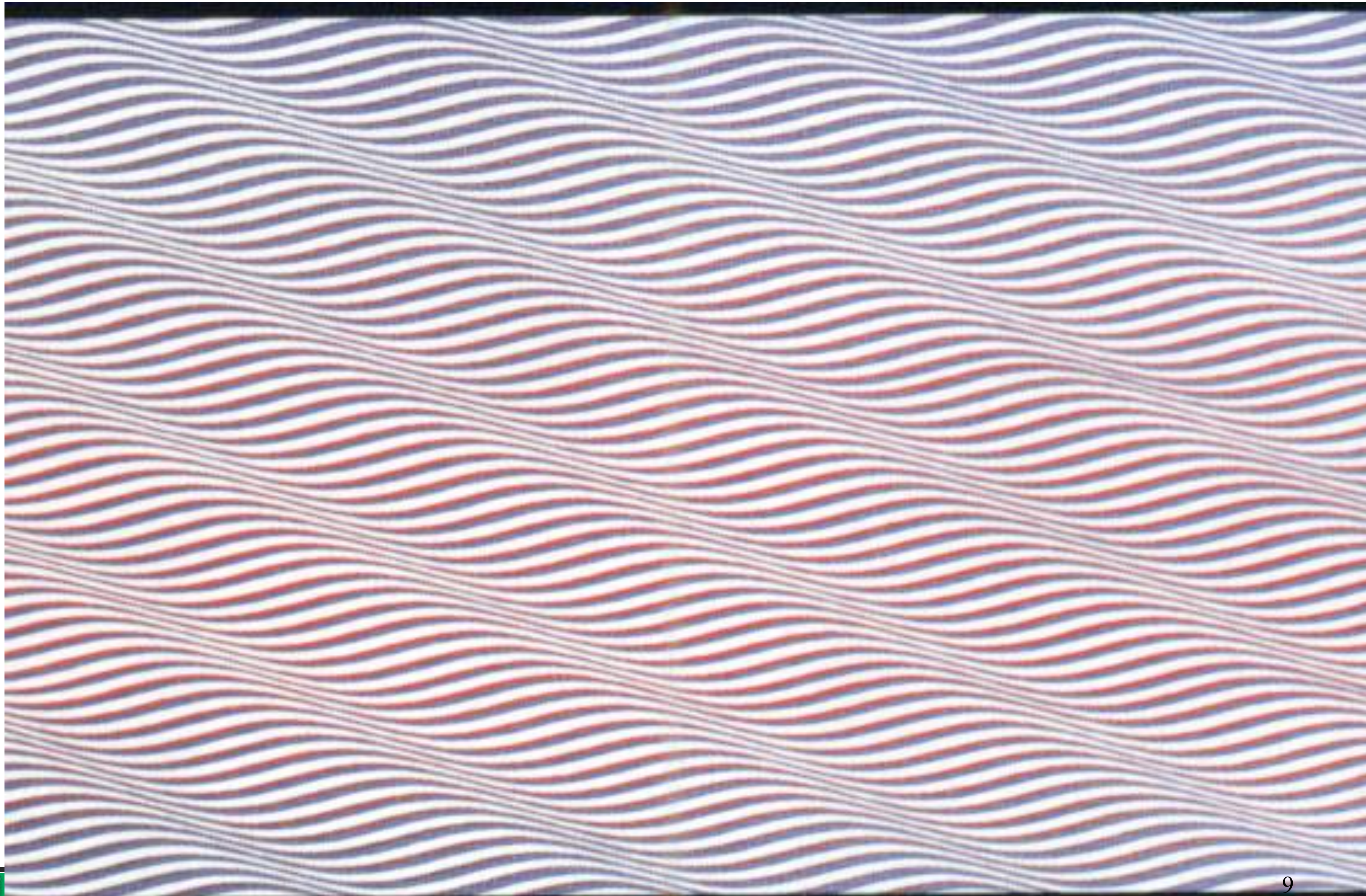
“Well-formed” Heuristics are close to the optimal solution.

Dislexy?

I cnduo't bvleiee taht I culod aulaclty uesdtannrd waht I was rdnaieg. Unisg the icndeblire pweor of the hmuan mnid, aocdcrnig to rseecrah at Cmabrigde Uinervtisy, it dseno't mttaer in waht oderr the lterets in a wrod are, the olny irpoamtnt tihng is taht the frsit and lsat ltteer be in the rhgit pclae. The rset can be a taotl mses and you can sitll raed it whoutit a pboerlm. Tihs is bucseae the huamn mnid deos not raed ervey ltteer by istlef, but the wrod as a wlohe. Aaznmig, huh? Yaeh and I awlyas tghhuot slelinpg was ipmorantt! See if yuor fdreins can raed tihs too.

Giant x 3D Ilusion?

Waves?



Simpathic?


GRIMASSE STEHT KOPF
Auf den ersten Blick scheint das Foto von Margaret Thatcher nicht ungewöhnlich. Das ändert sich, wenn Sie das Bild auf den Kopf stellen. Der verblüffende Effekt tritt auf, weil Augen und Mund auf dem Foto um 180 Grad gedreht sind – und damit genau die Merkmale, auf die das Gesichtserkennungsprogramm des Hirns besonders sensibel anspricht.
EMILE LUIDER / RAPHO / AGENTUR FOCUS (L.); PERCEPTION (R.)

Antipathic?


EMILE LUIDER / RAPHO / AGENTUR FOCUS (L.); PERCEPTION (R.)
GRIMASSE STEHT KOPF
Auf den ersten Blick scheint das Foto von Margaret Thatcher nicht ungewöhnlich. Das ändert sich, wenn Sie das Bild auf den Kopf stellen. Der verblüffende Effekt tritt auf, weil Augen und Mund auf dem Foto um 180 Grad gedreht sind – und damit genau die Merkmale, auf die das Gesichtserkennungsprogramm des Hirns besonders sensibel anspricht.

Introduction

- Connectionist Intelligent Systems

Artificial Intelligence

Science field that studies **paradigms**
that aims to explain how
intelligent behaviour
can **emerge**
from artificial implementations, in computers.

Intelligence:
learning, adaptation, comprehension

IA Paradigms

- AI Paradigms
  - Symbolist → (Lisp, Prolog) - Expert Systems
  - Connectionist (numerical)
    - Symbolic - Fuzzy
    - Sub-Symbolic - ANN

Intelligence :
learning, adaptation, comprehension

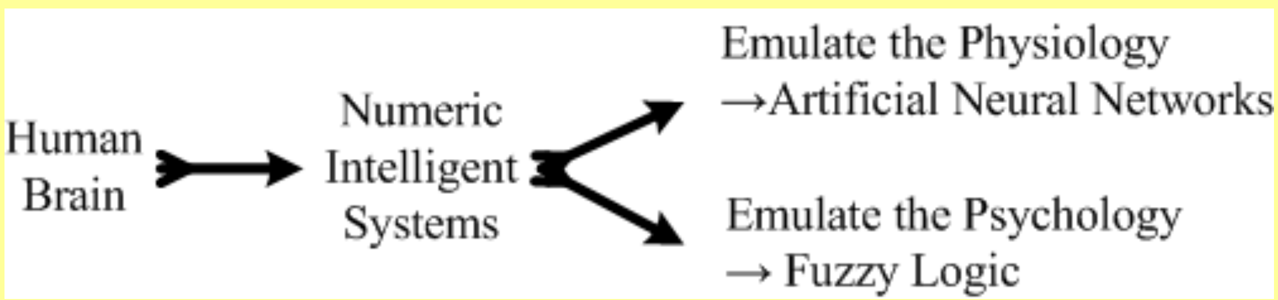
Connectionist Paradigm

Considers to be virtually impossible to ***transform in algorithms*** - i.e., to reduce to a sequence of logical and arithmetic steps many tasks that the human mind performs with ease and speed, for example:

- Face Recognition,
- Comprehend and translate natural languages,
- Memory evocation by association,
- Games...

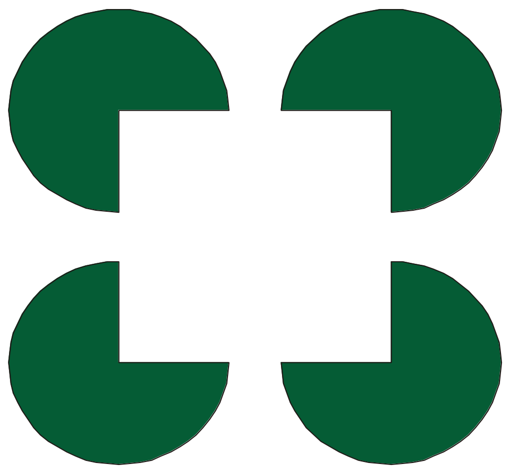
Connectionist Paradigm

The computational process have to mimic
the brain capacity of self-organization → **learn!**



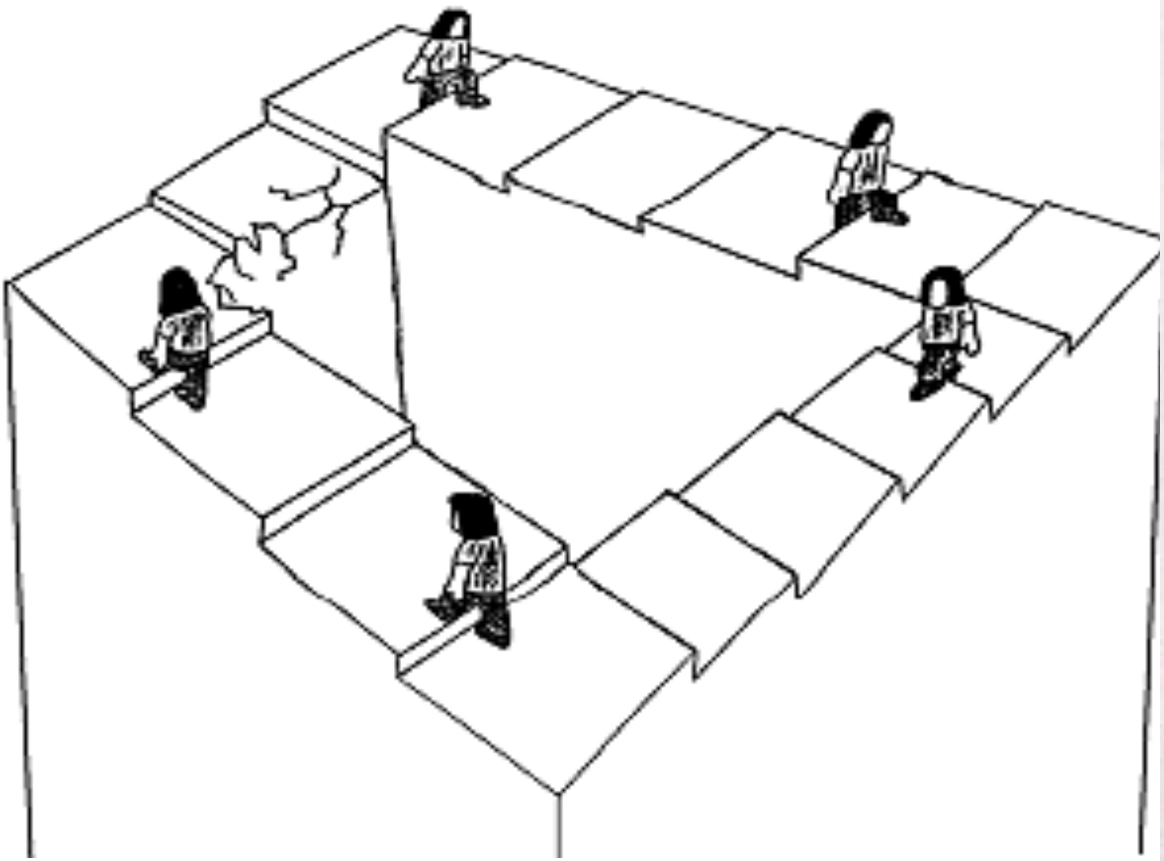
Symbolist versus Conectionist Paradigm

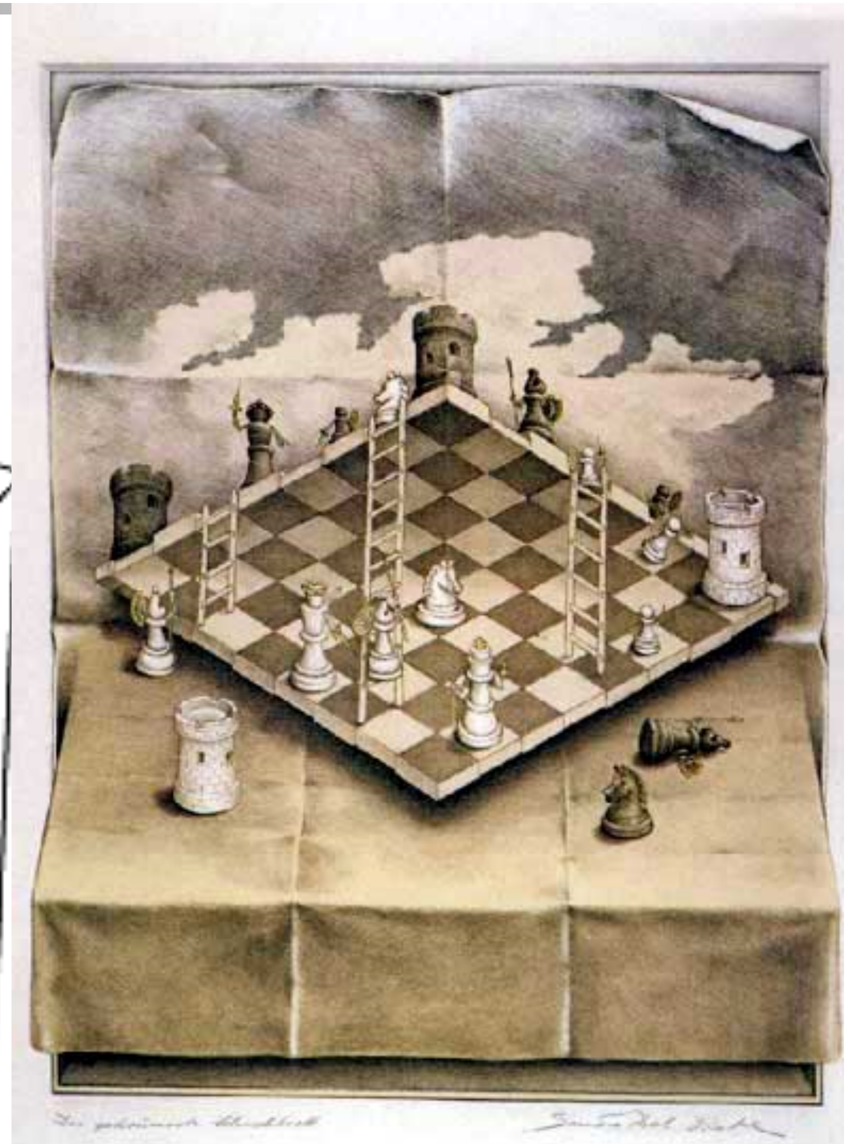
-Perception



The Kanizsa square, 1976

"Local Coherency –Global Paradox"





Sandro delPrete. Enigmas Visuais. Rio de Janeiro, 2004, p. 45

M.C.Escher

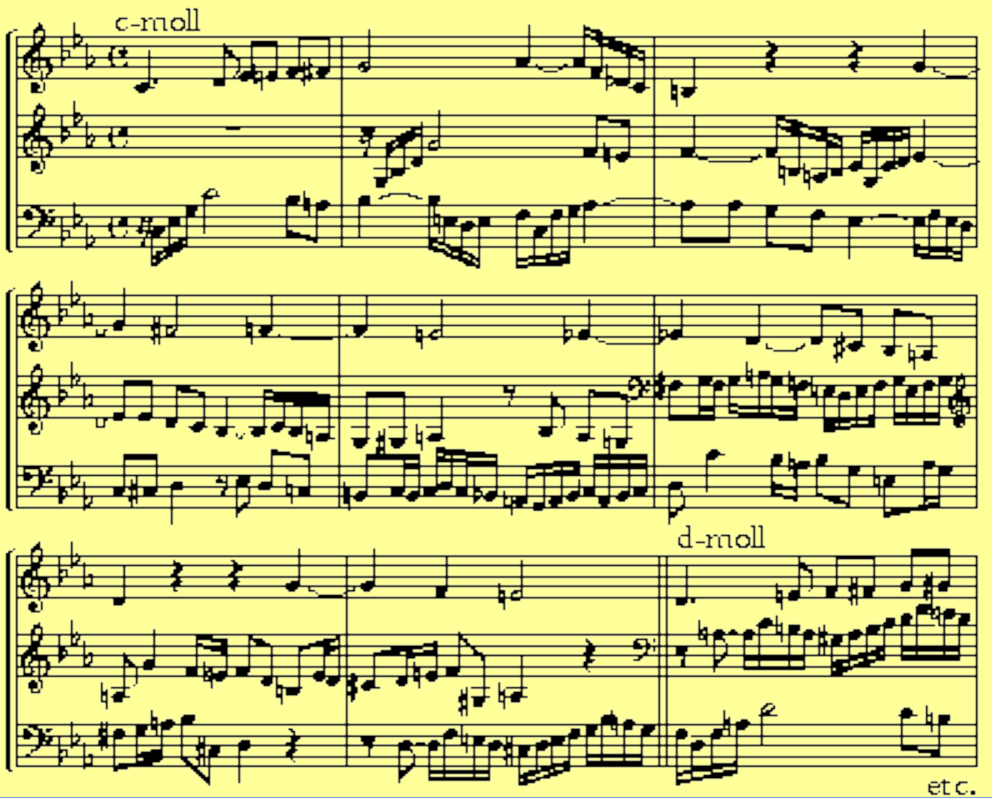
“Positive Truth X Negative Truth”

"Up stairs
X
Down stairs"

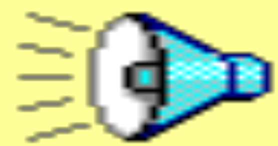
Paradigma
Simbolista versus Conexionista

- J.S. Bach

"Coerência Local - Paradoxo Global"



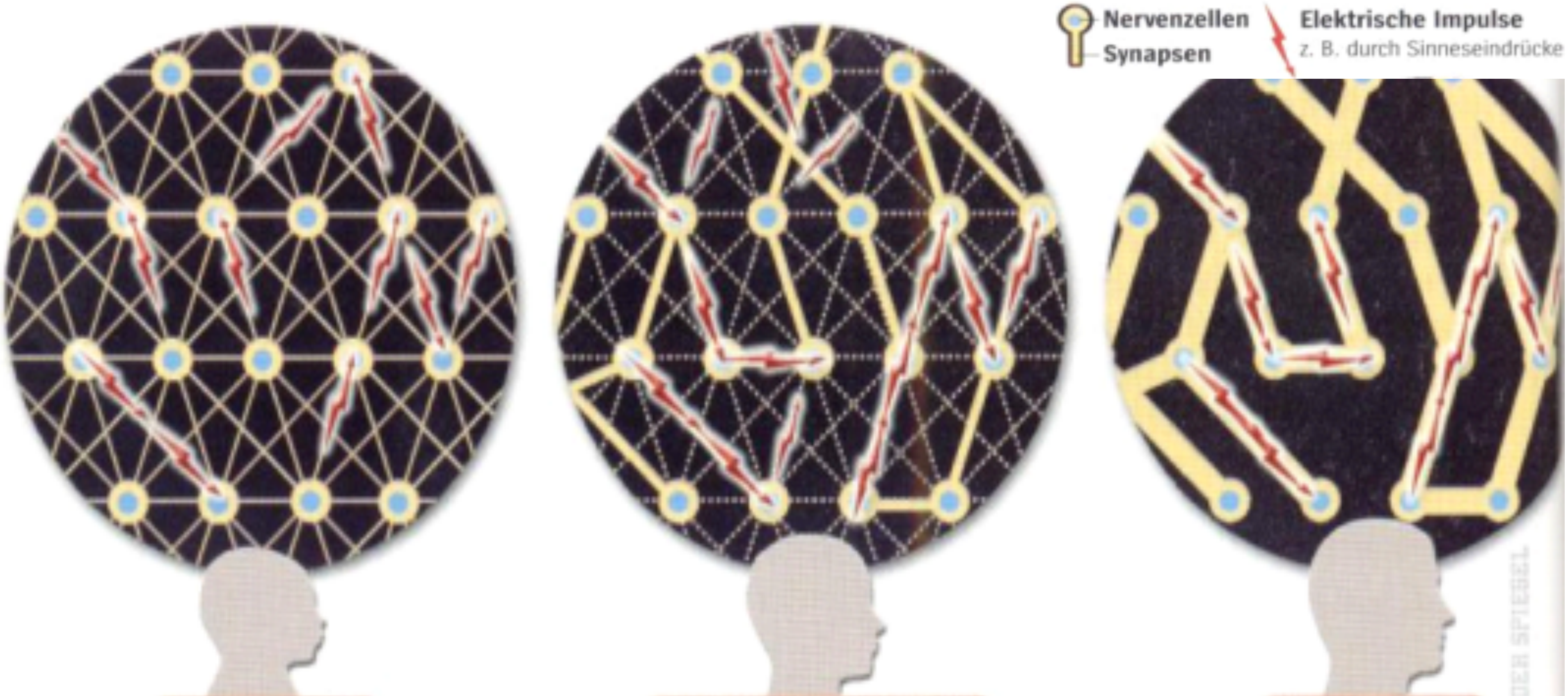
Shepard's scale



Pseudo-rising scale

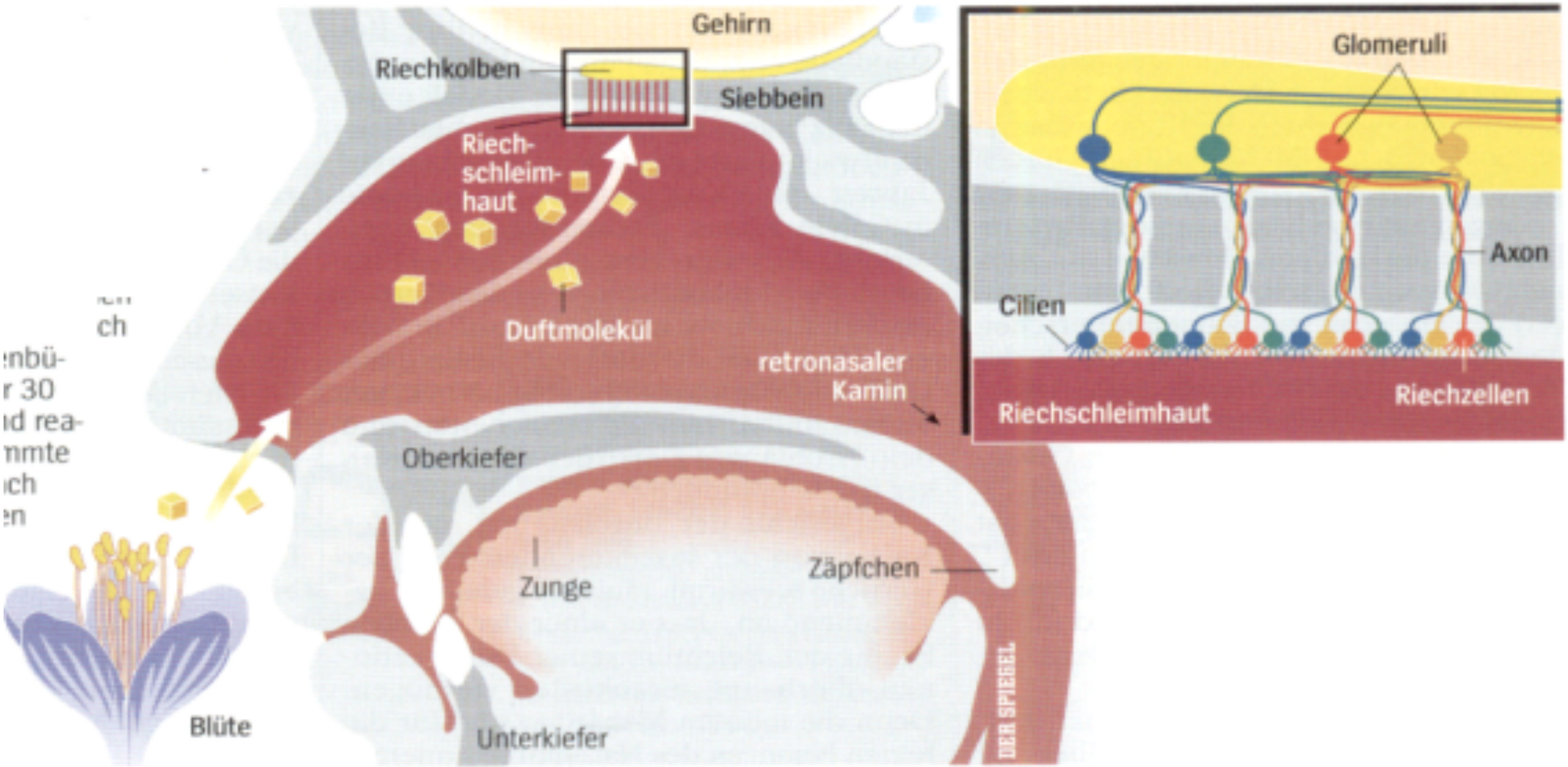
played on a vibraphone

Synapses Formation

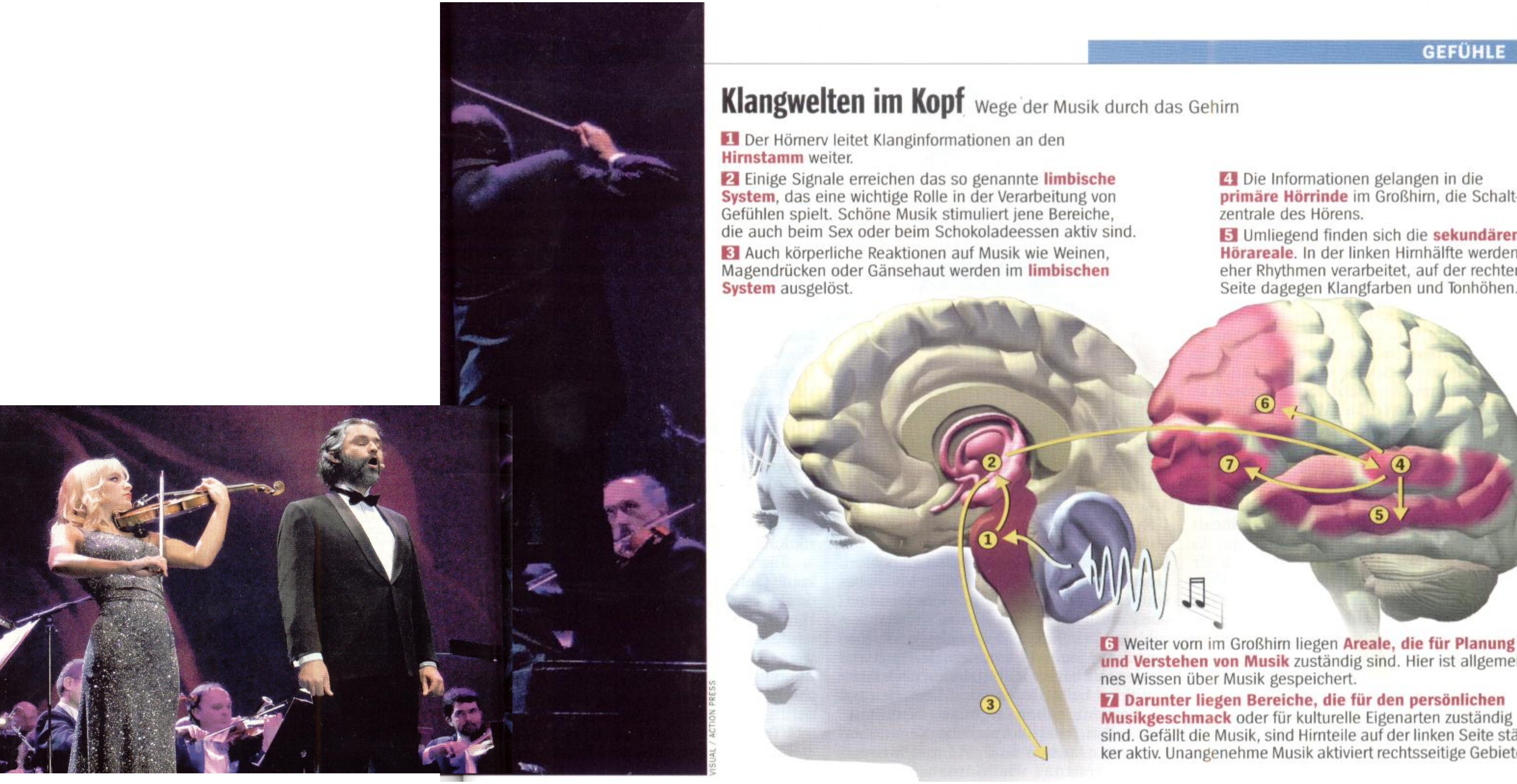


0-2 years | 2 years to puberty | Adult

Olfative Information Processing



Auditive Information Processing


GEFÜHLE

Klangwelten im Kopf Wege der Musik durch das Gehirn

1 Der Hörnerv leitet Klanginformationen an den **Hirnstamm** weiter.

2 Einige Signale erreichen das so genannte **limbische System**, das eine wichtige Rolle in der Verarbeitung von Gefühlen spielt. Schöne Musik stimuliert jene Bereiche, die auch beim Sex oder beim Schokoladeessen aktiv sind.

3 Auch körperliche Reaktionen auf Musik wie Weinen, Magendrücken oder Gänsehaut werden im **limbischen System** ausgelöst.

4 Die Informationen gelangen in die **primäre Hörrinde** im Großhirn, die Schaltzentrale des Hörens.

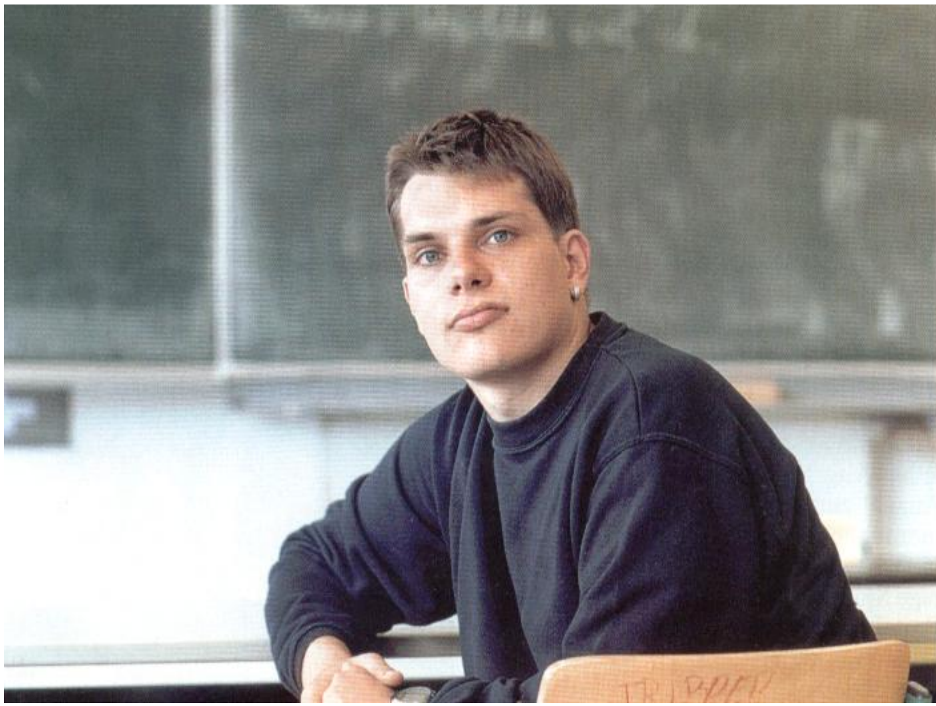
5 Umliegend finden sich die **sekundären Höräreale**. In der linken Hirnhälfte werden eher Rhythmen verarbeitet, auf der rechten Seite dagegen Klangfarben und Tonhöhen.

6 Weiter vorn im Großhirn liegen **Areale, die für Planung und Verstehen von Musik** zuständig sind. Hier ist allgemeines Wissen über Musik gespeichert.

7 **Darunter liegen Bereiche, die für den persönlichen Musikgeschmack** oder für kulturelle Eigenarten zuständig sind. Gefällt die Musik, sind Hirnteile auf der linken Seite stärker aktiv. Unangenehme Musik aktiviert rechtsseitige Gebiete.

VISUAL / ACTION PRESS

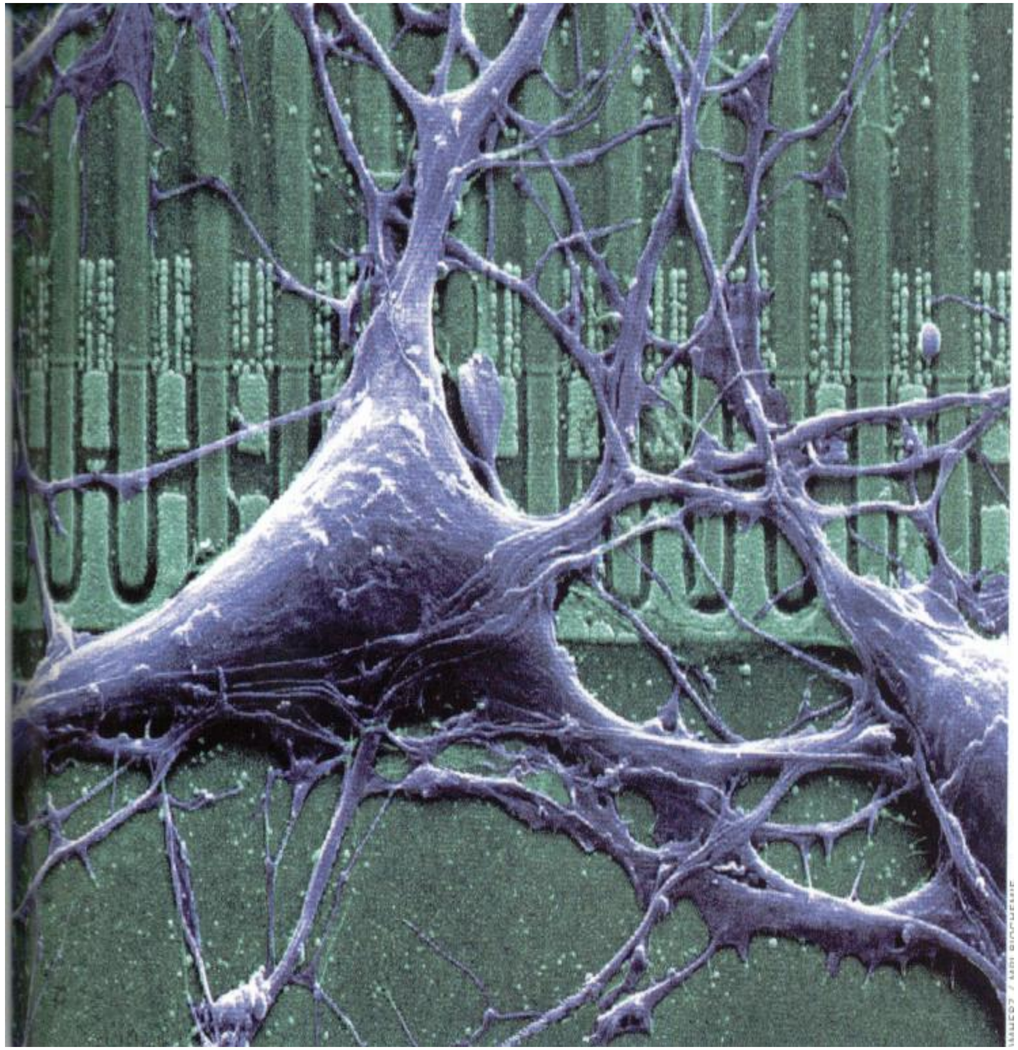
Epilepsy Patient –
without left brain hemisphere since 12 years age



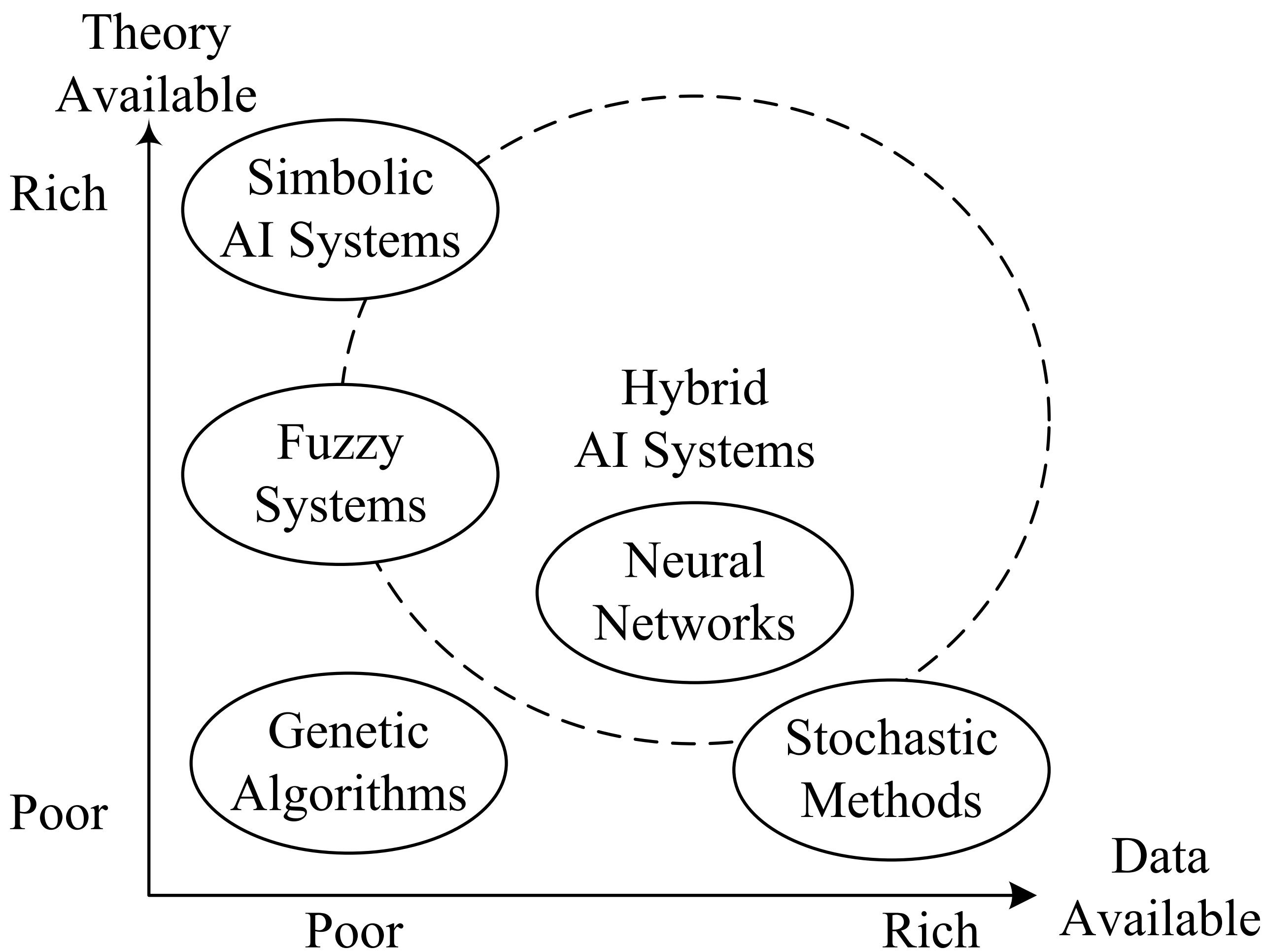
Leben ohne links

Seit zwölf Jahren lebt Philipp Dörr mit einem halben Großhirn. Trotzdem spielt er Schach, liest Goethe und taucht – ein verblüffendes Beispiel für die Wandlungsfähigkeit des Denkorgans.

A neuron over a Chip.



Methods in Knowledge Engineering



Gödel's Conjecture

Kurt Gödel presented in 1931 a conjecture
that shook the dominant mathematics conviction. In simple words Gödel's conjecture is:

"Every axiomatic formulation free of contradictions in number theory contains sentences that can not be verified nor denied"

Otherwise a **computer programs** could always be written **to solve any problem** that could be formalized!

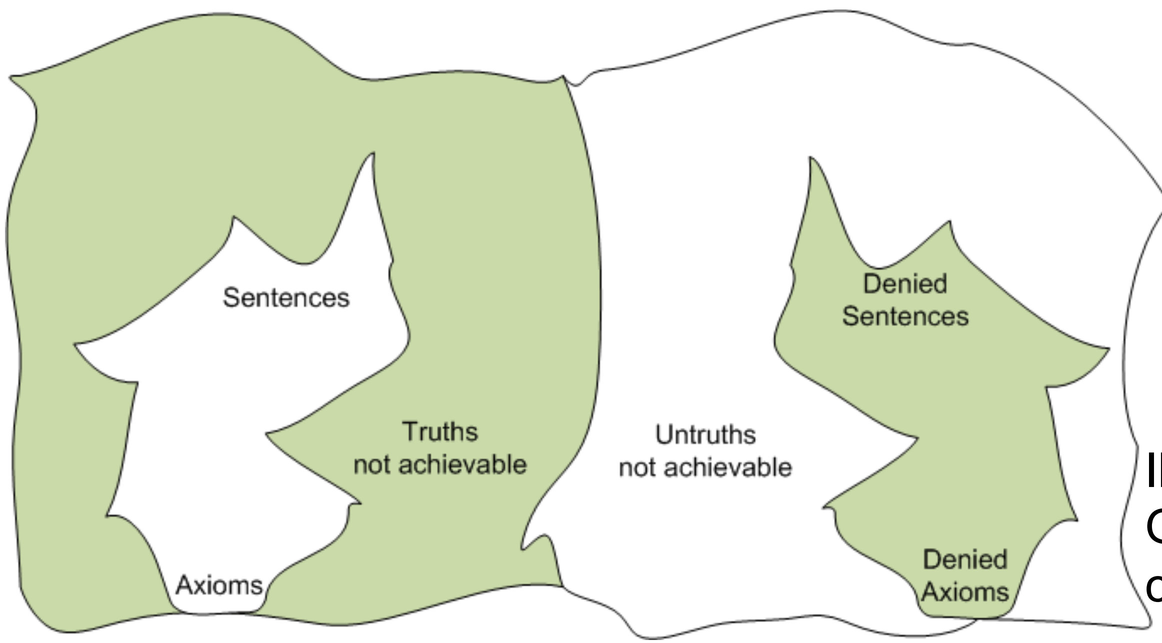


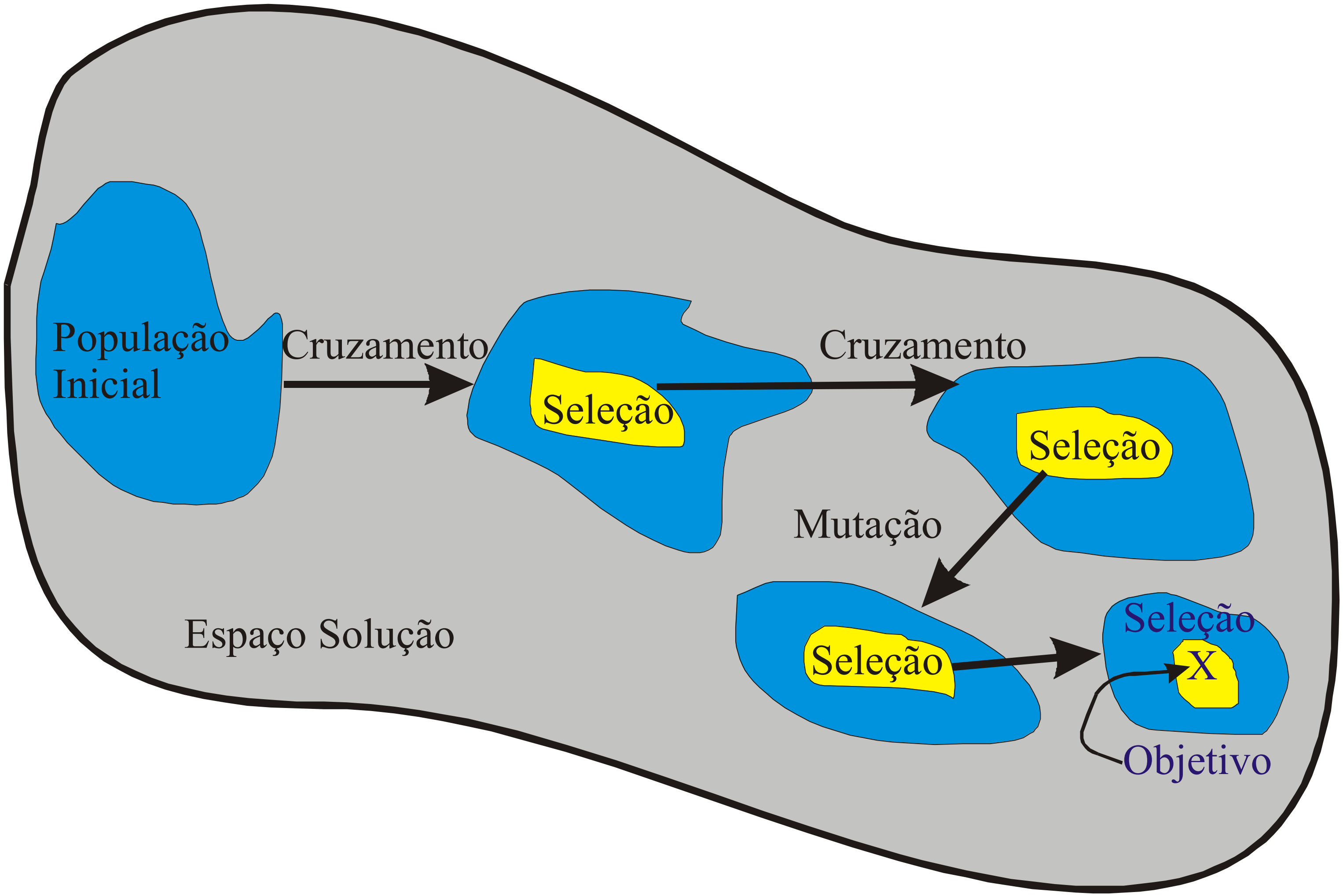
Illustration of Gödel's conjecture

Genetic Algorithms

- Genetic algorithms, as well as the so-called evolutionary computing, are **heuristic methods** for solving problems.
- John Holland, proposed GAs in 1975, inspired by Darwin's evolution theory

- Basic Concepts
  - Gen
  - Chromosome - represents an individual, a possible solution
  - Population
  - Crossing
  - Mutation
  - Evaluation Criteria
  - Selection

Graphical representation of the GA



Example: Mastermind game

guess the number 001 010

The selection criterion is the proximity to the number 001010 (number of correct bits).

Initial Population	Evaluation	Selection
A 010101	1	
B 111101	1	
C 011011	4	*
D 101100	3	*

New population in which the individuals C and D are the parents.

	New Population	Evaluation	Selection
C 01:1011	E 01 11 00	3	
D 10:1100	F 10 10 11	4	*
C 0110:11	G 01 10 00	4	*
D 1011:00	H 10 11 11	3	

Example: Mastermind game (cont.)

guess the number 001 010

New population in which the individuals F and G are the parents.

	New Population	Evaluation	Selection
F 1:0 10 11	I 11 10 00	3	
G 0:1 10 00	J 00 10 11	5	*
F 10 1:0 11	K 10 10 00	4	*
G 01 1:0 00	L 01 10 11	4	

New population in which the individuals J and K are the parents.

	New Population	Evaluation	Selection
J 00 10: 11	M 00 10 00	5	
K 10 10: 00	N 10 10 11	5	*
J 00 10 1:1	O 00 10 10	6	Success – END
K 10 10 0:0	P 10 10 01	3	

Success after 16 questions.

By exhaustive search we would have 2^6 = 64 possible questions.

Universal Approximators

A heuristic rule is of the form:

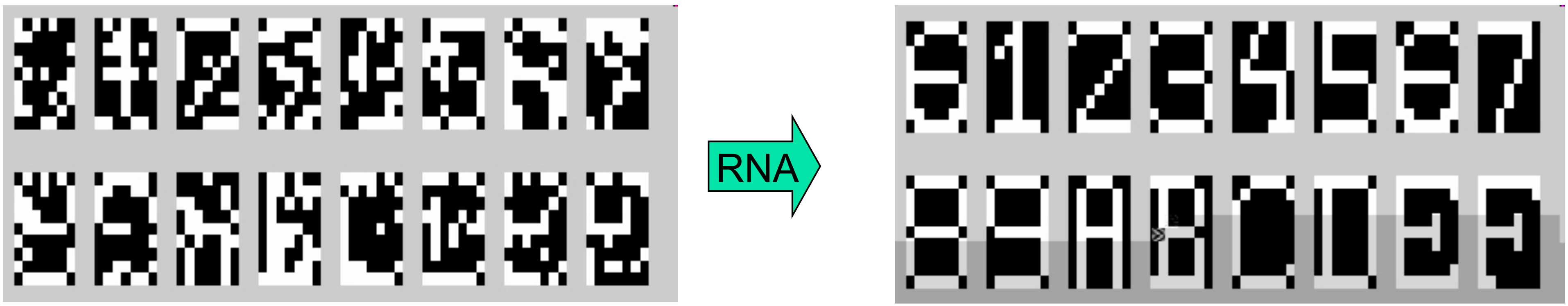
IF <condition> THEN <conclusão>

- fuzzy systems - heuristic knowledge represented by fuzzy rules.
- artificial neural networks - learn heuristics from the data.

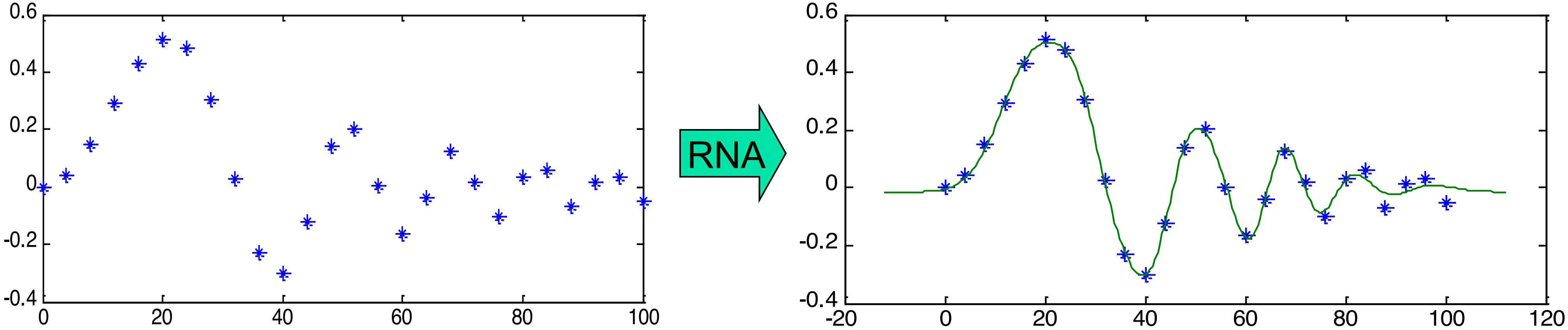
Both are ***universal approximators***,
that is, they can approximate any function
with **arbitrary precision**.

ANN Applications

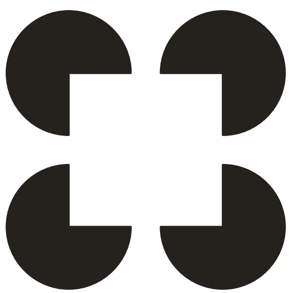
- Pattern Classification

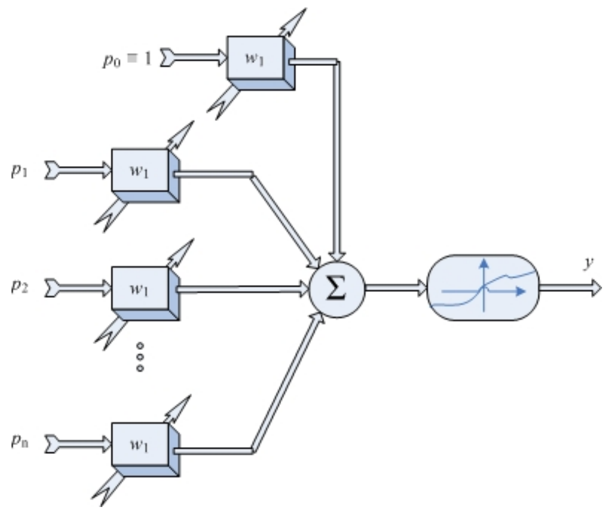


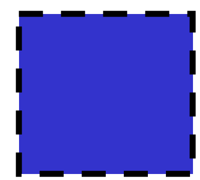
- Function Approximation (non linear)



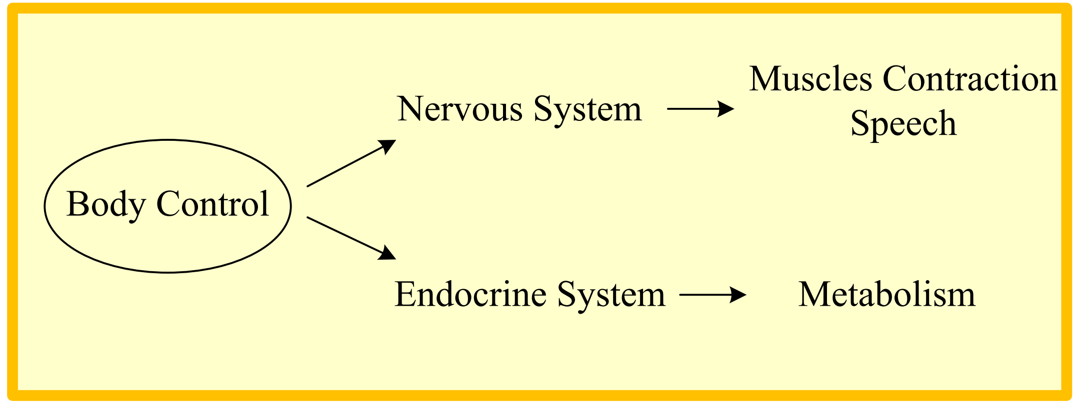
Part 2 – Artificial Neural Networks





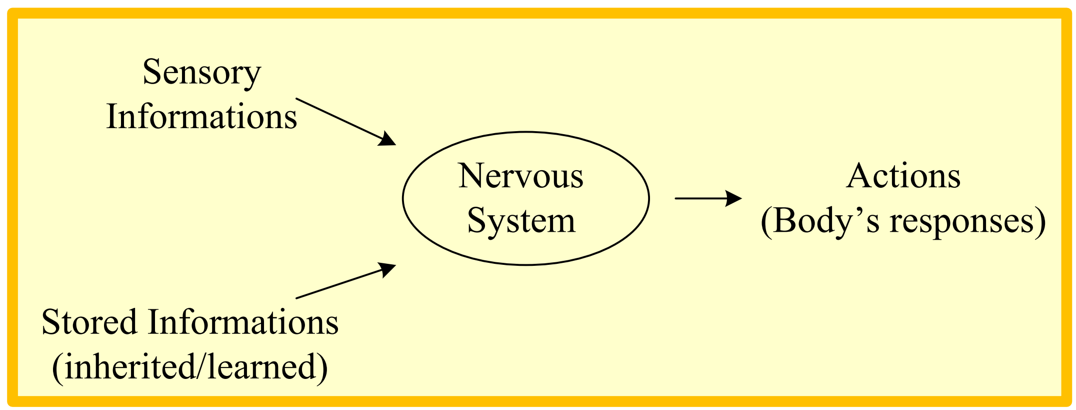


Biological Fundaments



The nervous systems and the endocrine systems control the body.

The nervous systems gatters information from the surroundings through sensors wich are combined with stored information to produce the actions of the body.

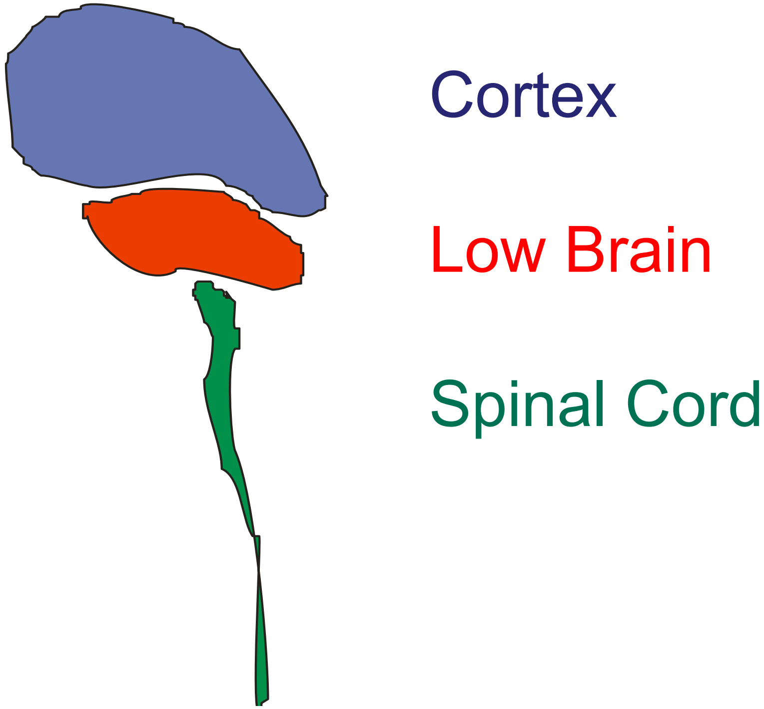


Sensors + Memory + Inference → Action.

Only a small part of the captured information is relevant to the well functioning of the body.

Biological Fundaments

The nervous system comprehends three levels.



Each one build by
neurons of different anatomies.

It is estimated that the human brain has about 10^{11} neurons,

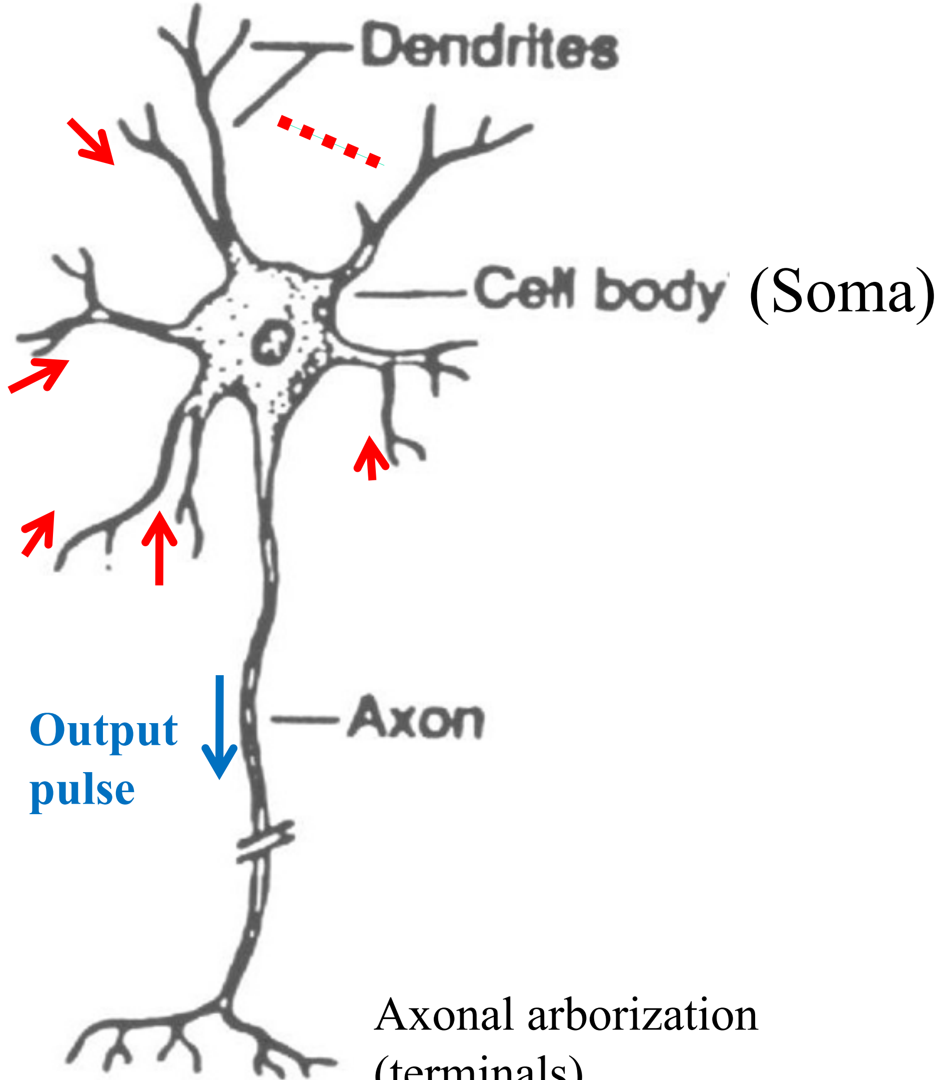
Whose summed lenght reaches 10^{14} meters.

Biological Fundaments

Brain Information
Processing Levels

- Structural
- Physiologic
- Cognitive

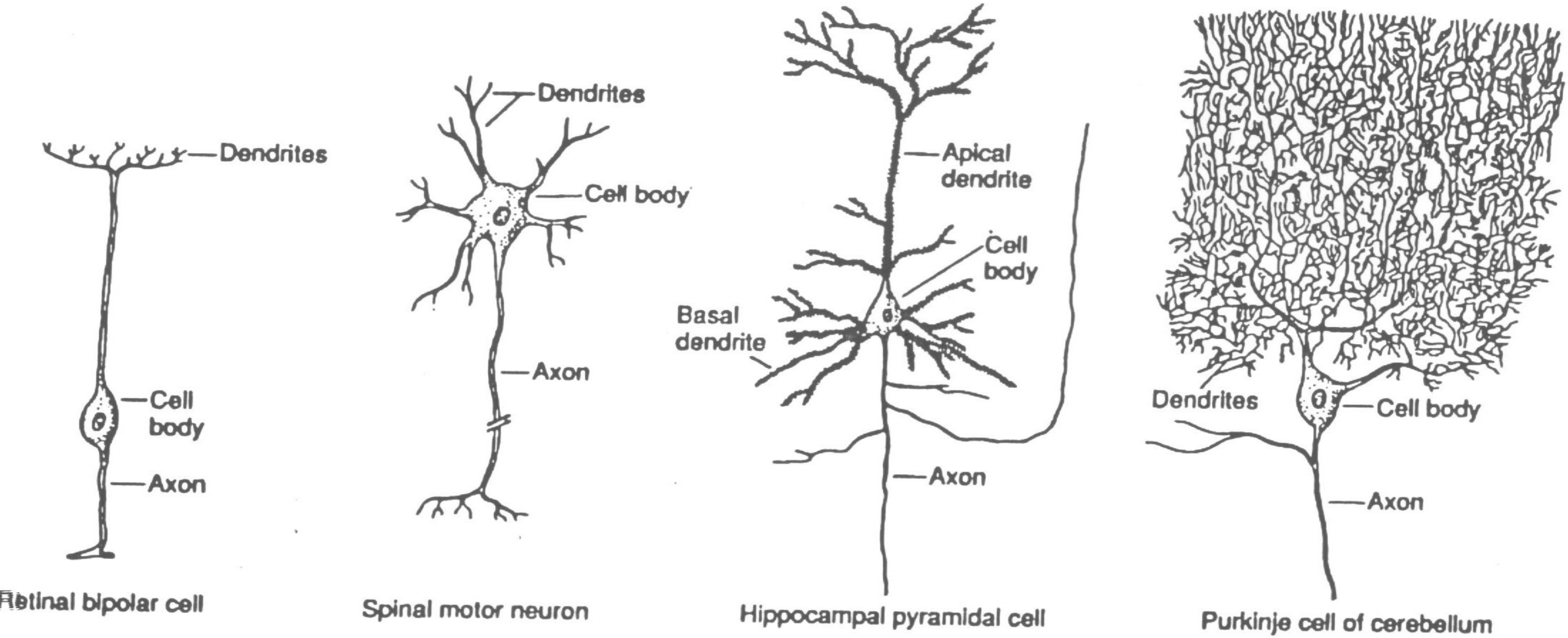
Input Pulses



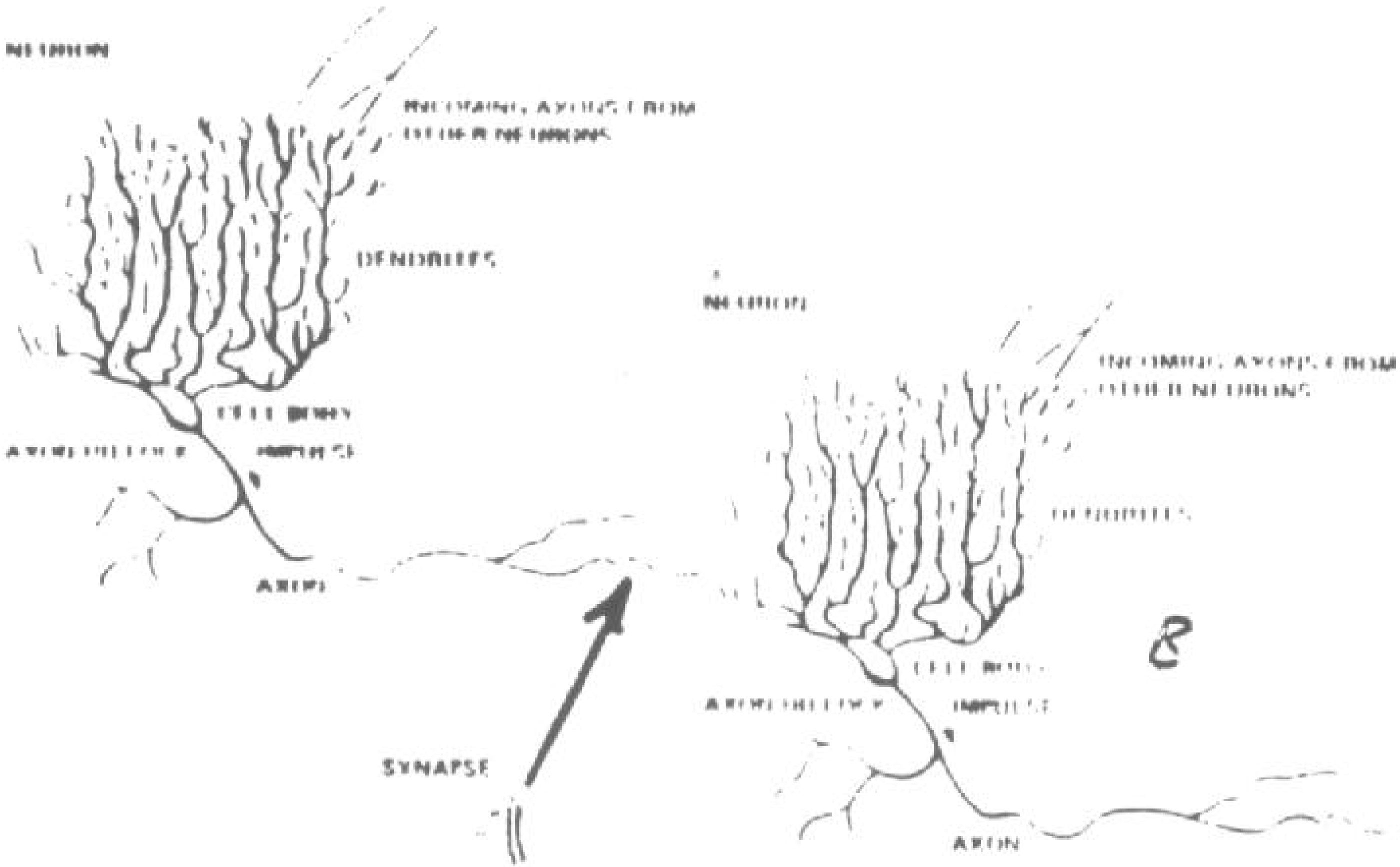
The information flow (electric current)
is always from the dendrites to the axon.

Biological Fundaments

Some kinds of neurons

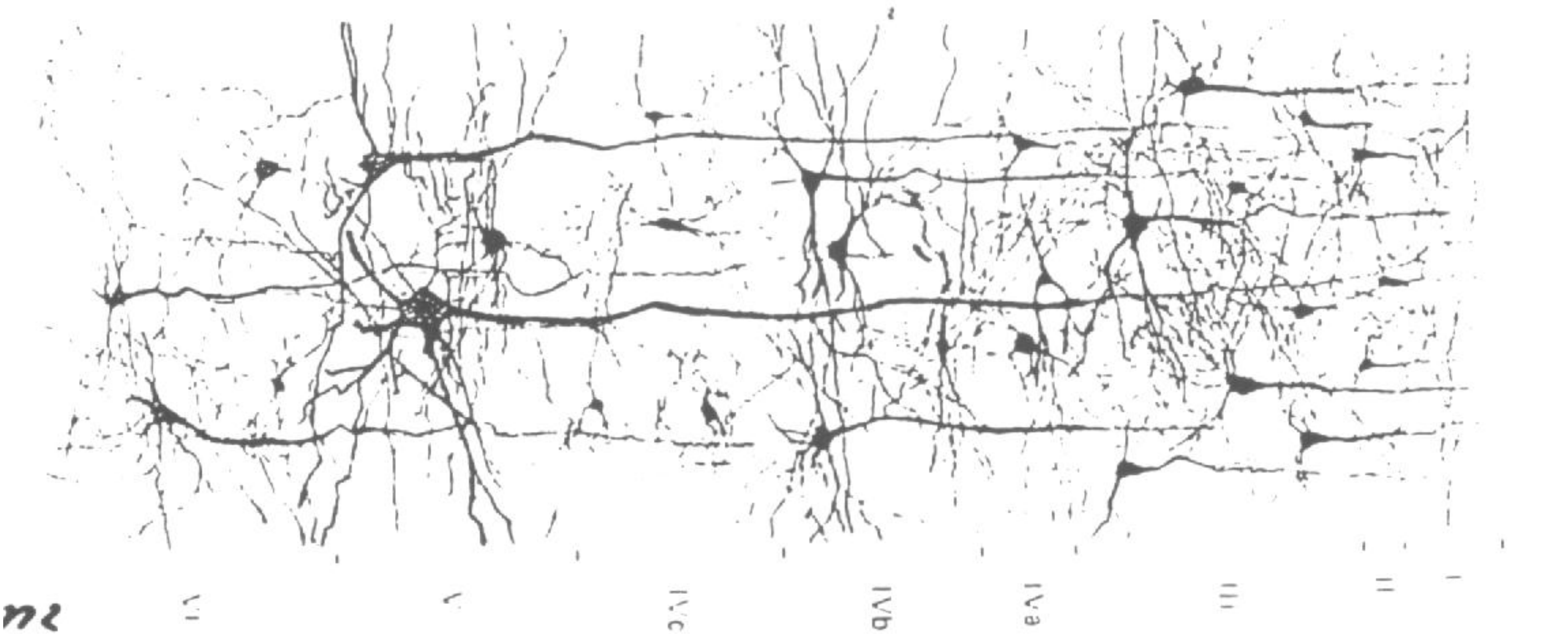


Synaptic Connection

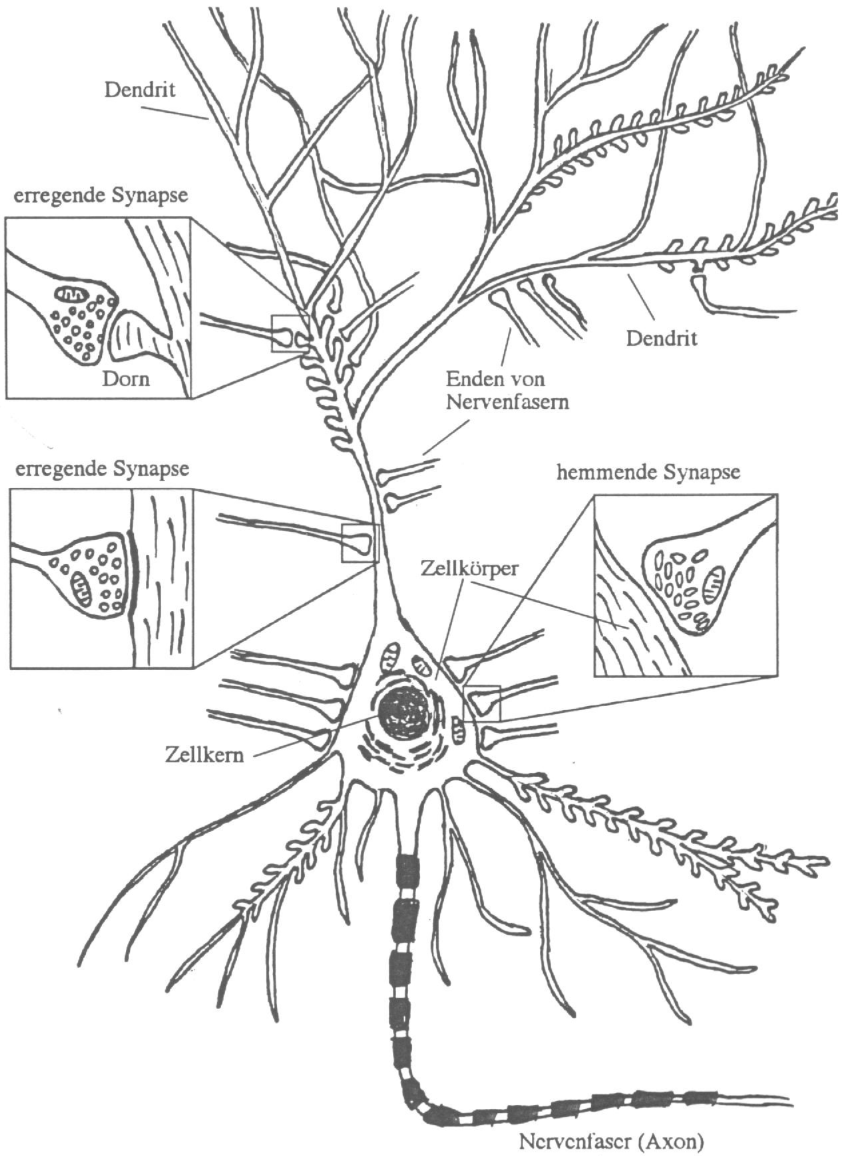


Biological Fundaments

Connection pattern: mostly in layers



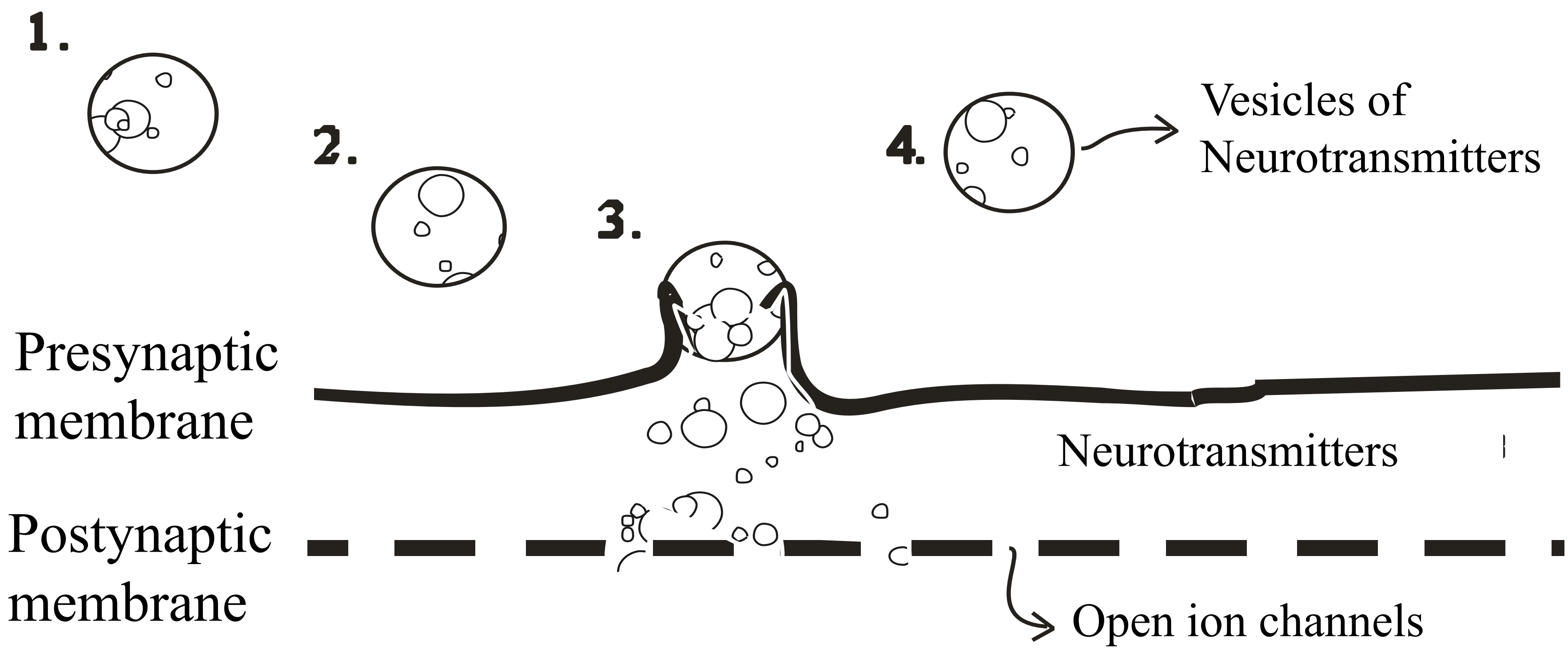
Excitatory and Inhibitory Synapses



	Excitatory	$A\uparrow \Rightarrow B\uparrow$
Synapses / \		
	Inhibitory	$A\uparrow \Rightarrow B\downarrow$

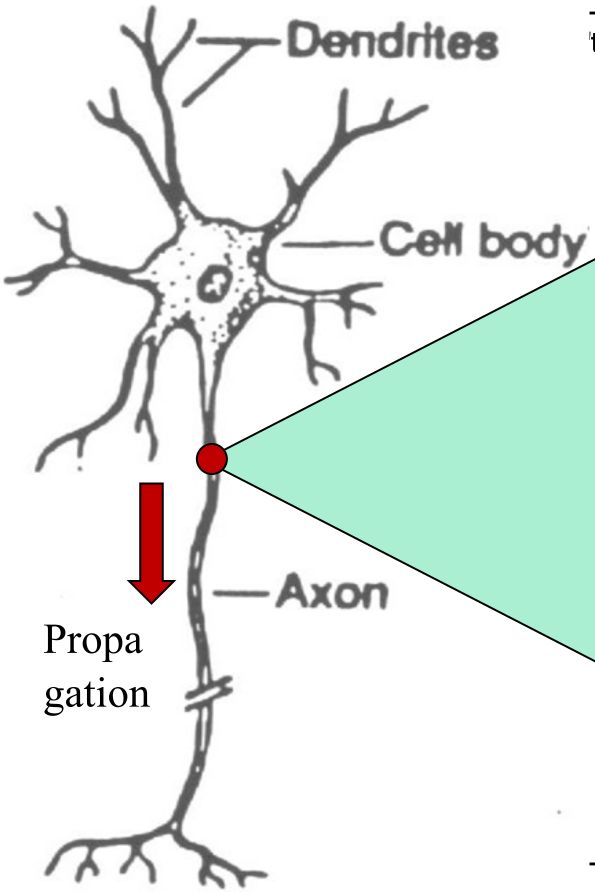
Biological Fundaments

Neurotransmiters in the synaptic gap



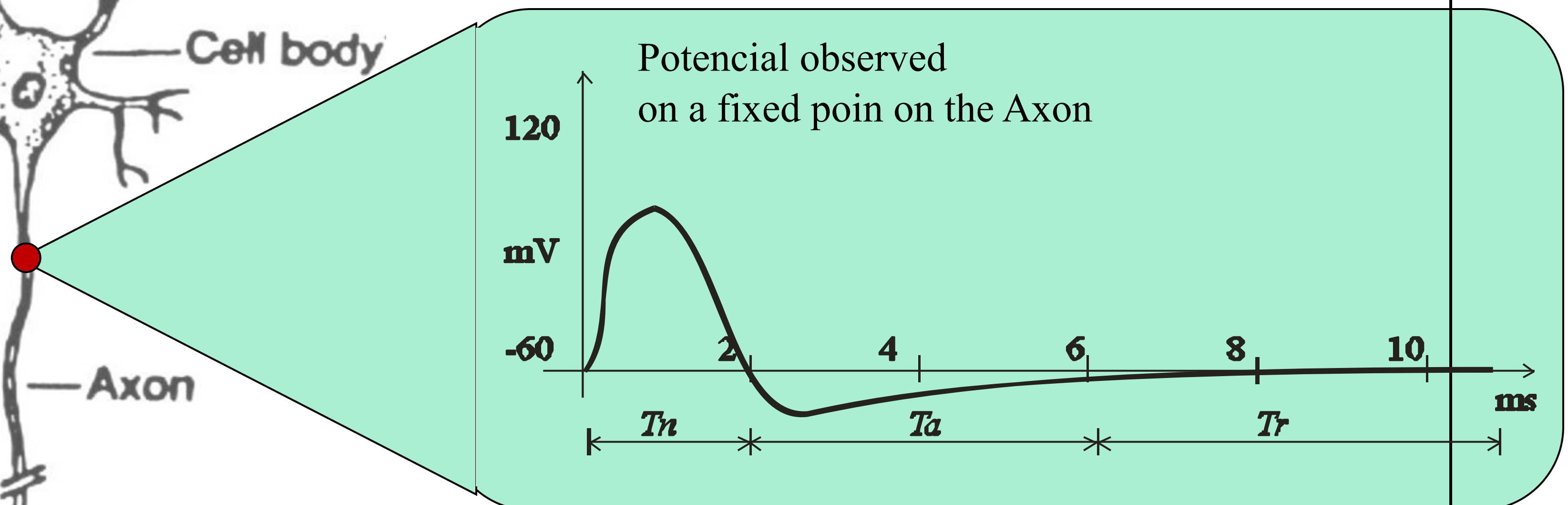
The Action Potential

Axon –
"distribuited R-C Transmission Line"



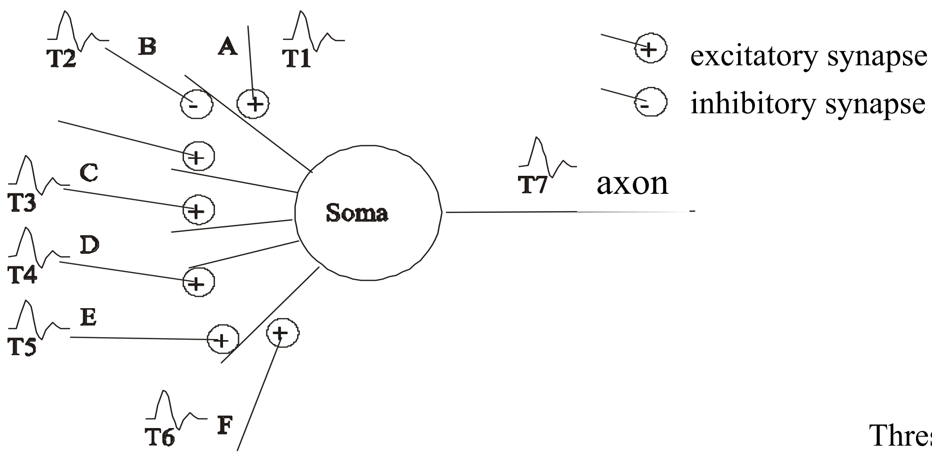
't be displayed.

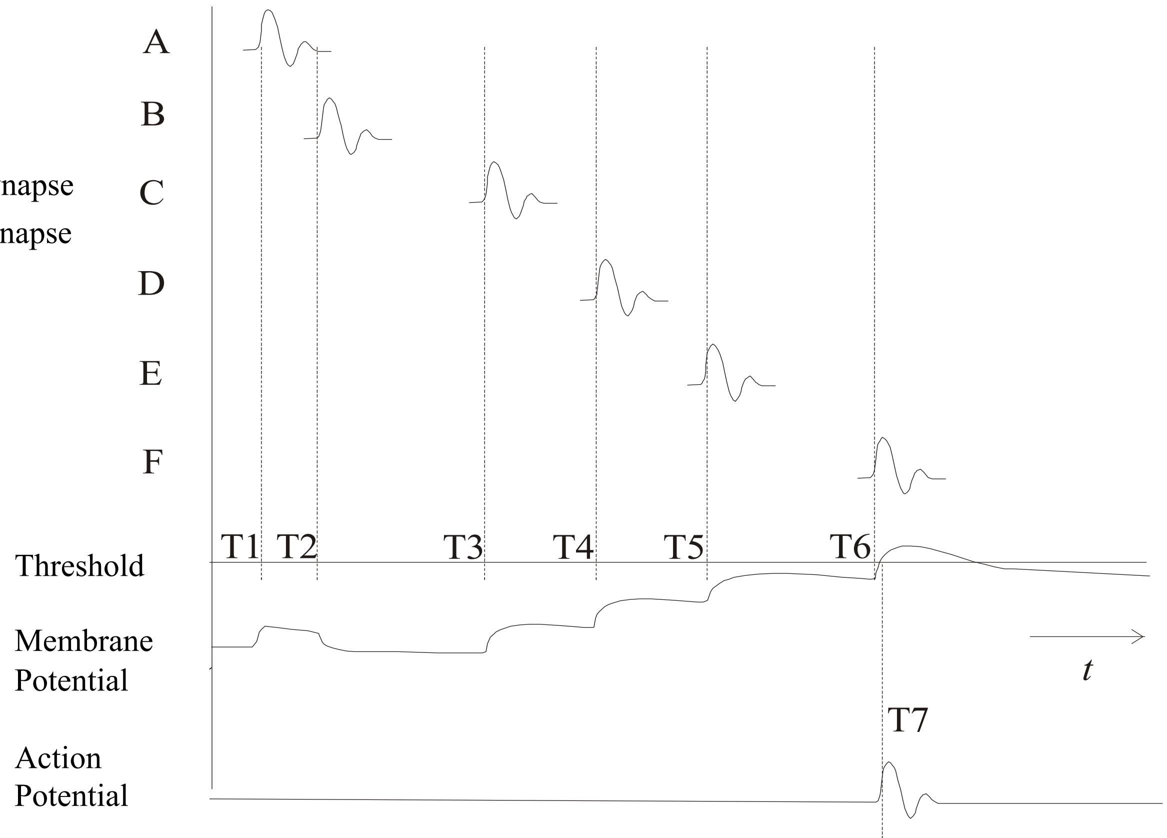
Propa
gation



T_n – length of the nerve impulse
T_a – period of absolute refraction
T_r – period of relative refraction

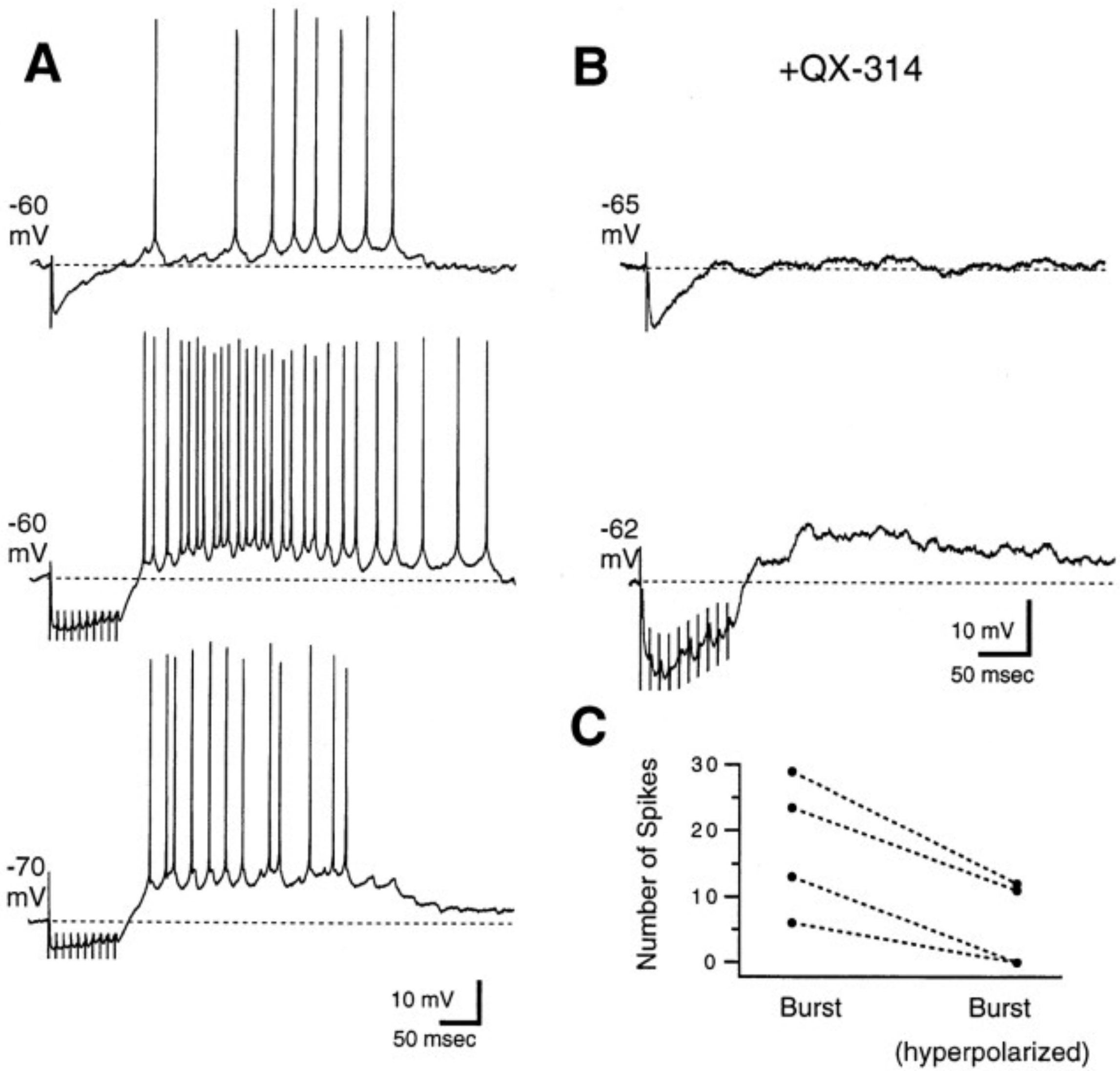
Space/Time Integration of the stimuli





Anatomy - structure
Physiology – function, operating modus

Pulse frequency as the information



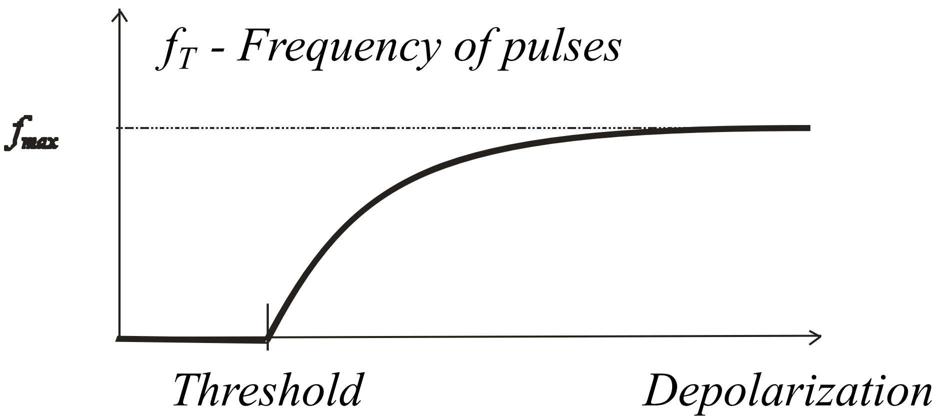
Biological System:
Information is coded using
Pulse Frequency Modulation

Aizenman et al.,
Polarity of Long-Term Synaptic Gain Change
is Related to Postsynaptic Spike Firing at a
Cerebellar Inhibitory Synapse,
Neuron, Vol. 21, 4, 1998.

Space/Time Integration

Maximum frequency of pulses in the axon

$$f_{\max} = \frac{1}{T_a + T_n}$$

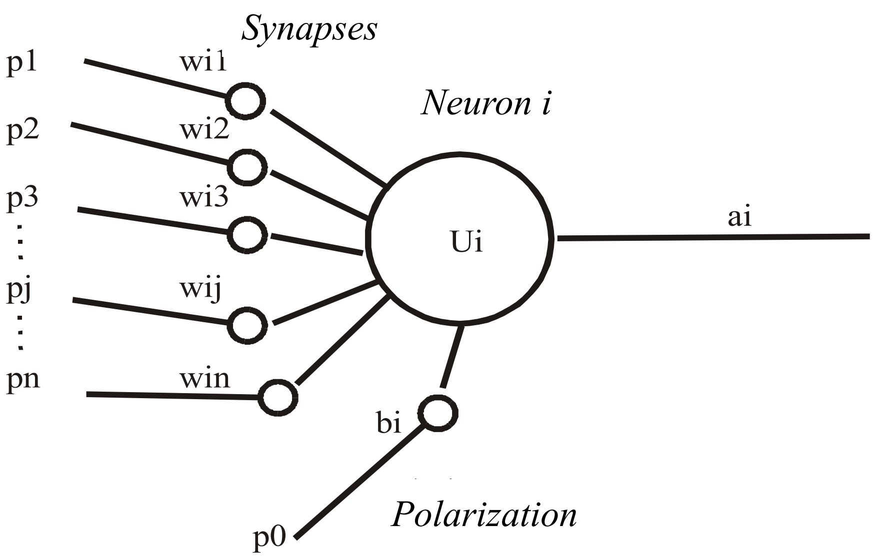


$$f_T = g\left(\int_0^T \sum_i \alpha_i(t) x_i(t) dt \right)$$

f_T – average frequency of nerve impulses in the time interval T,
$\alpha_i(t)$ – synaptic gains,
$x_i(t)$ – inputs of neurons.

Difficult to implement as an Electrical circuit!!

Basic model of an artificial neuron

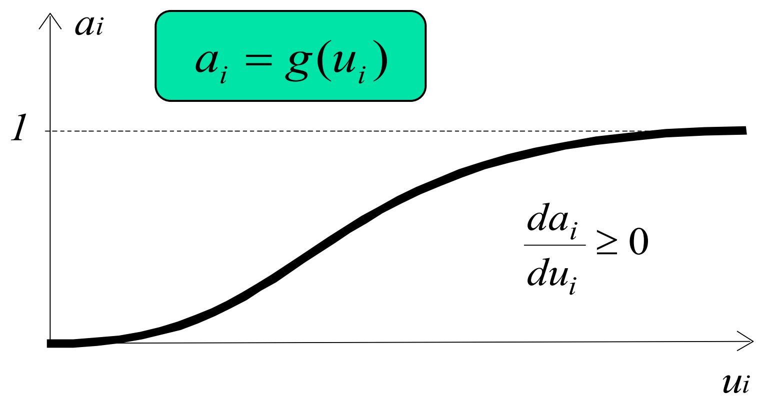


$$u_i = \sum_{j=1}^{n} w_{ij} p_j + b_i = \mathbf{w}_i^t \mathbf{p} + b_i$$

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{in} \end{bmatrix}.$$

Excitatory synapse $w_{ij} > 0$,

inhibitory synapse $w_{ij} < 0$.



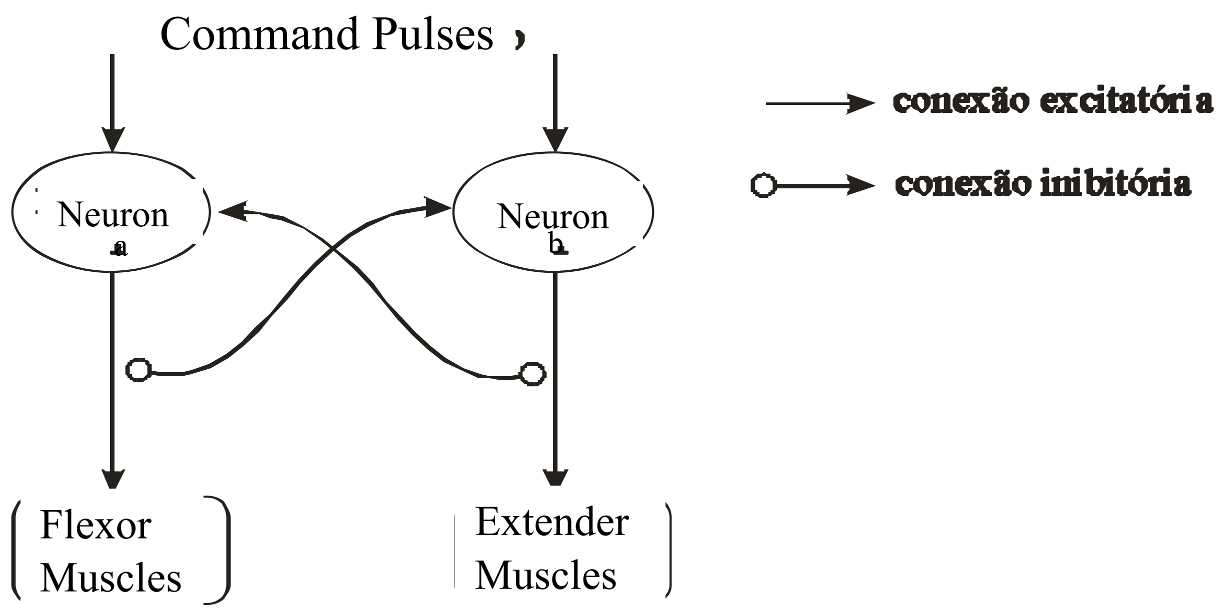
$$a_i = g(u_i)$$

$$\frac{da_i}{du_i} \geq 0$$

g(.) - Usually, a non-Linear activation function, e.g.: Sigmoid ("S" shaped)

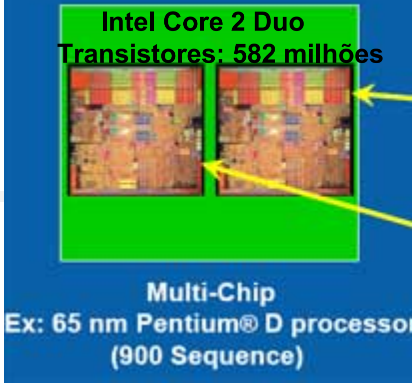
ANN – same funcionality
easy DSP implementation !!

Neurons with lateral connection



Neural circuit with antagonism inhibition

Comparison Brain x Computer



	Brain	*Computer*
# processing elements	~ 10^{11} neurons	~ 10^{9} transistors
Processing Form	Massively parallel	In general serial
Memory	Associative	Addressed
Switching time	~ 1 ms	~ 1 ns
Switchings /s	~ 10^{3} /s	~ 10^{9} /s
Total Switchings (theory)	~ 10^{14} /s	~ 10^{18} /s
Total Switchings (real)	~ 10^{12} /s	~ 10^{10} /s

100 steps rule

People recognize a familiar face in ~ 0.1 s.

Considering 1ms per neuron: 100 sequential steps to recognize the pattern.

⇒ parallel processing architectures!

Historical Perspective of ANNs

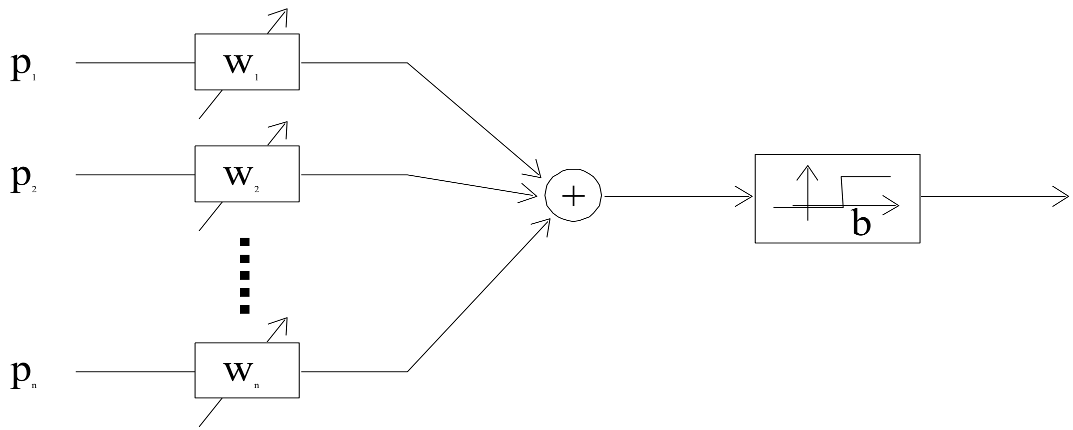
1943 McCulloch	Boolean neuron
1949 Hebb	Learning rule
1957 Rosenblatt	Perceptron
1960 Widrow-Hoff	ADALINE/MADALINE LMS
Rosenblatt	Multilayer Perceptron, without training
1969 Minsky-Papert	*Perceptrons*
1974 Werbos	*Error Backpropagation Algorithm – without repercussion*
1982 Hopfield	Network with feedback
1986 Rumelhart, Hinton & Williams	
PDP – MIT	*Backpropagation for Multilayer Perceptron*as Activation Funciont still continuous sigmoid
1987 Kosko	BAM

Starting Enthusiasm

Disenchantment

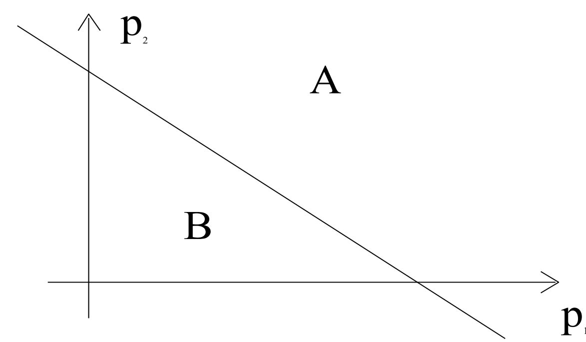
Resurgence

The McCulloch Neuron (1943)



for n=2

$$w_1 p_1 + w_2 p_2 = b$$

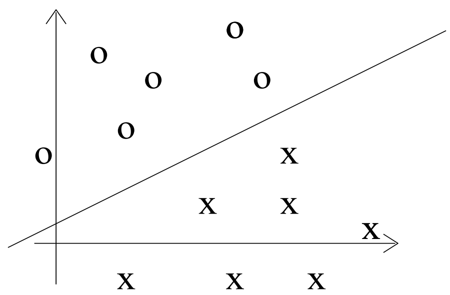


$$a = g\left(\sum_{i=1}^{n} w_i p_i - b \right) = g(\mathbf{w}^t \mathbf{p} - b) \rightarrow a \in [0;1]$$

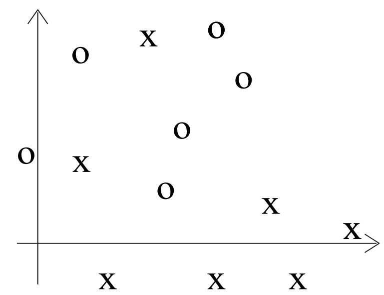
g = *step function*

The euclidian space $\Re^n$ is divided in two regions A and B

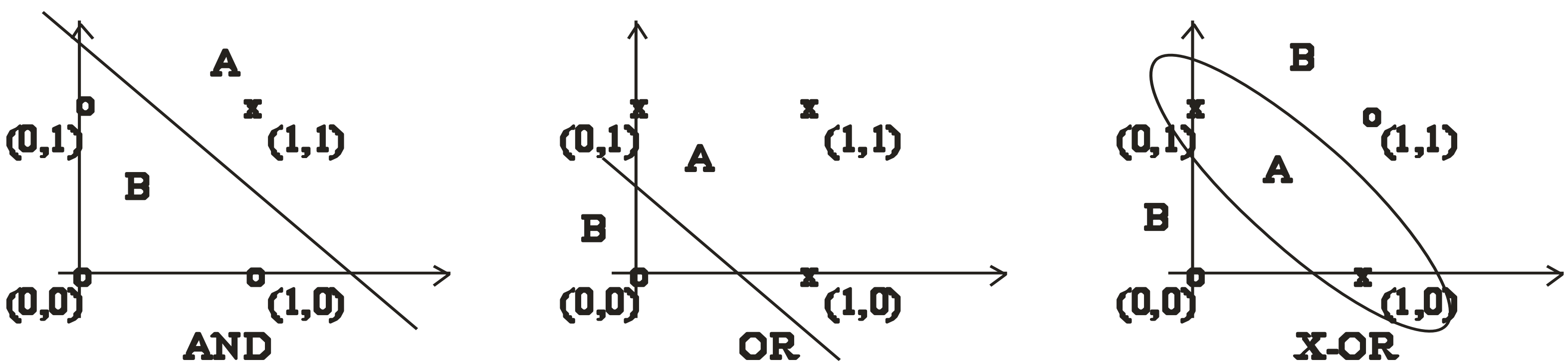
The McCulloch Neuron
– as patterns classifier



Linearly separable collections



Linearly dependent (non-separable) collections



Some Boolean functions of two variables represented in a binary plan.

Linear and Non-Linear Classifiers

There exist $2^m = 2^{2^n}$ possible logical functions connecting n inputs to one binary output.

n	# of binray patterns	# of logical functions	# linearly separable	% linearly separable
1	2	4	4	100
2	4	16	14	87,5
3	8	256	104	40,6
4	16	65536	1.772	2,9
5	32	$4{,}3 \times 10^{9}$	94.572	$2{,}2 \times 10^{-3}$
6	64	$1{,}8 \times 10^{19}$	5.028.134	$3{,}1 \times 10^{-13}$

The logical functions of one variable:

$A,\ \overline{A},\ 0,\ 1$

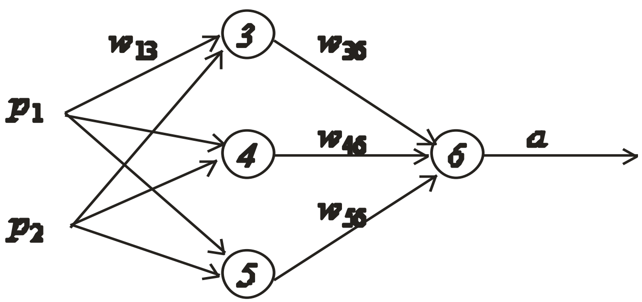
The logical functions of two variables:

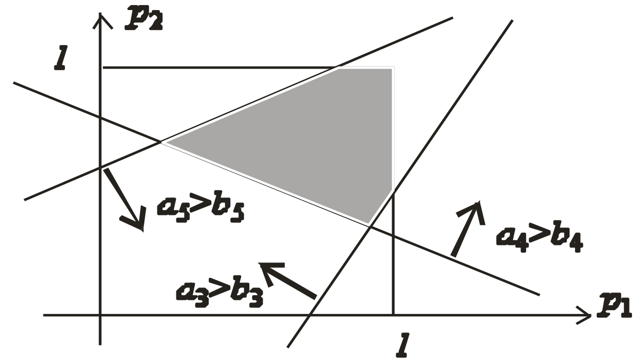
$A,\ B, \overline{A}, \overline{B},\ 0,\ 1$

$A \vee B, A \wedge B, \overline{A} \vee B, \overline{A} \wedge B,$

$A \vee \overline{B}, A \wedge \overline{B}, \overline{A} \vee \overline{B}, \overline{A} \wedge \overline{B},\ A \oplus B, \overline{A \oplus B}$

Two Step Binary Perceptron





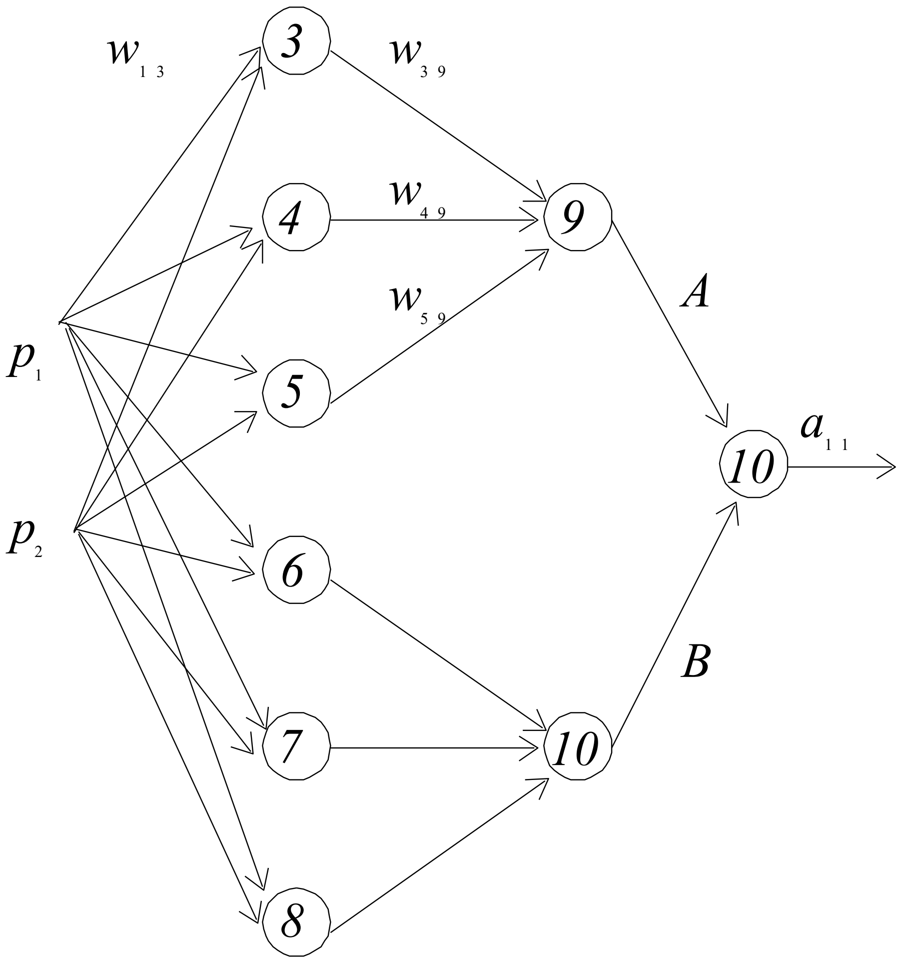
The neuron 6 implements a logical AND function by choosing

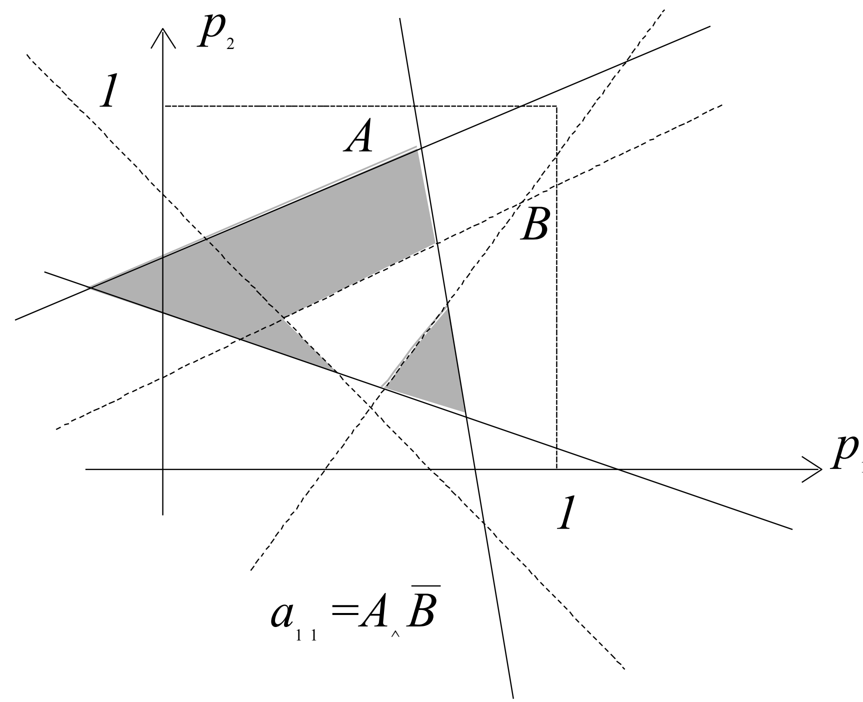
$$b_6 = \sum_{i=3}^{5} w_{i6} \; .$$

For example:

$$w_{36} = w_{46} = w_{56} = \frac{1}{3}; \quad b_6 = 1 \quad \Rightarrow a_6 = 1 \; if \;\; and \;\; only \;\; if \quad a_3 = a_4 = a_5 = 1$$

Three Step Binary Perceptron





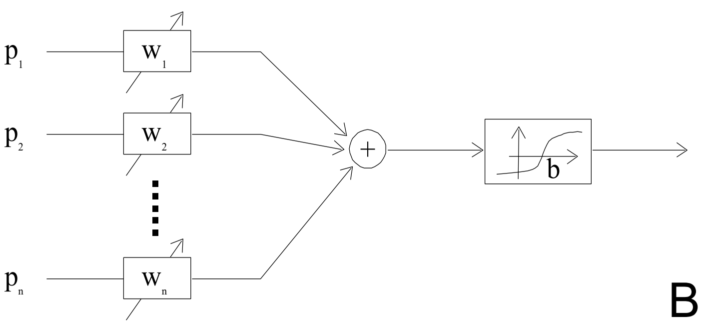
Neurons and Artificial Neural Networks

- Micro-structure

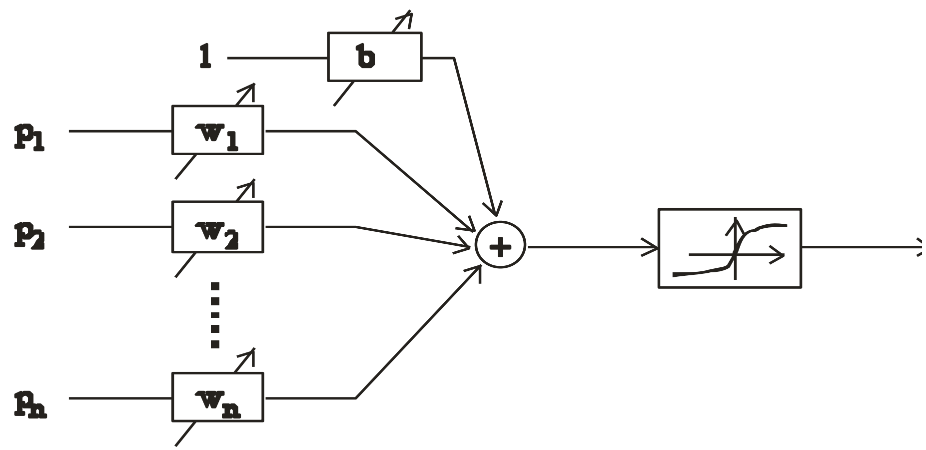
  characteristics of each neuron in the network
- Meso-Structure

  organization of the network
- Macro-Structure

  association of networks, eventually with some analytical processing approach for complex problems

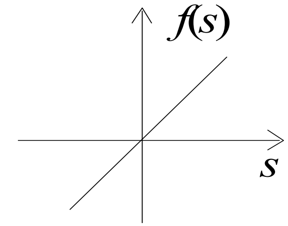
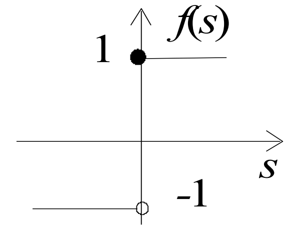
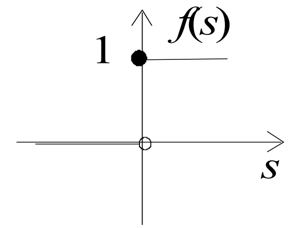
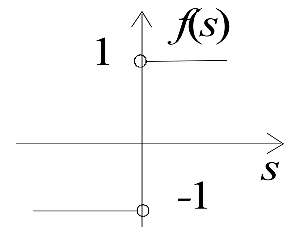


Bias input

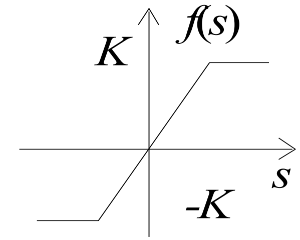
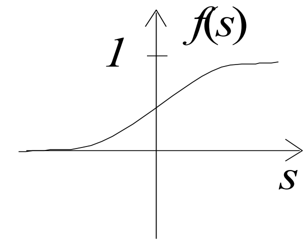
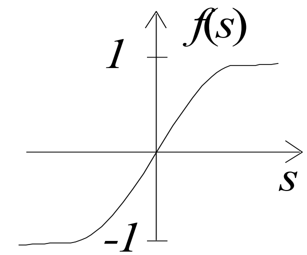


Bias:
with p=0,
output $\neq 0$ still possible !

Typical activation functions

Linear	$f(s) = s$	Hopfield BSB	purelin	
Signal	$f(s) = \begin{cases} +1 \ se \ s \geq 0 \\ -1 \ se \ s < 0 \end{cases}$	Perceptron	hardlims	
Step	$f(s) = \begin{cases} +1 \ se \ s \geq 0 \\ 0 \ se \ s < 0 \end{cases}$	Perceptron BAM	hardlim	
Hopfield/ BAM	$f(s) = \begin{cases} +1 \ se \ s > 0 \\ -1 \ se \ s < 0 \\ unchanged \ \ if \ \ s = 0 \end{cases}$	Hopfield BAM		

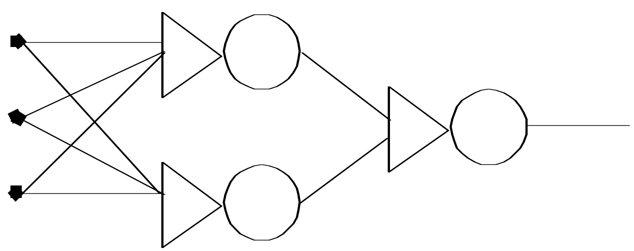
Typical activation functions

BSB or Logical Threshold	$f(s)=\begin{cases}-K \ \ se\ s\le -K\\ s\ \ se\ -K<s<+K\\ +K\ \ se\ s\ge +K\end{cases}$	BSB	satlin satlins	
Logístics	$f(s)=\dfrac{1}{1+e^{-s}}$	Perceptron Hopfield BAM, BSB	logsig	
Hiperbolic Tangent	$f(s)=\tanh(s)=\dfrac{1-e^{-2s}}{1+e^{-2s}}$	Perceptron Hopfield BAM, BSB	tansig	

Meso-Structure – Network Organization...

neurons per layer
network layers
connection type (forward, backward, lateral).

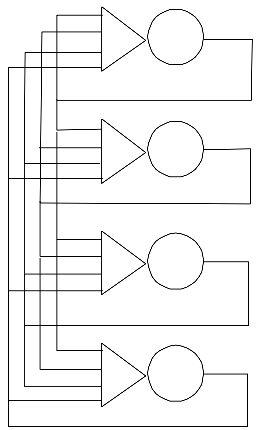
1- Multilayer *Feedforward*



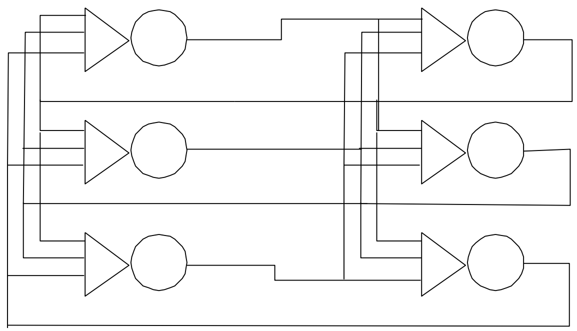
Multilayer Perceptron (MLP)

Meso-Structure – Network Organization...

2- Single Layer laterally connected (BSB (self-feedback), Hopfield)

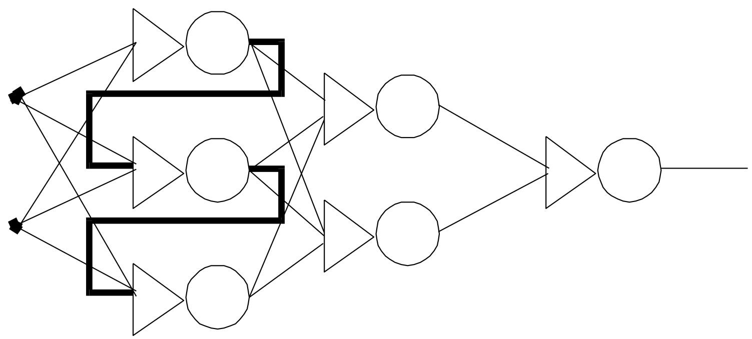


3 – Bilayers *Feedforward/Feedbackward*

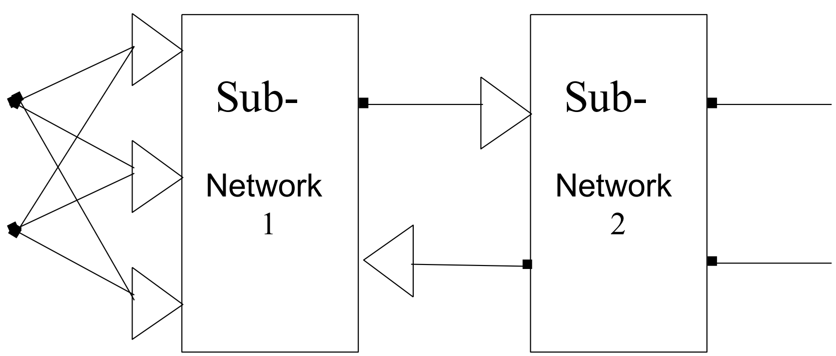


Meso-Structure – Network Organization

4 – Multilayer Cooperative/Comparative Network

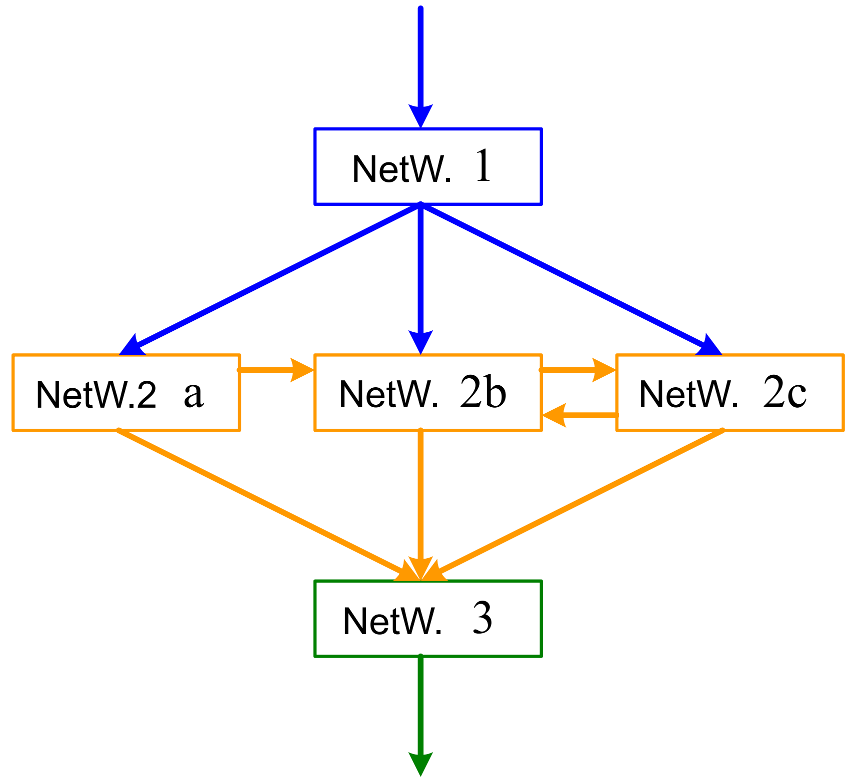


5 – Hybrid Network

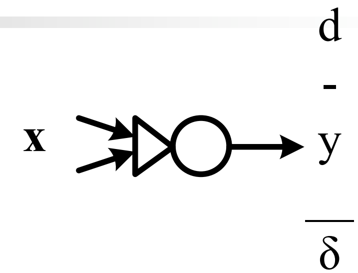


Neural Macro-Structure

- # networks
- connection type
- size of networks
- degree of connectivity



Supervised Learning



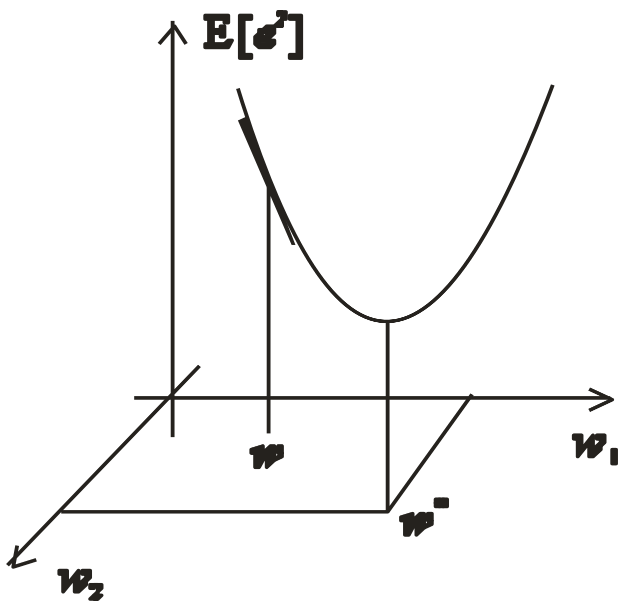
- Delta Rule → Perceptron

$$w \leftarrow w + \mu \delta x \qquad \delta \equiv d - y$$

μ – learning rate

- Widrow-Hoff delta rule (LMS) → ADALINE, MADALINE

- Generalized Delta Rule

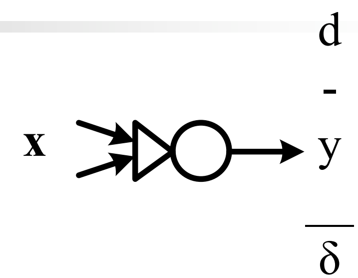


$$w_{ij} \leftarrow w_{ij} + \frac{\mu \delta_j x_{ij}}{\sum x_k^2}$$

Widrow-Hoff Delta Rule (LMS)

Delta rule → Perceptron

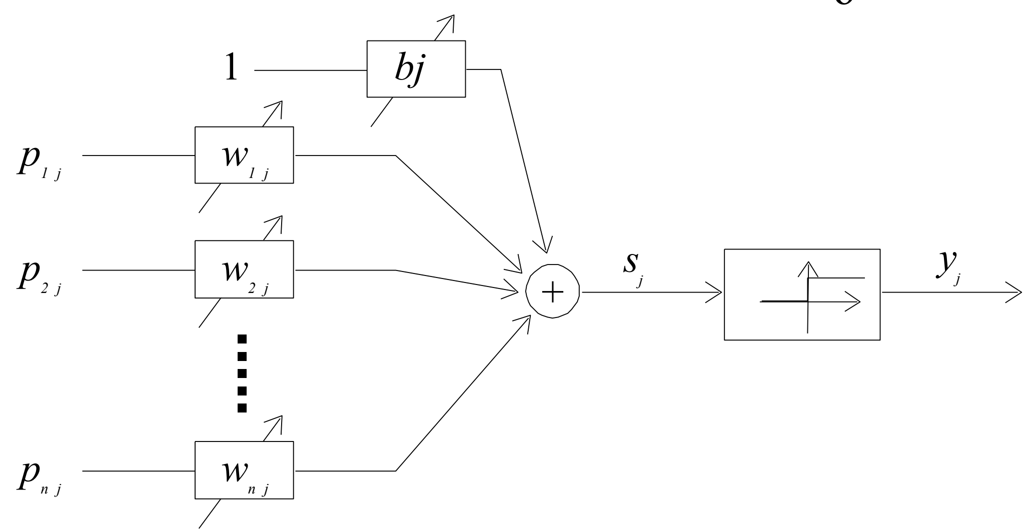
Perceptron – Rosenblatt, 1957



Dynamics:

$$s_j = \sum_i w_{ij} p_{ij} + b_j$$

$$y_j = f(s_j) = \begin{cases} +1 \ se \ s_j \geq 0 \\ 0 \ se \ s_j < 0 \end{cases}$$



$$\delta_j = d_j - y_j$$

$$w_{ij} \leftarrow w_{ij} + \mu \, \delta_j x_{ij}$$

Delta Rule

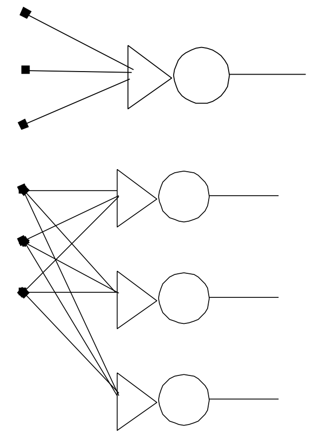
<u>Psychology Reasoning:</u>
- positive reinforcement
- negative reinforcement

μ - learning rate

$\delta_j = 0 \rightarrow$ *the weight is not changed.*

ADALINE and MADALINE

Widrow & Hoff, 1960 – (Mult.) Adaptive Linear Element



$$y_j = \sum_i w_{ij} p_{ij} + b_j$$

Training:

$$\varepsilon_j = d_j - s_j = d_j - \left(\sum w_{ij} p_{ij} + b_j \right)$$

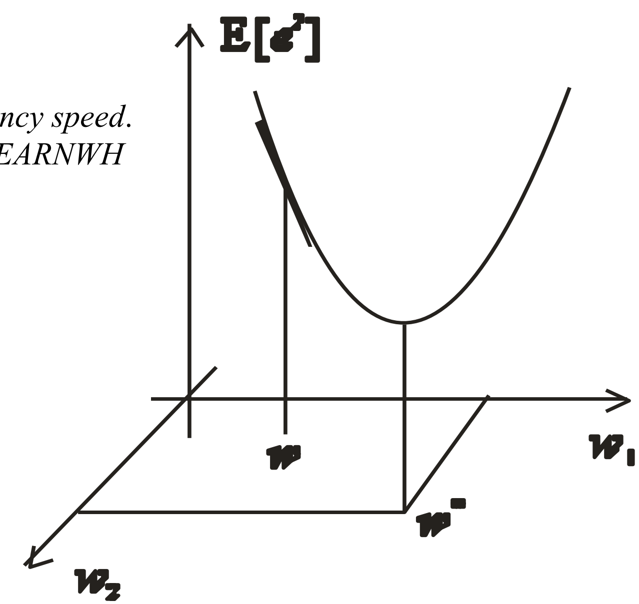
$$w_{ij} \leftarrow w_{ij} + \frac{\mu \varepsilon_j x_{ij}}{\sum x_k^2}$$

Widrow-Hoff delta rule

LMS – Least Mean Squared algorithm

$0.1 < \mu < 1$ – stability and convergency speed.

MatLab: NEWLIN, NEWLIND, ADAPT, LEARNWH



$$Obs: \ \varepsilon_j \equiv \delta_j \qquad w_{ij} \leftarrow w_{ij} + \mu \, \delta_j x_{ij}$$

Delta Rule

LMS Algorithm

Objective: learn a function $f: \Re^n \rightarrow \Re$ from the samples ($\boldsymbol{x}_k$, d_k)

$\{ \boldsymbol{x}_k \}$, $\{ d_k \}$ and $\{ e_k \}$ $\rightarrow$ stationary stochastic processes

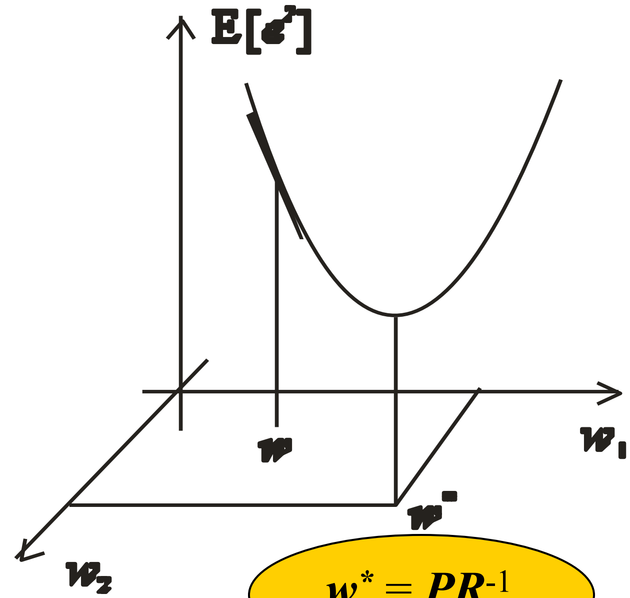
$e = d - y \rightarrow$ actual stochastic error

$\rightarrow$ Linear neuron

$$y = \sum_{i=1}^{n} x^i w^i = \mathbf{x}\mathbf{w}^t$$

Expected value

$$
\begin{aligned}
E[e^2] &= E[(d-y)^2] \\
&= E[(d-\boldsymbol{x}\boldsymbol{w}^t)^2] \\
&= E[d^2] - 2E[d\boldsymbol{x}]\boldsymbol{w}^t + \boldsymbol{w}E[\boldsymbol{x}^t\boldsymbol{x}]\,\boldsymbol{w}^t
\end{aligned}
$$



$\boldsymbol{w}^* = \boldsymbol{P}\boldsymbol{R}^{-1}$

Optimal analytic solution of the optimization (solvelin.m)

Assuming $\boldsymbol{w}$ deterministic.
With
E $[\boldsymbol{x}^t\boldsymbol{x}] \equiv \boldsymbol{R} \rightarrow$ autocorrelation input matrix
E $[d\boldsymbol{x}] \equiv \boldsymbol{P} \rightarrow$ cross correlated vector

$$E[e^2] = E[d^2] - 2\boldsymbol{P}\boldsymbol{w}^t + \boldsymbol{w}\boldsymbol{R}\boldsymbol{w}^t$$

$$\mathbf{0} = 2\boldsymbol{w}^*\boldsymbol{R} - 2\boldsymbol{P}$$

(Partial derivatives equal 0 for optimal w*)

Iterative LMS Algorithm

Objective: adaptively learn a function $f : \Re^n \to \Re$ from the samples ($\boldsymbol{x}_k$, d_k)

Knowing $\boldsymbol{P}$ and $\boldsymbol{R}$, ∍ $\boldsymbol{R}^{-1}$, then for some $\boldsymbol{w}$:

$$\nabla_w \, \mathrm{E}[e^2] = 2\boldsymbol{wR} - 2\boldsymbol{P}$$

Post-multiplyting by ½ $\boldsymbol{R}^{-1}$

$$\tfrac{1}{2} \nabla_w \, \mathrm{E}[e^2] \, \boldsymbol{R}^{-1} = \boldsymbol{w} - 2\boldsymbol{P}\,\boldsymbol{R}^{-1} = \boldsymbol{w} - \boldsymbol{w}^*$$

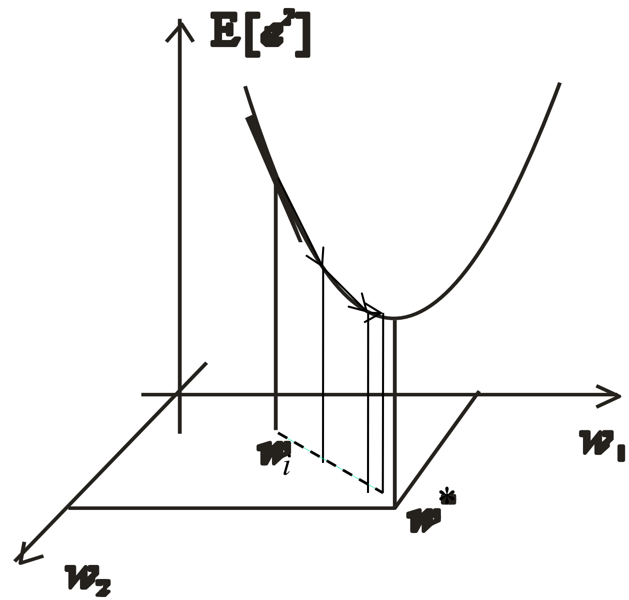
$$\boldsymbol{w}^* = \boldsymbol{w} - \tfrac{1}{2} \nabla_w \, \mathrm{E}[e^2] \, \boldsymbol{R}^{-1}$$

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - c_k \nabla_w \, \mathrm{E}[e^2] \, \boldsymbol{R}^{-1}$$

($c_k = ½ \to$ Newton's method)

LMS Hypothesis:

$\mathrm{E}[e^2_{k+1} | \, e^2_0, e^2_1, \ldots e^2_k] = e^2_k$



How to, ***cautiously*** find new (better) values for w_i, the free parameters ?

Iterative LMS Algorithm...

assuming $\boldsymbol{R} = \boldsymbol{I} \rightarrow$ estimated ***steppest decent algorithm***:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - c_k \nabla_w e^2_k$$

Gradient of e^2_k with respect to $\boldsymbol{w}$

$$\nabla_w e^2_k = \left[\frac{\partial e_k^2}{\partial w_1}, \quad \cdots \quad \frac{\partial e_k^2}{\partial w_n} \right]$$

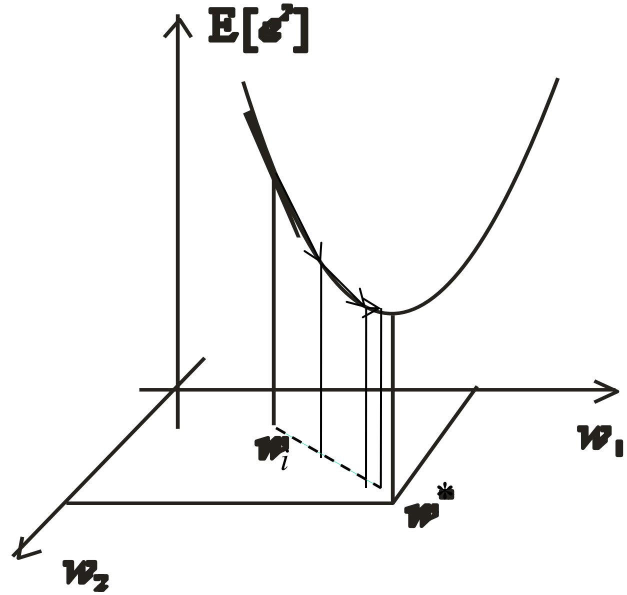
$$= \left[\frac{\partial (d_k - y_k)^2}{\partial w_1}, \quad \cdots \quad \frac{\partial (d_k - y_k)^2}{\partial w_n} \right]$$

$$= \left[-2(d_k - y_k)\frac{\partial y_k}{\partial w_1}, \quad \cdots \quad -2(d_k - y_k)\frac{\partial y_k}{\partial w_n} \right]$$

$$= -2e_k \left[\frac{\partial y_k}{\partial w_1}, \quad \cdots \quad \frac{\partial y_k}{\partial w_n} \right]$$

$$= -2e_k \left[x_k^1, \quad \cdots \quad x_k^n \right] = -2 e_k \mathbf{x}_k \qquad (y_k = \mathbf{x}_k \mathbf{w}_k^t)$$

LMS algorithm reduces to $\boxed{\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + 2c_k e_k \boldsymbol{x}_k}$



$$\boxed{w_{ij} \leftarrow w_{ij} + \frac{\mu \delta_j x_{ij}}{\sum x_k^2}}$$

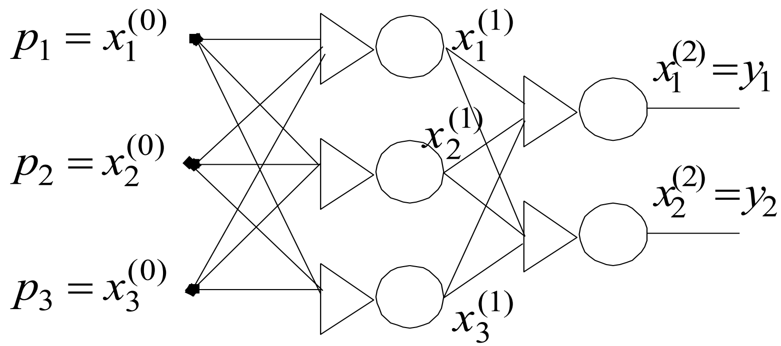
Norma-
lization

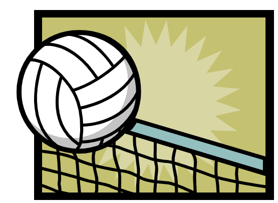
Iterative (adaptive) solution
(The optimal solution is never reached!)
MADALINE i-input, j-neuron

The Multilayer Perceptron

- The Generalized Delta Rule

Rumelhart, Hinton e Williams, PDP/MIT, 1986





Neuron Dynamics:

Processing Element (PE) j
in layer k
input i

$$s_j^{(k)} = w_{0j}^{(k)} + \sum_i w_{ij}^{(k)} x_i^{(k-1)}$$

$$x_j^{(k)} = f(s_j^{(k)})$$

with f (activation function)
continuous differentiable

Turning Point Question:
How to find the error associated with an internal neuron??

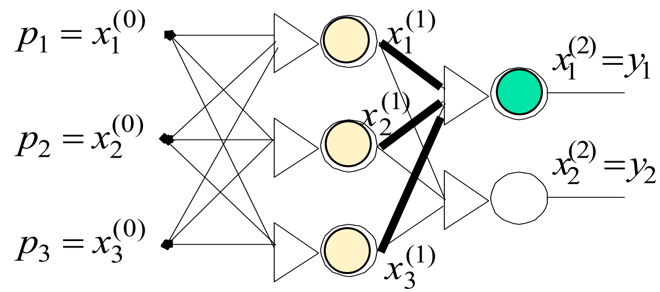
The generalized delta rule

Training

$$\varepsilon^2 = \sum_{j=1}^{m} (d_j - y_j)^2$$ - quadratic error

$$\mathbf{w}_j^{(k)} = (w_{oj}^{(k)}, w_{1j}^{(k)}, \ldots, w_{mj}^{(k)})$$ - weigths of PE j

$$\mathbf{x}_j^{(k-1)} = (1, x_{1j}^{(k-1)}, \ldots, x_{nj}^{(k-1)})$$ - input vector of PE j



With $s_j^{(k)} = \mathbf{w}_j^{(k)} \mathbf{x}_j^{(k-1)} \rightarrow \dfrac{\partial s_j^{(k)}}{\partial \mathbf{w}_j^{(k)}} = \mathbf{x}_j^{(k-1)}$

Instantaneous gradient:

$$\nabla_j^{(k)} = \frac{\partial \varepsilon^2}{\partial \mathbf{w}_j^{(k)}} = \left[\frac{\partial \varepsilon^2}{\partial w_{0j}^{(k)}}, \quad \frac{\partial \varepsilon^2}{\partial w_{1j}^{(k)}}, \quad \cdots \quad \frac{\partial \varepsilon^2}{\partial w_{mj}^{(k)}} \right]$$

$$\nabla_j^{(k)} = \frac{\partial \varepsilon^2}{\partial \mathbf{w}_j^{(k)}} = \frac{\partial \varepsilon^2}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial \mathbf{w}_j^{(k)}}$$

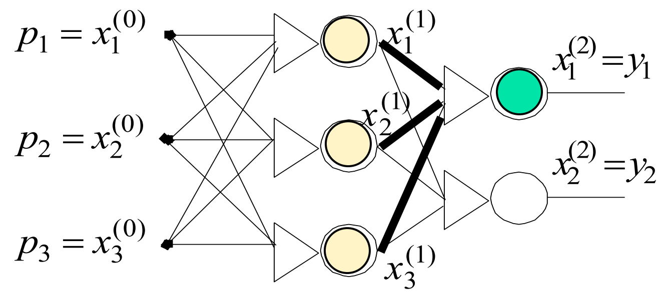
so $\nabla_j^{(k)} = \dfrac{\partial \varepsilon^2}{\partial \mathbf{w}_j^{(k)}} = \dfrac{\partial \varepsilon^2}{\partial s_j^{(k)}} \mathbf{x}_j^{(k-1)}$

Defining the ***quadratic derivative error*** as $\boxed{\delta_j^{(k)} = -\dfrac{1}{2} \dfrac{\partial \varepsilon^2}{\partial s_j^{(k)}}}$

$$\boxed{\nabla_j^{(k)} = -2 \delta_j^{(k)} \mathbf{x}_j^{(k-1)}}$$

Gradient of the error with respect to the weights as function of the ***former layer*** signals!!

The generalized delta rule...



For the ***output layer***, the quadratic derivative error is:

$$\delta_j^{(k)} = -\frac{1}{2}\frac{\partial \sum_{i=1}^{N_k}(d_i - y_i)^2}{\partial s_j^{(k)}} = -\frac{1}{2}\frac{\partial \sum_{i=1}^{N_k}(d_i - f(s_i^{(k)}))^2}{\partial s_j^{(k)}}$$

The partial derivatives are 0 for $i \neq j$

$$\delta_j^{(k)} = -\frac{1}{2}\frac{\partial (d_j - f(s_j^{(k)}))^2}{\partial s_j^{(k)}} = -(d_j - f(s_j^{(k)}))\frac{\partial (d_j - f(s_j^{(k)}))}{\partial s_j^{(k)}} = (d_j - x_j^{(k)})f'(s_j^{(k)})$$

The output error associated with PE_j, in the last layer:

$$\varepsilon_j^{(k)} = d_j - x_j^{(k)} = d_j - y_j$$

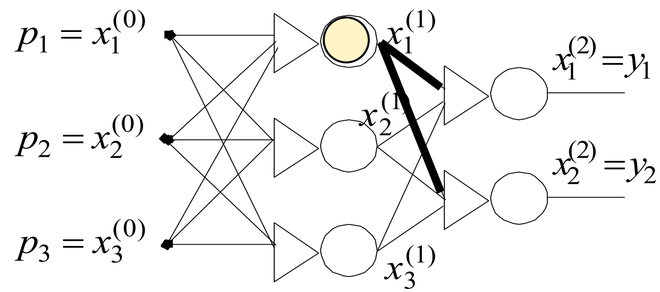
Giving:

$$\boxed{\delta_j^{(k)} = \varepsilon_j^{(k)} . f'(s_j^{(k)})}$$

Remember,
"activation function, *f, **continuous differentiable***"

The generalized delta rule...

For a ***hidden layer*** k, the quadratic derivative error can be calculated using the linear outputs of layer $k+1$:



$$\delta_j^{(k)} = -\frac{1}{2}\frac{\partial \varepsilon^2}{\partial s_j^{(k)}} = -\frac{1}{2}\sum_{i=1}^{N_{k+1}}\left(\frac{\partial \varepsilon^2}{\partial s_i^{(k+1)}}\frac{\partial s_i^{(k+1)}}{\partial s_i^{(k)}}\right) \qquad (Chain\ Rule)$$

$$= \sum_{i=1}^{N_{k+1}}\left(\left(-\frac{1}{2}\frac{\partial \varepsilon^2}{\partial s_i^{(k+1)}}\right)\frac{\partial s_i^{(k+1)}}{\partial s_i^{(k)}}\right) = \sum_{i=1}^{N_{k+1}}\left(\delta_i^{(k+1)}\frac{\partial s_i^{(k+1)}}{\partial s_i^{(k)}}\right)$$

Taking into account that $\quad s_j^{(k)} = w_{0j}^{(k)} + \sum_{i=1}^{N_k} w_{ij}^{(k)} x_i^{(k-1)}$

$$\delta_j^{(k)} = \sum_{i=1}^{N_{k+1}}\left(\delta_i^{(k+1)}\frac{\partial}{\partial s_i^{(k)}}\left(w_{0i}^{(k+1)} + \sum_{l=1}^{N_k} w_{li}^{(k+1)} f\left(s_l^{(k)}\right)\right)\right)$$

$$\delta_j^{(k)} = \sum_{i=1}^{N_{k+1}}\left(\delta_i^{(k+1)}\sum_{l=1}^{N_k} w_{li}^{(k+1)}\frac{\partial}{\partial s_i^{(k)}} f\left(s_l^{(k)}\right)\right)$$

considering

$$\frac{\partial}{\partial s_j^{(k)}} f\left(s_l^{(k)}\right) = 0 \text{ if } l \neq j \text{ and that } \frac{\partial}{\partial s_j^{(k)}} f\left(s_j^{(k)}\right) = f'\left(s_j^{(k)}\right)$$

We have: $\displaystyle \delta_j^{(k)} = \underbrace{\left(\sum_{i=1}^{N_{k+1}}\left(\delta_i^{(k+1)} w_{ji}^{(k+1)}\right)\right)}_{\equiv \varepsilon_j^{(k)}} .f'\left(s_j^{(k)}\right)$

Finally, the ***quadratic derivative errror*** for a hidden layer:

$$\boxed{\delta_j^{(k)} = \varepsilon_j^{(k)} . f'\left(s_j^{(k)}\right)}$$

The "Error Backpropagation" algorithm

1. $w_{ij}^{(k)} \leftarrow random$, initialize the network weigths
2. for ($\mathbf{x}$,$\mathbf{d}$), training pair, obtain y. Feedforward propagation: $\varepsilon^2 = \sum_{j=1}^{m}(d_j - y_j)^2$
3. k $\leftarrow last\ \ layer$
4. for each element j in the layer k do:

   Compute $\varepsilon_j^{(k)}$ using $\varepsilon_j^{(k)} = d_j - x_j^{(k)} = d_j - y_j$ if k is the last layer,

   $$\varepsilon_j^{(k)} = \sum_{i=1}^{N_{k+1}} \delta_i^{(k+1)} w_{ji}^{(k+1)} \text{ if it is a hidden layer;}$$

   Compute $\delta_j^{(k)} = \varepsilon_j^{(k)}.f'(s_j^{(k)})$
5. $k \leftarrow k-1$ if k > 0 go to step 4, else continue.
6. $\mathbf{w}_j^{(k)}(n+1) = \mathbf{w}_j^{(k)}(n) + 2\mu\delta_i^{(k)}\mathbf{x}_i^{(k)}$
7. For the next training pair go to step 2.

The Backpropagation Algorithm ***in practice***

1 – In the standard form BP is ***very slow***.

2 – BP Pathologies: ***paralysis*** in regions of small gradient.

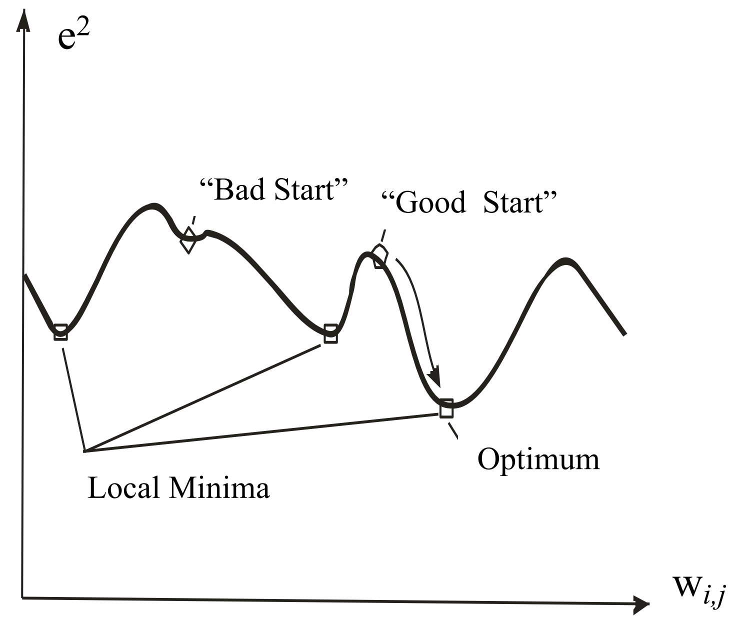
3 – ***Initial conditions*** can lead to local minima.

4 – Stop conditions – number of epochs, $\Delta w_{ij} < \epsilon$

5 – BP variants

- trainbpm (with momentum)
- trainbpx (adaptive learning rate)
- ....
- trainlm (Levenberg-Marquard – *J*, Jacobian)

$$\Delta \mathbf{W}_j^{(k)} = (J^T J + \mu J)^{-1} J^T e$$



$e^2(w_{i,j})$ - Illustrative quadratic error as function of the weights

Obs: the error surface is, normally, unknown.

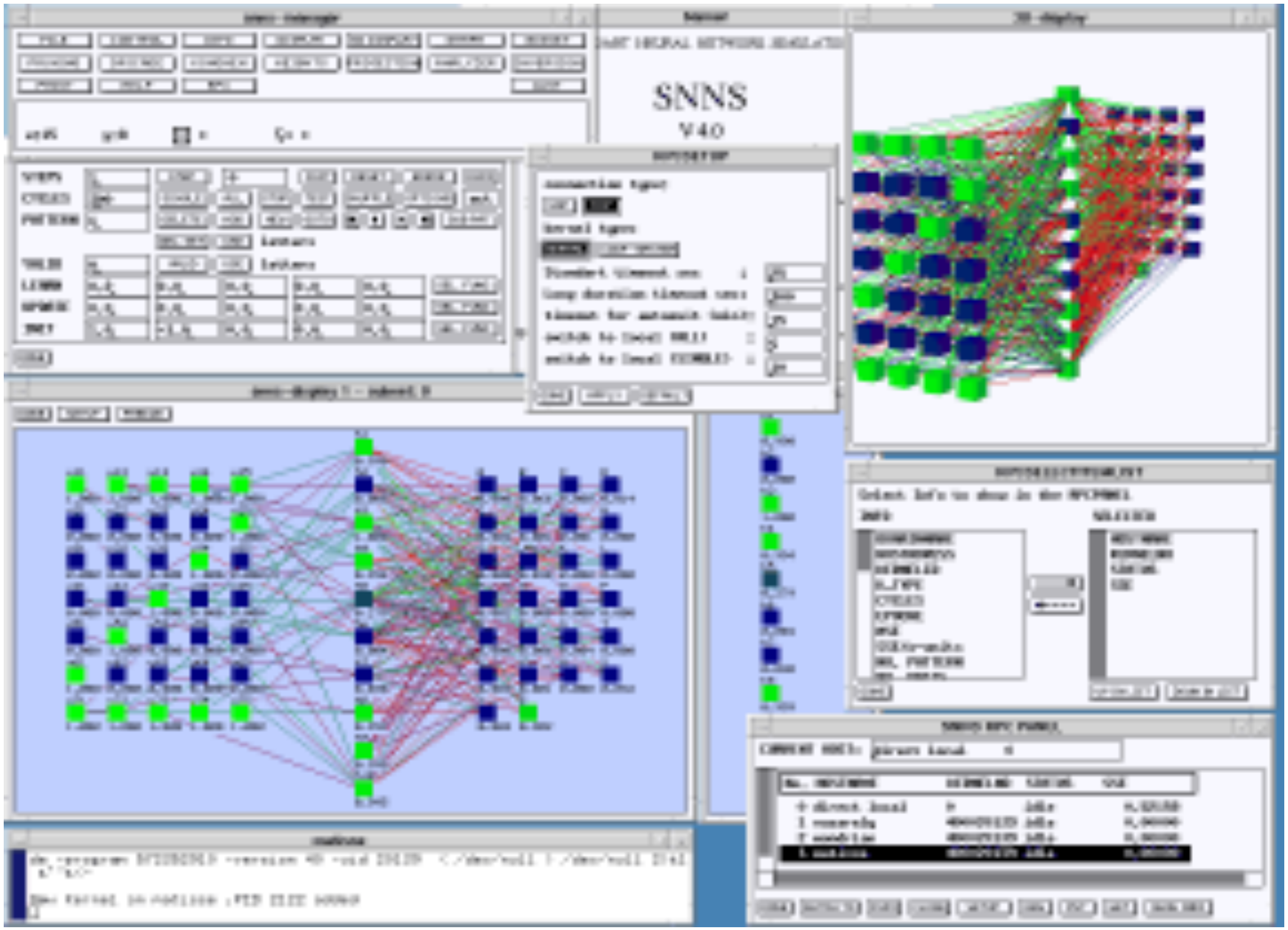
Steepest descent → go in the opposite direction of the ***local*** gradient ("downhill").

Computational Tools

- SNNS
- MatLab
  - Neural Network Toolbox
- NeuralWorks
- Java
- C++

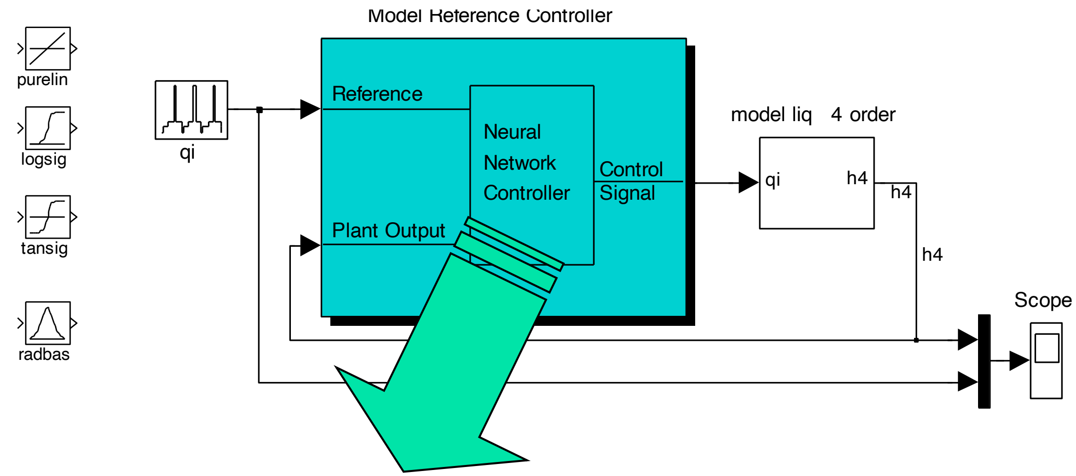
- Hardware Implementations of RNAs

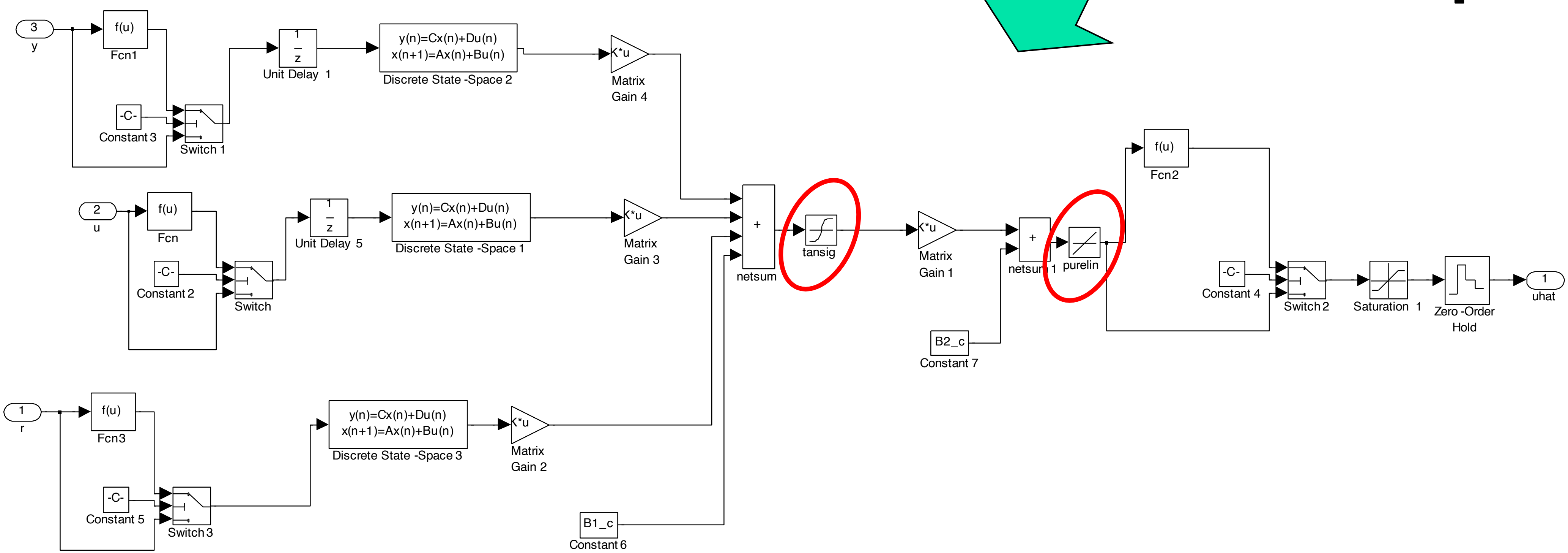
SNNS - Stuttgarter Neural Network Simulator



MatLab

- complete environment
  - -System Simulation
  - -Training
  - -Control





Demonstration - Perceptron

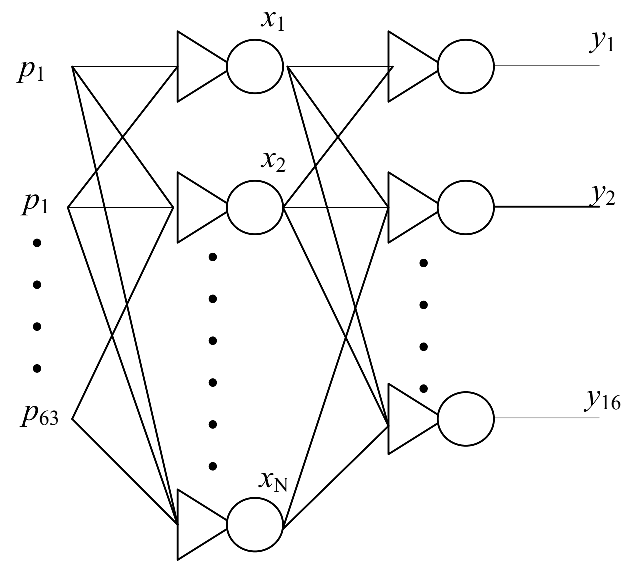
```
% Perceptron
% Training an ANN to learn to classify a non-linear problem
% Input Pattern
P=[ 0    0    0    0    1    1    1    1
    0    0    1    1    0    0    1    1
    0    1    0    1    0    1    0    1]
% Target
%T=[1 0 1 1 1 0 1 0]     % Linear separable
  T=[1 0 0 1 1 0 1 0]     %     non separable

% Try with Rosenblat's Perceptron
net=newp(P,T,'hardlim')

% train the network
net=train(net,P,T)

Y=sim(net,P)
```

T =	1	0	1	1	1	0	1	0
Y =	1	0	1	1	1	0	1	0

T =	1	0	0	1	1	0	1	0
Y =	1	0	1	0	0	0	1	0

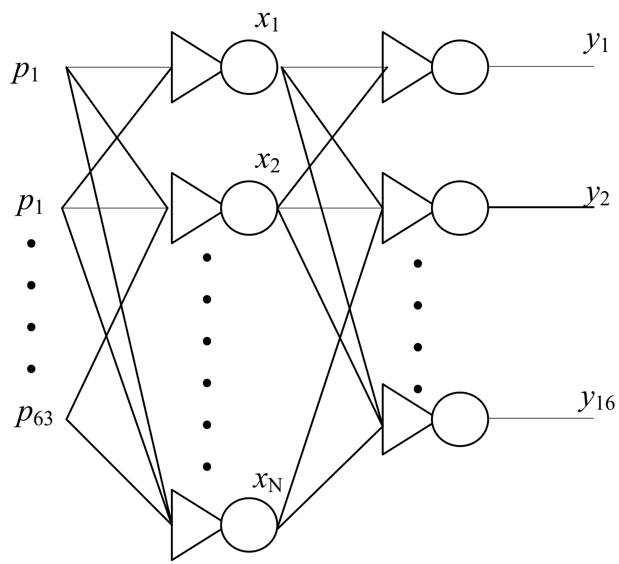
Demonstration - OCR

Training Vector

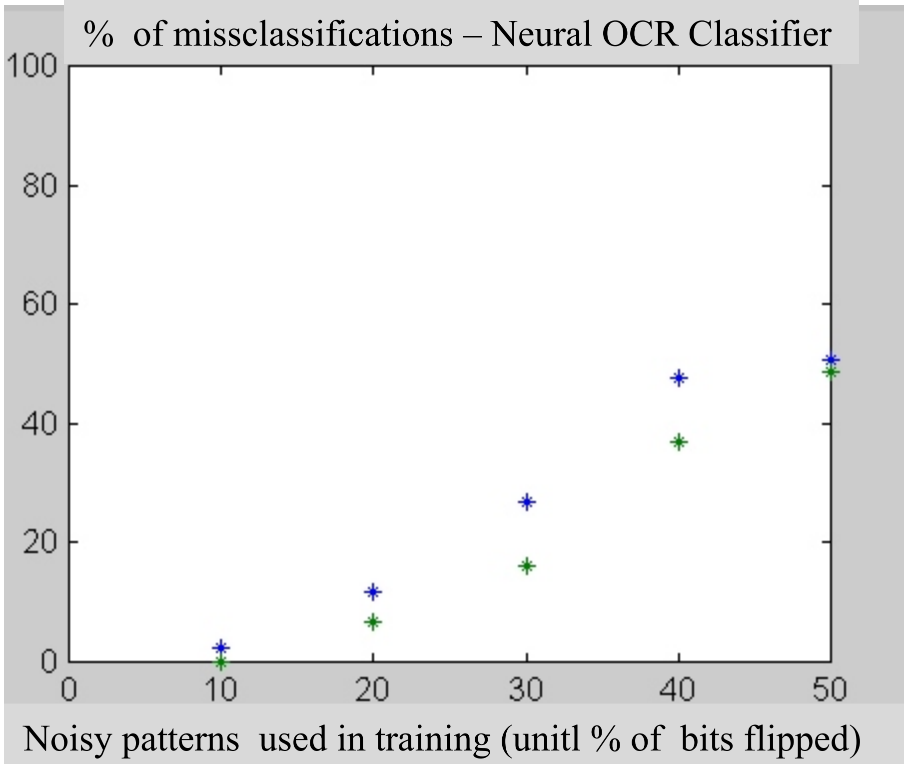


20 % Noise

Demonstration – OCR...



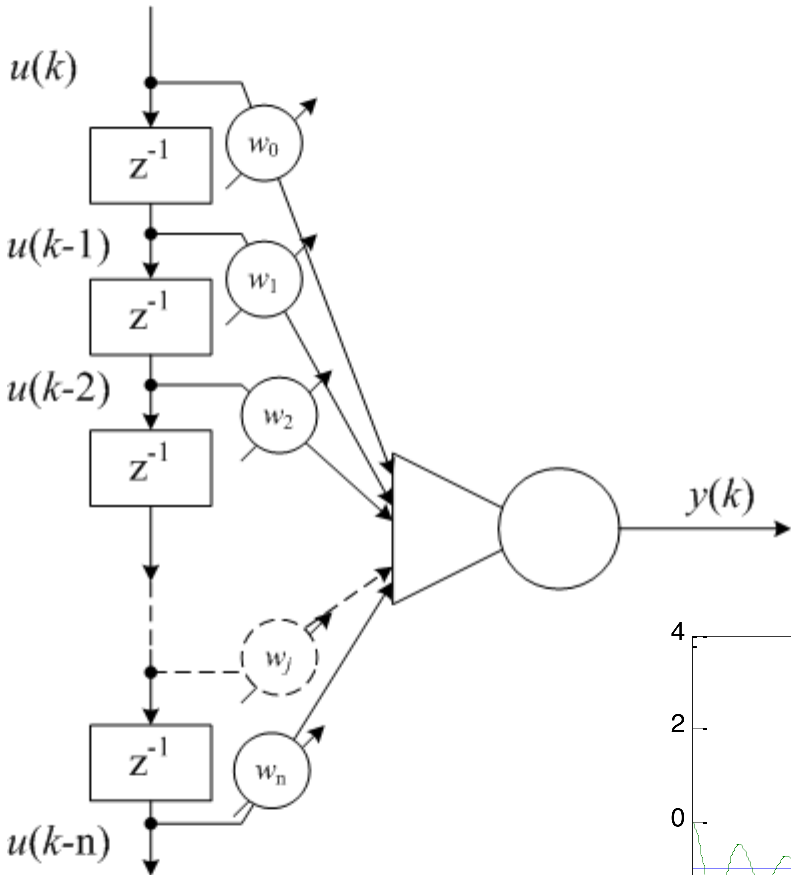
Training with 10 x (0,10,20,30,40,50) % noise



* - error without noisy training patterns
* - error using noisy training patterns

With Some Noisy Training Pattern
→ Learns how to treat "any" noise

Demonstration – LMS, ADALINE, FIR



(TDL – Time Delay Line)

$$y(k) = w_0 u(k) + w_1 u(k-1) + w_2 u(k-2) + \cdots w_n u(k-n)$$

$$\frac{Y(z)}{U(z)} = w_0 + w_1 z^{-1} + w_2 z^{-2} + \cdots w_n z^{-n}$$

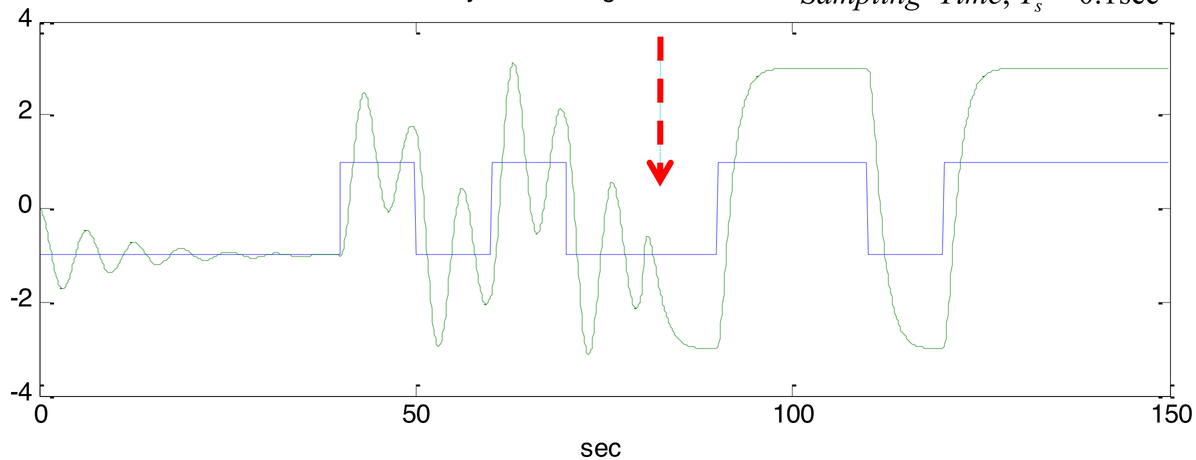
FIR Model (always stable, only Zeros)

Obs : IIR Model is more compact, but can be unstable!

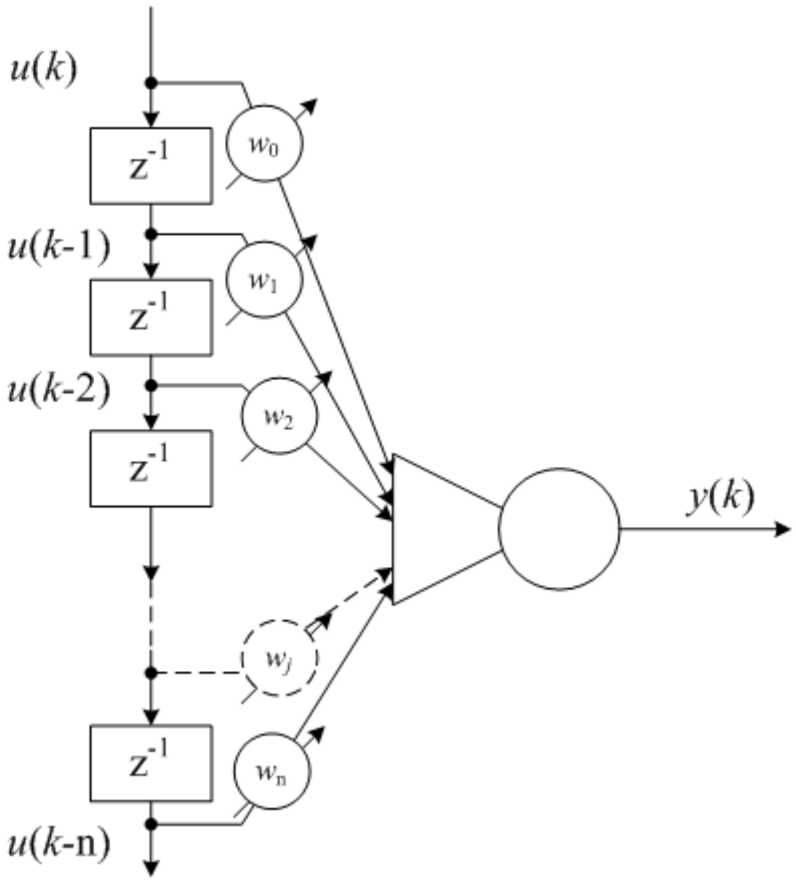
$$g_1(0-79.9\,\text{sec}) = \frac{1}{s^2 + 0.2s + 1} \qquad g_2(80-150\,\text{sec}) = \frac{3}{s^2 + 2s + 1}$$

System changes at 80 sec

Sampling Time, $T_s = 0.1\,\text{sec}$

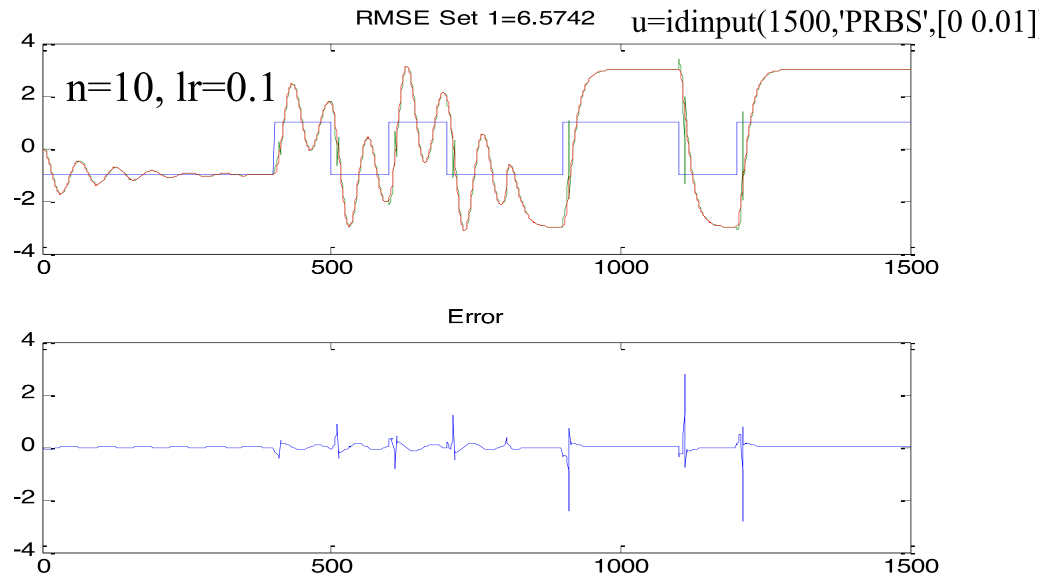


Demo – LMS, ADALINE, FIR...

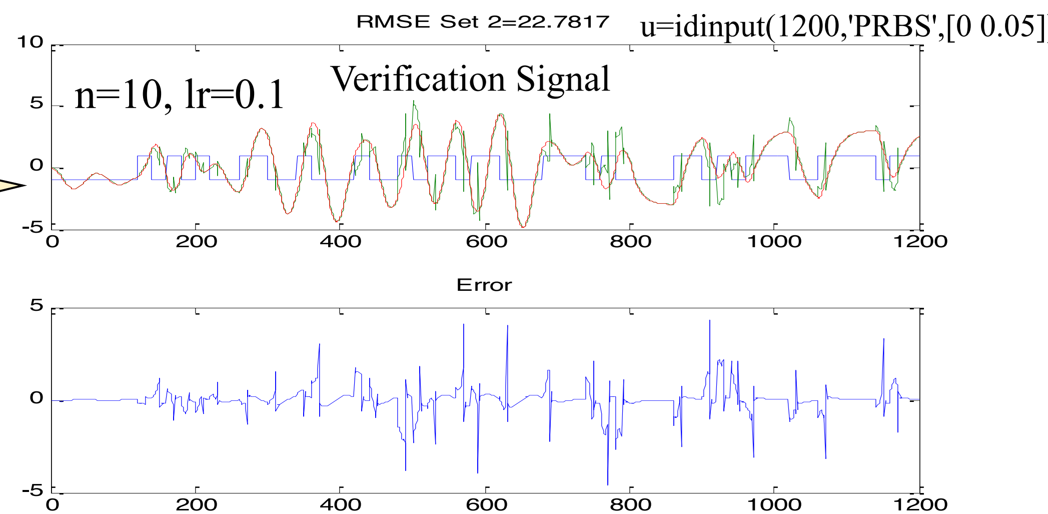



```
% ADALINE - Adaptive dynamic system identification
% First sampled system - until 80 sec
g1=tf(1,[1 .2 1]), gd1=c2d(g1,.1)
 % Sytem changes dramatically - after 80 sec
g2=tf(3,[1 2 1]),gd2=c2d(g2,.1)

% Pseudo Random Binary Signal - good for identification
u=idinput(120*10,'PRBS',[0 0.01],[-1 1]);

% time vector
...
[y1,t1,x1]=lsim(gd1,u1,t1);
[y2,t2,x2]=lsim(gd2,u2,t2,x1);


% Creates new adaline nework with delayed inputs (FIR)
% Learning Rate = 0.09

net=newlin(t,y,[1 2 3 4 5 6 7 8 9 10],0.09)
[net,Y,E]=adapt(net,t,y)


% design an average transfer function
netd=newlind(t,y)
```

Demo – LMS, ADALINE, FIR...

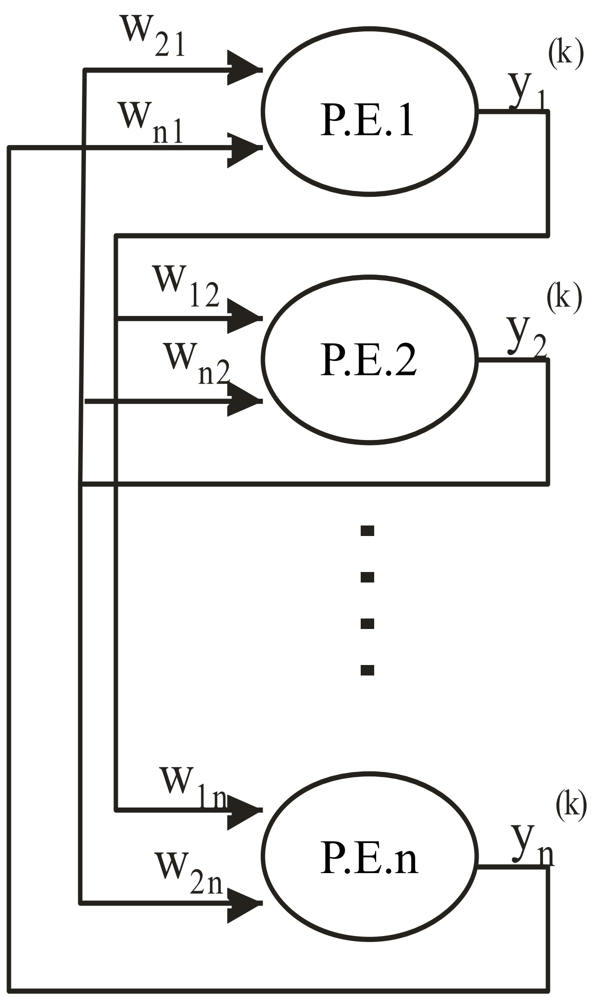


ADALINE
Learns System AND
also Changes in the Dynamics!!



But, in other frequency range
not so good...
(needs to Adjust *TDL*, *lr*, T_s)

Hopfield Network– Recurrent Netorks



Hopfield Network with *n* Processing Elements

Auto-Associative Memory:
Given an initial *n* bit pattern
returns the closest stored (associated) pattern.
No P.E. self-feedback!

$$Dynamics: \; s_j^{(k)} = \sum_{i=1}^{n} w_{ij} y_i^{(k)}$$

$$y_j^{(k+1)} = f\left(s_j^{(k)}\right)$$

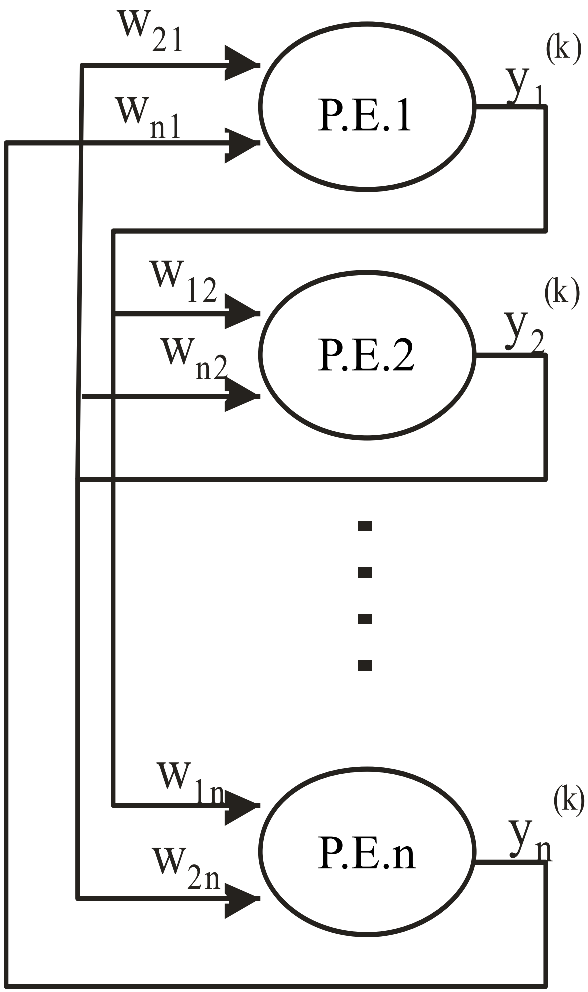
$$Network\ Initialization: \quad \mathbf{y}^{(0)} = \mathbf{x}$$

$$Output\ Vector: \quad \mathbf{y}^{(k)} = \left[y_i^{(k)}\right]$$

Binary activation function:

$$f(s_j) = \begin{cases} 1 \ if \ s_j > L_j \\ 0 \ if \ s_j < L_j \\ hold \ \ previous \ value, \ if \ s_j = L_j \end{cases}$$

Hopfield Network...

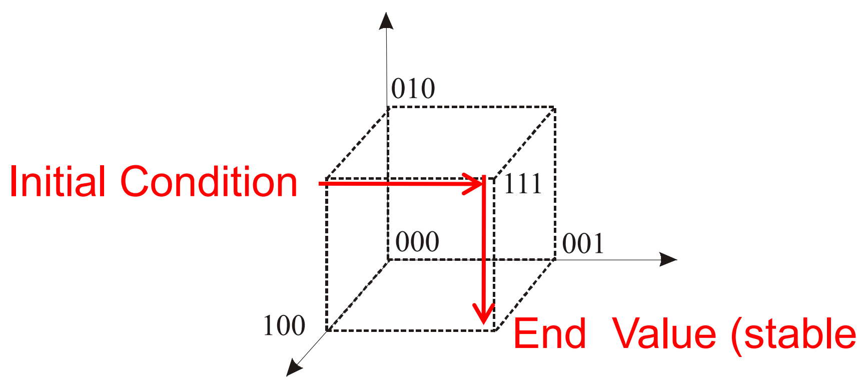


Hopfield Network with *n* Processing Elements

- Fast training and fast data recovery
- IIR system with no input (only I.C.)
- Guaranteed stability
- Good for VLSI implementation

-Operating Forms (firing order)

- Asynchronous
- Synchronous
- Sequential



Possible Hopfield Network states (8) with 3 Processing Elements
(Illustration of a typical recovery state evolution. From I.C. to E.V.)

Hopfield Network...

Learning:

The patterns to be stored in the associative memory are chosen a priori.

m distinct patterns. Each of the form:

$A_p = \lfloor a_1^p \quad a_2^p \quad \ldots \quad a_n^p \rfloor$, with $a_i^p = 0$ or 1. ($L = 0$, *usually*)

$$\boxed{w_{ij} = \sum_{p=1}^{m} (2a_i^p - 1)(2a_j^p - 1)}$$

Obs: $(2a_i^p - 1)$ converts 0/1 to –1/+1

w_{ij} is incremented by 1 if $a_i^p = a_j^p$ otherwise it is decremented
Procedure is repeated for each i,j for every A_p.

Learning is analogous to *reinforcement learning*

Hopfield Network - Example

Patterns to be stored as 3x3 matrices:

a_1	a_2	a_3
a_4	a_5	a_6
a_7	a_8	a_9

1		
1		
1	1	1

1	1	1
	1	
	1	

	1	
1	1	1
	1	

Symbol	Training Vector
L	$A_1 = [1\ 0\ 0\ \ 1\ 0\ 0\ \ 1\ 1\ 1]$
T	$A_2 = [1\ 1\ 1\ \ 0\ 1\ 0\ \ 0\ 1\ 0]$
+	$A_3 = [0\ 1\ 0\ \ 1\ 1\ 1\ \ 0\ 1\ 0]$

$$w_{ij} = \sum_{p=1}^{m} (2a_i^p - 1)(2a_j^p - 1)$$

Weigth Matrix

$$W = \begin{bmatrix} 0 & -1 & 1 & -1 & -1 & -3 & 1 & 1 & 1 \\ -1 & 0 & 1 & -1 & 3 & 1 & -3 & 1 & -3 \\ 1 & 1 & 0 & -3 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -3 & 0 & -1 & 1 & 1 & 1 & 1 \\ -1 & 3 & 1 & -1 & 0 & 1 & -3 & 1 & -3 \\ -3 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & -3 & -1 & 1 & -3 & -1 & 0 & -1 & 3 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & -1 \\ 1 & -3 & -1 & 1 & -3 & -1 & 3 & -1 & 0 \end{bmatrix}$$

Hopfield Network - Example

New pattern presented to the trained network:

1		1
1		
	1	1

$$\mathbf{x} = \mathbf{y}^{(0)} = [\,1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1\,]$$

$$W = \begin{bmatrix} 0 & -1 & 1 & -1 & -1 & -3 & 1 & 1 & 1 \\ -1 & 0 & 1 & -1 & 3 & 1 & -3 & 1 & -3 \\ 1 & 1 & 0 & -3 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -3 & 0 & -1 & 1 & 1 & 1 & 1 \\ -1 & 3 & 1 & -1 & 0 & 1 & -3 & 1 & -3 \\ -3 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & -3 & -1 & 1 & -3 & -1 & 0 & -1 & 3 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & -1 \\ 1 & -3 & -1 & 1 & -3 & -1 & 3 & -1 & 0 \end{bmatrix}$$

Sequential operation of the network:

Fired P.E.	P.E. Sum	P.E. Output	New output vector
1	2	1	1 0 1 1 0 0 0 1 1
2	-3	0	1 0 1 1 0 0 0 1 1
3	-4	0	1 0 **0** 1 0 0 0 1 1
4	1	1	1 0 0 1 0 0 0 1 1
5	-4	0	1 0 0 1 0 0 0 1 1
6	-4	0	1 0 0 1 0 0 0 1 1
7	4	1	1 0 0 1 0 0 **1** 1 1
8	0	1	1 0 0 1 0 0 1 1 1
9	4	1	1 0 0 1 0 0 1 1 1
1	2	1	1 0 0 1 0 0 1 1 1
2	-8	0	1 0 0 1 0 0 1 1 1

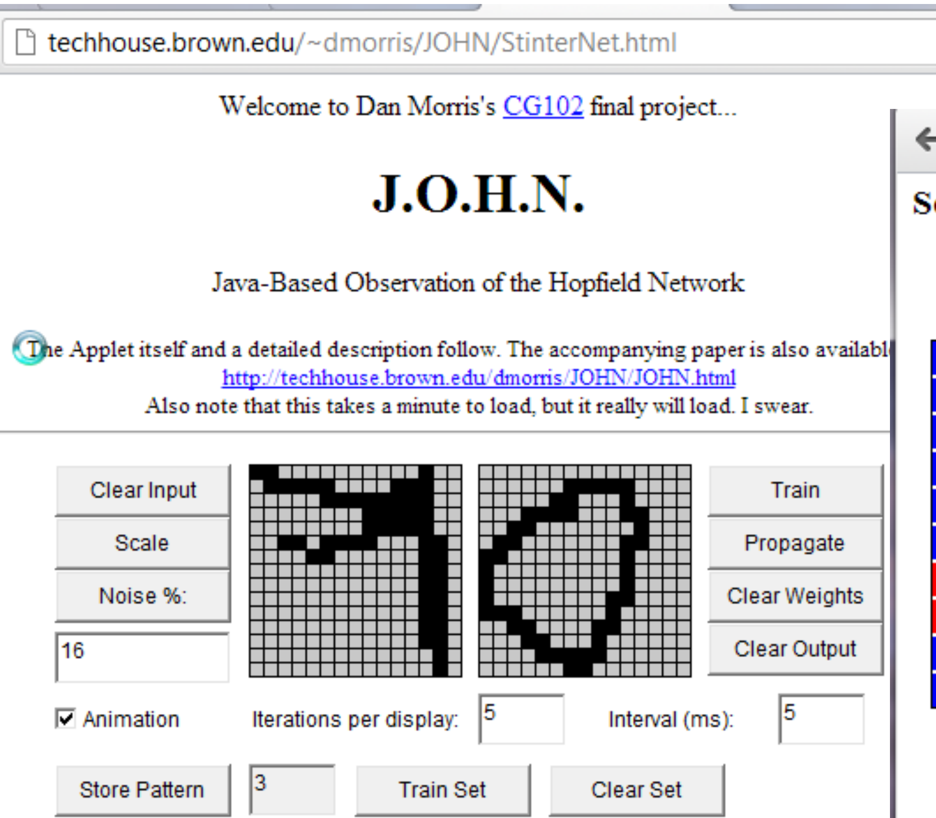
Convergence to “L” Pattern

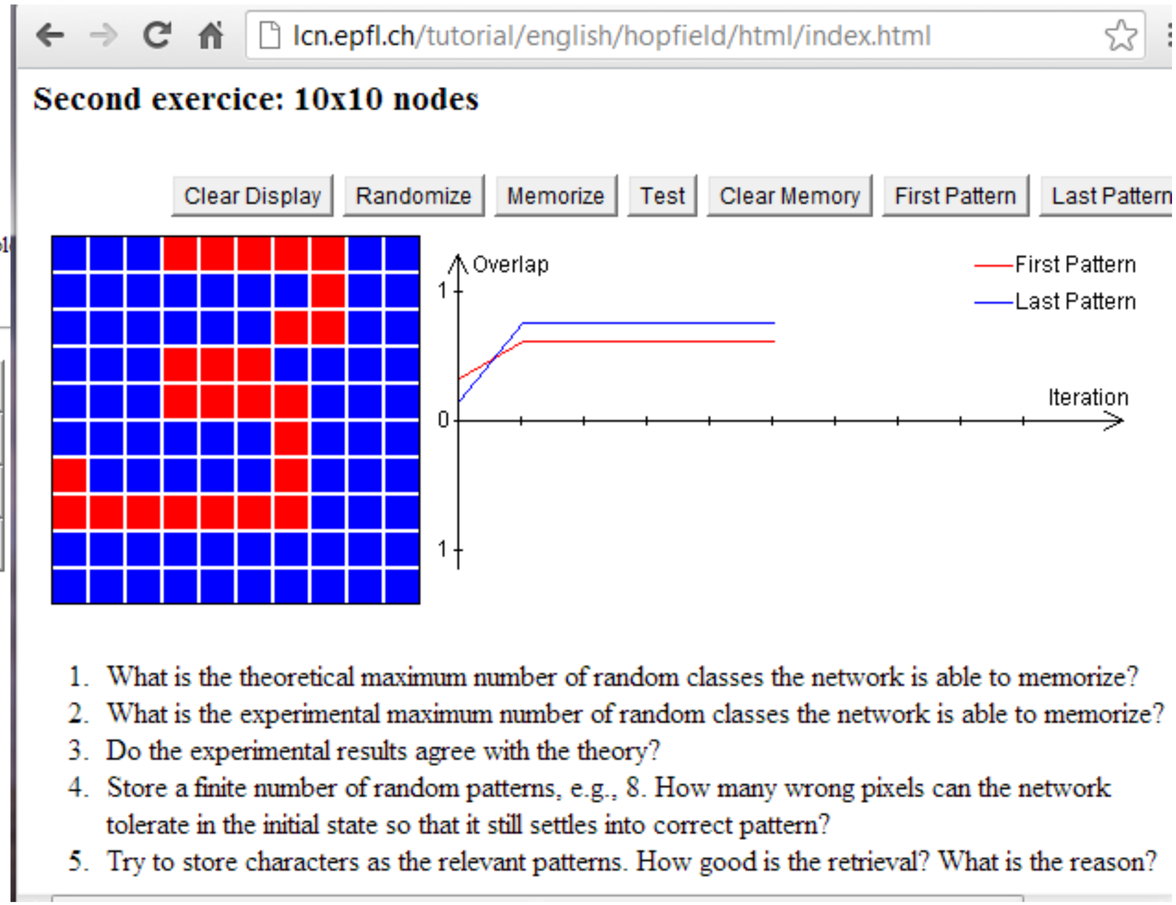
$$Remember - Binary\ activation\ function,\ L_j = 0:$$

$$f(s_j) = \begin{cases} 1\ if\ s_j > 0 \\ 0\ if\ s_j < 0 \\ hold\ previous\ value,\ if\ s_j = 0 \end{cases}$$

Hopfield Network – java demos

Demonstrations available in the www, e.g.:





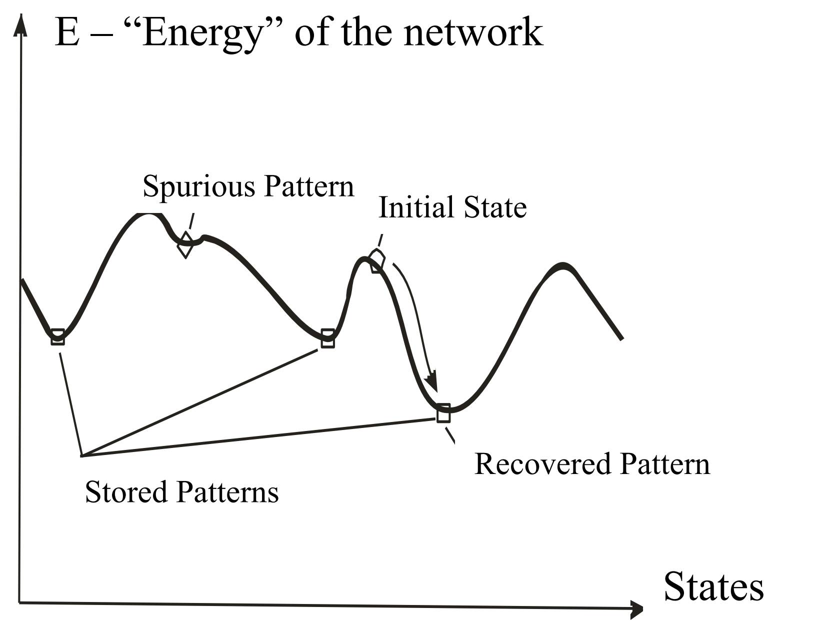
Hopfield N. – final considerations

Stability proof – Cohen and Grossberg, 1983.

W symmetric with zero diagonal

"Energy funcition" always decreases.

$$E = -\frac{1}{2}\sum_i \sum_j w_{ij} y_i y_j - \sum_j y_j L_j$$



Typical Energy and Patterns illustration for Hopfield Newtworks

Hopfield Network Limitations:

- Not necessarily the closest pattern is returned.
- Differences between patterns. Not all patterns have equal emphasis (size of attraction basins).
- Spurious patterns, i.e., patterns evoked that are not part of the stored set.
- Maximum number of stored patterns is limited.

$m \leq 0{,}5n / \log n$, m patterns, n bits network

Radial Basis Functions

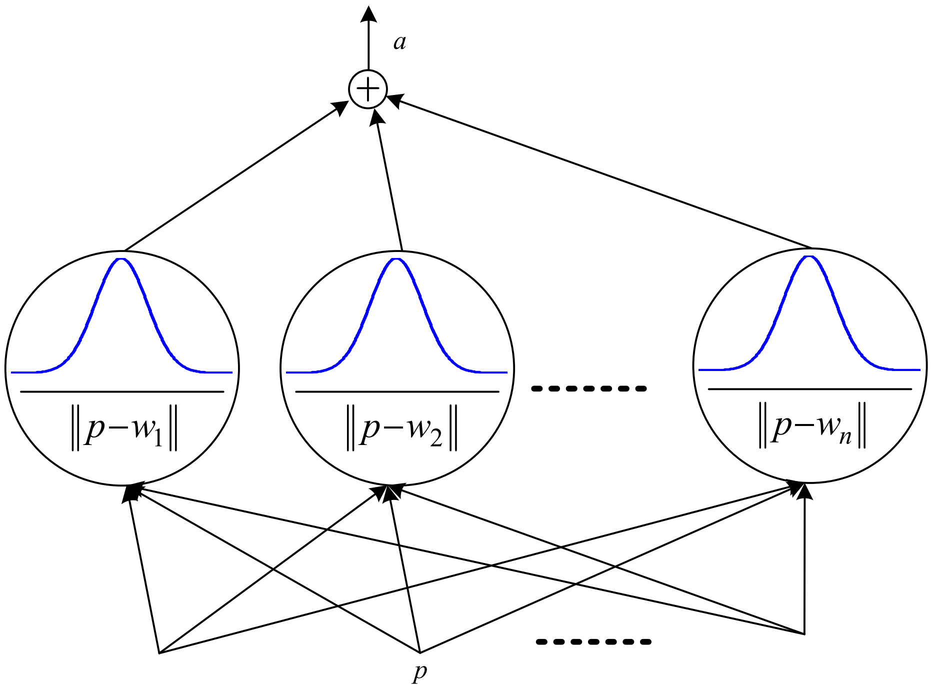
- Moody & Darken, 1989,...
- Function Approximators
- Inspiration: sensoricc overlapped reception fields in the cortex
- ***Localized activity*** of the processing elements

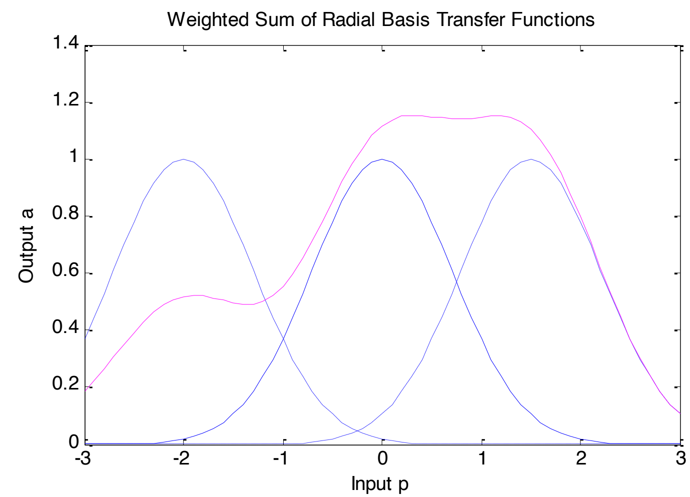
$$a_i = e^{-\frac{\|x_i-\mu_i\|}{\sigma_i^2}}$$

Gaussian
(Average, Variance)

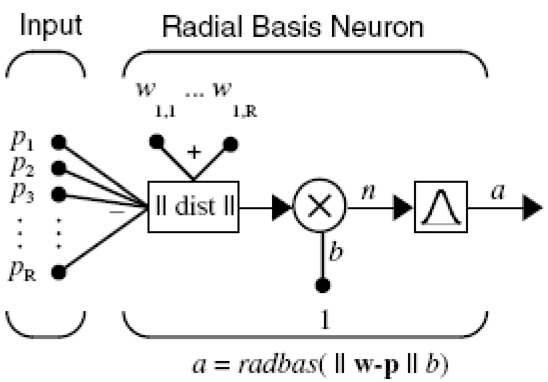
$$a_i = e^{-\|p_i-w_i\|b}$$

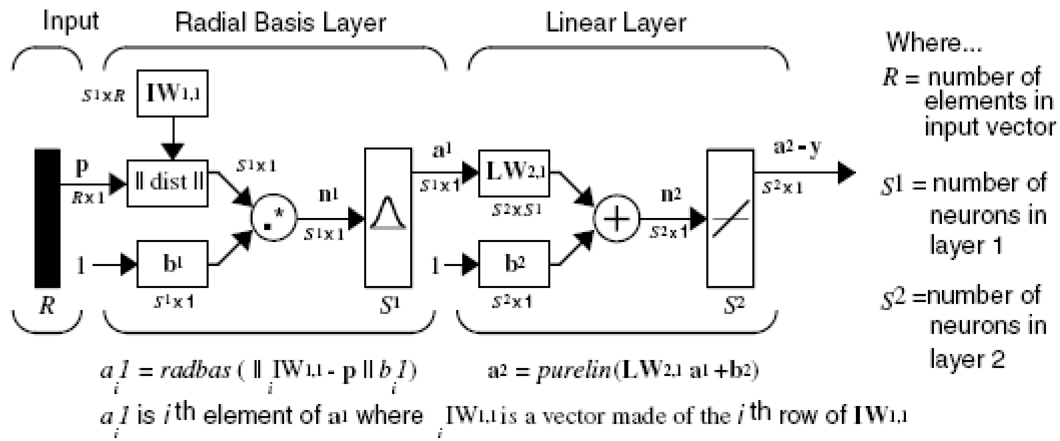
ml:
w – weigth
b – bias

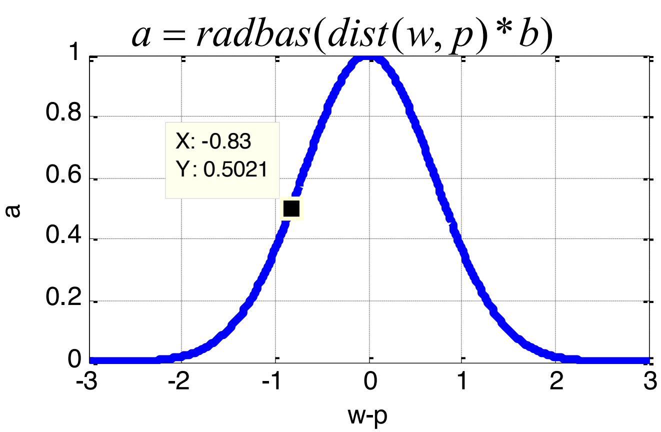




Radial Basis Functions...



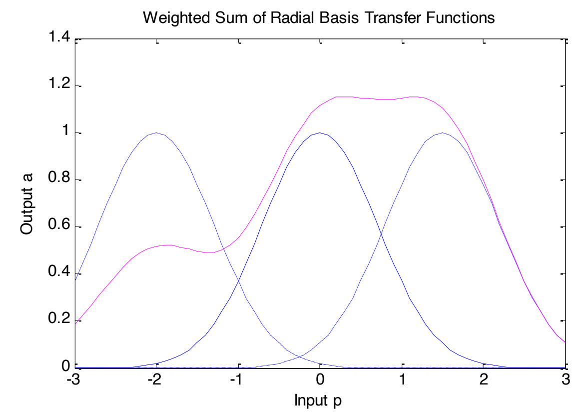




MatLab Implementation

[net,tr] = newrb(P,T,GOAL,SPREAD,MN)

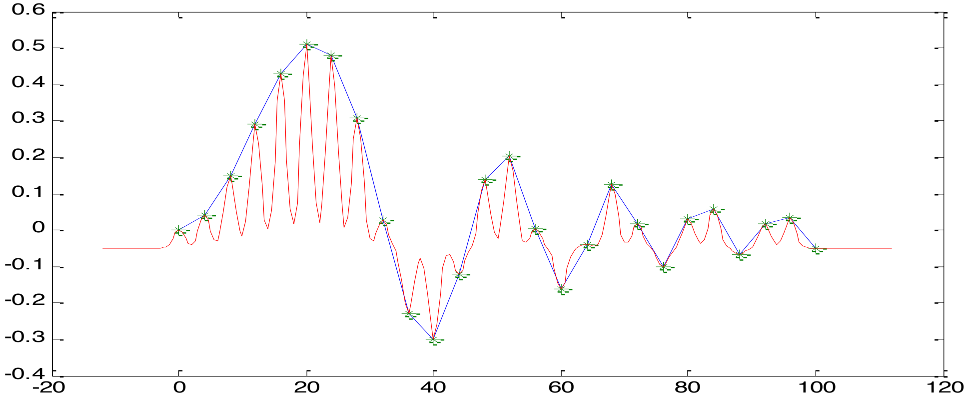
P - RxQ matrix, Q input vectors (“pattern”),
T - SxQ matrix, Q objective vectors (“target”),
GOAL - desired mean square error, default = 0.0,
SPREAD - radial basis function spread, default = 1.0,
MN - Maximum number of neurons, default is Q.



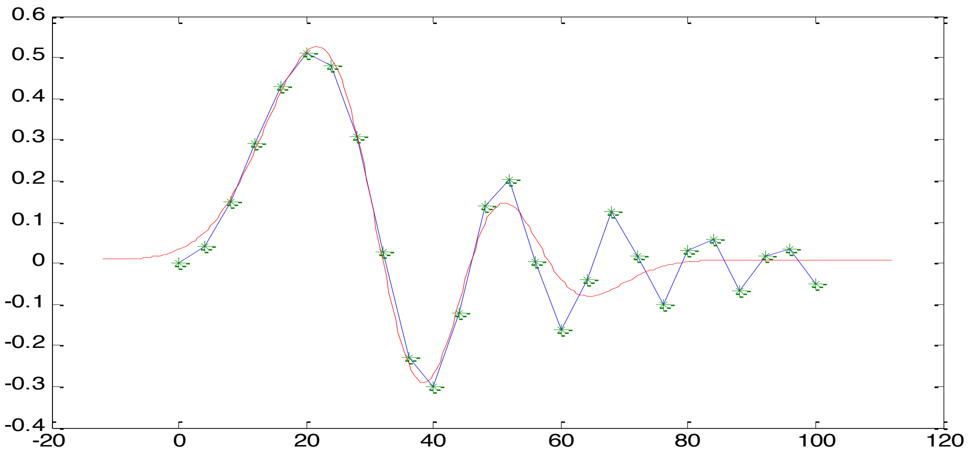
Radial Basis Functions...

Learning: add neurons incrementally

- Where? To obtain the largest quadratic errror reductions at each step $\rightarrow \min \sum e^2 = 0$

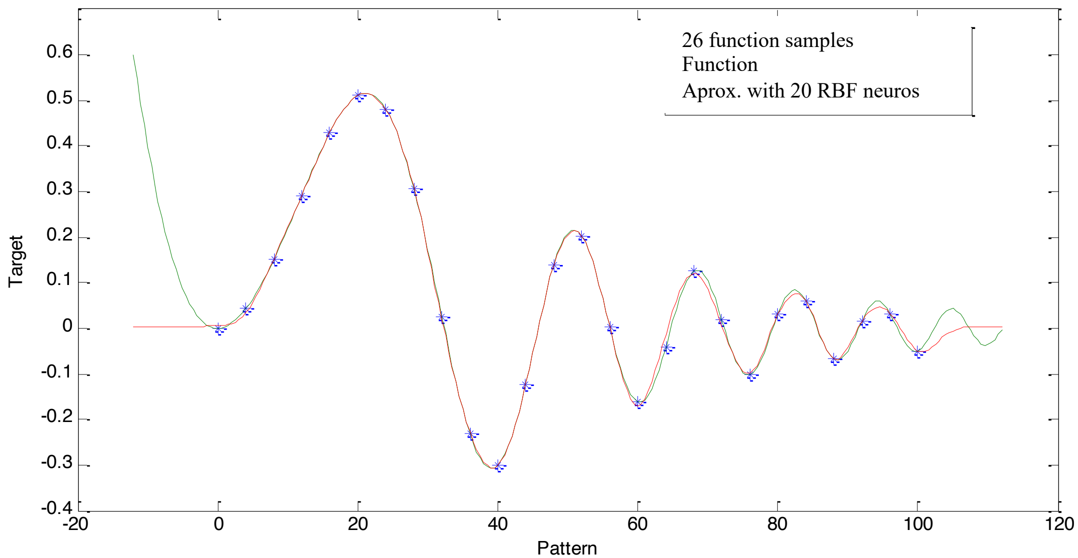


spread → to low
- Good **fitting** (at the training points)!
- Bad **interpolation!**



spread → to high
- Good **fitting at low frequencies**
- Good **interpolation in some ranges!**

Radial Basis Functions...



Heuristics:

$$|x_{i+1} - x_i| < SPREAD < |x_{\max} - x_{\min}|$$

spread → OK
- Good **fitting**!
- Good **interpolation!**

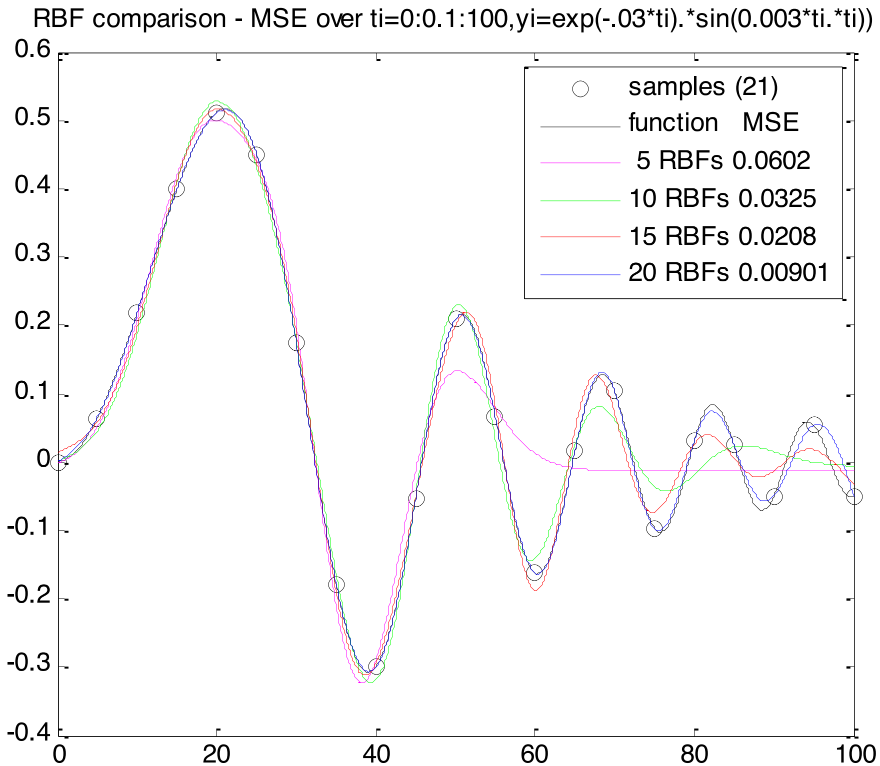
-Bad extrapolation
(is a very difficult task)

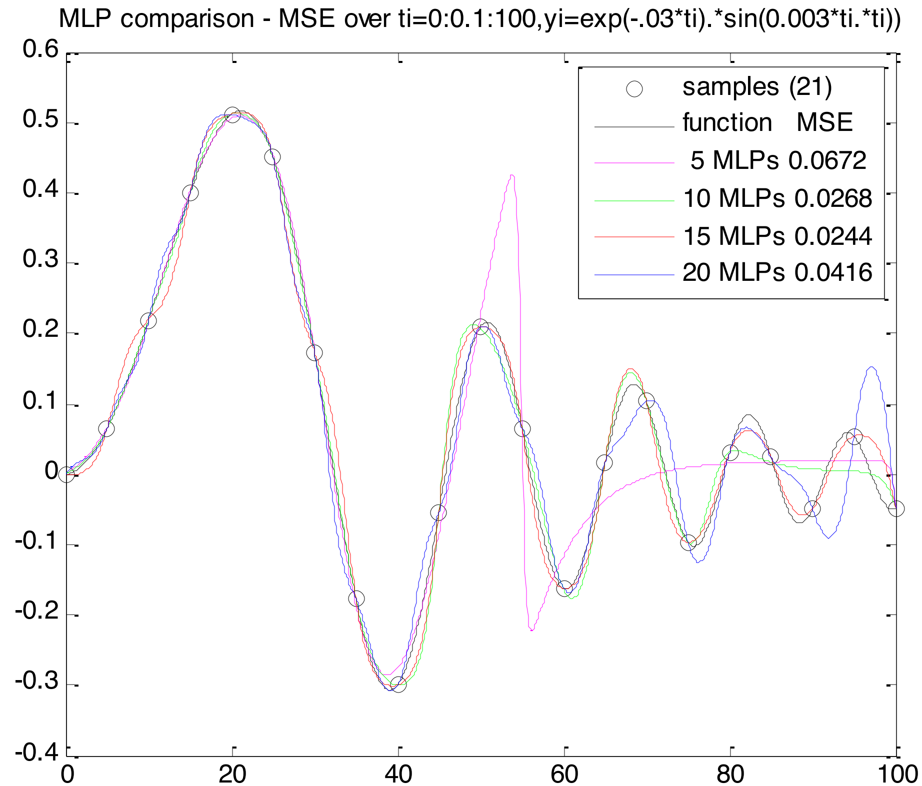
Conclusions

- Faster training faster, but uses more neurons than MLP.
- Incremental Training, new points can be learned without losing prior knowledge.
-You can use *a priori* knowledge to locate neurons (which is not possible in a MLP).

- Fixed spread – Incremental traning → **suboptimal solution**!!

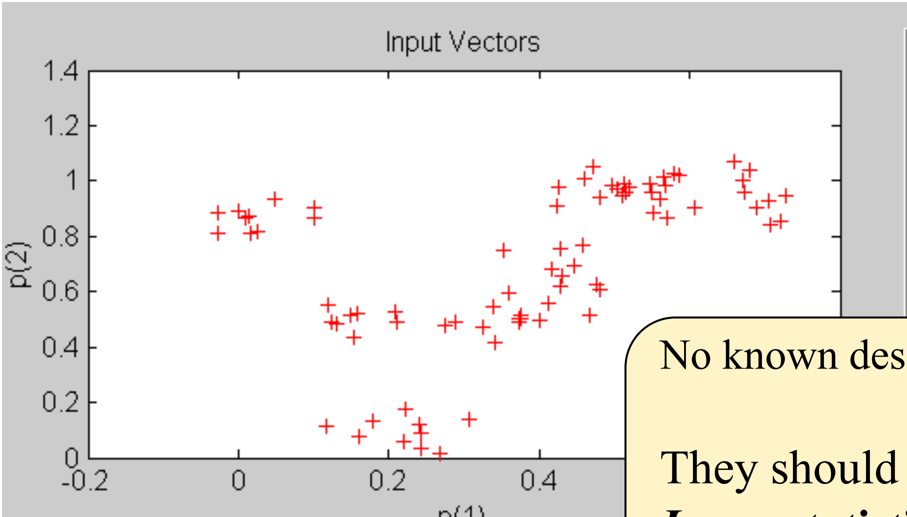
Comparison RBF x MLP

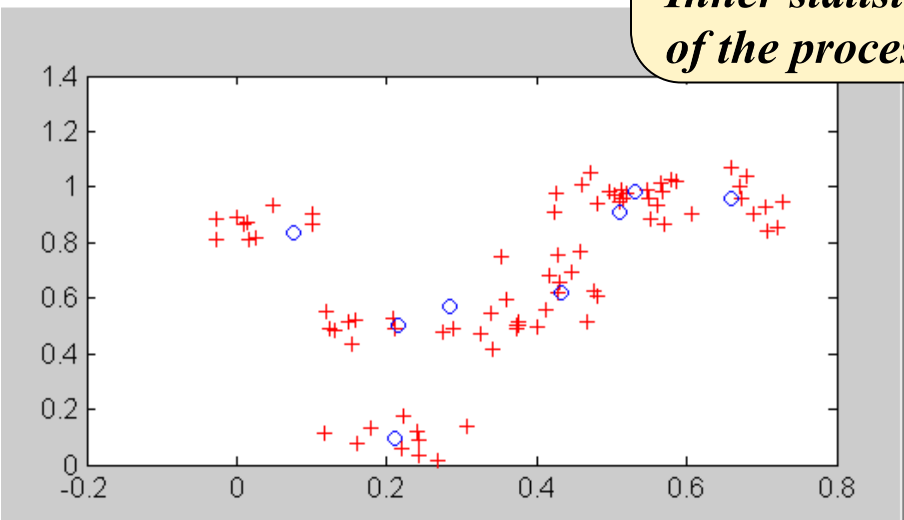




RBF – more neurons better fitting → best solution newrbe (exact fitting!)
MLP – too much neurons → worse fitting (bad interpolation)

Unsupervised Learning





No known desired Code Vectors

They should reflect the
Inner statistical distribution of the process

Competitive Layer

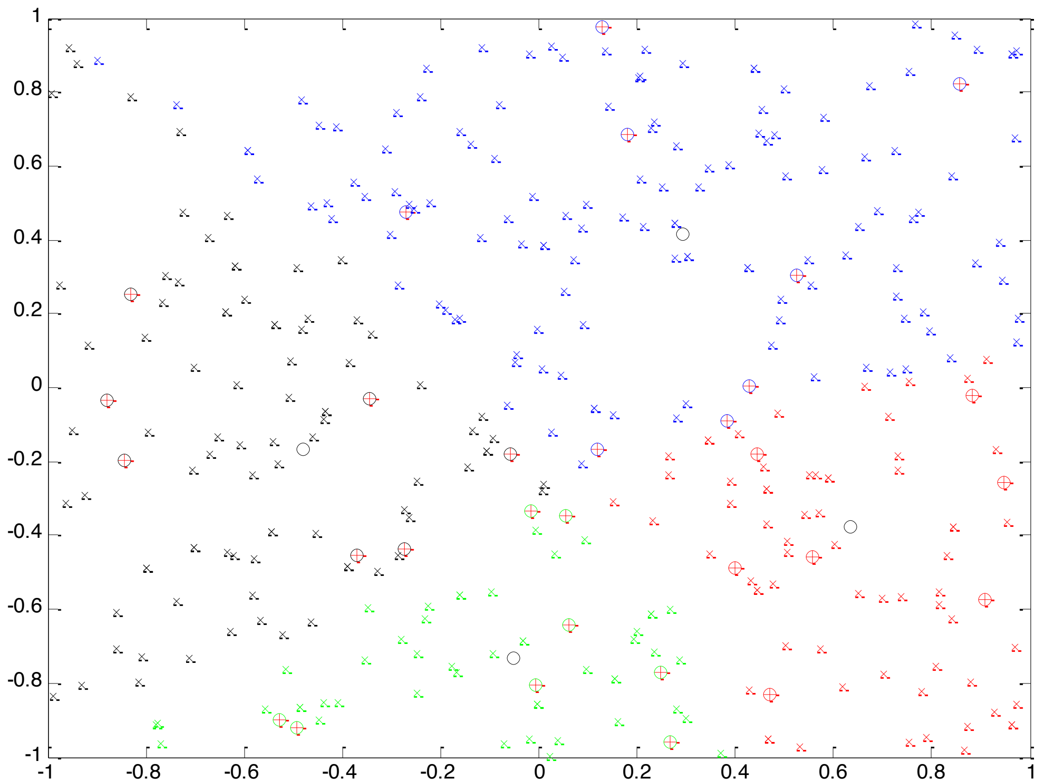
Find vector codes that describe the data distribution

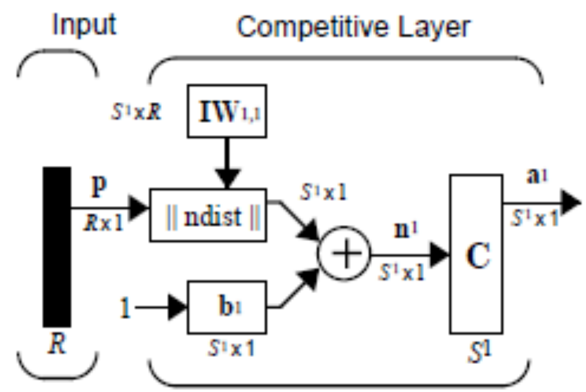
Used in Data Compression

Example:
Code symbols that will be transmitted over a communication channel.

For the comprehension the variability of the signal is considered as "noise", and so, discarded.

Competitive Layer



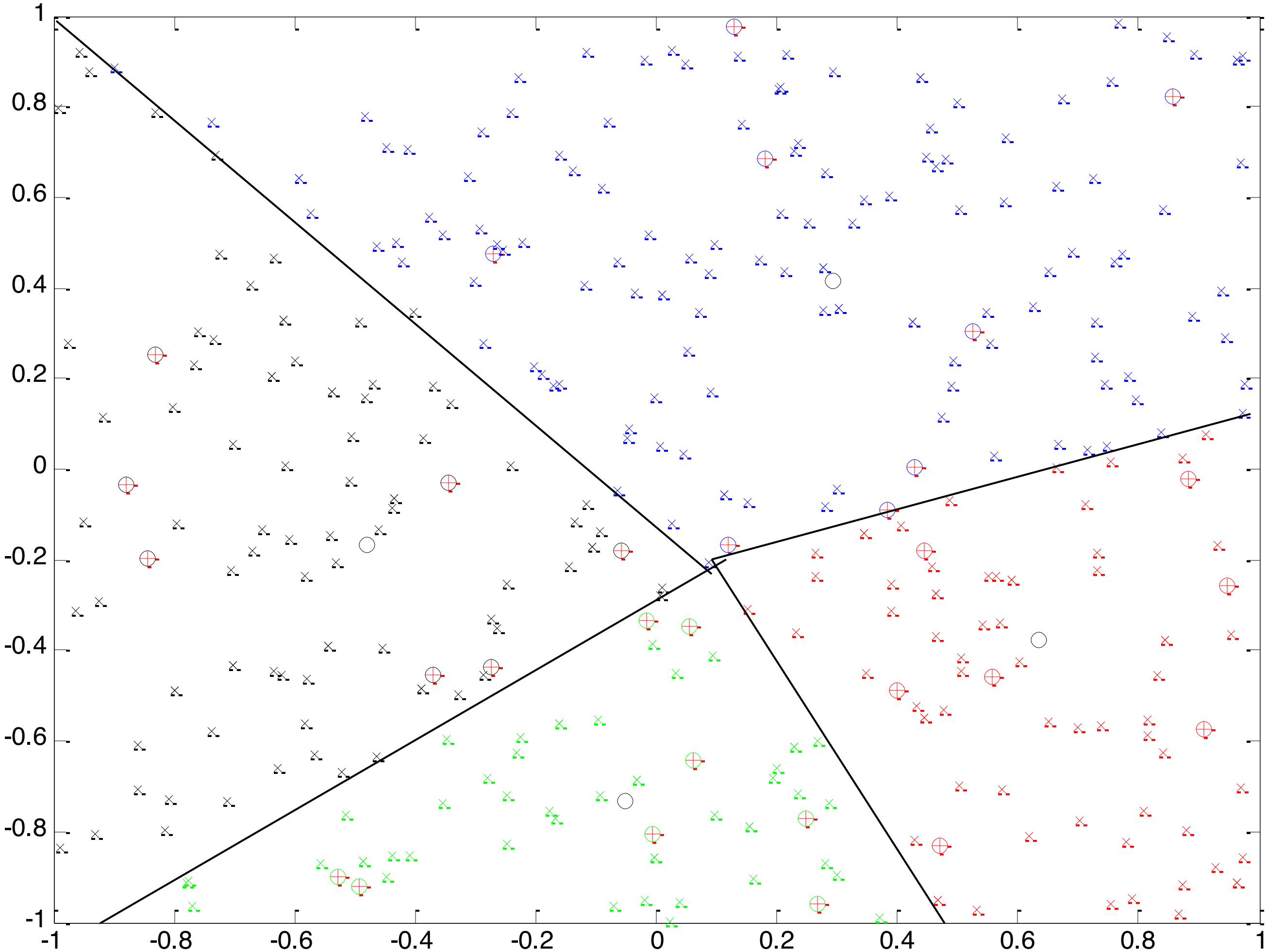


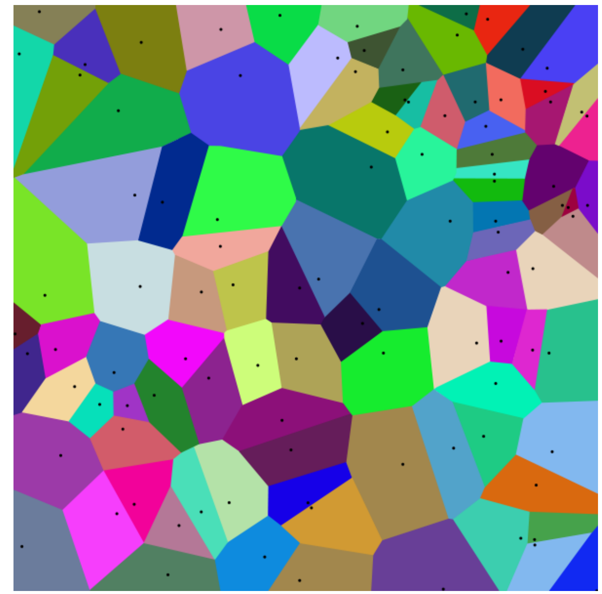
o code vectors

⊕ training vectors

x test vectors

Borders



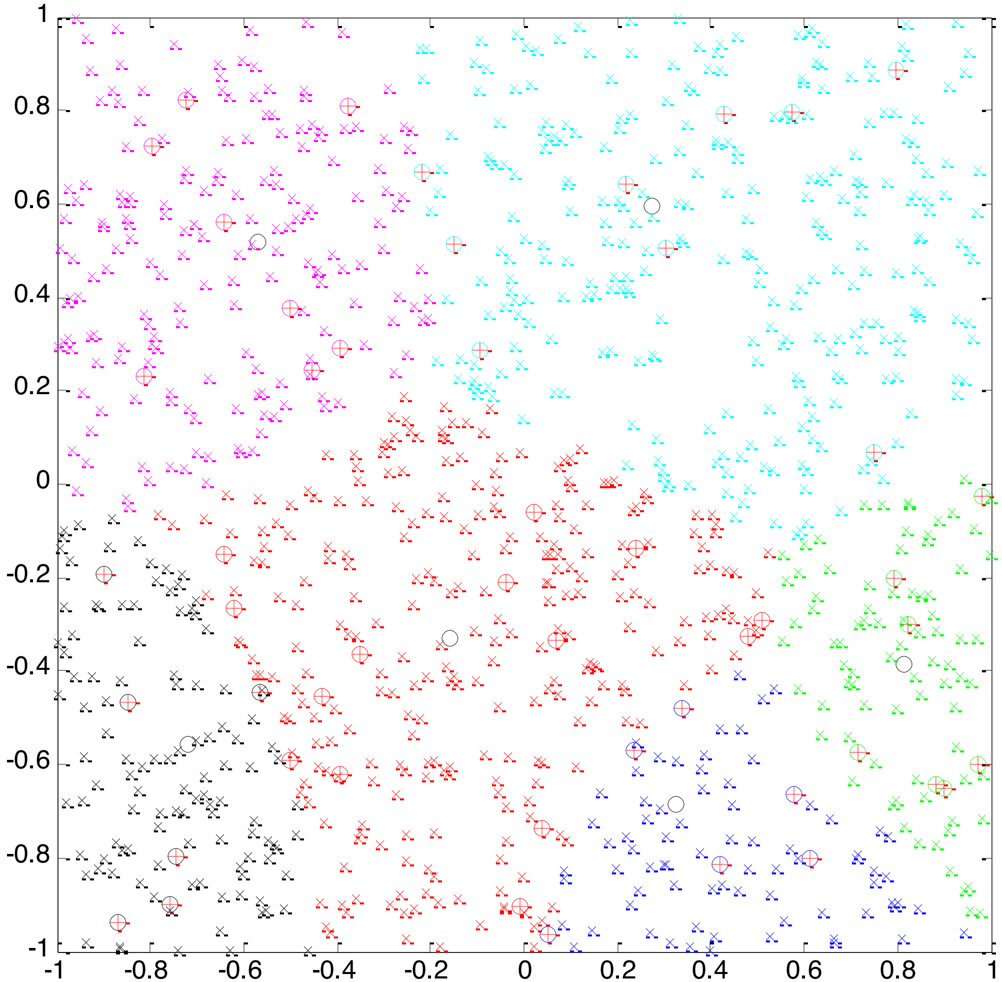


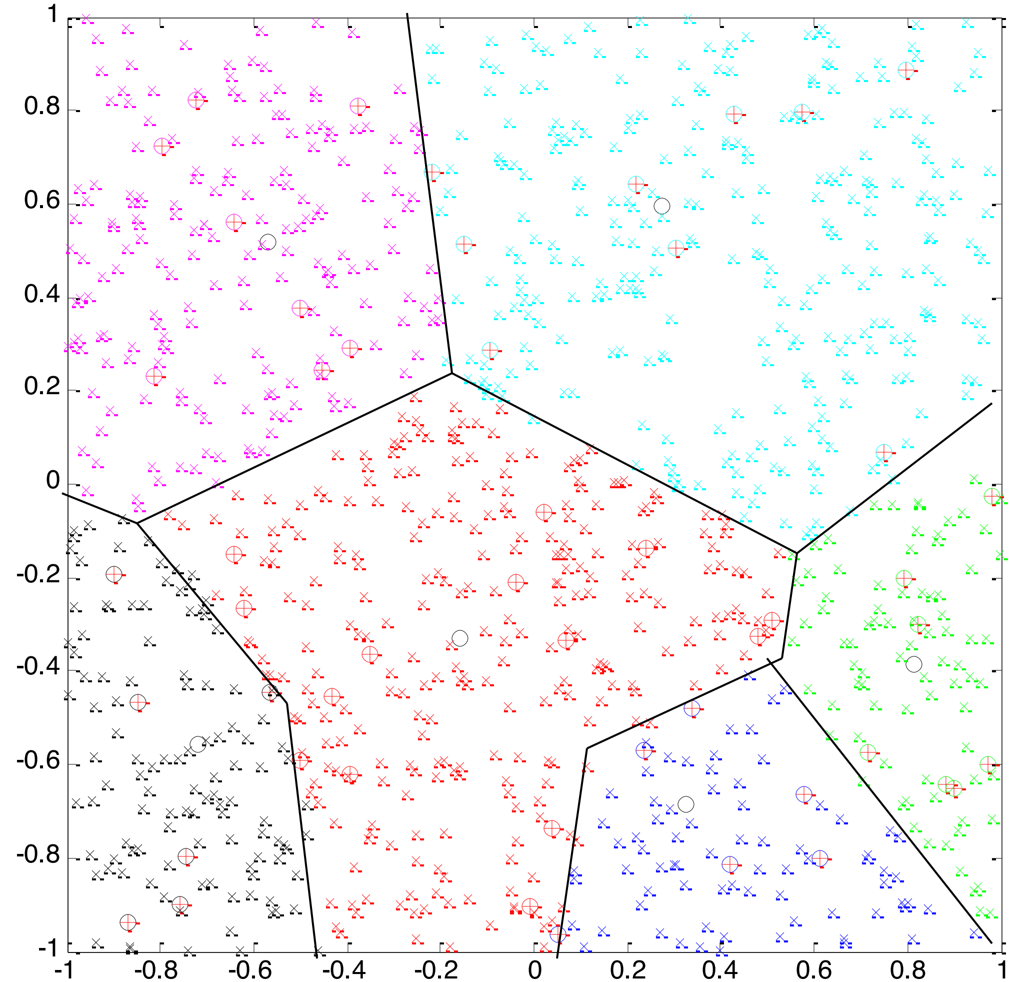
(Voronoi Diagram)

o code vectors
⊕ training vectors
x test vectors

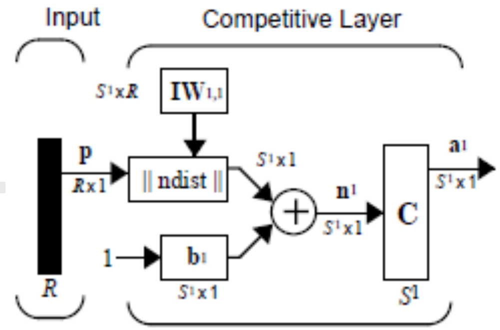
Ex. Classification Borders

o code vectors
⊕ training vectors
x test vectors

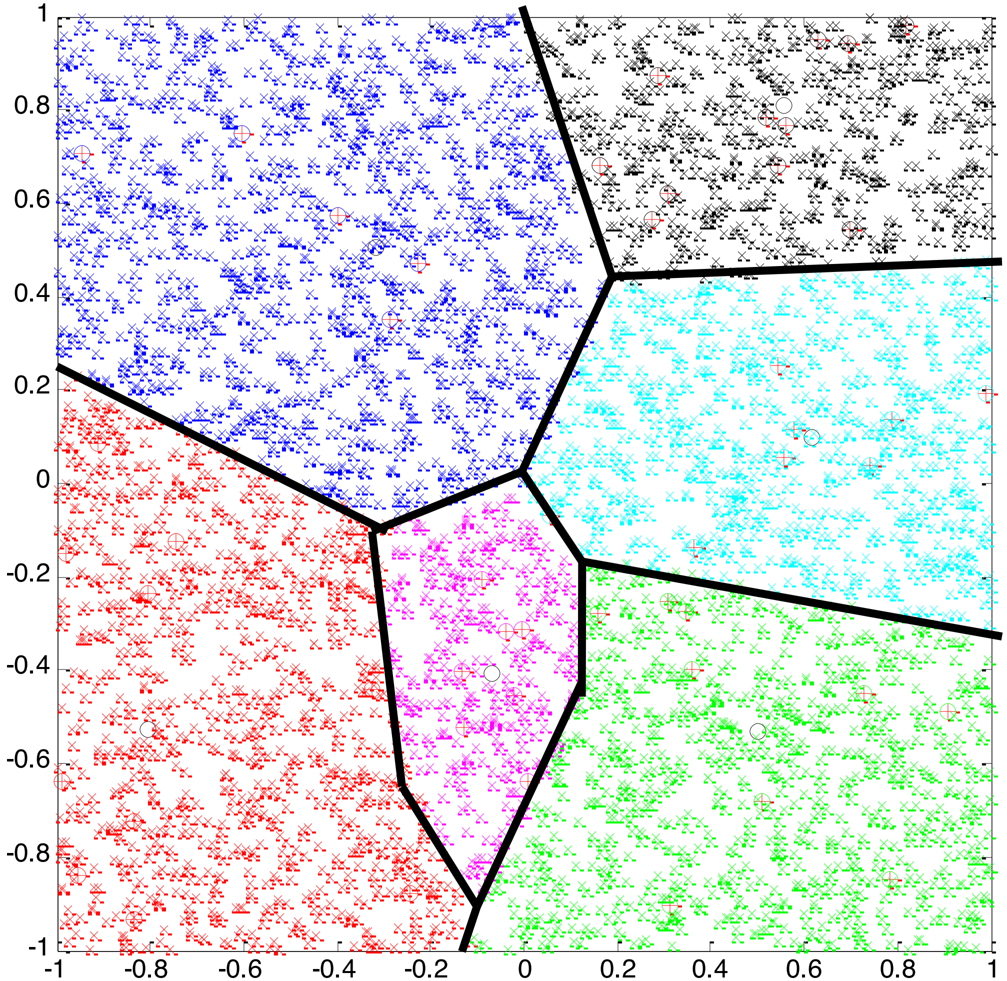




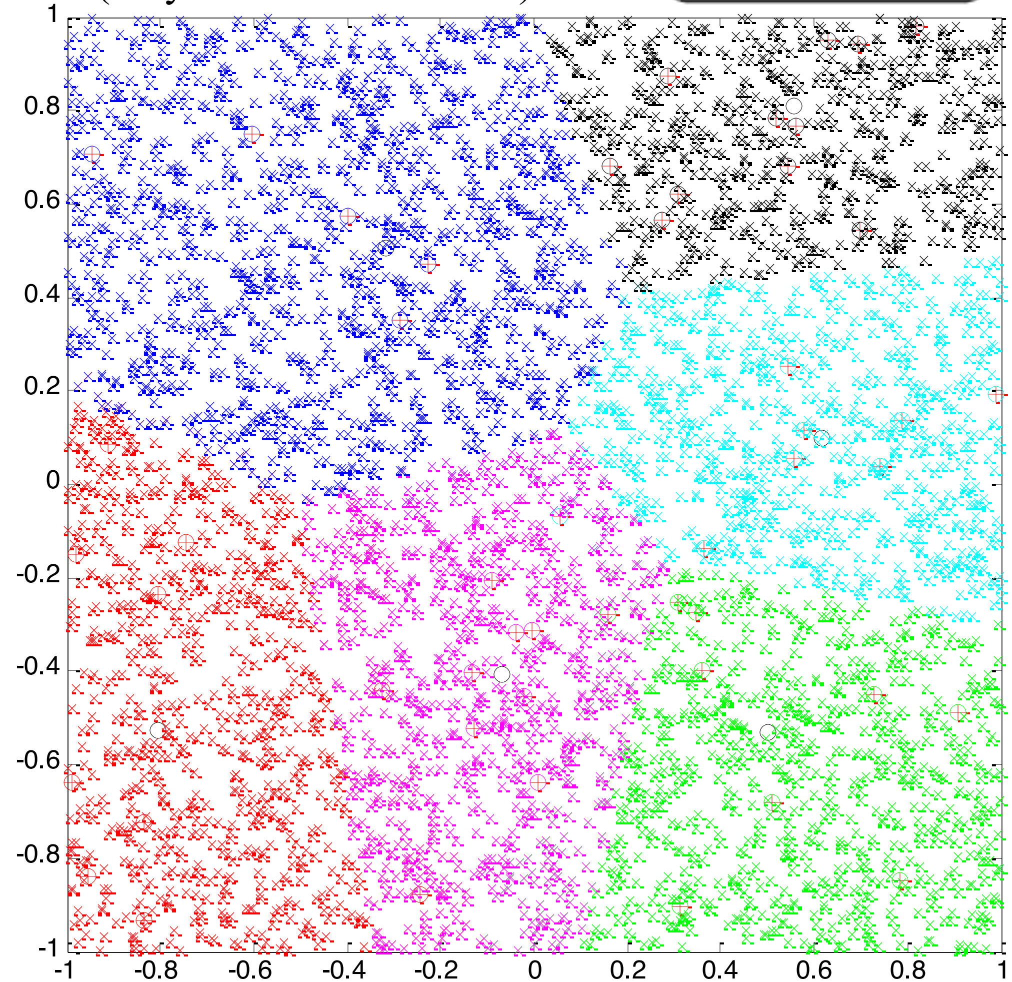
Bias



Bias adjustment to to help "weak" neurons



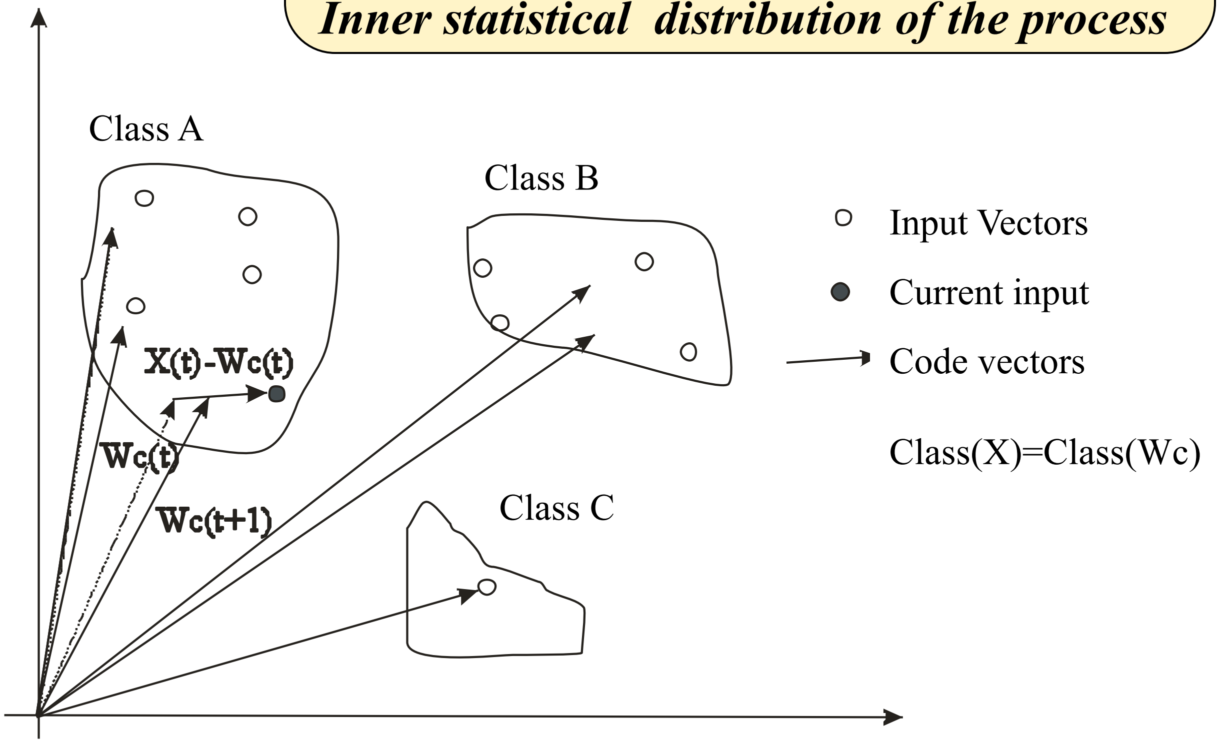
Bias = 0
(only Euclidean distance)



Learning Vector Quantization

No known desired Code Vectors

Data points belong to Classes
Code Vectors should reflect the
Inner statistical distribution of the process



LVQ1, LVQ2.1, LVQ3, OLVQ

"Enhanced Algorithms"

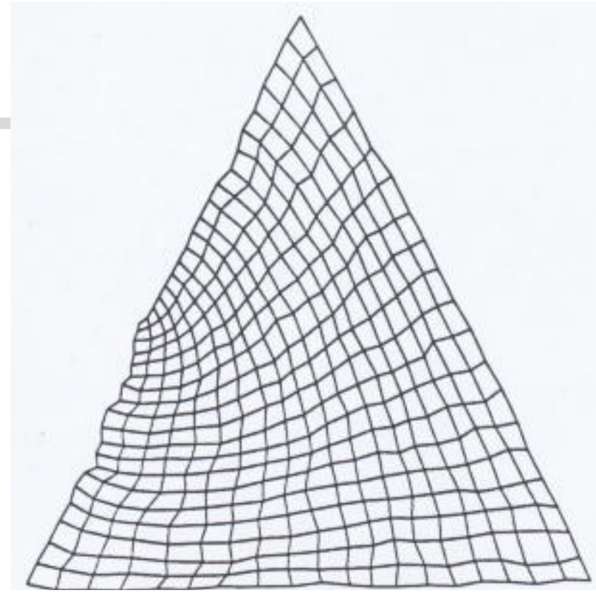
-Dead neurons
-Neighborhood definition

Self Organizing Maps

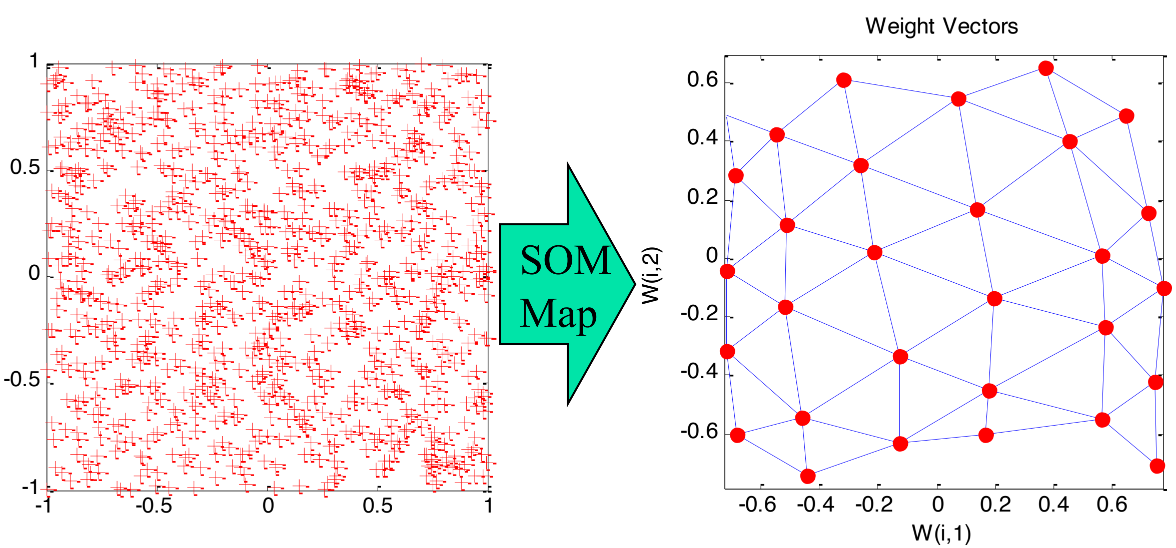
Kohonen, 1982 – Unsupervised learning

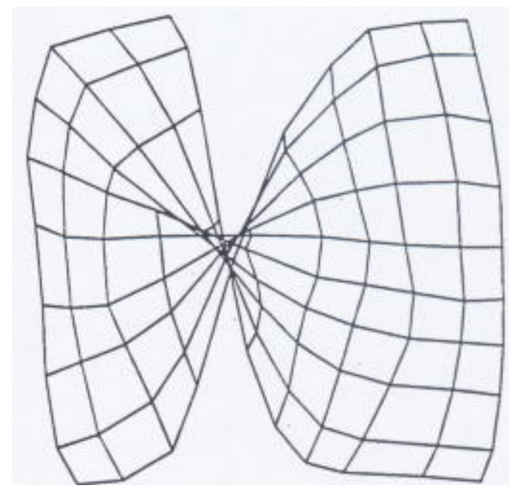
One active layer with *neighborhood* constrains

Code Vectors should reflect the
Inner statistical distribution of the process



Triangular distribution





Weird unsuccessful training

ANN General Characteristics

- Positive
  - Learning
  - Parallelism
  - Distributed knowledge
  - Fault Tolerant
  - Associative Memory
  - Robust against Noise
  - No exhaustive modelling
- Negative
  - Knowledge acquisition only by learning
    (“E.g., Wich topology is best suit?)
  - Introspection is not possible
    (“What is the contribution of this neuron?)
  - The logical inference is hard to obtain
    (“Why this output for this situation?”)
  - Learning is slow
  - Very sensitive to initial conditions

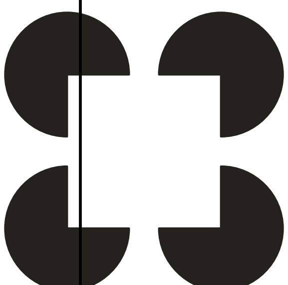
To obtain successful ANN
a good process knowledge is
recommended in order to design
experiments that produce useful data sets!

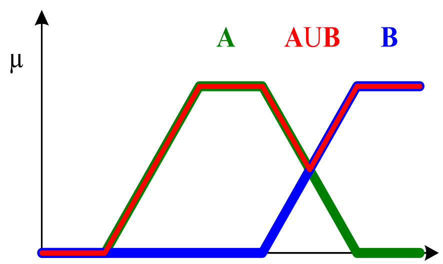
There is no free lunch!

The picture can't be displayed.

The picture can't be displayed.

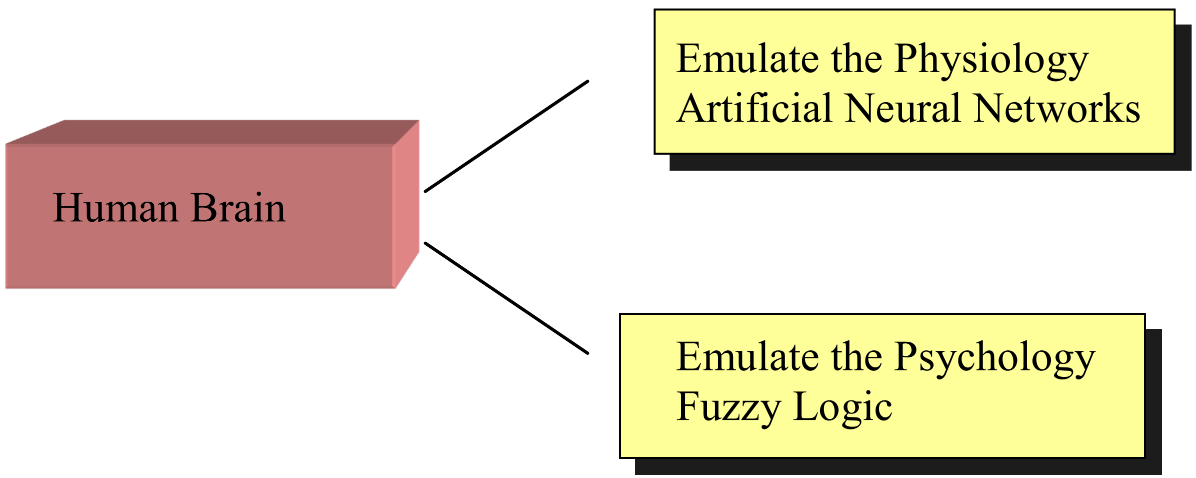
Part 3 –Fuzzy Logic and Fuzzy Systems





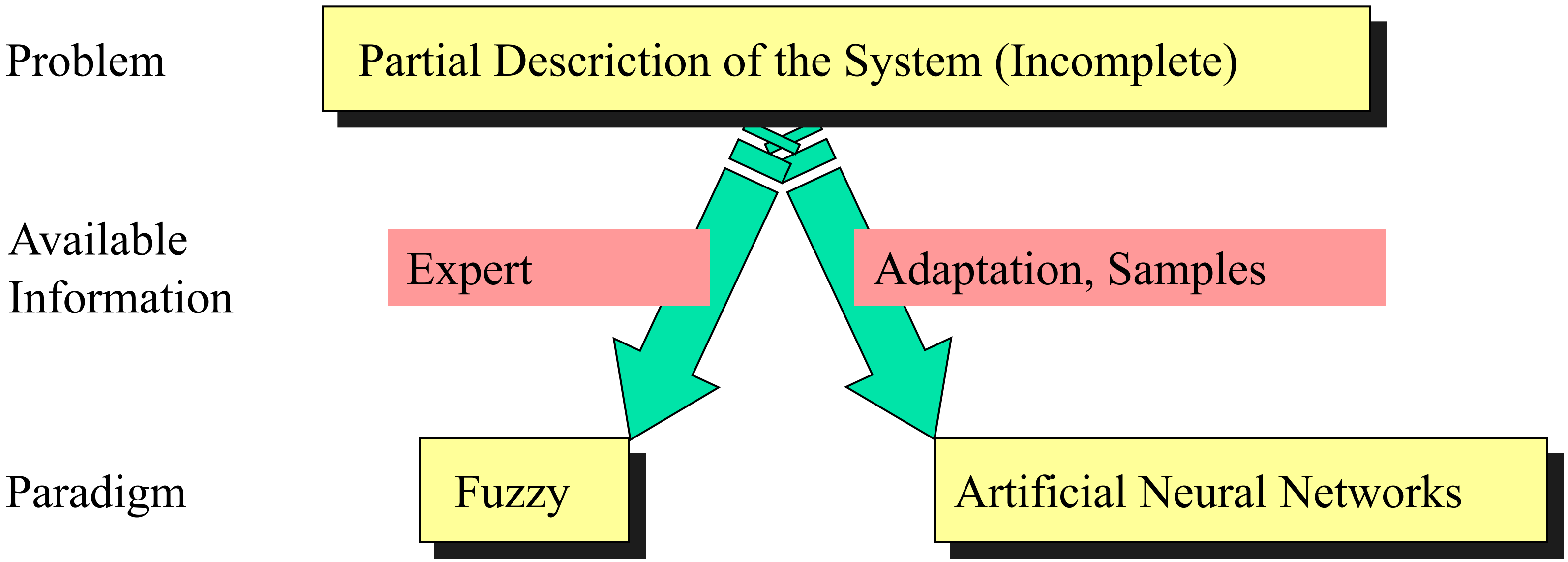
Fuzzy Logic

- The fuzzy set theory was proposed by Lotfi Zadeh in 1965.
- Was long misunderstood.
- In the mid-80s used to design Mamdani fuzzy controllers

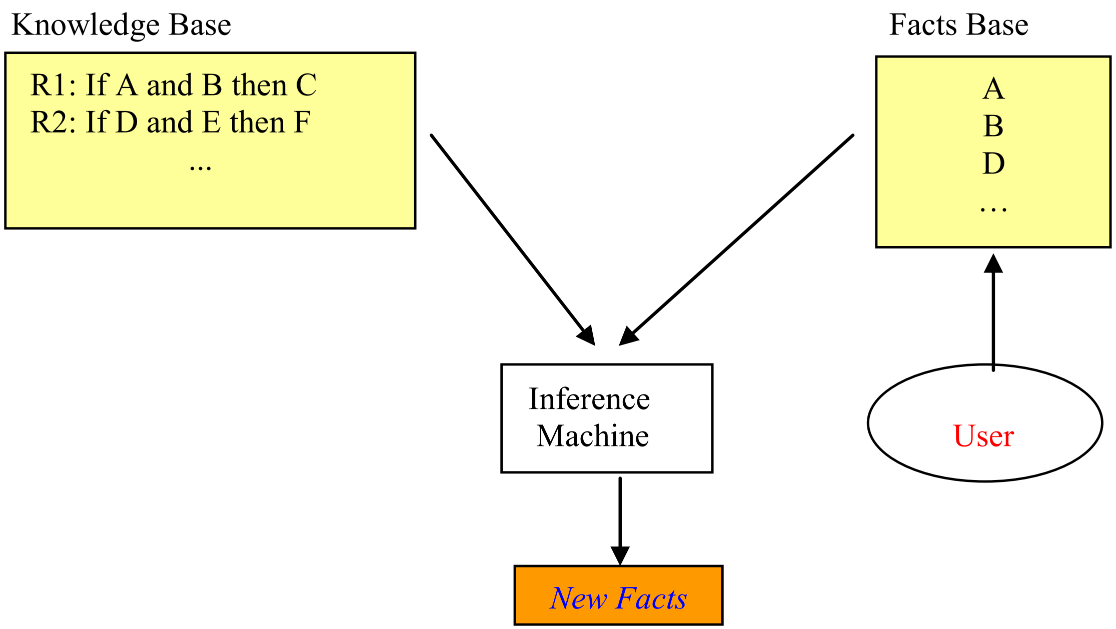


Fuzzy Logic

According to the availability of
an ***expert*** Or ***samples*** of a system
the fuzzy or the RNA paradigm is indicated.

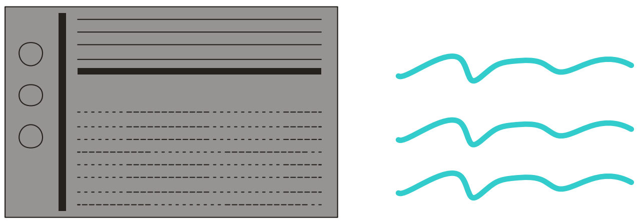


Expert Systems

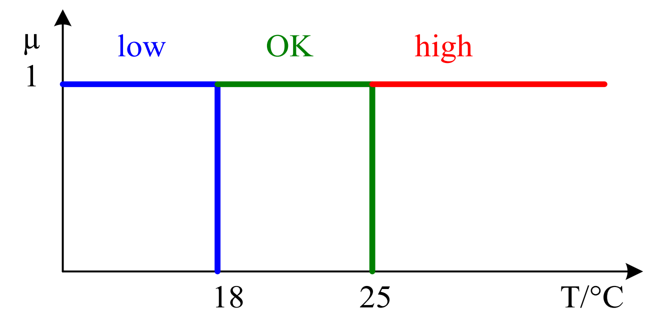


1. Extraction of Knowlegde of Expert (build knowledge base)
2. I.T. expert creates the environment (shell)
3. “Normal” human presents facts and questions (over and over)
4. Response of ES similar to the human expert!

Example of "If-Then" Rules



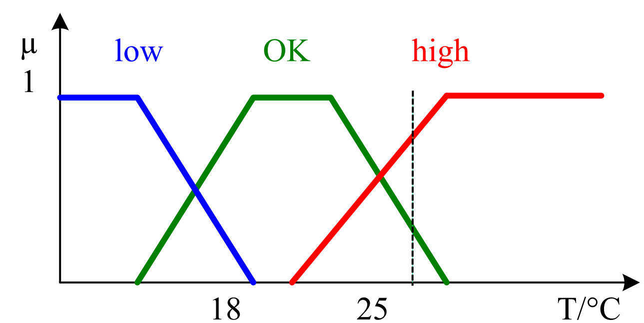
Air Conditioner



If temperature is **low**
Then reduce air conditioning

If temperature is **OK**
Then do nothing

If temperature is **high**
Then increase air conditioning power



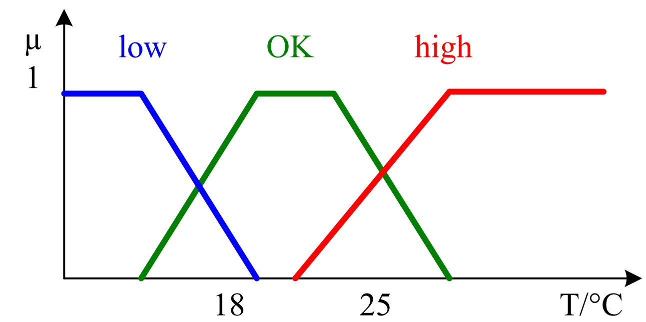
Partial membership to both linguistic concepts!

Fuzzy Sets – Membership Fuction

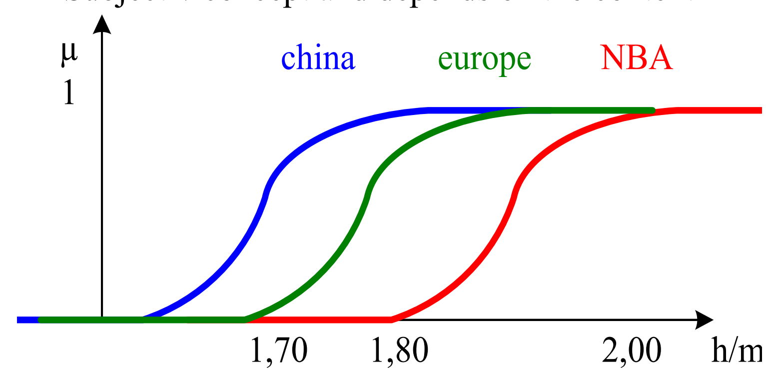
$$\mu_A(x): X \to [0,1]$$

$$\mu_A(x) = \begin{cases} 1 & if\ \ x\ is\ a\ \ fully\ member\ of\ \ A \\ (0,1) & if\ x\ belongs\ \ partially\ to\ \ A \\ 0 & if\ \ x\ is\ not\ \ a\ \ member\ of\ \ A \end{cases}$$

Fuzzy Sets



Tal person:
Subjectiv concept and depends on the context



Extension of the Boolean Logic – Historical perspective:

~ 1930, Lukasiewicz : {0,1/2,1}, [0,1]
1937, Black : Membership fuction
1965, Lotfi Zadeh : Fuzzy Sets
Multivalent Set Theory
~ 1988, Commercial Products : "third wave" of interest

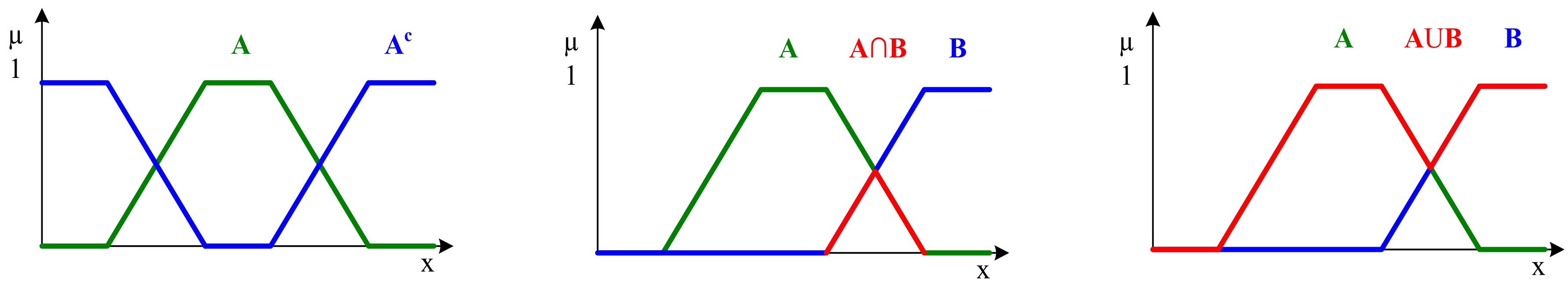
Fuzzy Sets – operations

Ex.: Complement, Intersection and Union

$$\mu A^c(x) = 1 - \mu A(x)$$

$$\mu A \cap B(x) = \min(\mu A(x), \mu B(x))$$

$$\mu A \cup B(x) = \max(\mu A(x), \mu B(x))$$



Fuzzy Sets – Properties

Involution	$(A^C)^C = A$	
Comutativity	$A \cup B = B \cup A$	$A \cap B = B \cap A$
Associativity	$A \cup (B \cup C) = (A \cup B) \cup C$	$A \cap (B \cap C) = (A \cap B) \cap C$
Distributivity	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Idempotency	$A \cup A = A$	$A \cap A = A$
Absorption	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$
Identity	$A \cup \Phi = A$	$A \cap \Omega = A$
Absorption by Ω e Φ	$A \cup \Omega = \Omega$	$A \cap \Phi = \Phi$
De Morgan's Law	$(A \cap B)^C = A^C \cup B^C$	$(A \cup B)^C = A^C \cap B^C$

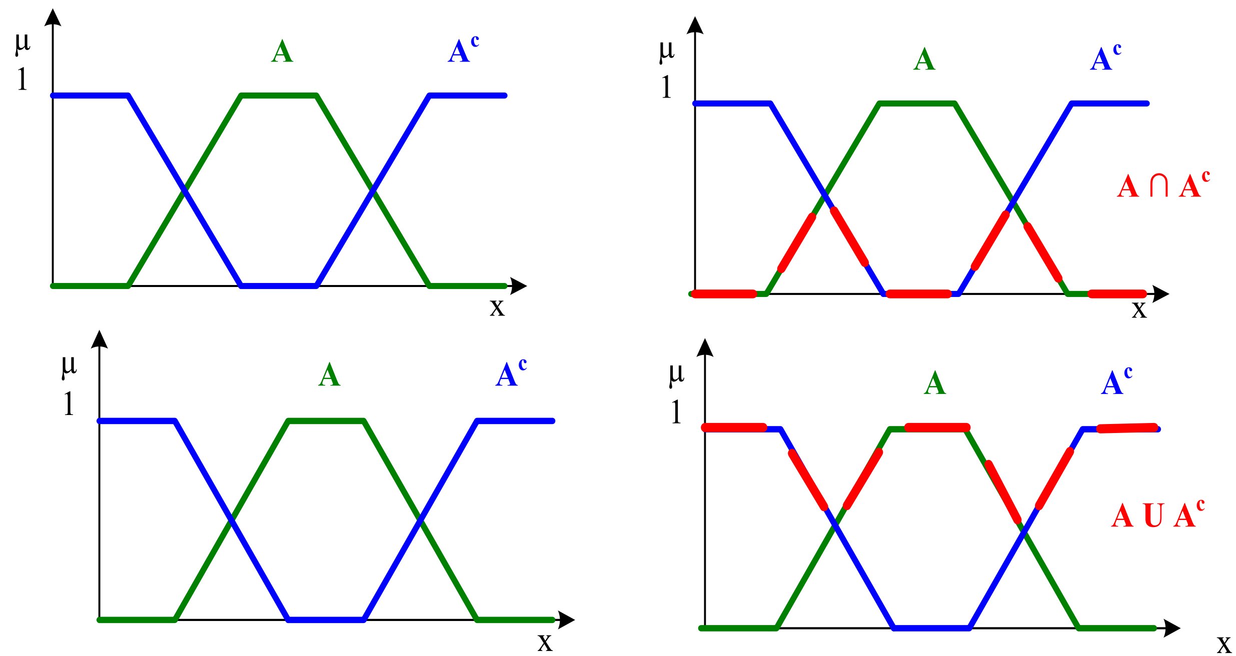
However:

$A \cap A^C \neq \Phi$	Does not satisfy the law of no-contradiction
$A \cup A^C \neq \Omega$	Does not satisfy the law of third excluded
$A \cup (A^C \cap B) \neq A \cup B$	Does not satisfy the absorption of the complement
$A \cap (A^C \cup B) \neq A \cap B$	Does not satisfy the absorption of the complement

Fuzzy Sets – Examples of “strange” behavior

$A \cap A^C \neq \Phi$ Does not satisfy the law of non-contradiction

$A \cup A^C \neq \Omega$ Does not satisfy of the third excluded



Sentence Calculus – Classical Logic

In Classical Logic, the truth values of propositions (sentence calculus) are obtained by the following truth table ("modus ponens" – afirmative modus).

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$ $(\neg A \vee B)$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

Sentence Calculus – Fuzzy Logic

When the information is inaccurate, the inference engine implements the so-called approximate reasoning.

Fuzzy logic implements approximate reasoning in the context of fuzzy sets ("generalized modus ponens").

fact:	A`	Tomatoes are very red
rule:	$A \rightarrow B$	If tomatoes are red then they are mature
consequence	B`	The tomatoes are very ripe

$\neg A = n(A)$	n – negation
$A \wedge B = T(A,B)$	T – t -norm
$A \vee B = S(A,B)$	S – t-conorm
$A \rightarrow B = I(A,B)$	I – implication

Implication Operators

“If <premise> Then <conclusion>”

$I : [0,1]^2 \to [0,1]$, $\mu A : X \to [0,1]$, $\mu B : Y \to [0,1]$

$\mu A \to B\ (x,y) = I(\mu A(x), \mu B(y))$

Implication	Name
max (1-a,b)	Kleene-Dimes
min(1-a+b,1)	Lukasiewicz
min(a.b)	Mamdani
a.b	Larsen
…	

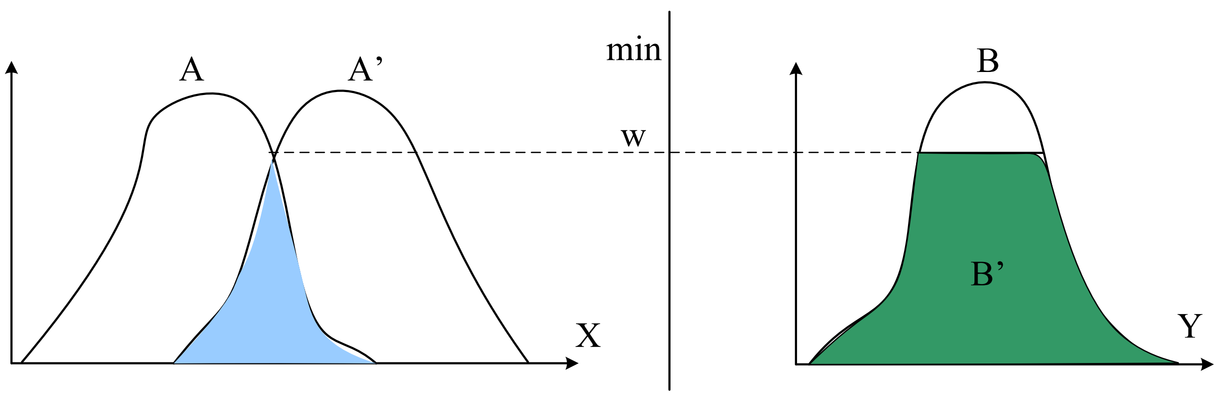
Fuzzy Reasoning
based on Max-Min composition

Definition: Let A, A' and B fuzzy sets on X, X and Y respectively. Assume that the fuzzy implication A → B is expressed by the fuzzy relation R on X x Y, then the fuzzy set B' is induced "x is A'" and the fuzzy rule "**if x is A then y is B**" is defined by:

$$\mu B'(y) = \max_x \min[\mu A'(x), \mu R(x,y)]$$
$$= V_x [\mu A'(x) \wedge \mu R(x,y)],$$

that means: $B' = A' \circ R = A' \circ (A \to B)$

One fuzzy rule with one antecedent

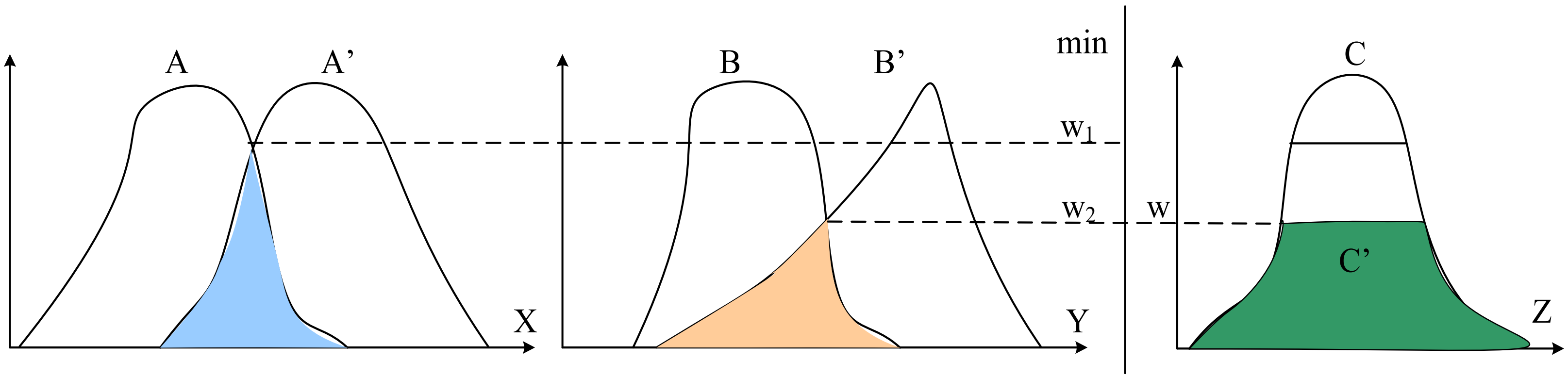


w→ degree of membership to the rule

Fuzzy Reasoning
based on Max-Min composition

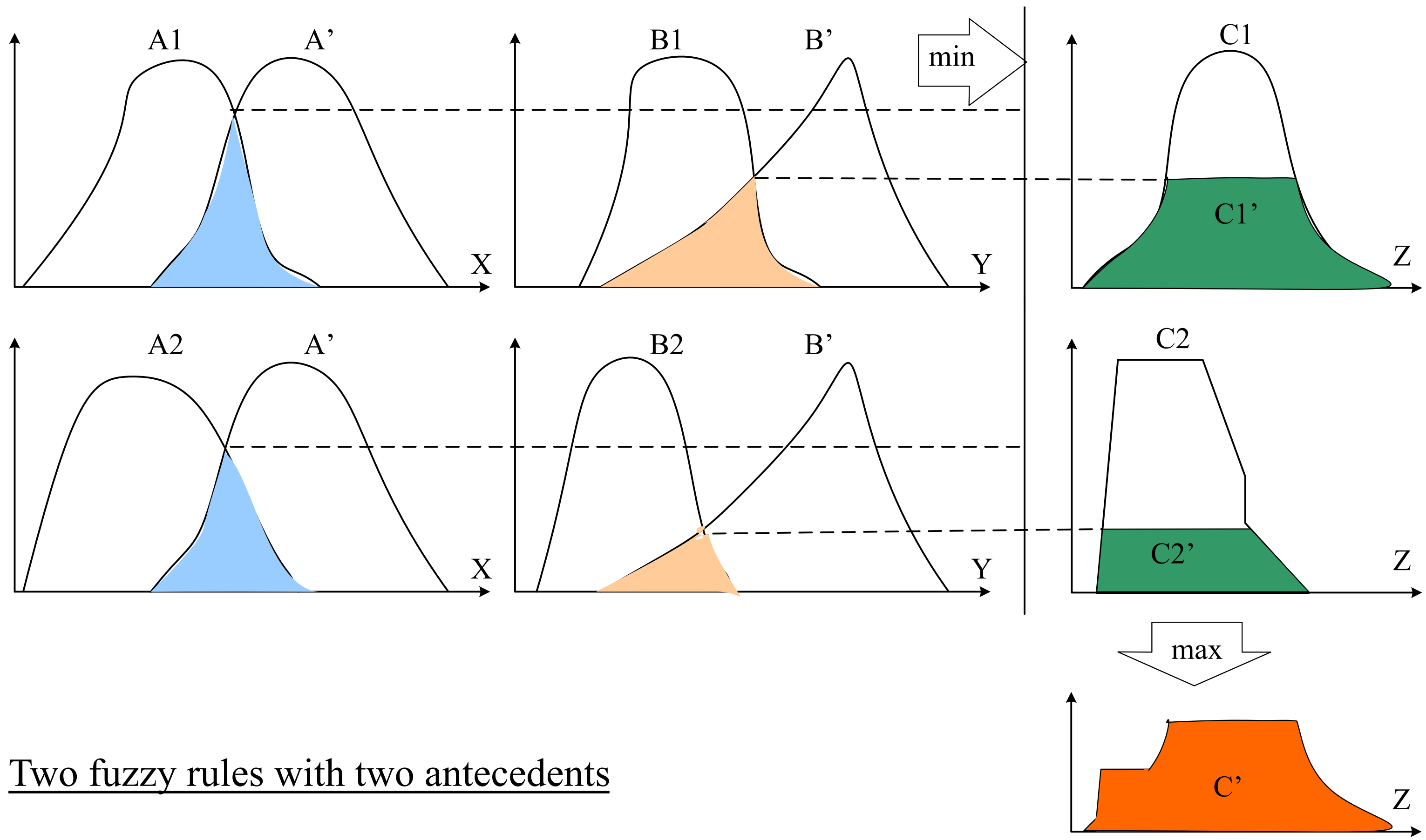
<u>One fuzzy rule with two antecedents</u>

"if x is A and y is B then z is C"



w_1, w_2 → degrees of membership to the respective rules.

Max-Min Fuzzy Reasoning



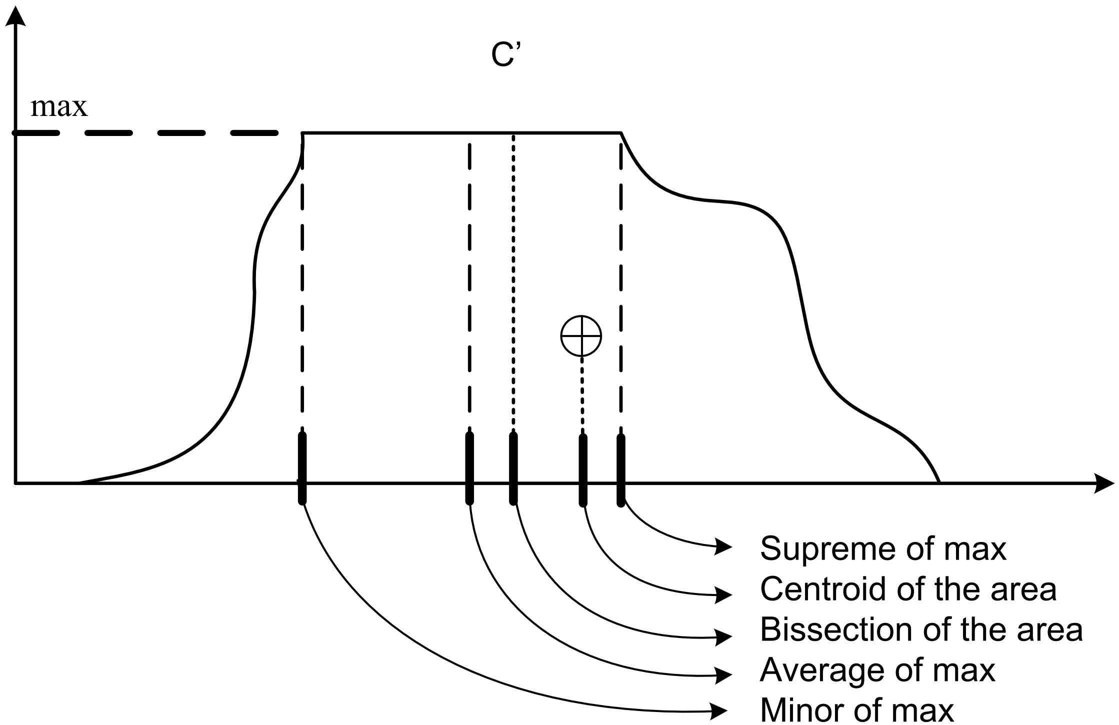
Two fuzzy rules with two antecedents

"If x is A1 and y is B1 then z is C1"
"If x is A2 and y is B2 then z is C2"

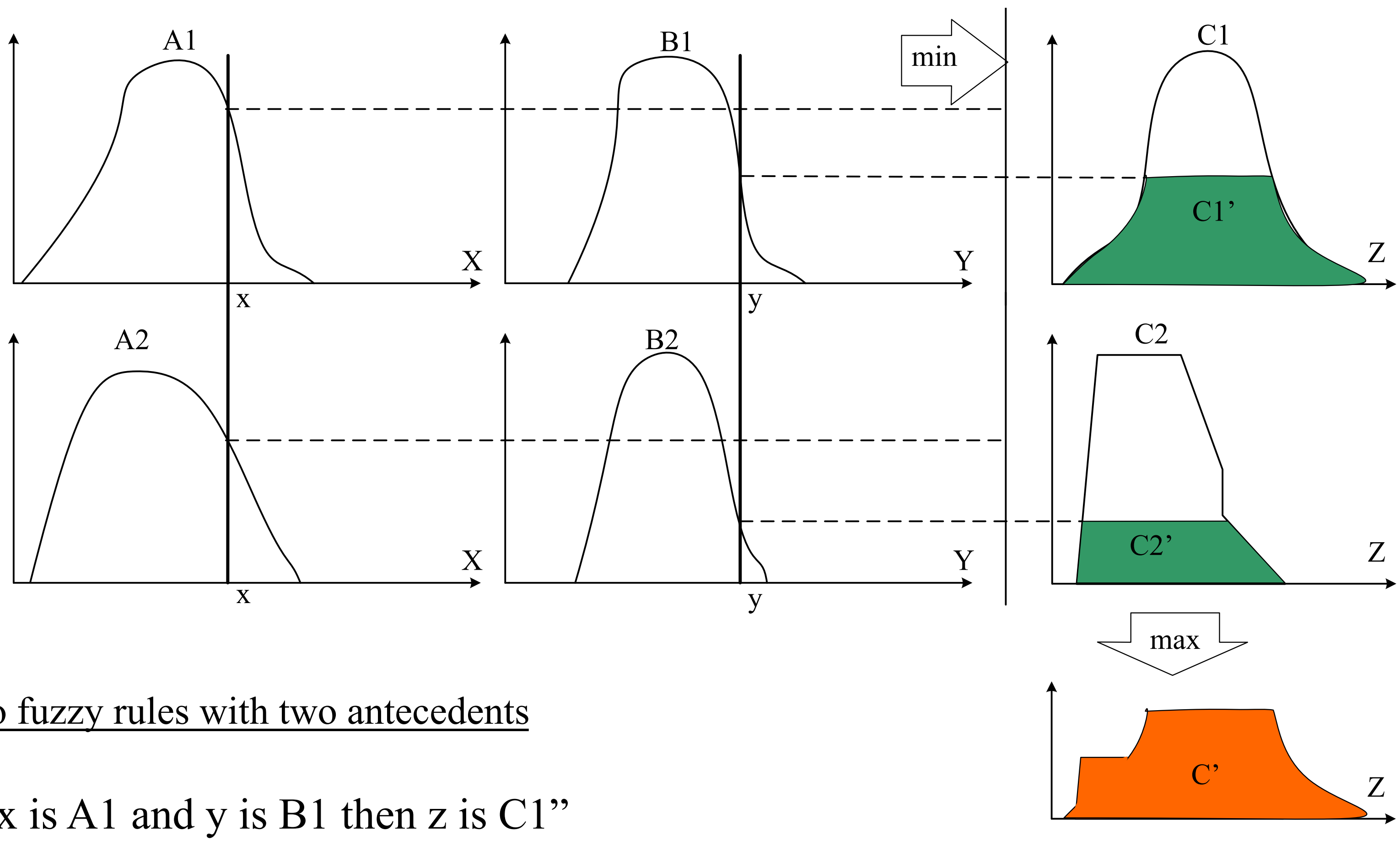
Result: C'

Defuzzyfication Schemes

Associate a numeric value
→ output of the fuzzy inference machine



Fuzzy Inference with exact A' ("crisp")
- Mamdani Model



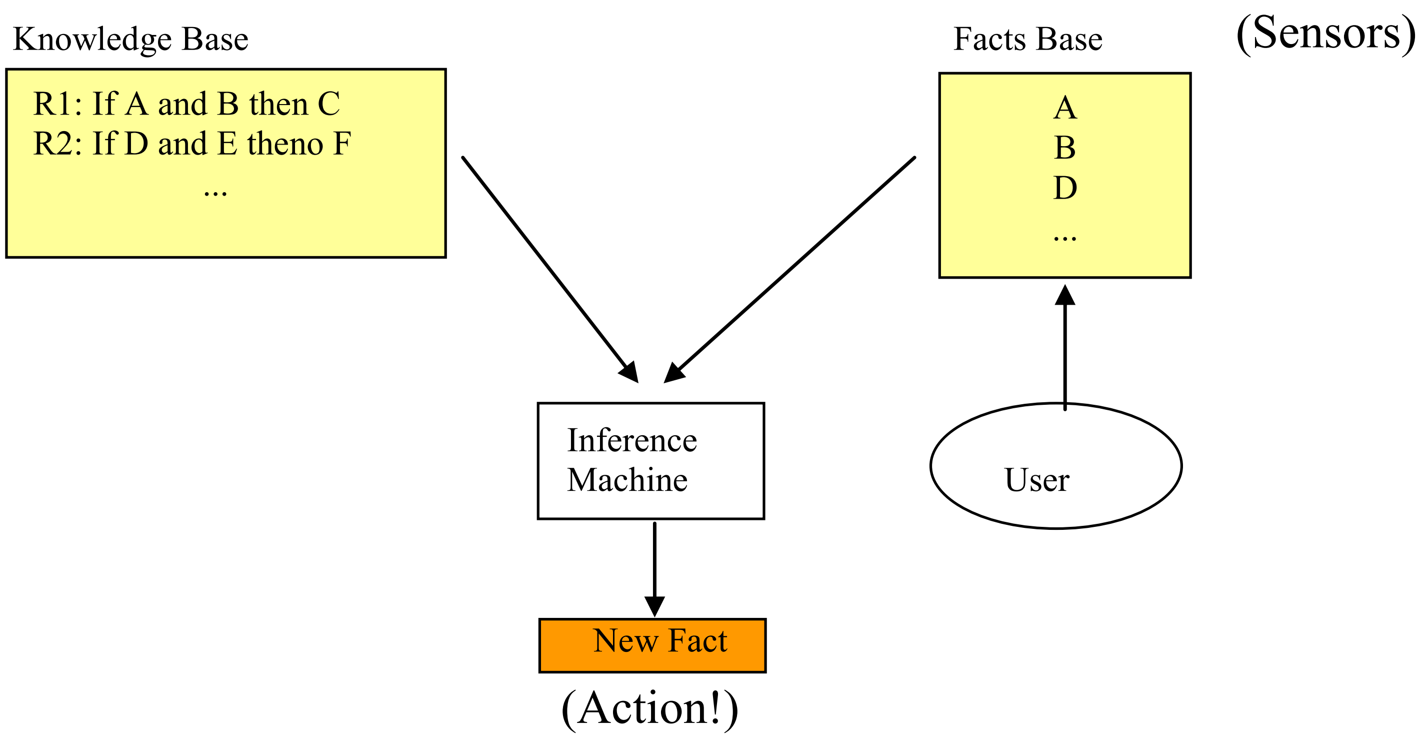
Two fuzzy rules with two antecedents

"if x is A1 and y is B1 then z is C1"
"if x is A2 and y is B2 then z is C2"

Result: C'

Fuzzy Inference Systems

• Fuzzy systems are knowledge-based systems (like Expert Systems).



Fuzzy Inference Machine

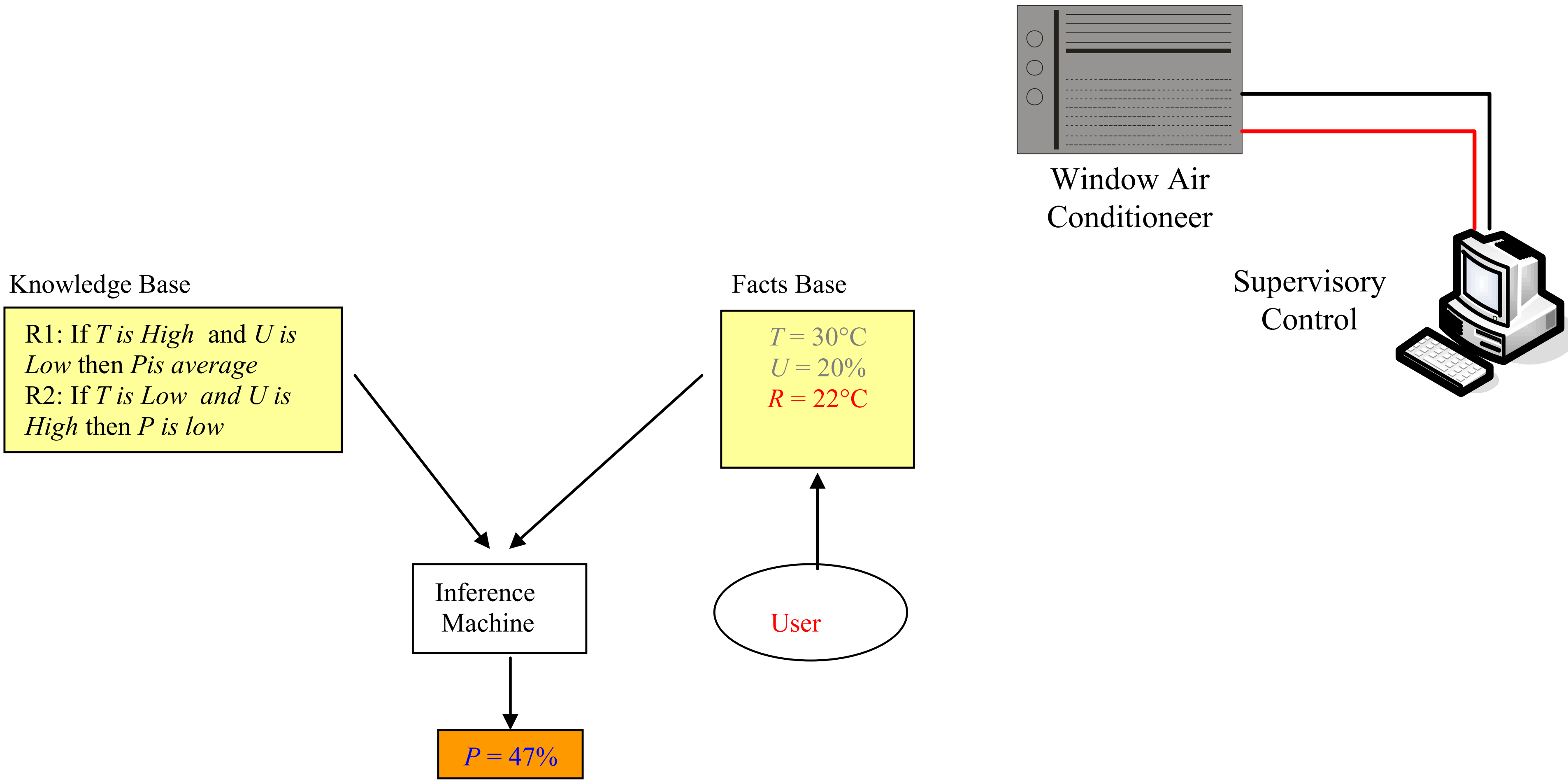
The fuzzy inference machine follows these steps to obtain the Inference Result given a set of facts:

1. facts with premises (antecedents)
2. compatibility degree of each rule
3. belief in each rule
4. aggregation

For the aggregation four methods are popular:

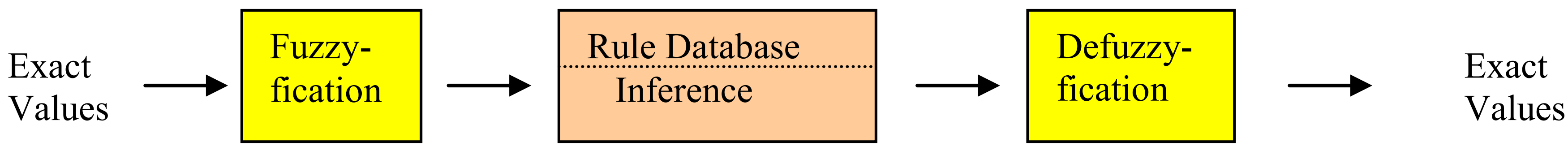
a) Mamdani (Max-Min)
b) Larsen
c) Tsukamoto
d) Takagi-Sugeno

Exemple: Fuzzy Control (revisited)
– Air Conditioneer

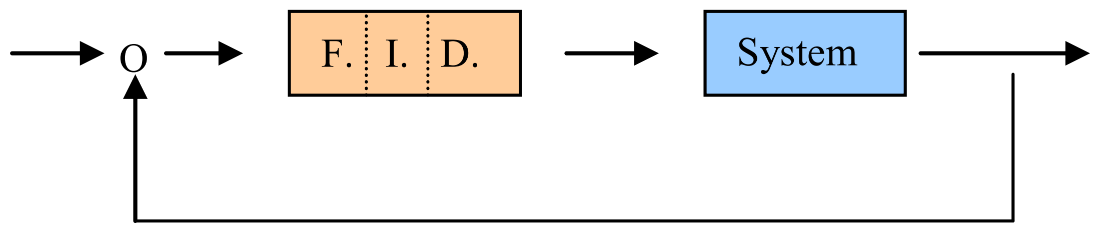


Interface with the Real World

- Fuzzyfication and Defuzzyfication



A feedback controller based on fuzzy logic (Intelligent Controller) would have the following structure, where F. I. D. means: Fuzzyfication, Inference and Defuzzyfication.

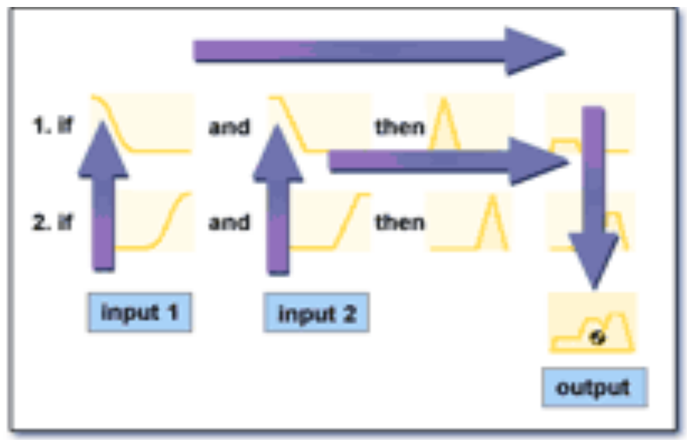


Computational Tools

Xfuzzy http://www.imse.cnm.es/Xfuzzy/download.htm

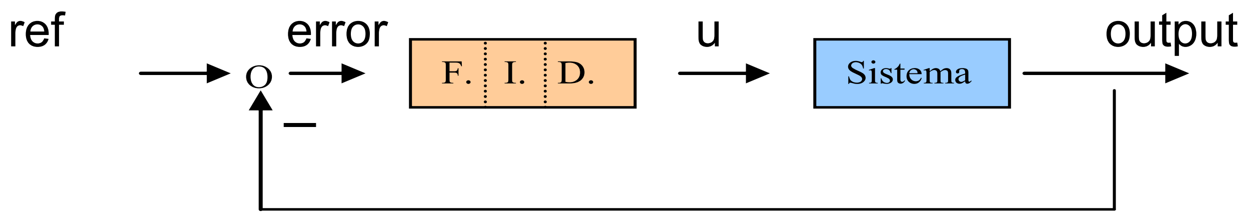
XFuzzy System for Unix developed by the
Instituto de Microelectrónica de Sevilla – Espanha

http://www.mathworks.com MatLab®

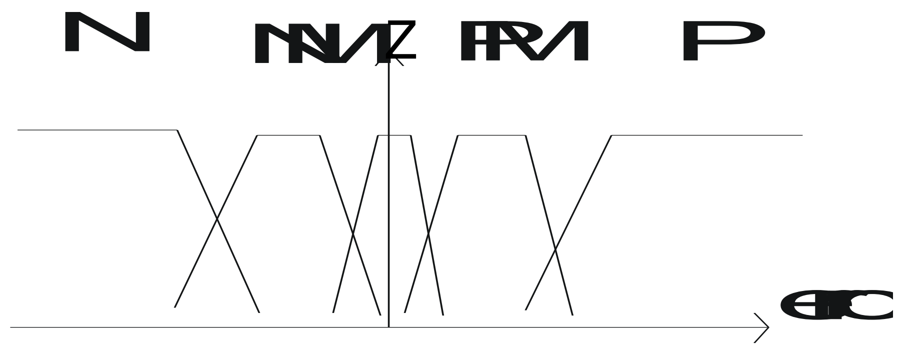


http://www-rocq.inria.fr/scilab/ SciLab

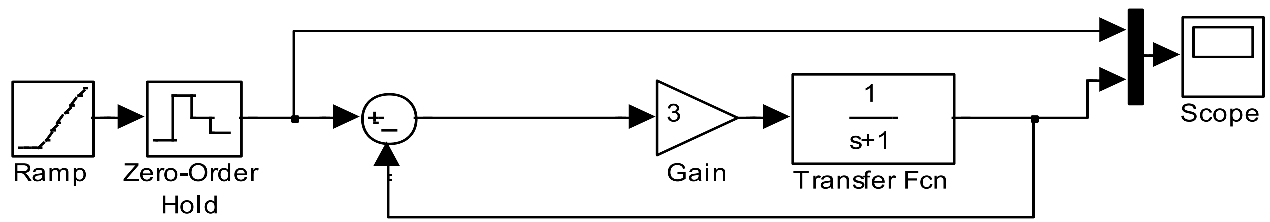
Ex: Fuzzy Proportional Control

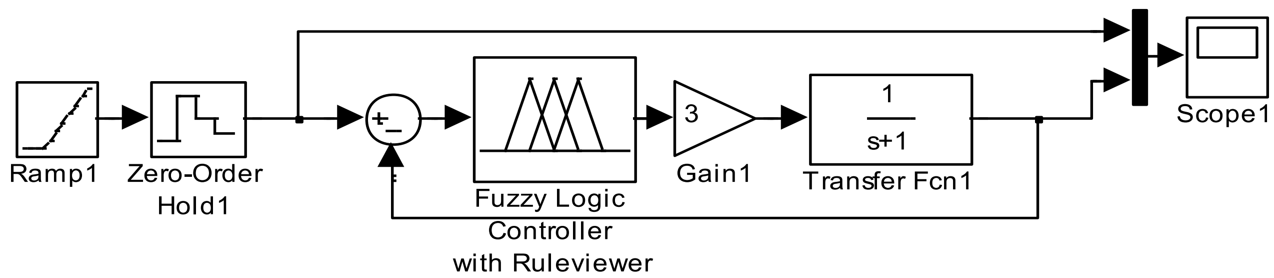


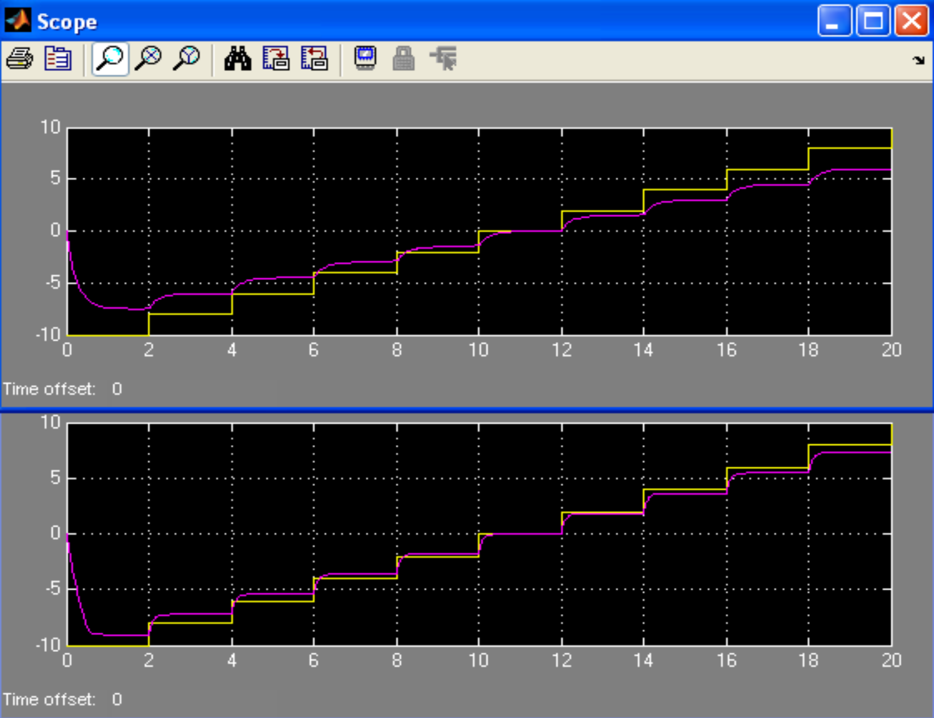
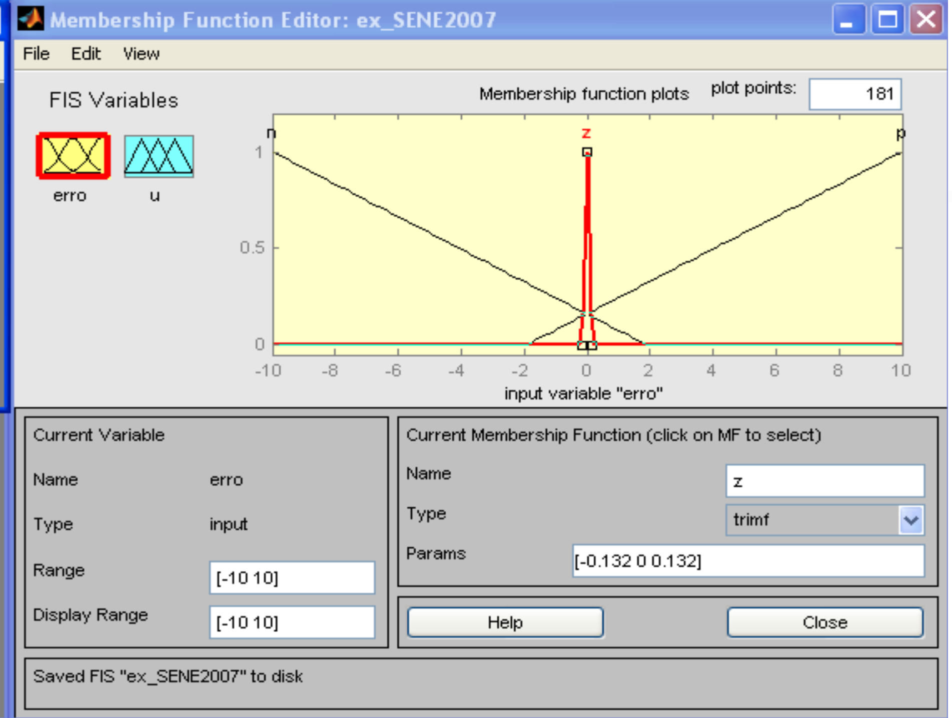
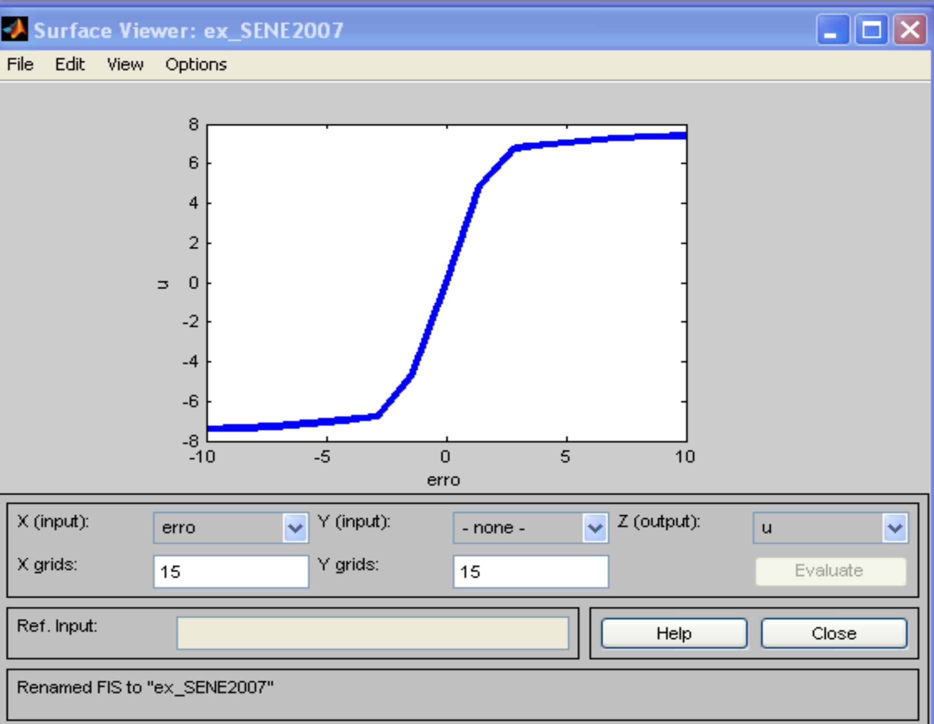
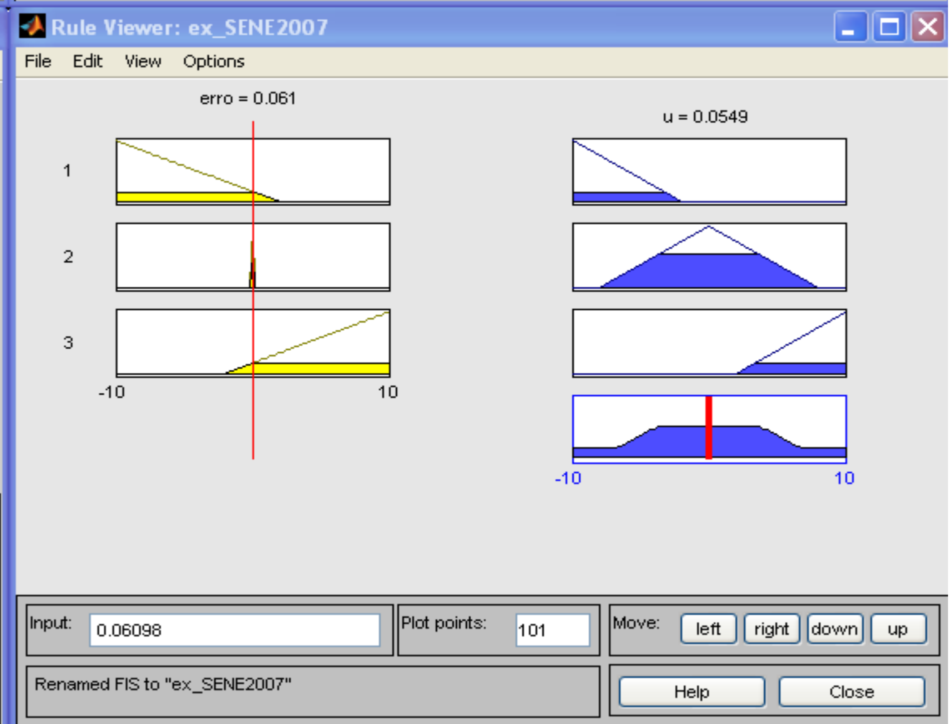
Closed Loop Control



Ex: Fuzzy Proportional Control

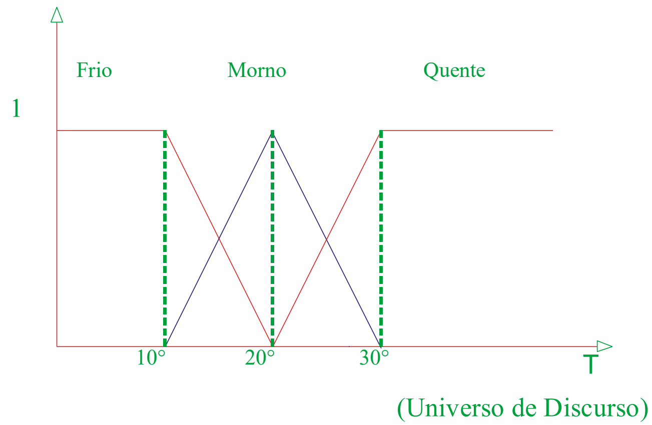


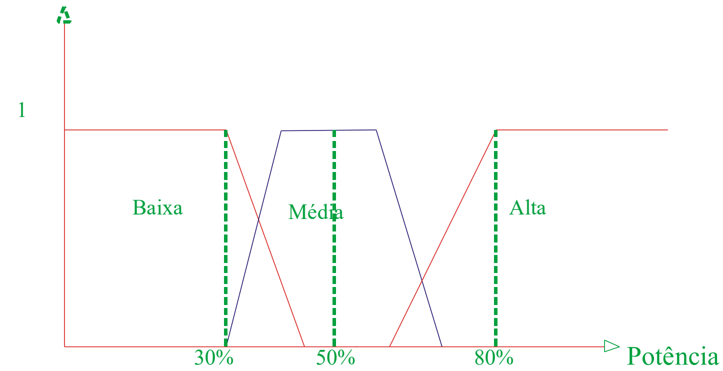


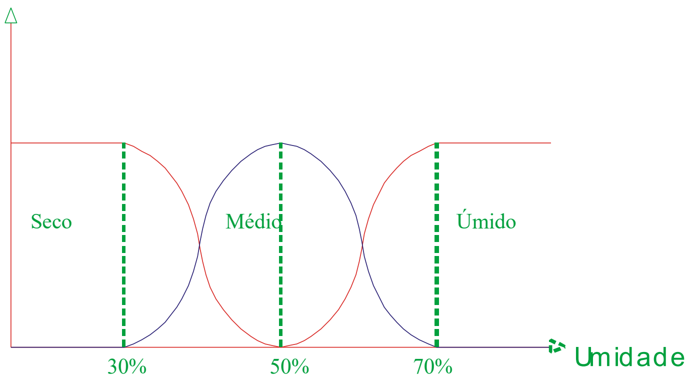

Scope
Time offset: 0
Time offset: 0

Membership Function Editor: ex_SENE2007
File Edit View
FIS Variables
erro
u
Membership function plots
plot points: 181
input variable "erro"
Current Variable
Name erro
Type input
Range [-10 10]
Display Range [-10 10]
Current Membership Function (click on MF to select)
Name z
Type trimf
Params [-0.132 0 0.132]
Help
Close
Saved FIS "ex_SENE2007" to disk

Surface Viewer: ex_SENE2007
File Edit View Options
u
erro
X (input): erro
Y (input): - none -
Z (output): u
X grids: 15
Y grids: 15
Evaluate
Ref. Input:
Help
Close
Renamed FIS to "ex_SENE2007"

Rule Viewer: ex_SENE2007
File Edit View Options
erro = 0.061
u = 0.0549
Input: 0.06098
Plot points: 101
Move: left right down up
Renamed FIS to "ex_SENE2007"
Help
Close

Ex: Multivariable *Fuzzy* Controller

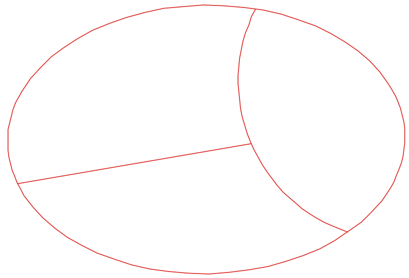
Air conditioning system for an office environment.







Conjunto Fuzzy
Partição do Universo de discurso

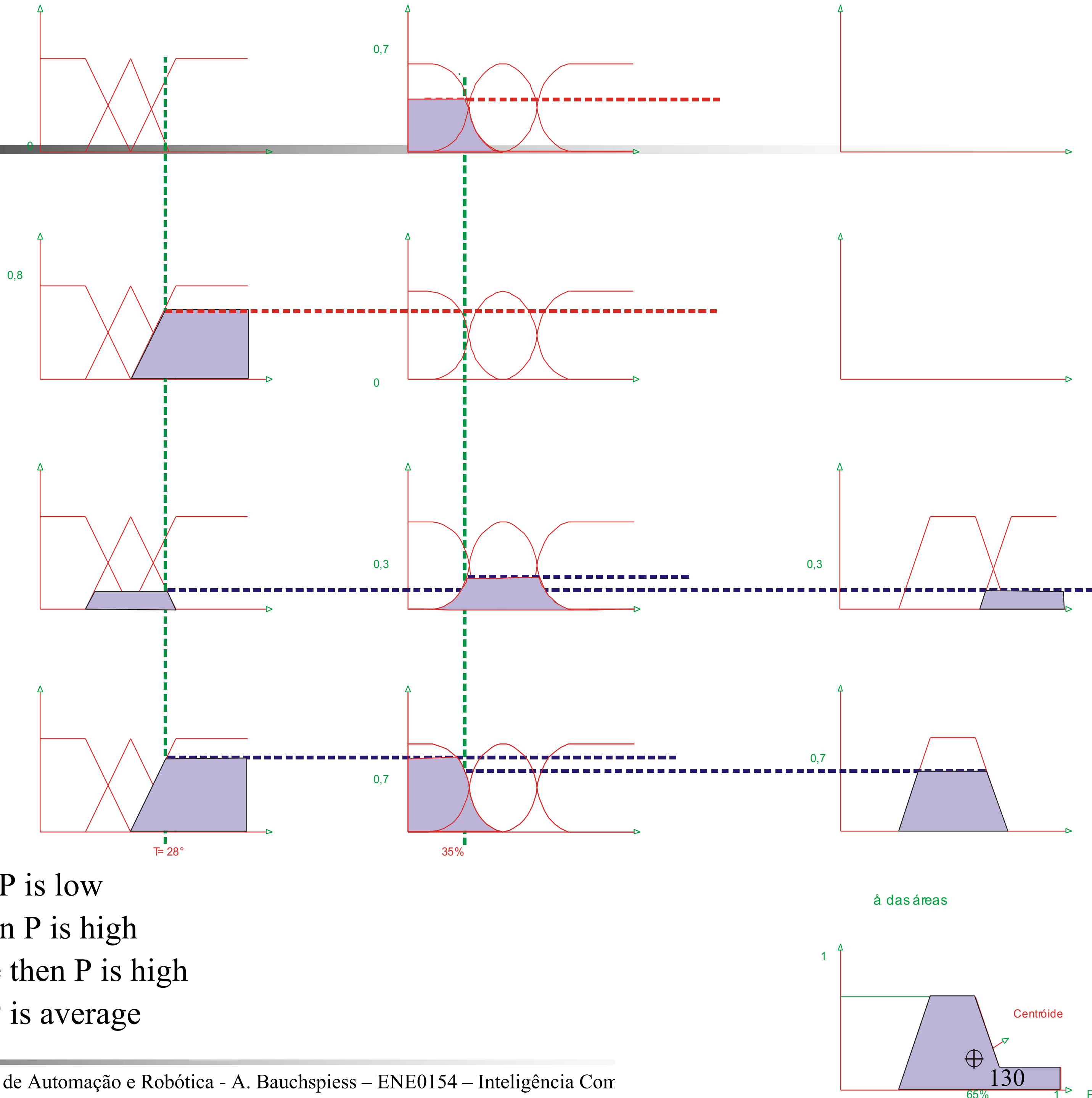


Membership functions used for the temperature control.

Fuzzy Inference:

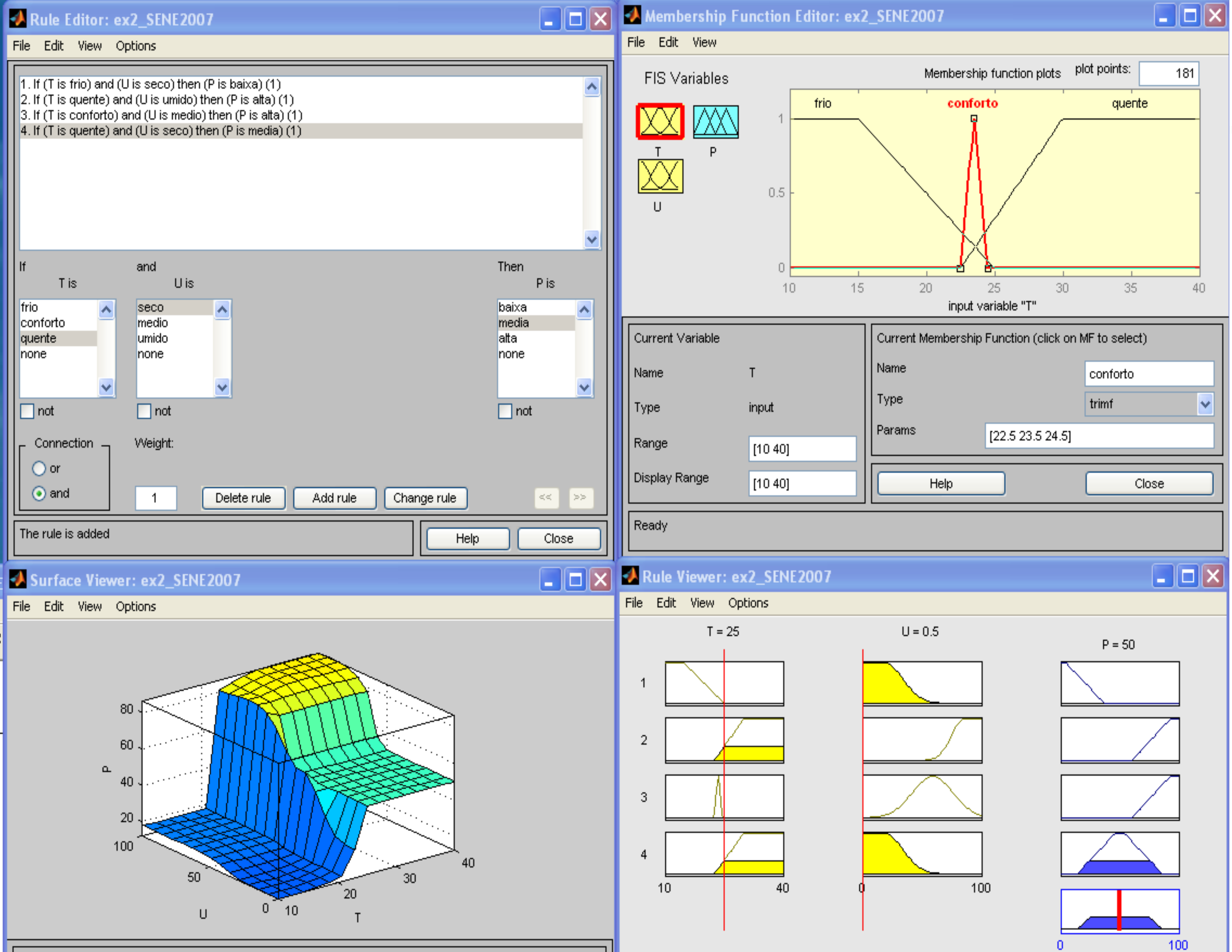
Temperature is 28º C
Relative humidity is 35%

The calculated Power is 65%.



Rule Basis

If T is cold and U is dry then P is low
If T is hot and U is humid then P is high
If T is warm and U is average then P is high
If T is hot and U is dry then P is average

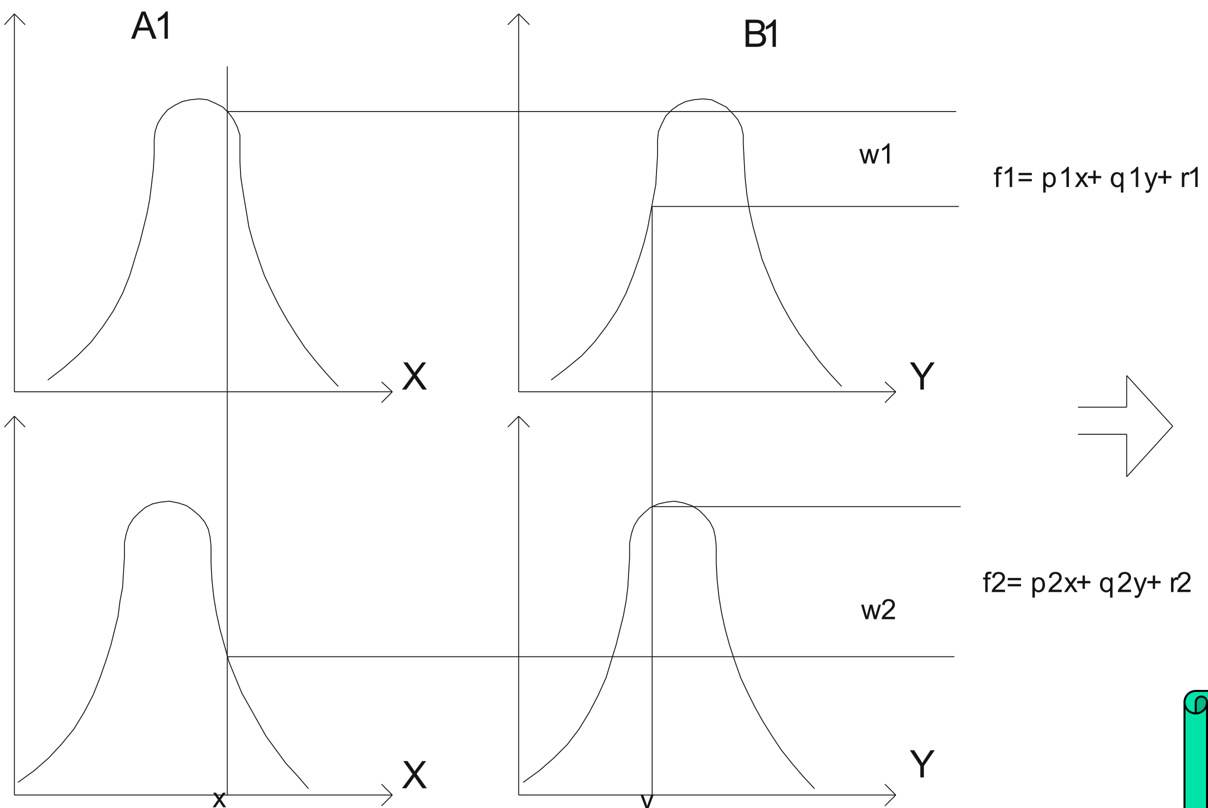

Rule Editor: ex2_SENE2007
File Edit View Options
1. If (T is frio) and (U is seco) then (P is baixa) (1)
2. If (T is quente) and (U is umido) then (P is alta) (1)
3. If (T is conforto) and (U is medio) then (P is alta) (1)
4. If (T is quente) and (U is seco) then (P is media) (1)
If
T is
frio
conforto
quente
none
not
and
U is
seco
medio
umido
none
not
Then
P is
baixa
media
alta
none
not
Connection
or
and
Weight:
1
Delete rule
Add rule
Change rule
The rule is added
Help
Close
Membership Function Editor: ex2_SENE2007
File Edit View
FIS Variables
T
P
U
Membership function plots
plot points:
181
frio
conforto
quente
input variable "T"
Current Variable
Name T
Type input
Range [10 40]
Display Range [10 40]
Current Membership Function (click on MF to select)
Name conforto
Type trimf
Params [22.5 23.5 24.5]
Help
Close
Ready
Surface Viewer: ex2_SENE2007
File Edit View Options
P
U
T
Rule Viewer: ex2_SENE2007
File Edit View Options
T = 25
U = 0.5
P = 50

1st Order Sugeno Fuzzy Inference System

R_1: IF x is A_1 AND y is B_1 THEN $f_1 = p_1x + q_1y + r_1$
R_2: IF x is A_2 AND y is B_2 THNE $f_2 = p_2x + q_2y + r_2$

Consequent: linear combination of the inputs

(pi,qi,ri) instead of output M.F.



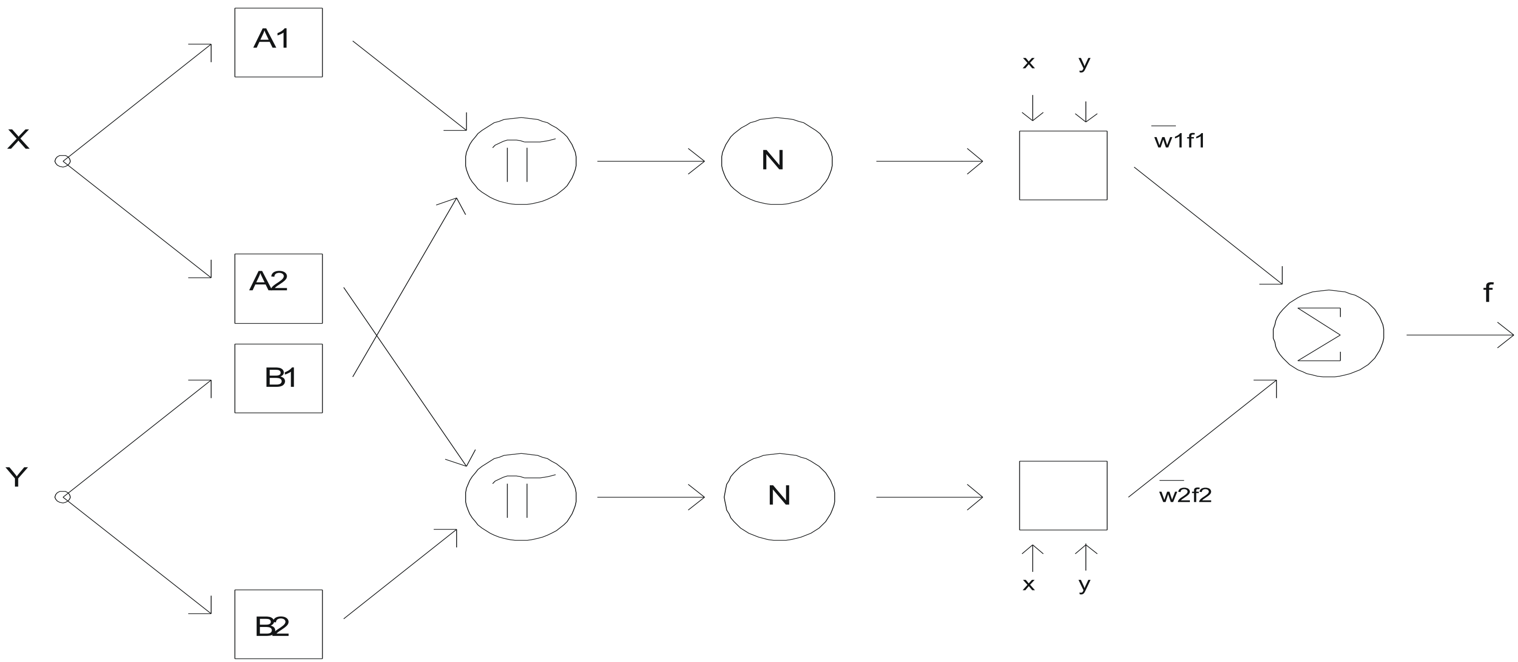
$$f = \frac{w1f1 + w2f2}{w1 + w2} = \overline{w1}f1 + \overline{w2}f2$$

Degrees of compatibility (w1,w2) weigth the rule interpolation

Adaptive Neuro Fuzzy Inference System (ANFIS)

R_1: IF x is A_1 AND y is B_1 THEN $f_1 = p_1x + q_1y + r_1$

R_2: IF x is A_2 AND y is B_2 THNE $f_2 = p_2x + q_2y + r_2$



Adaptive feed-forward network for the Sugeno fuzzy model

ANFIS Layers

Layer 1: Adaptive Nodes

$$o_{1,i} = \mu_{Ai}(x), \text{ for } i = 1, 2.$$
$$o_{1,i} = \mu_{B\,i-2}(y), \text{ for } i = 3, 4.$$

Ai – generalized bell function

$$\mu_{Ai}(x) = \frac{1}{1 + [(x-c_i)^2/(a_i)^2]^{b_i}}$$

$\{a_i, b_i, c_i\}$ set of premise parameters

Layer 2: Fixed Nodes

$$o_{2,i} = \omega_i = \underbrace{\mu_{Ai}(x) \;\; x \;\; \mu_{B\,i}(y)}_{\text{T-Norm}}, \; i = 1, 2$$

Layer 3: Fixed Nodes

$$o_{3,i} = \omega_i\,(\text{médio}) = \frac{\omega_i}{\omega_1 + \omega_2}, \qquad i = 1,2$$

Layer 4: Adaptive Nodes

$$o_{4,i} = \frac{\omega_i}{\omega_1 + \omega_2} . f_i = \frac{\omega_i}{\omega_1 + \omega_2} . (p_i x + qiy + r_i)\,, \qquad i = 1,2$$

$\{ p_{i,} \, qi, \, r_i\}$ set of the consequent parameters

Layer 5: Fixed Node

$$o_{5,i} = \Sigma_i \frac{\omega_i}{\omega_1 + \omega_2} . f_i = \frac{\Sigma_i \, \omega_i f_i}{\Sigma_i \, \omega_i}$$

ANFIS Hybrid Learning

1 – Fix the premisse parameters

$\Rightarrow$ Output is a linear combination of the consequence parameters

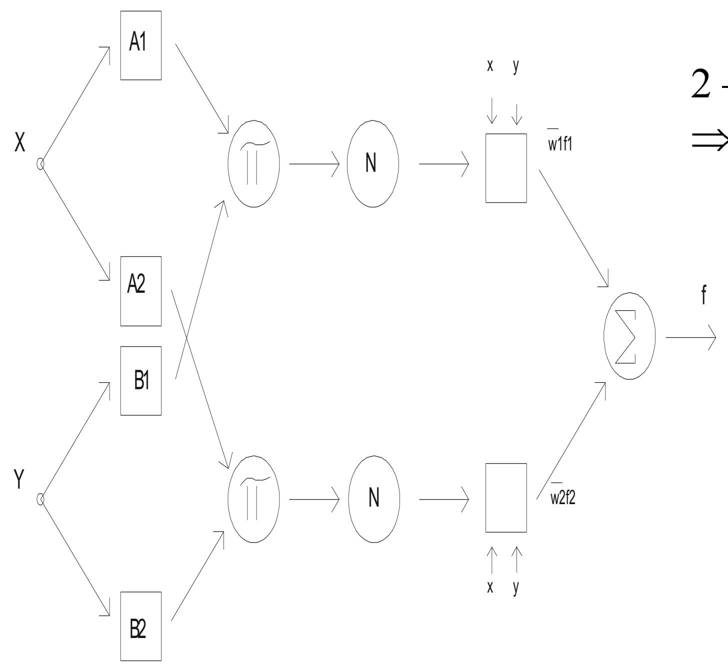
$$f = \frac{\omega_1}{\omega_1 + \omega_2} . f_1 + \frac{\omega_2}{\omega_1 + \omega_2} . f_2$$

$$= \frac{1}{\omega_1 + \omega_2} . (\omega_1 x p_1 + \omega_1 y q_1 + \omega_1 r_1 + \omega_2 x p_2 + \omega_2 y q_2 + \omega_2 r_2)$$

$\Rightarrow$Identify consequence paraters using Least Mean Squares method.

2 – Backpropagtion of the errror signals toadapt the premisse parameters

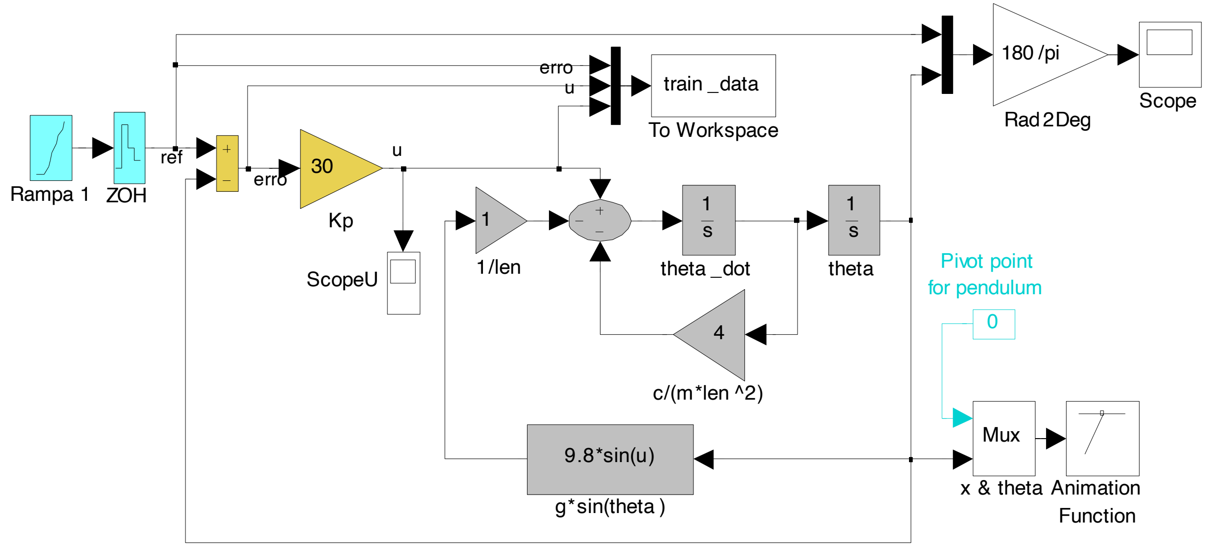
$\Rightarrow$ Gradient descent method.

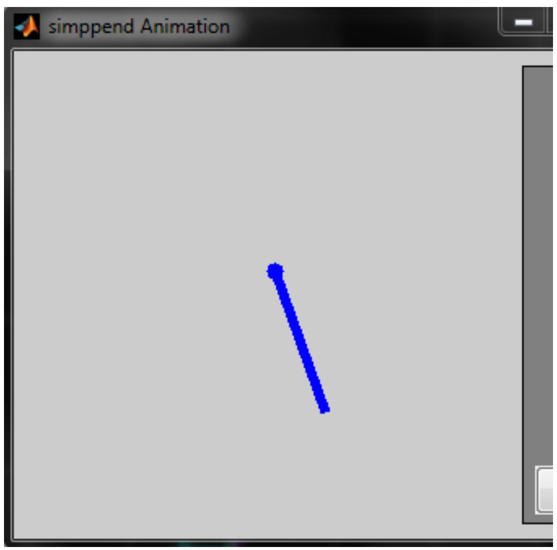


	Forward Step	Backward Step
Premisse Parameters	Fixed	Gradiente descent
Consequence Parameters	LMS Estimate	Fixed
Signals	Output nodes	Error Signals

ANFIS

Adaptive Neuro Fuzzy Inference System



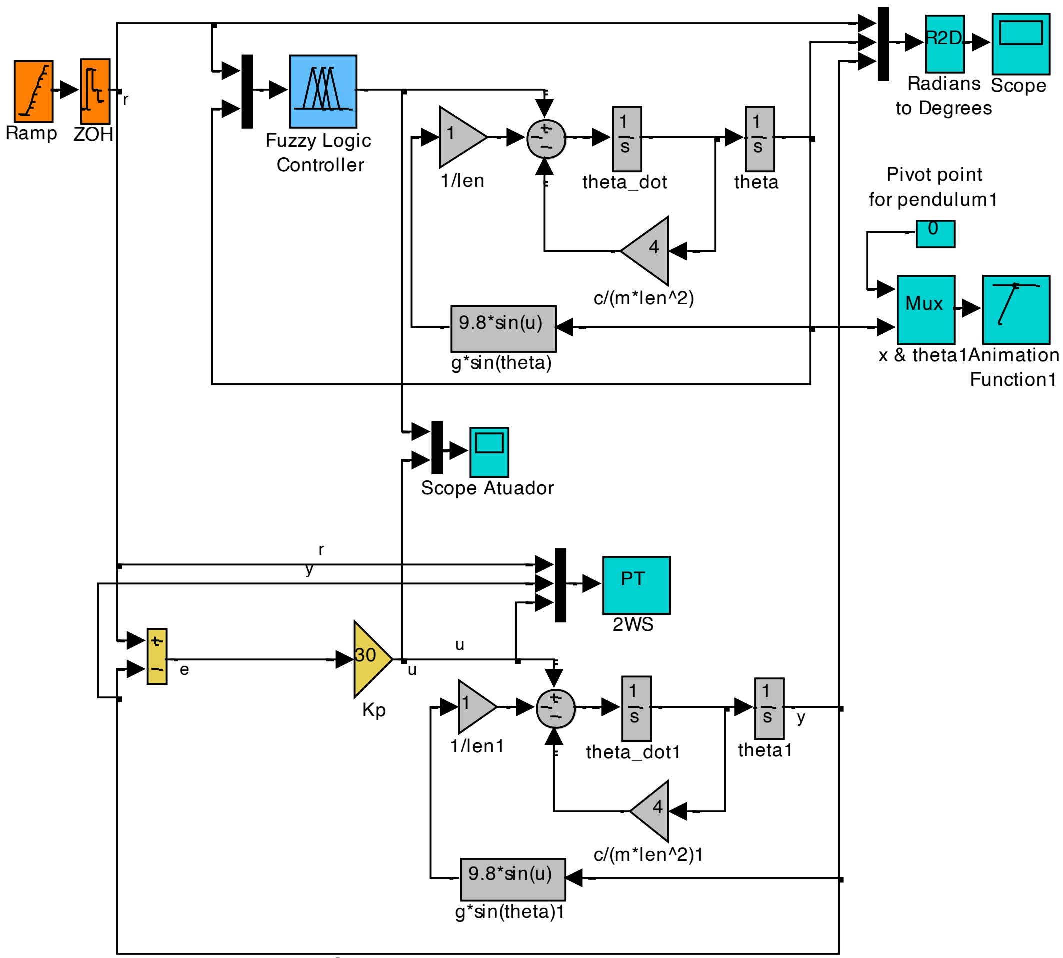


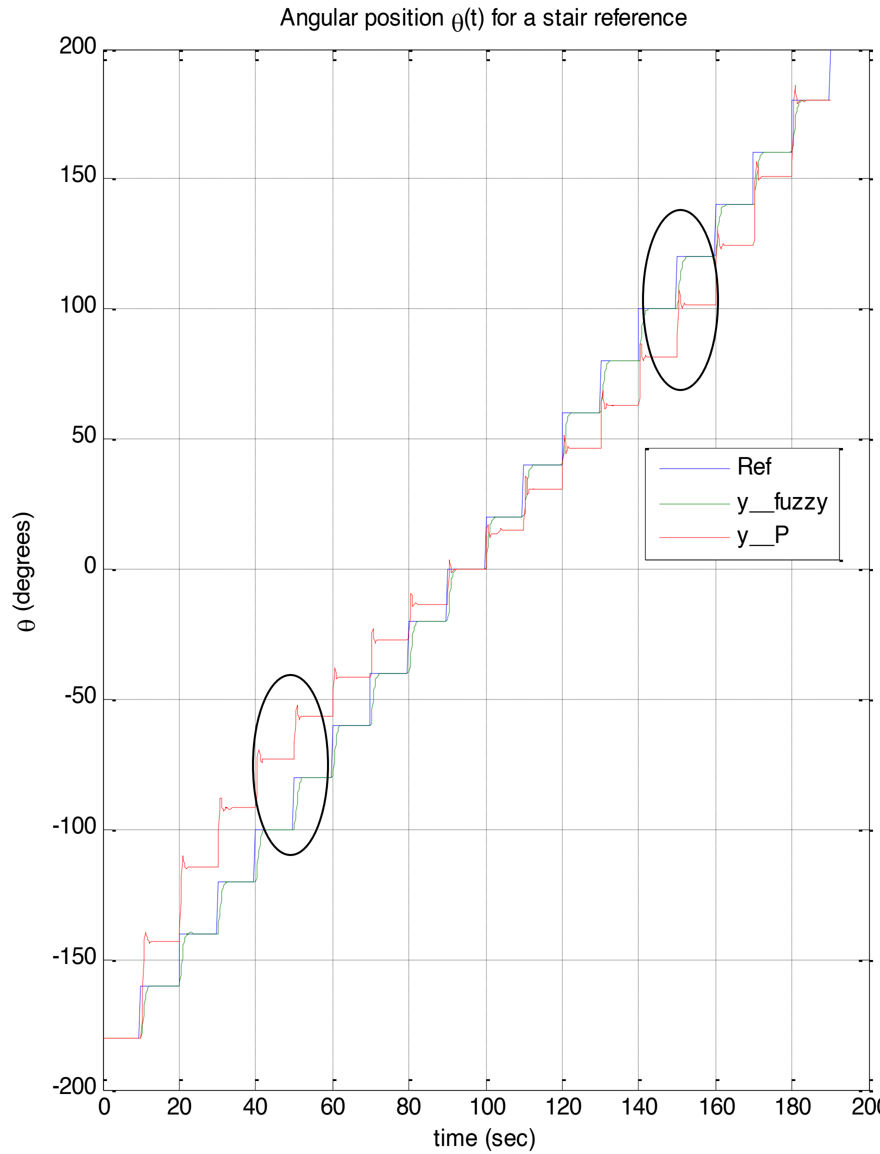
$$\ddot{\theta} = -\frac{c}{ml^2}\dot{\theta} - \frac{g\sin(\theta)}{l} + u$$

$$\ddot{\theta} + 4\dot{\theta} + 9.8\sin(\theta) = u$$

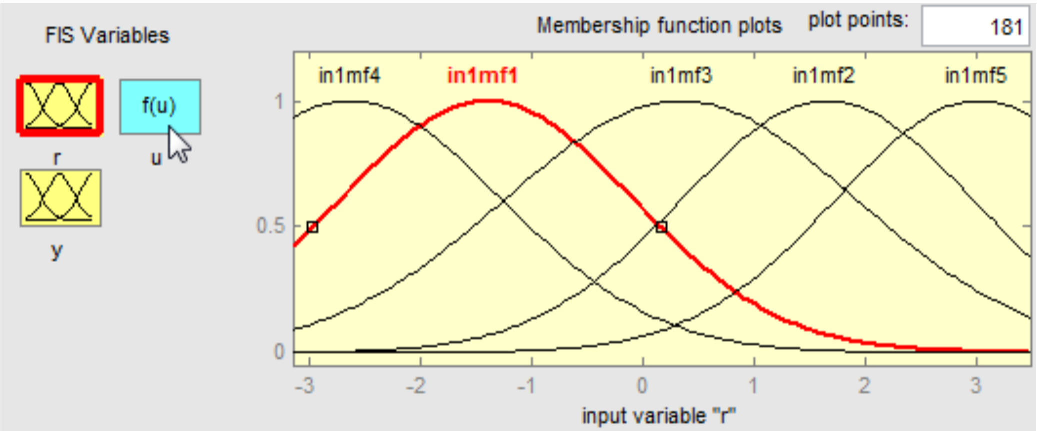
$$Control\ law \qquad u = 30(r - \theta)$$

ANFIS x P-Controller

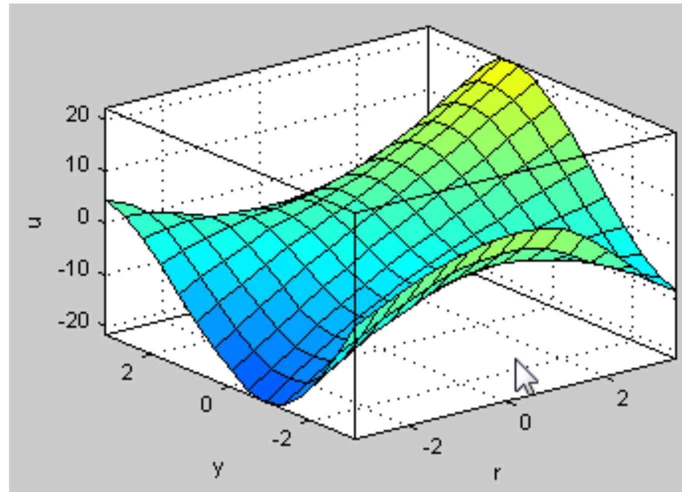




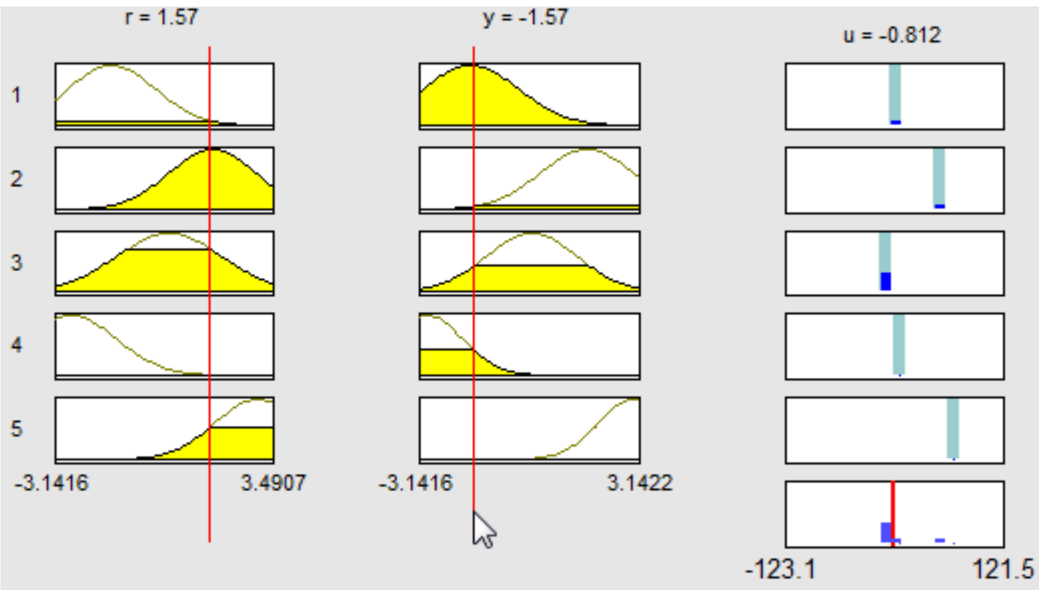
ANFIS Controller



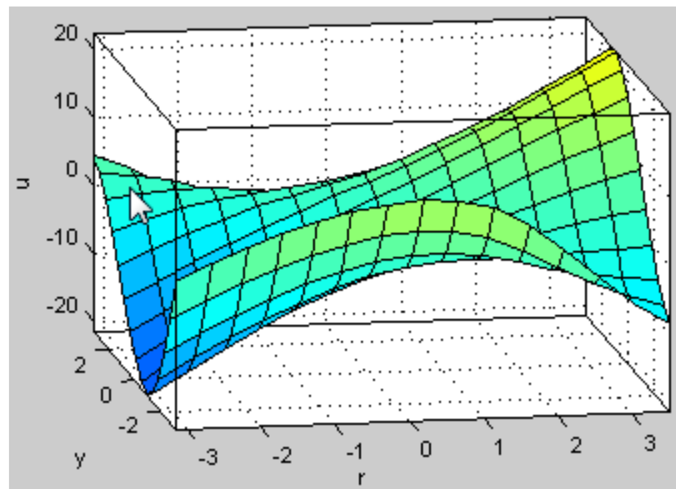
Membership functions of r (reference)



View of the ***non-linear*** control surface

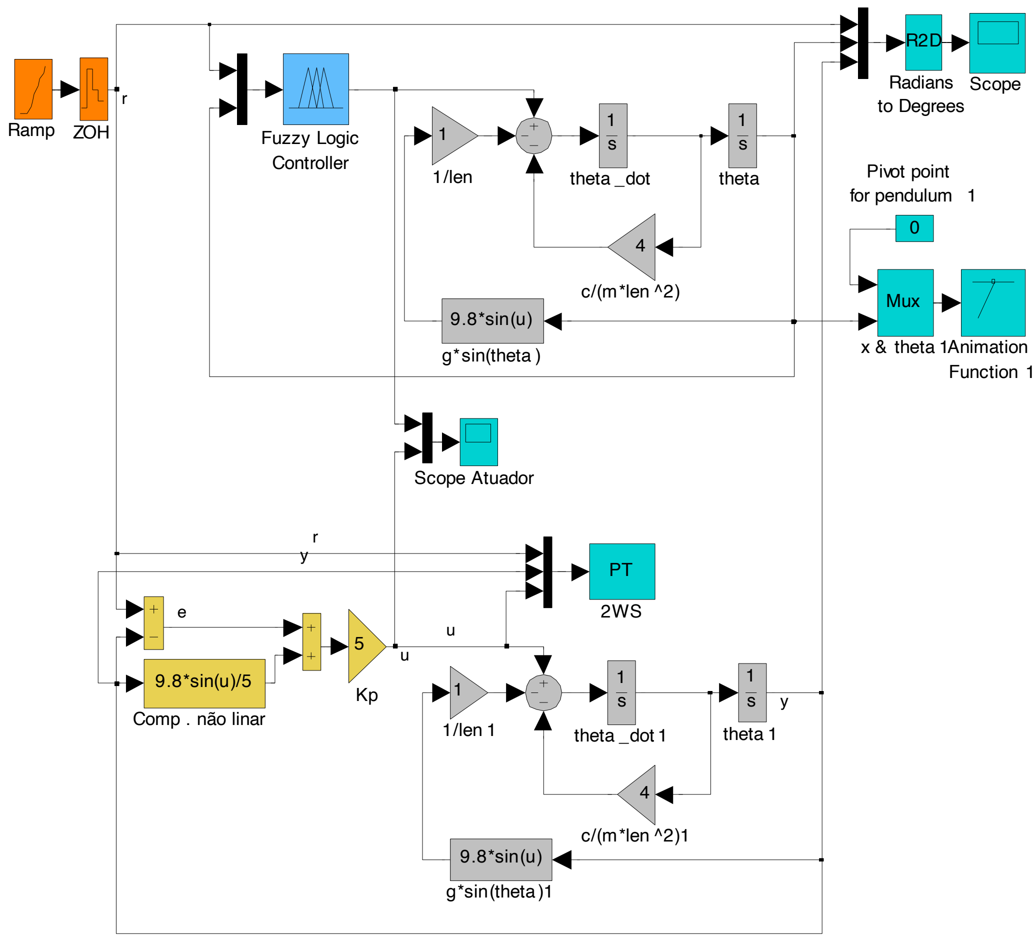


Rule Viewer



Another view of the ***non-linear*** control surface

ANFIS – Who is the Expert?



Expert Knowlege

Linear behavior if the control signal, u, can cancel the non-linear dynamics of the process.

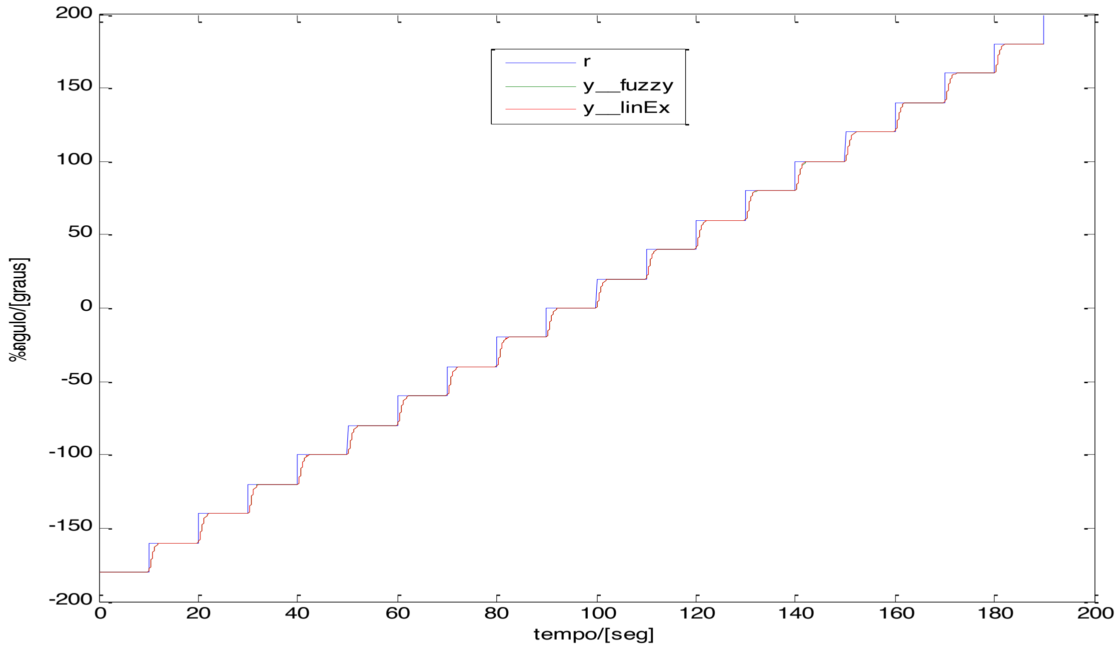
→ "Exact Linearization" (E.L.)
(not only operating point)

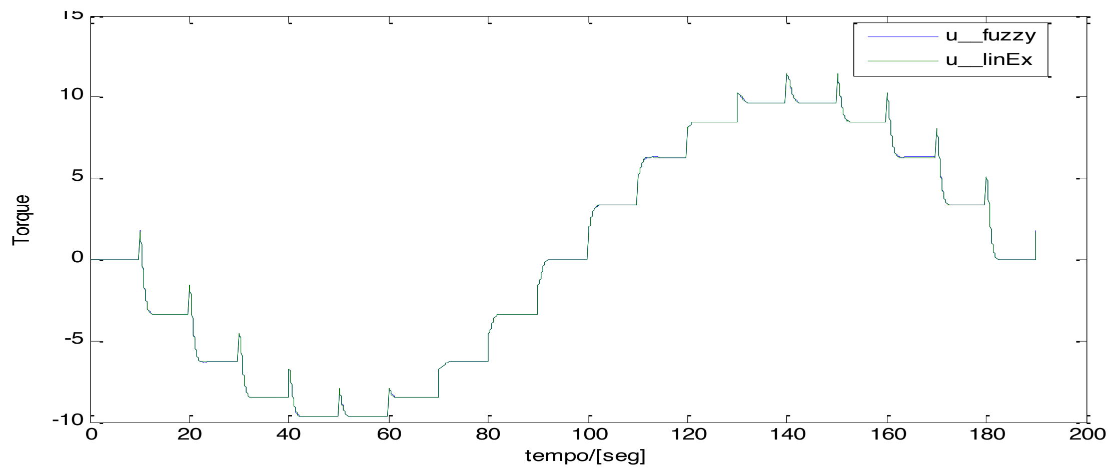
$$\ddot{\theta} + 4\dot{\theta} + 9.8\sin(\theta) = u$$

$$P-Control\ law \quad u = K_p(r-\theta)$$

$$E.L.-Control\ law \quad u = K_p(r-\theta-9.8\sin(\theta))$$

ANFIS – Controller





Rules Trained by an ANN!

-You can explain and add new rules (*fuzzy*)

-You can train with real data (ANN)

-Drawback *more parameters...*

Part 4 –Applications

- Nondestructive inspection of structures
- Visual inspection of transmission lines
- Liquid Level Process
- Water treatment plant
- Automatic Car Guiding
- Consumer Electronics
- Path Planning
- Building Automation – (Ambient Intelligence)

Applications:
Nondestructive inspection of structures

DAMAGE DETECTION USING AN HIBRID FORMULATION BETWEEN CHANGES IN CURVATURE MODE SHAPES AND NEURAL NETWORK.

Miguel Genovese, Adolfo Bauchspiess, José L.V. de Brito,Graciela N. Doz

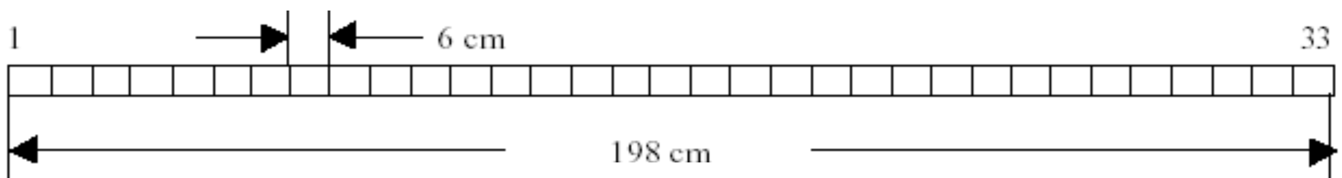
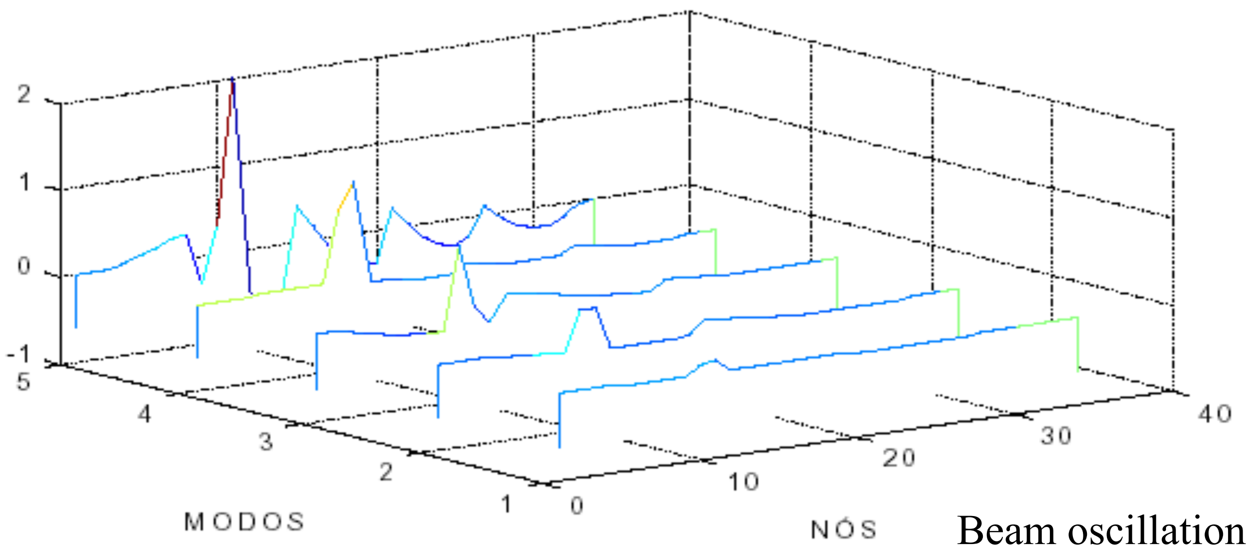


Figura 2: Discretization of the test beam

- strain gauges
- Hammer hit
- Signal acquisition

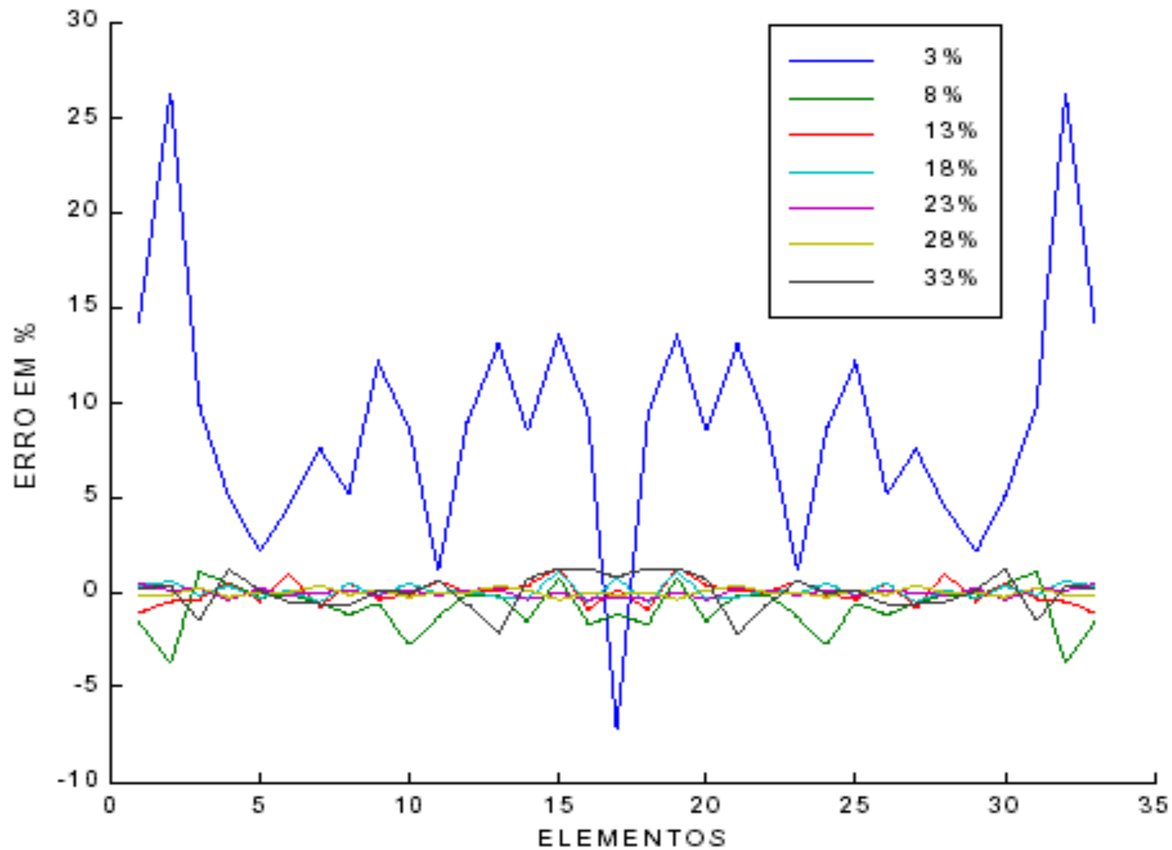
Applications:
Nondestructive inspection of structures



Beam oscillation frequencies (Hz): with and without damage

Harmonics	Beam without damage	Beam with 20% moment of inertia reduction at element 10
First	67.76	67.48
Second	184.22	182.72
Third	354.01	352.62
Fourth	570.26	569.59
Fifth	825.83	821.93

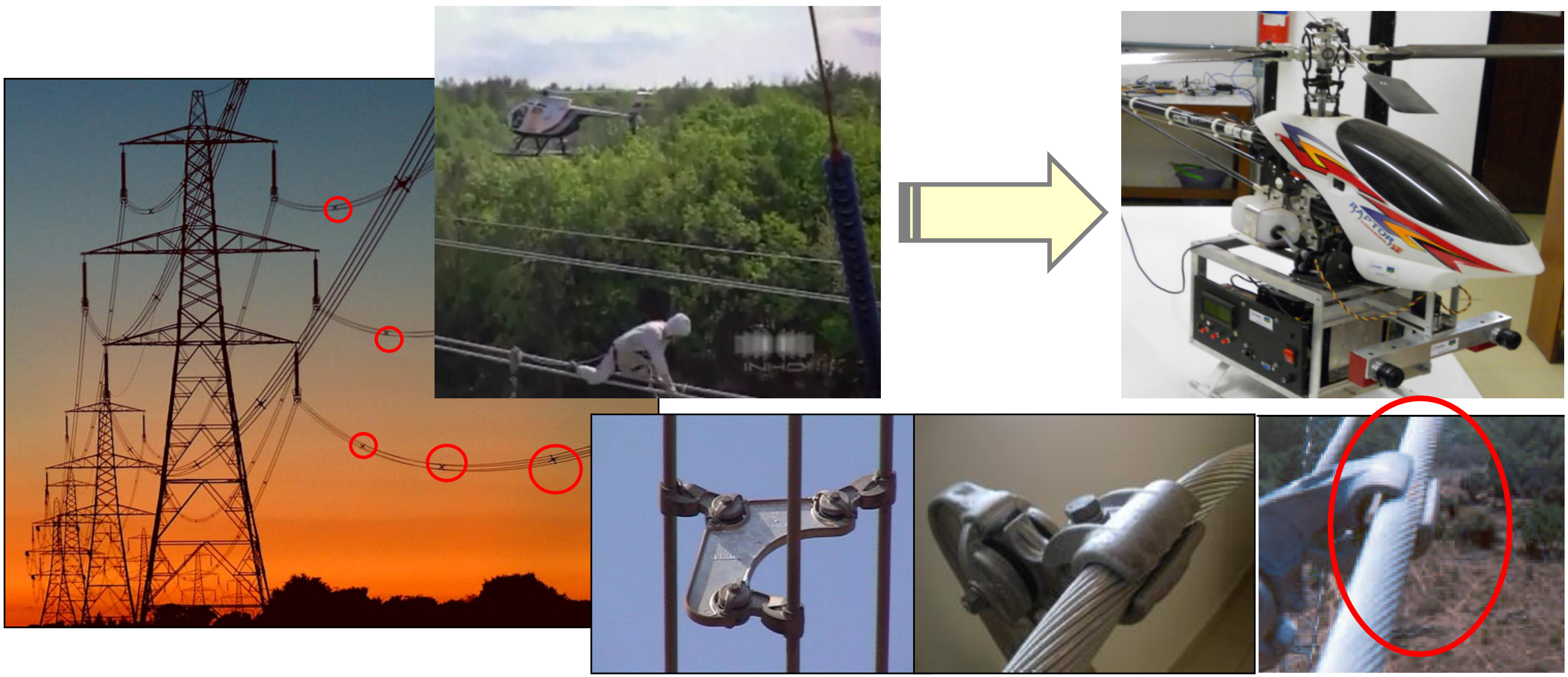
Applications: Nondestructive inspection of structures



ANN error at the training data

Inspection of Transmission Lines

- Autonomous system - visual inspection of electricity transmission lines
- Detection of flaws in the gripper of the line spacers



Need Maintenance!!

Inspection of Transmission Lines

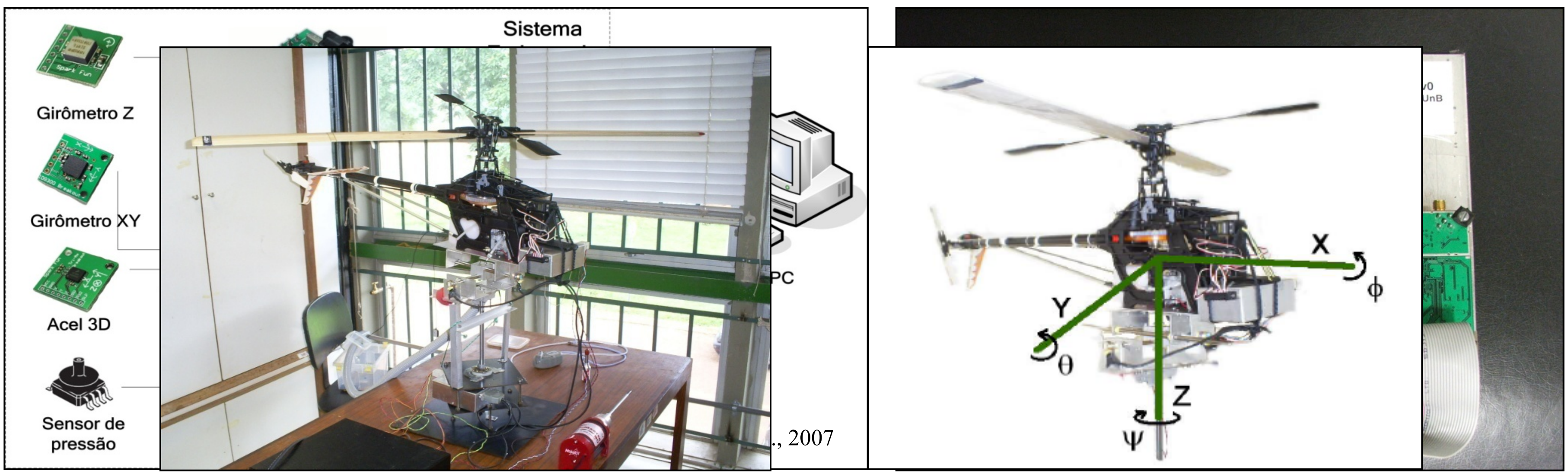
- Traditional inspection of transmission lines:
  - Aerial survey using a helicopter
  - Staff onshore
- Costly and expensive

Inspection of Transmission Lines

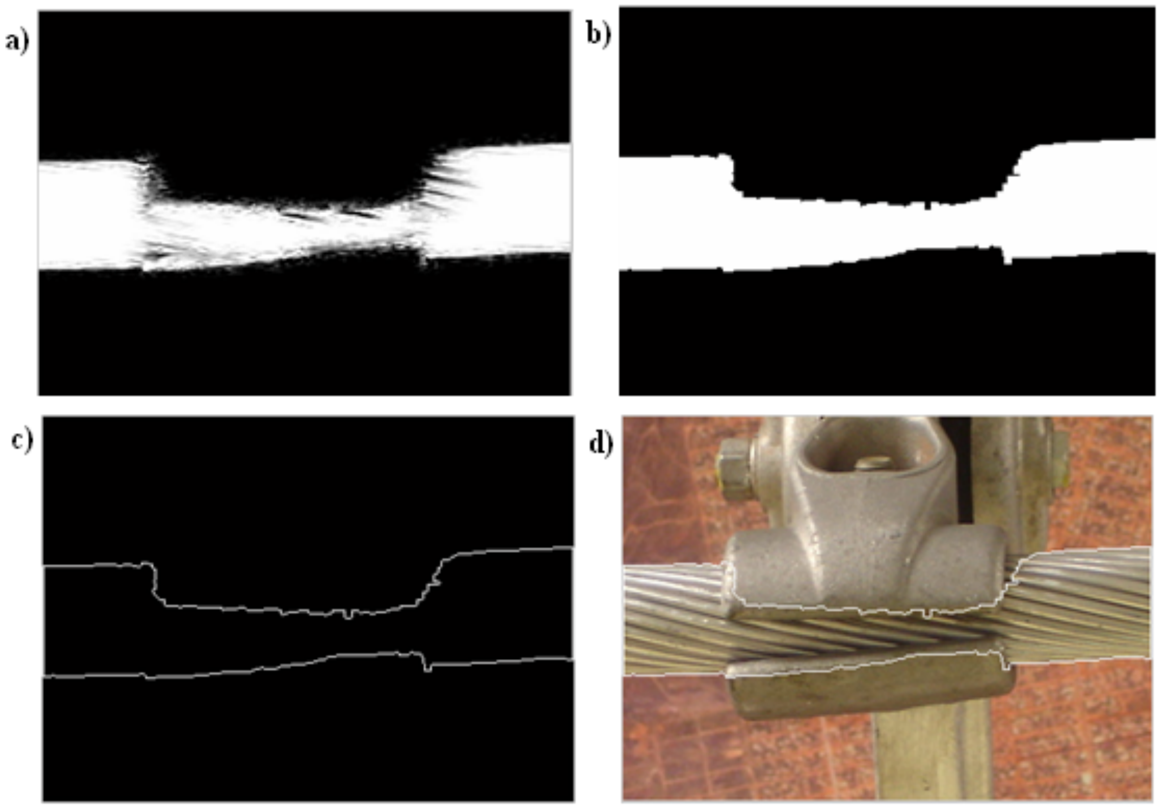
UAV – LARA/UnB

Inspection of Transmission Lines

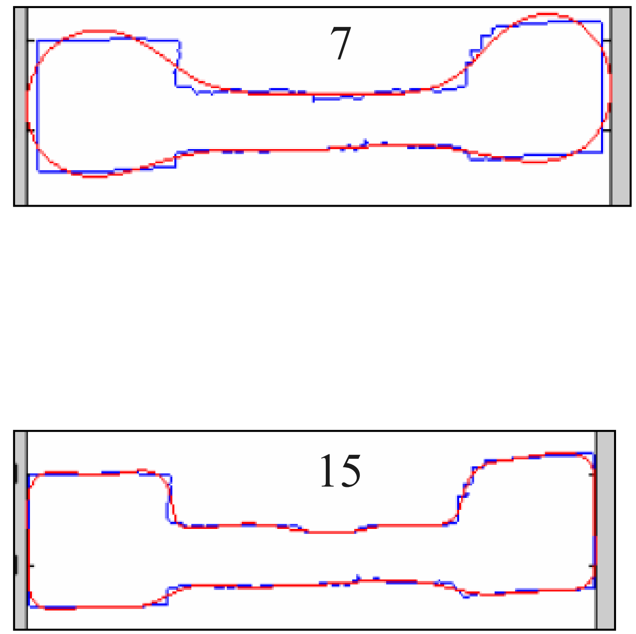
- Adaptation of Unmanned Aerial Vehicles (UAVs)
- Research project UNB / ANEEL - Expansion
  - Development of an UAV to aid inspecting transmission lines



Gripped cable contour:
FFT coefficients of directional chains

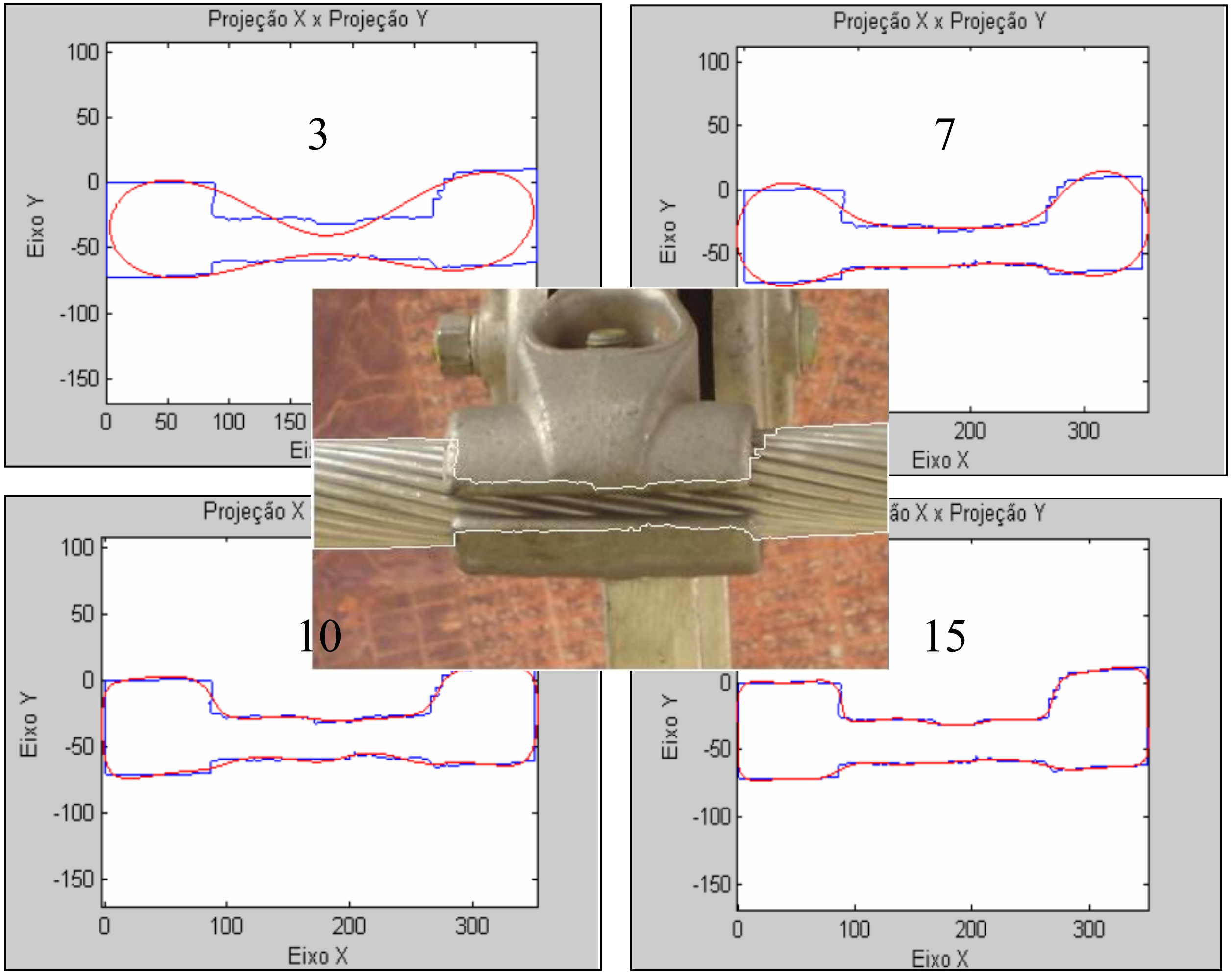


a) Gabor – b) Closing – c) Border – d)Image

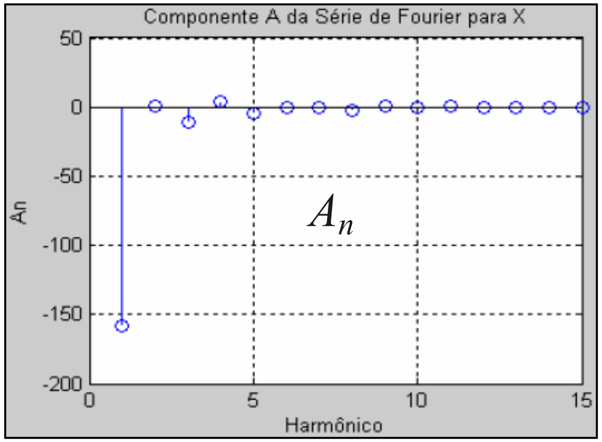


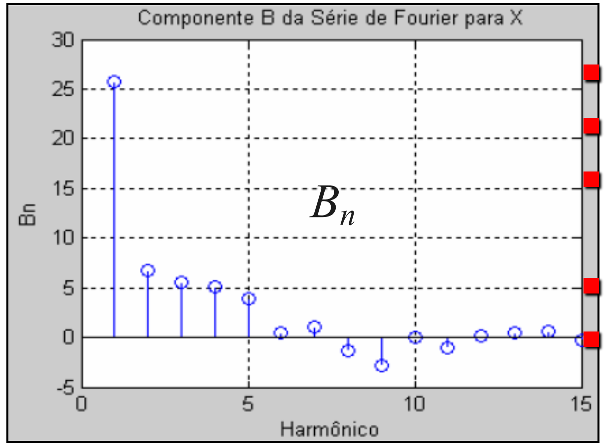
Reconstruction -
7 and 15 Harmonics

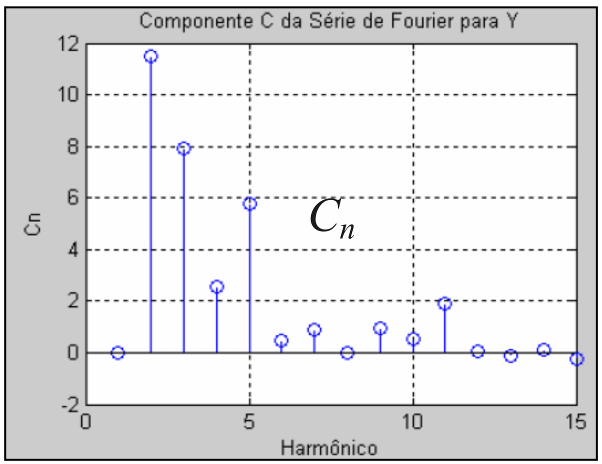
Results – Representation of the contour

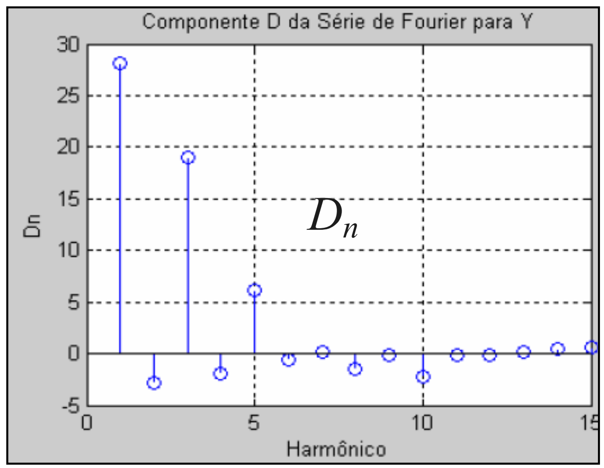


Gripped cable contour:
FFT coefficients of directional chains

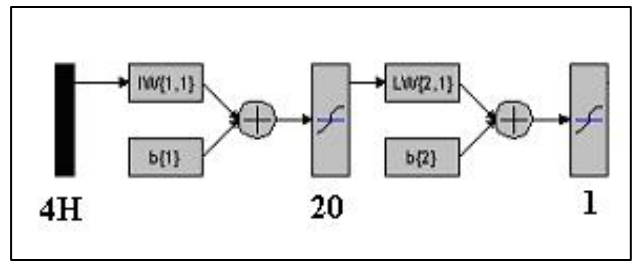








- ANN – "Need Maintenance" classification
- Training, Test, Validation 80, 25, 25 images

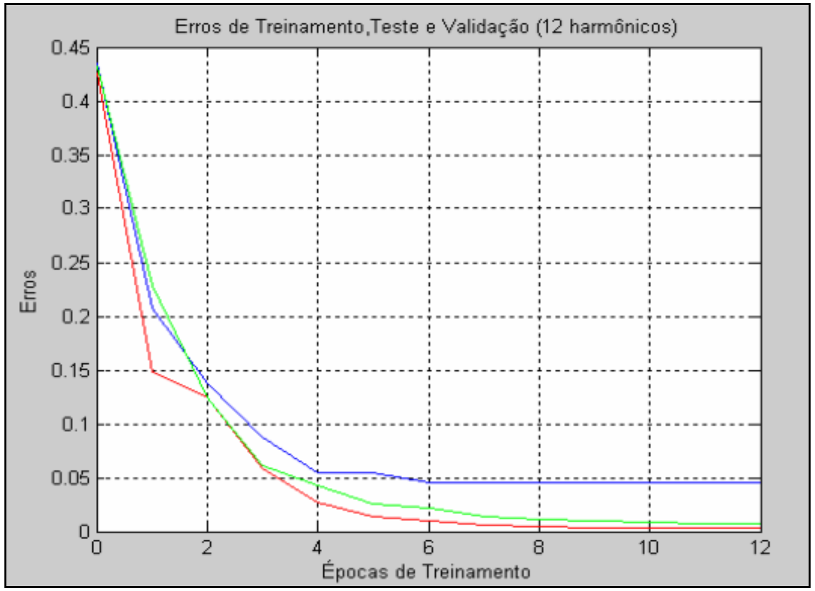


H=10 ➛
2 Misclassified images

H=12 ➛
1 Misclassified image

Flávio Oliveira & Alexandre Dias, 2007

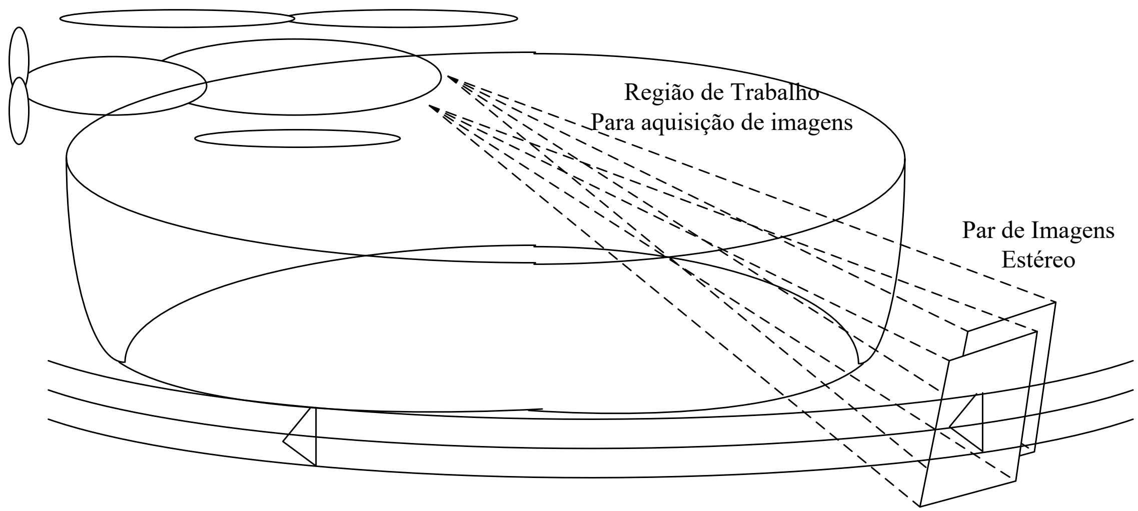
Results – Neural Network training



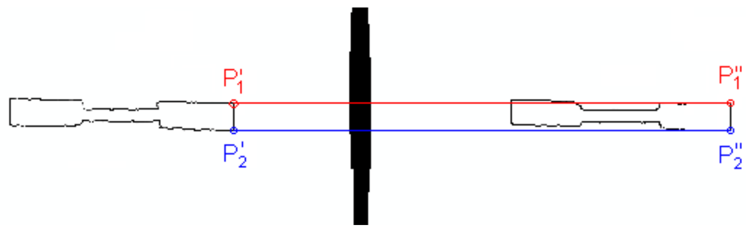
- Simulation of the validation set for the network trained with 10 harmonics
  - Misclassification of 2 images
- Simulation of the validation set for the network trained with 12 harmonics
  - Misclassification of 1 image

Gripper inspection with 3D reconstr.

- It is not possible to train an ANN for every position/orientation in the visual field of the VANT.
- ANN trained for a fixed point of view.
- Build 3D contour model
- Reproject 3D contour to ANN point of view
- Classify with ANN



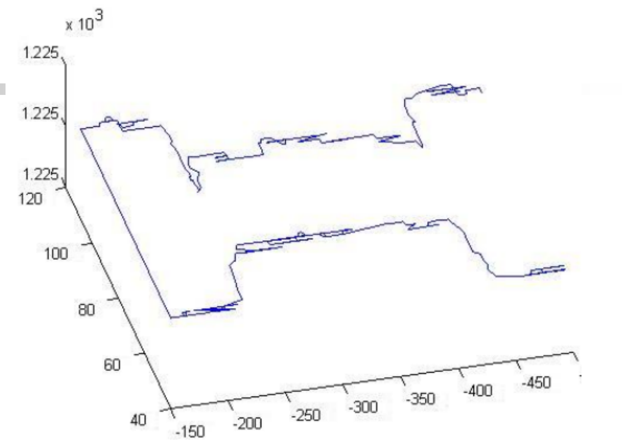
Different ROI's



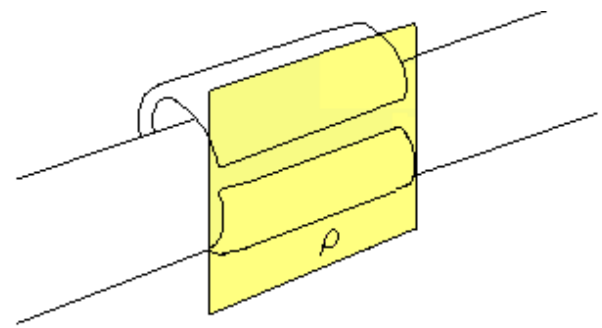
Correspondence –ROI contour in stereo pair

3D gripped cable

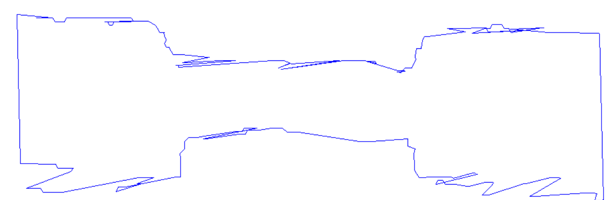
3D Reconstruction



ANN data bank Image plane



Reprojected contour for ANN



Results:

20 stereo pairs – 1 false pos., 1 false neg.

Elder Oroski, 2011

Liquid Level Process

Components

-3 Reservoirs (5x25x35 cm)

-1 Supply Tank

-3 Level Sensors

-2 Pumps (0 to 10 V)

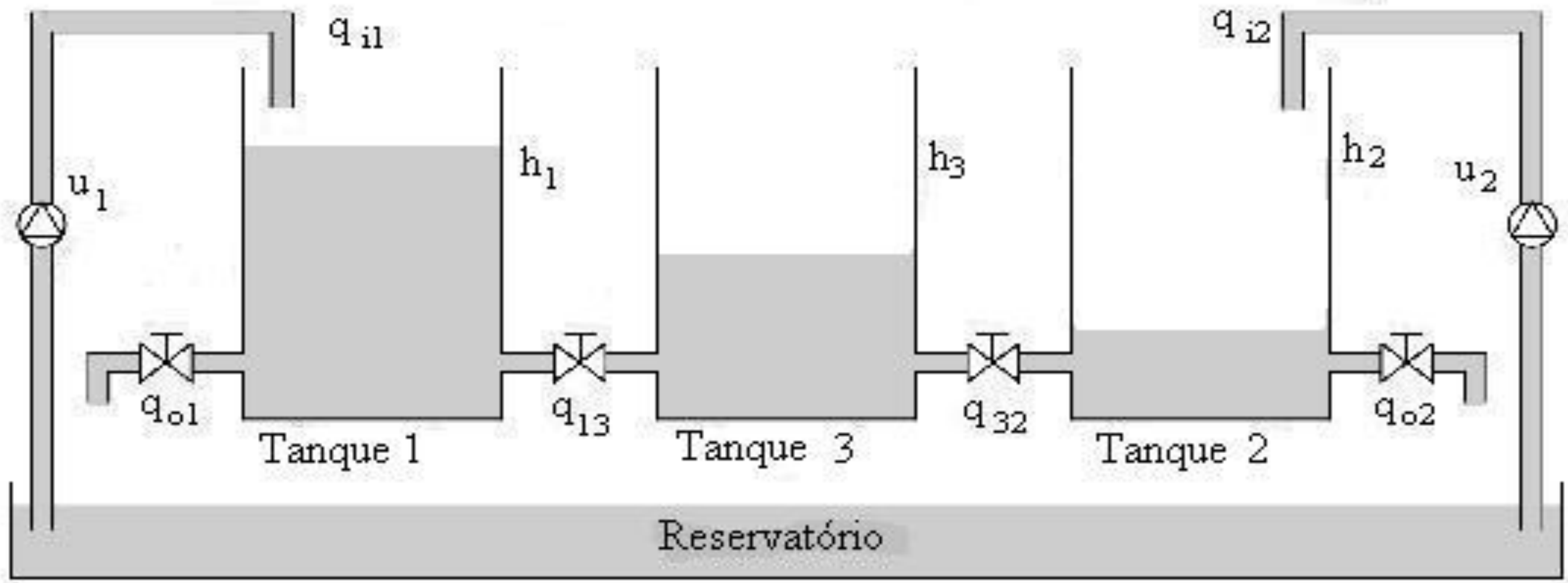
-2 Power Circuits

-A/D & D/A Interface

- Time Constant = 5 *min*

-Sampling Rate = 2Hz

Schematic Diagram



Dynamics

Bernoulli:

$$A\frac{dh_1}{dt} = q_{i1} + signal(h_3 - h_1)k\sqrt{|h_3 - h_1|} - k\sqrt{h_1},$$

$$A\frac{dh_2}{dt} = q_{i2} + signal(h_3 - h_2)k\sqrt{|h_3 - h_2|} - k\sqrt{h_2},$$

$$A\frac{dh_3}{dt} = -signal(h_3 - h_1)k\sqrt{|h_3 - h_1|}$$

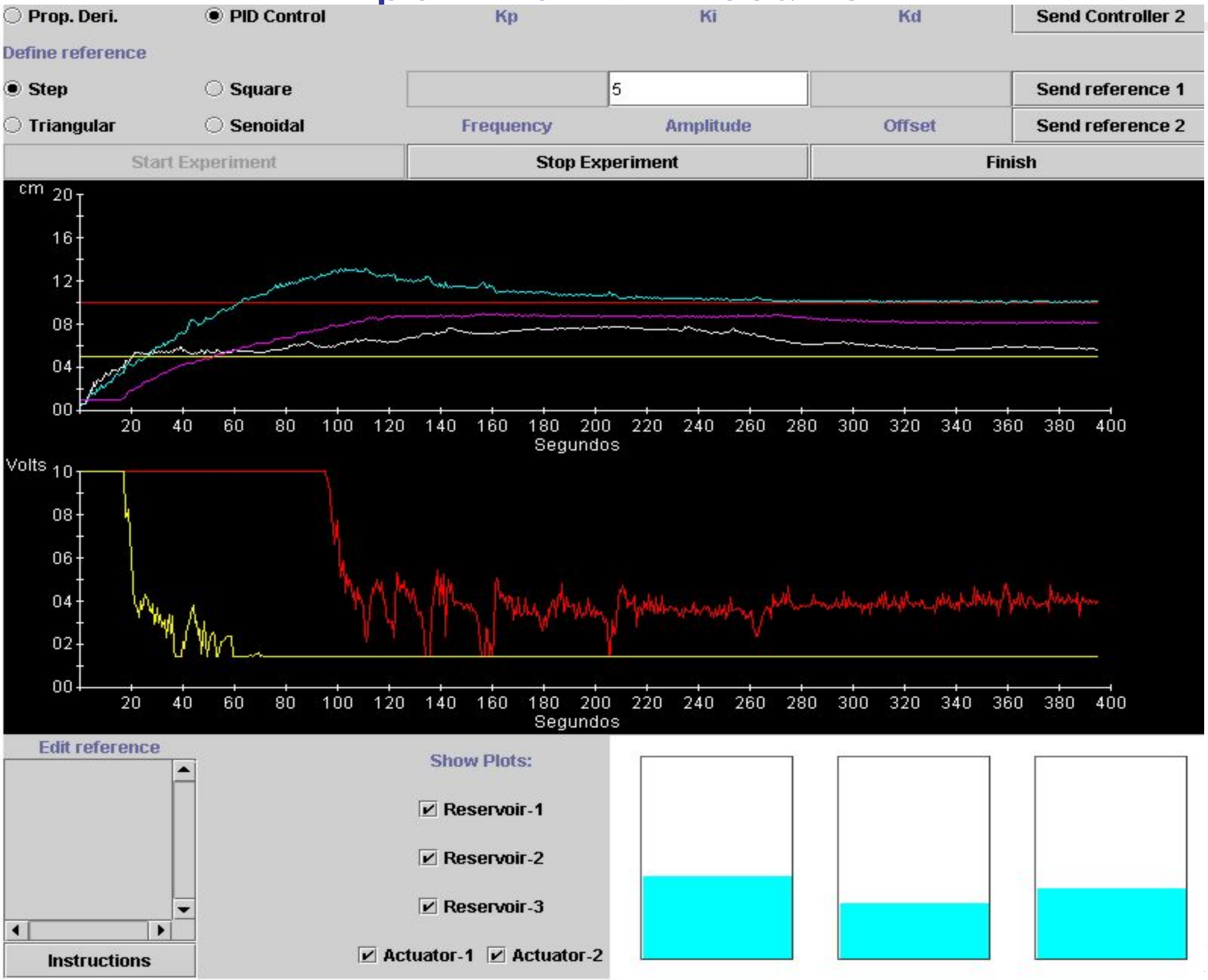
$$- signal(h_3 - h_2)k\sqrt{|h_3 - h_2|}$$

Non-Linear, Coupled e Multivariable

Remotely operated process - www

Client Process Controller-PC Server

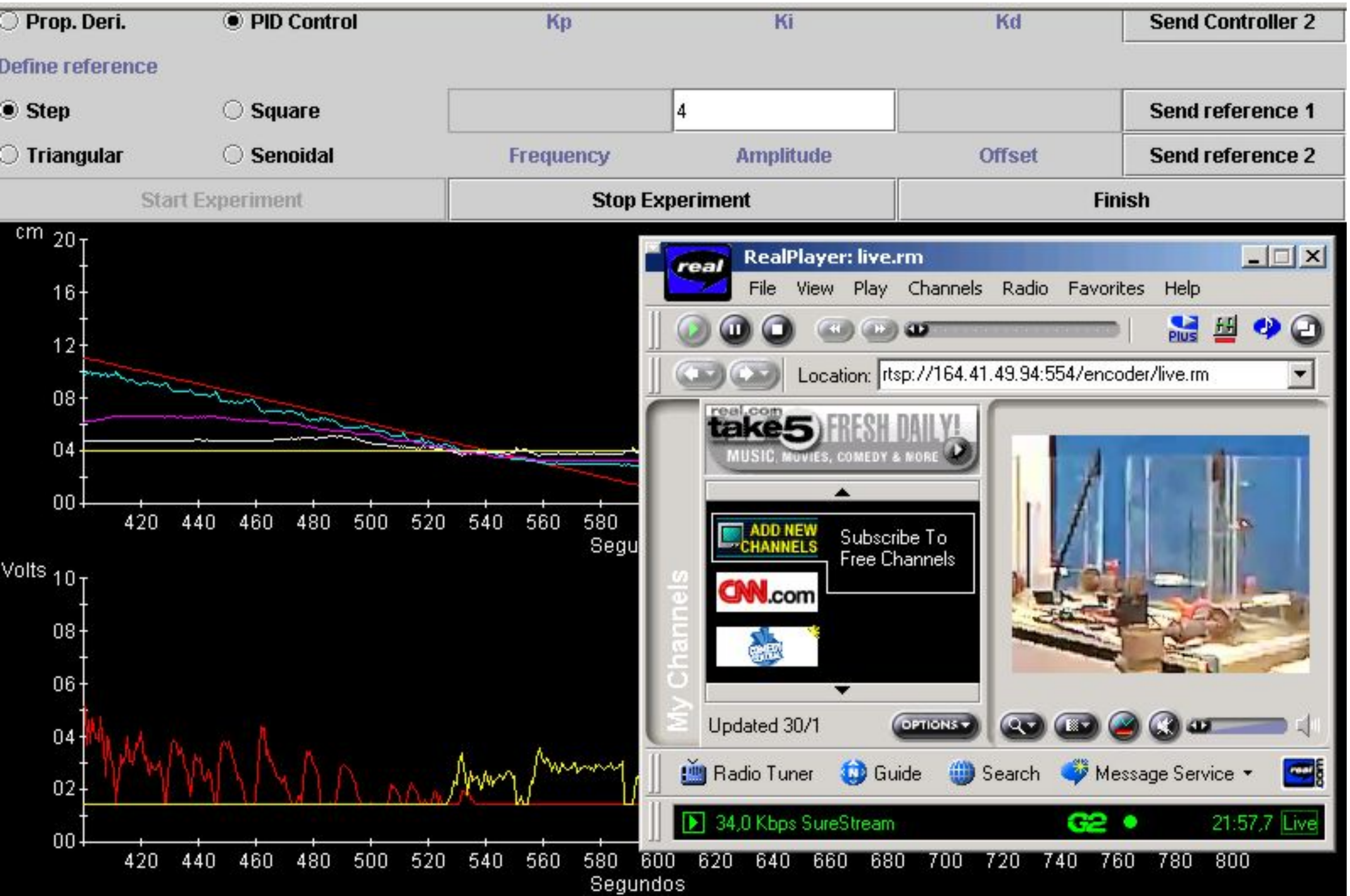
Experimental Results



Step Response

- 10 cm (tank 1)
- 05 cm (tank 2)

Experimental Results



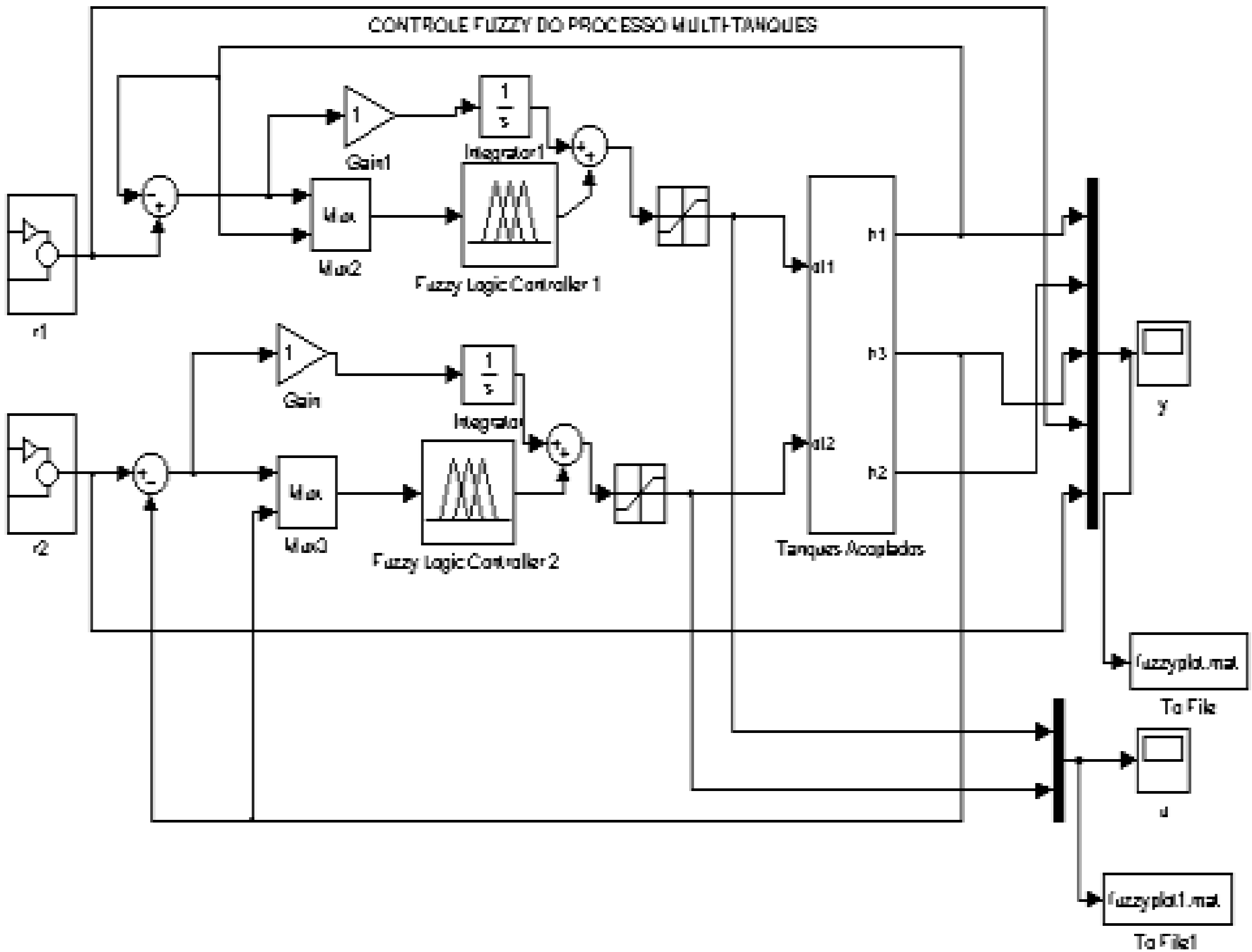
Reference

Step and Triangular

Live Video

Streaming

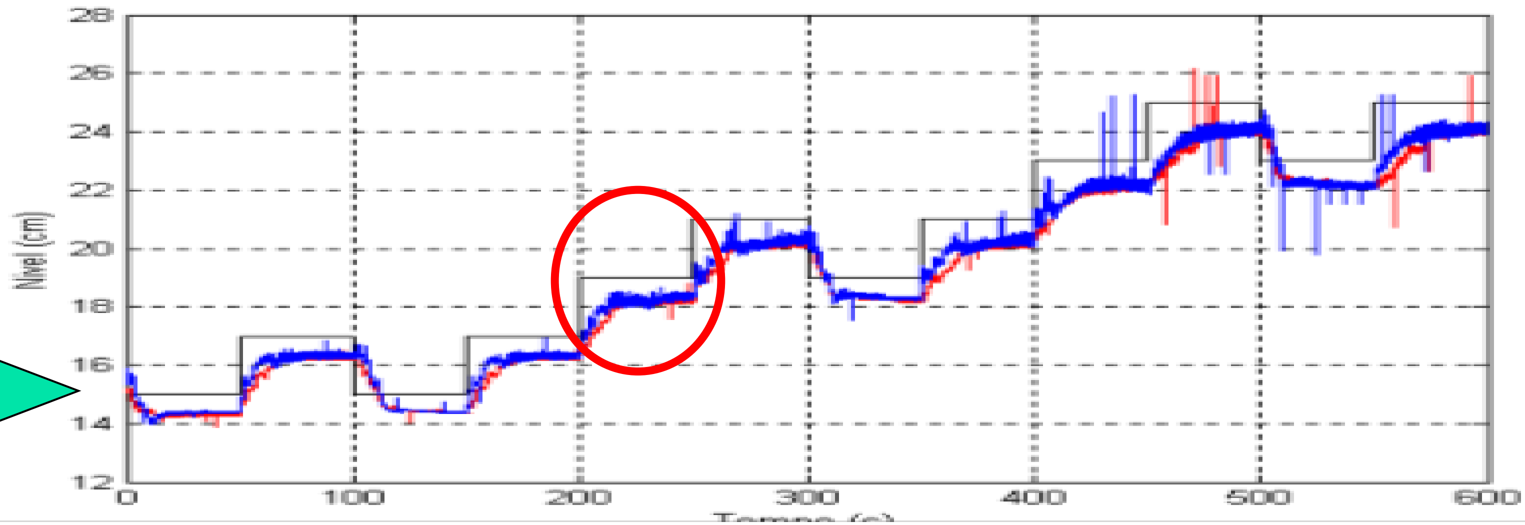
Fuzzy Control in Simulink



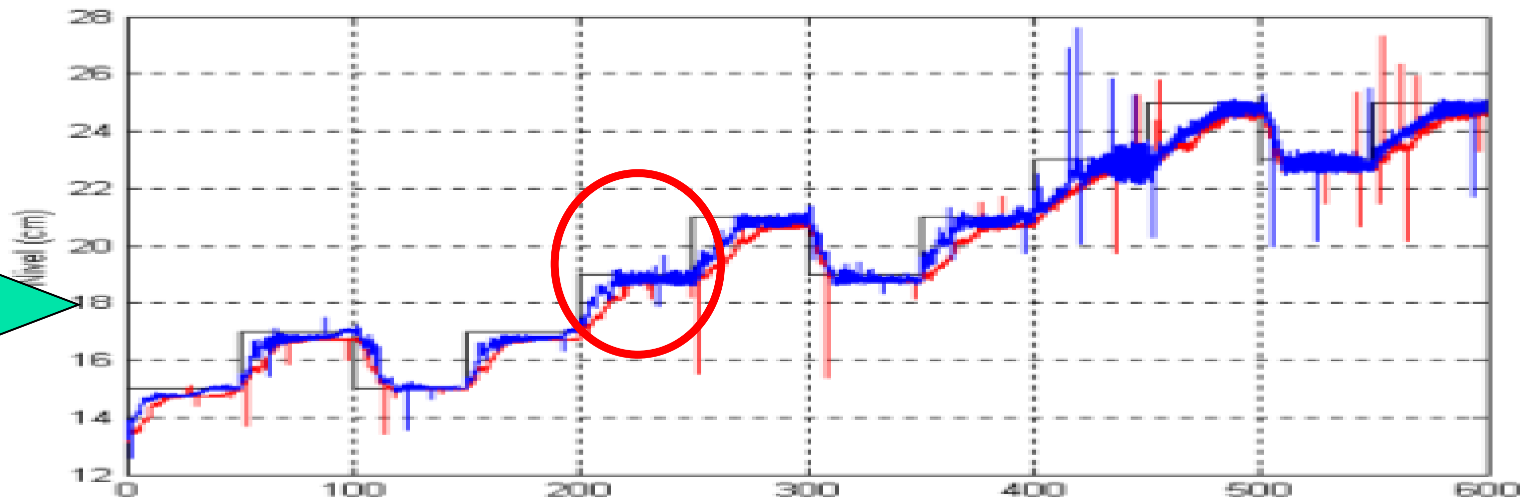
Fuzzy Control

Step Response to different levels:

Controle PI

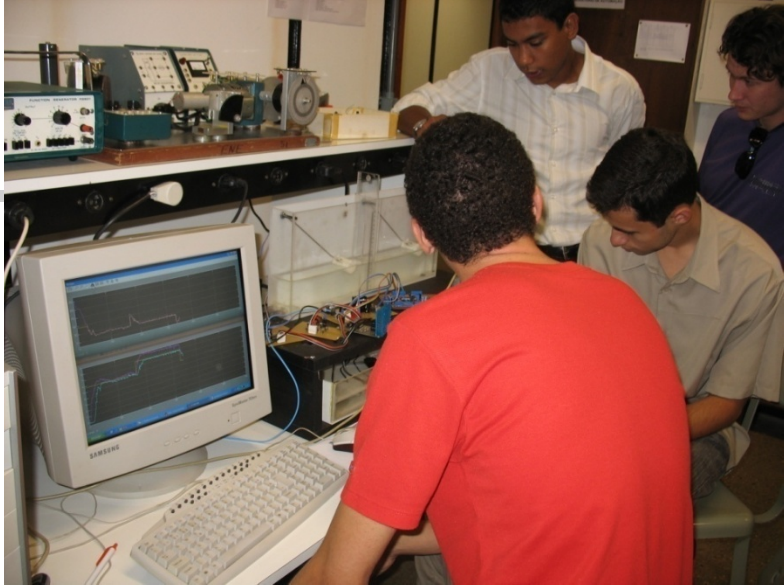


Fuzzy Control



LEARn

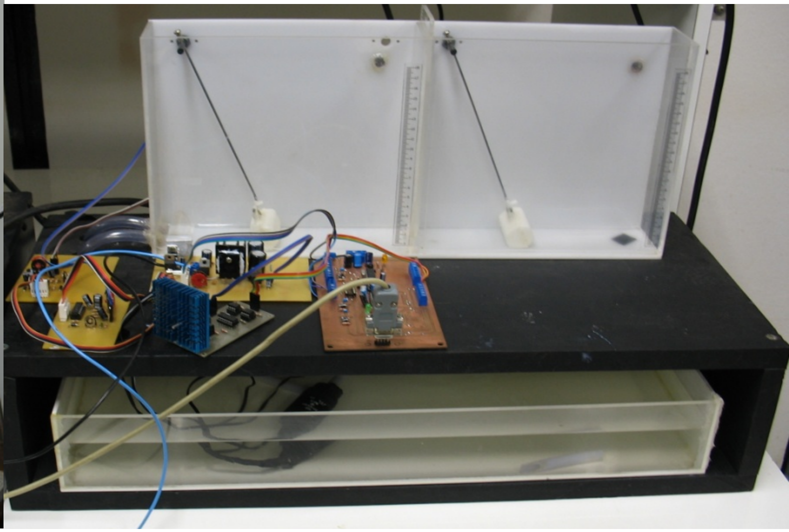
Remotely operated Automation Laboratory
(Laboratório de Ensino de Automação Remoto)



4th order

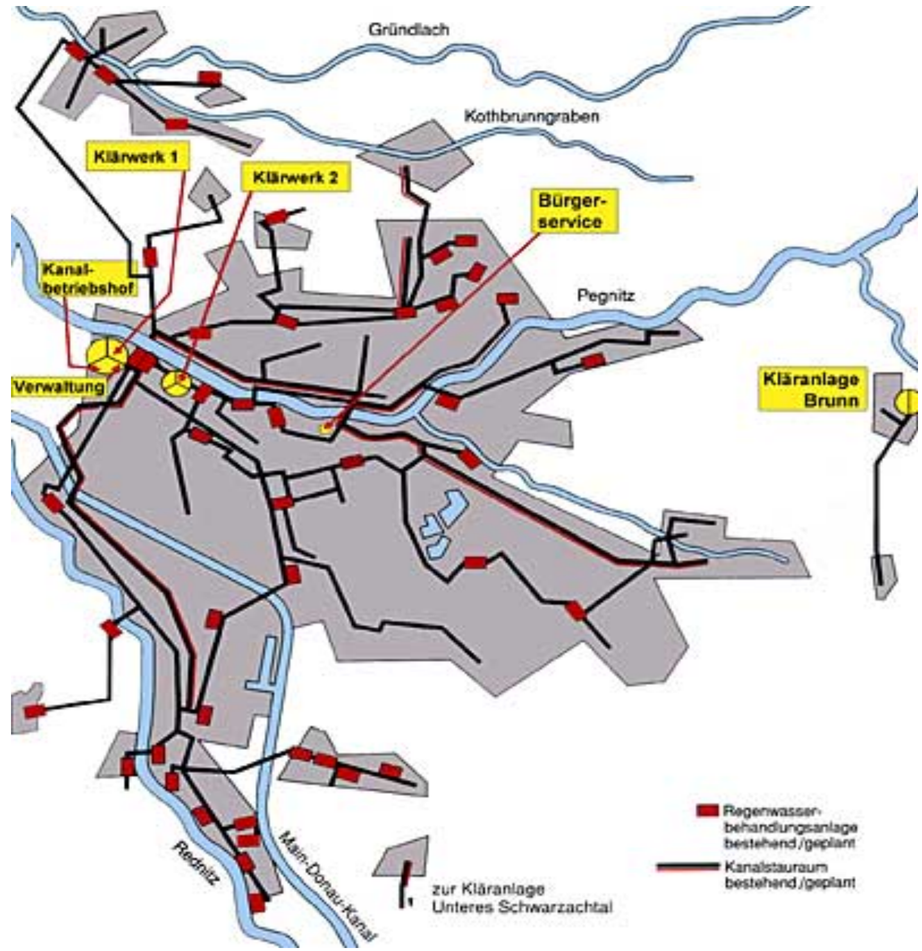
3rd order

2nd order



Water Treatment Station –

www.abwasser.nuernberg.de

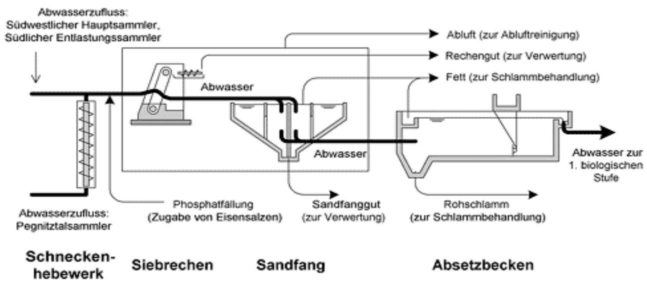


Water Treatment Station –

www.abwasser.nuernberg.de

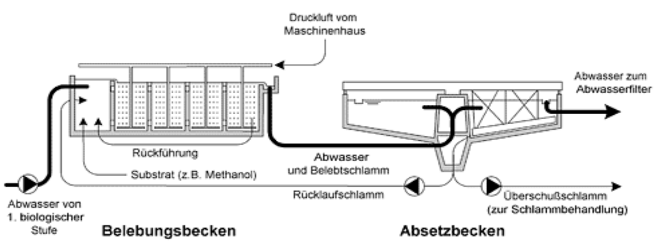
Klärwerk 1

Mechanik (Rechen, Sandfang, Vorklärung)

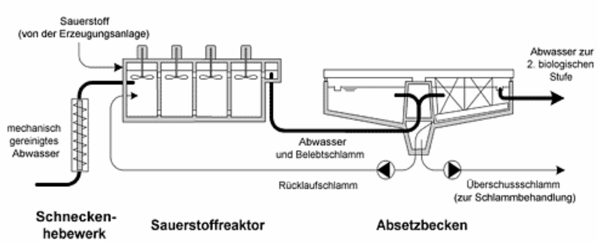


Klärwerk 1

2. biologische Stufe (Schwachlastbelebungsanlage)

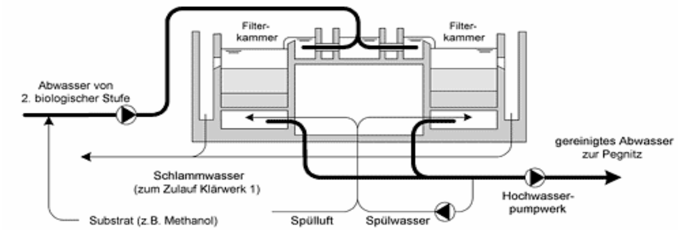


1. biologische Stufe (Hochlastbelebungsanlage)



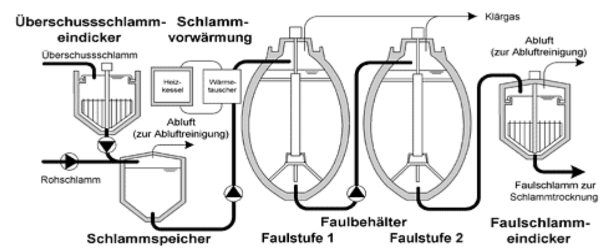
Klärwerk 1

Abwasserfilter



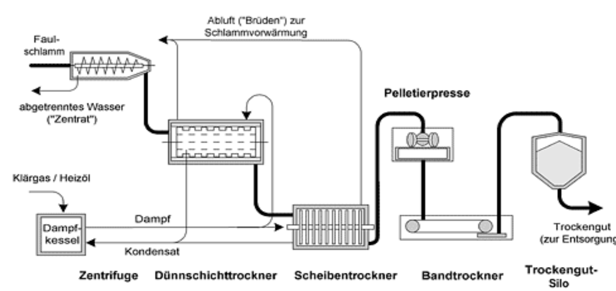
Klärwerk 1

Schlammfaulung



Klärwerk 1

Schlammtrocknung



Automatic guiding - BMW

BMW 645 ci - www.bmw.de

Cruise Control

automatic transmission

User Profile

- sporting
- economic
- cautious

Proximity Sensor

- front
- back
- side

Fuzzy Air Conditioner

A High Definition Of Design And Performance.

ENERGY SAVING, BETTER STARTING CURRENT, PLASMA FILTER, DESIGNER PANEL

LP-K2465QC

- 4 WAY SWING
- DEHUMIDIFICATION
- FUZZY LOGIC
- JET COOL

WHERE TO BUY

Rs.49826
MRP

ADD TO COMPARE

Camera

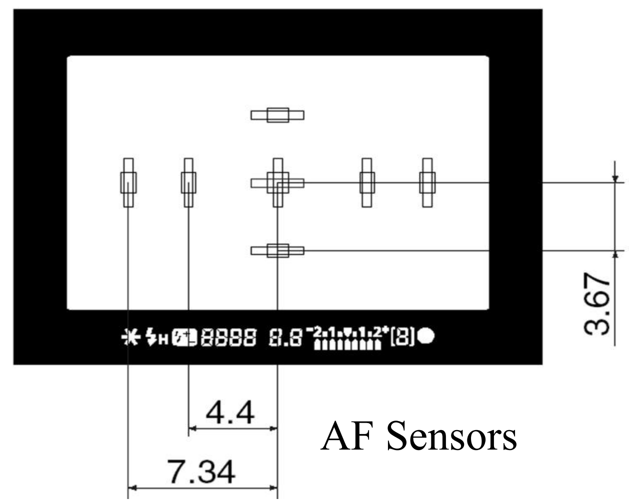
Olympus IS-5
Auto Focus SLR Camera - 28-140mm 5x zoom lens, Date imprinting capability, Panorama Mode - w/Case & Batteries

Features

...

Programmed Auto Exposure lets you choose between Full Auto, Stop Action, Portrait, Night Scene and Landscape modes

TTL metering system: ***Fuzzy logic*** ESP, center-weighted average, Spot



AF Sensors

Specifications

...

Focus TypeTTL phase-difference detection system with autofocus focus lock. Auto focus beep available. Auxiliary flash activation in low light.

Focus Range0.6 m to infinity in macro shooting; 0.6m to infinity at wide angle and 0.9m to infinity at telephoto in standard shooting. Predictive autofocus (in Stop Action mode only)

Washing machine

- Modern washing machines automatically determine the optimum settings to get your clothes clean with the use of ***fuzzy logic***. That's the 'skill' that gets machines to make 'best case' decisions based on incomplete information.
- Previously, washing machines were manually set. You had to make trial-and-error decisions on the amount of washing detergent, the size of the load, and the length of washing time. A fuzzy logic controller, comprising sensors, microchips and software algorithms, mathematically works out the amount of dirt and type of dirt on the clothes with the help of an optical sensor, which measures the transparency of the water.
- When the clothes are loaded into the washing machine and water added, the sensor checks to see how dirty the water is - dirtier clothes mean dirtier water, naturally. It also checks the type of dirt on the clothes by how fast the water gets saturated by the dirt. With this input, the fuzzy logic controller determines how soiled the load is, decides how much detergent is needed and how long it must wash the clothes.

Vacuum cleaner

Samsung VC-8930EN

Power Consumption : 2000W
Suction Power : 450W
Digital Auto Power Control (***Fuzzy Logic***)
Variable Power Control
5-Stage HEPA-Filter System
Exbug : Mite Killing Function
LED Display Panel
2 Step Smart Brush
Aluminium Telescopic Tube
Smart Protector
3 Built-in Accessories
2-Way Parking System
With Twister System

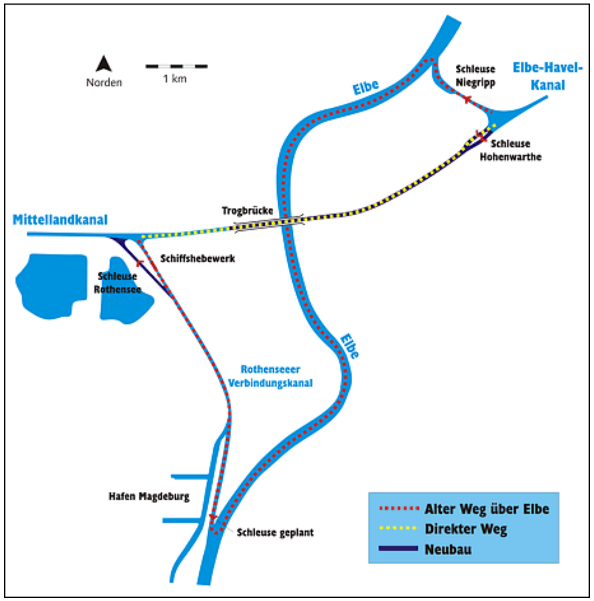
Digital Wrist Pressure Monitor

Model WS 501

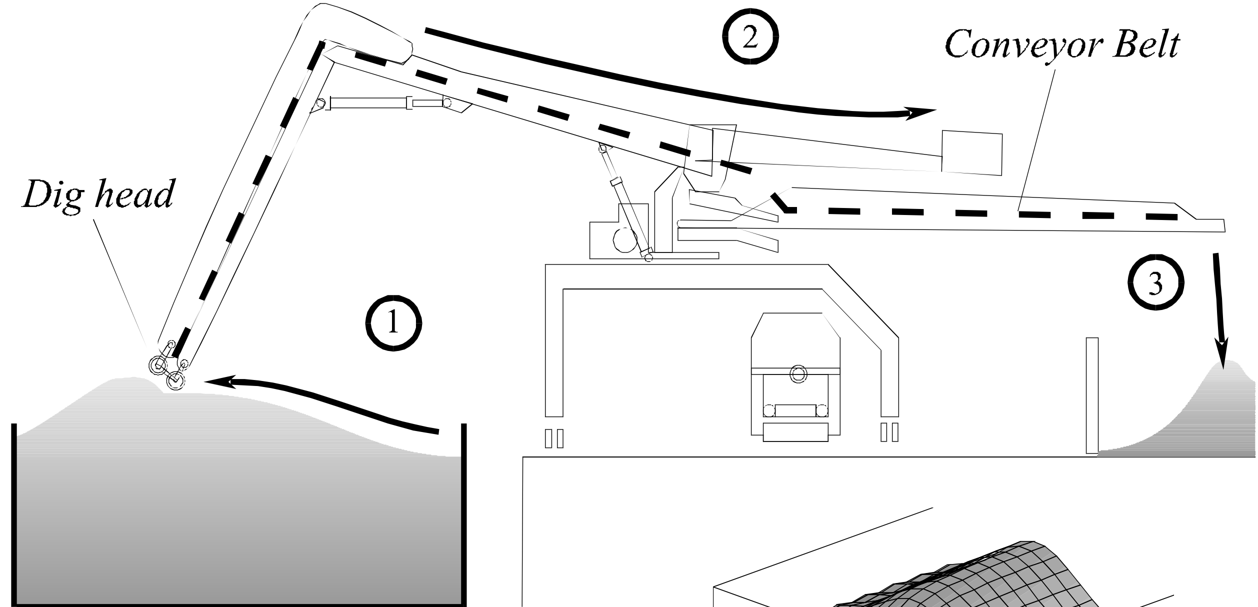
It has 60 memories with date and time (digital clock) that facilitates distance monitoring between doctor and patient.
Battery charge indicator.
Japanese FUZZY LOGIC technology of the latest generation. R$220,00

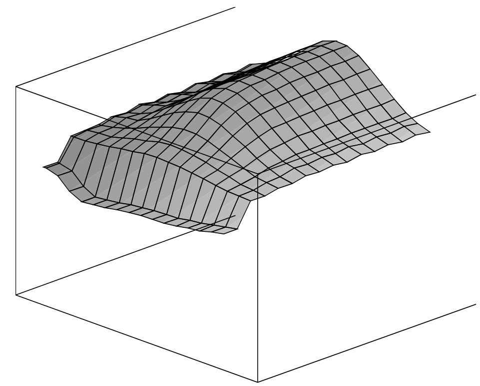
www.etronics.com.br/detalhes.asp?codpro=495

Coal Unloading – Erlangen/Germany



River Crossing – Minden, Elbe, Germany





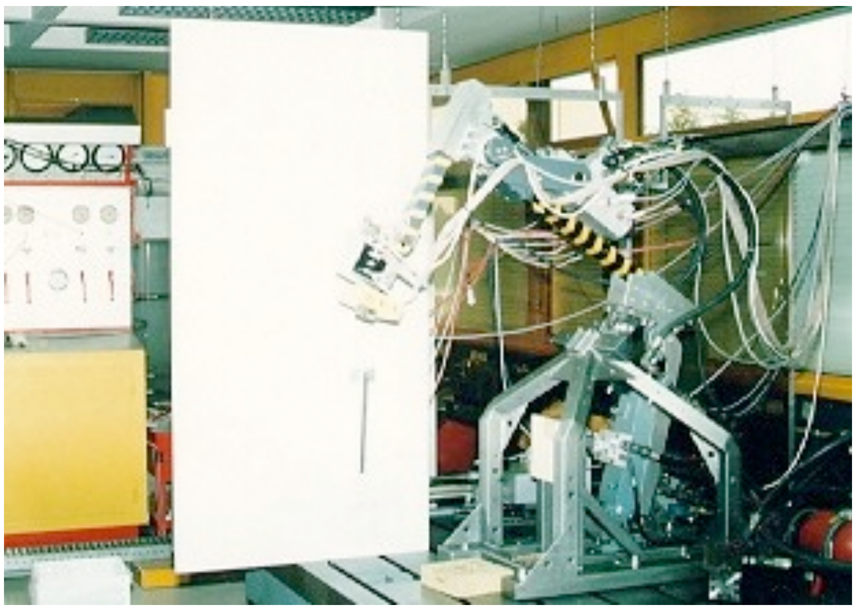
*Trajectory Planning - **Fuzzy***

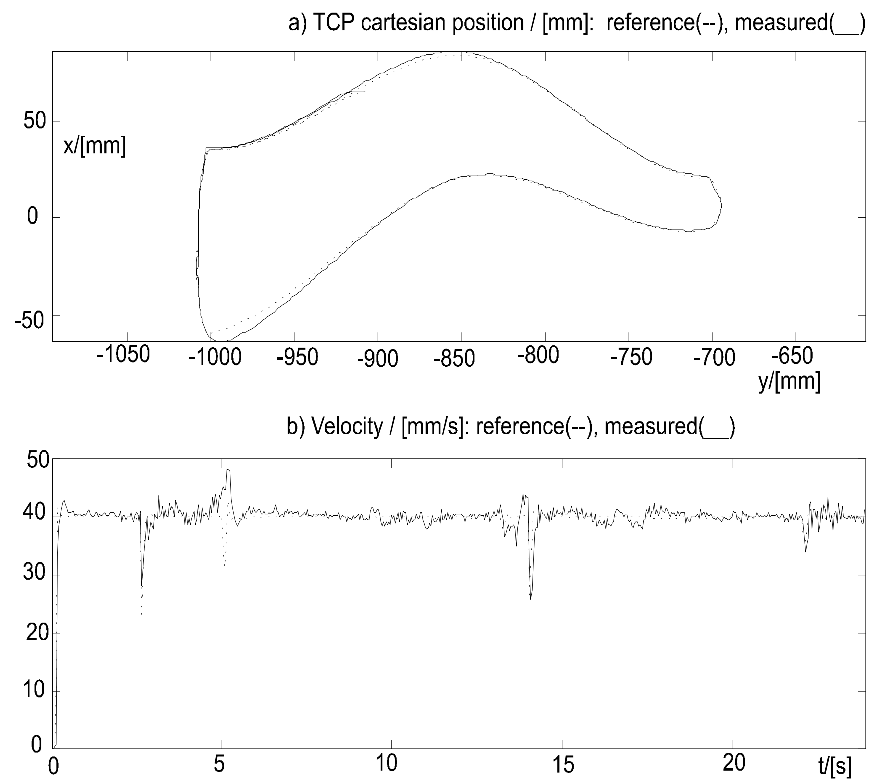
“Redundant Sensor Guided Unloading Crane”– MAN

Bauchspiess, 1995

Universität – Erlangen-Nürnberg

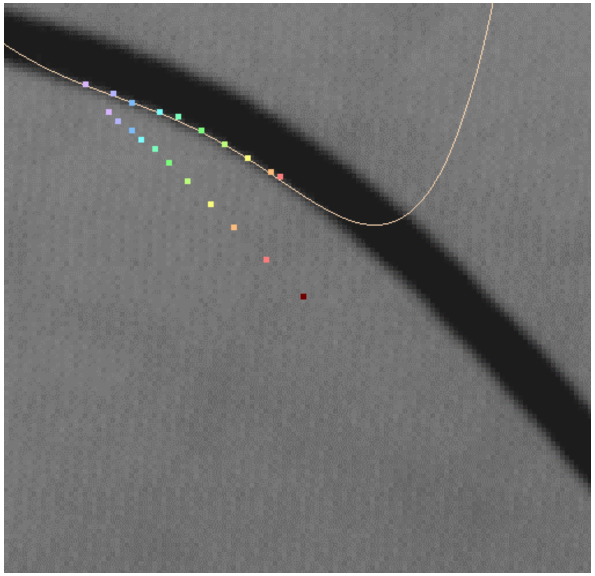
Sensor guided Hydraulic Robot

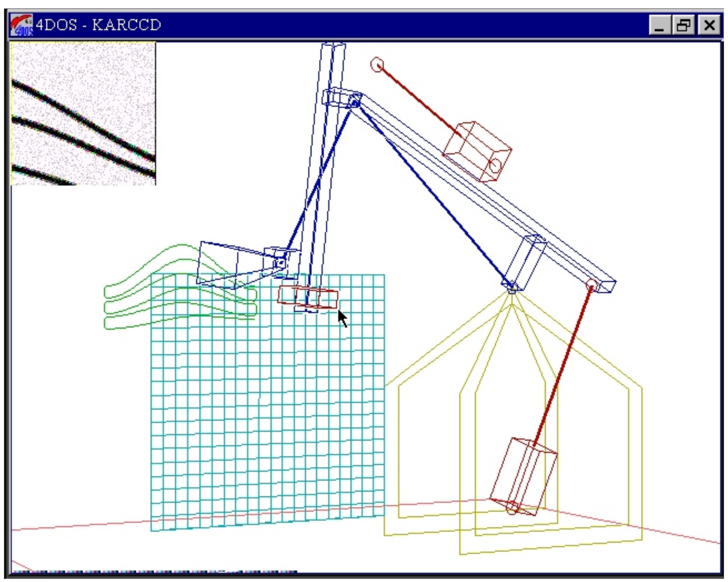


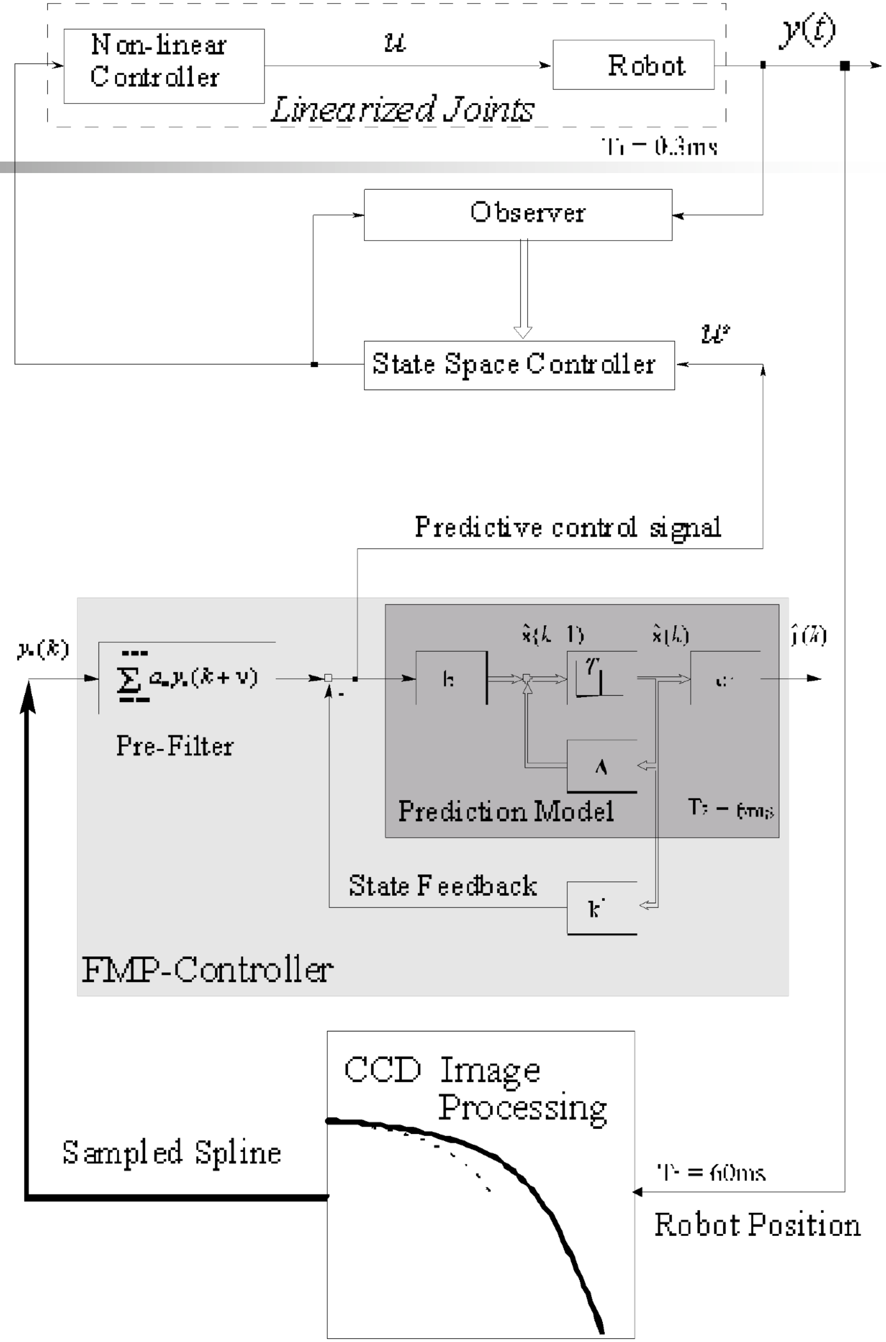


Bauchspiess, 1995

Path tracking



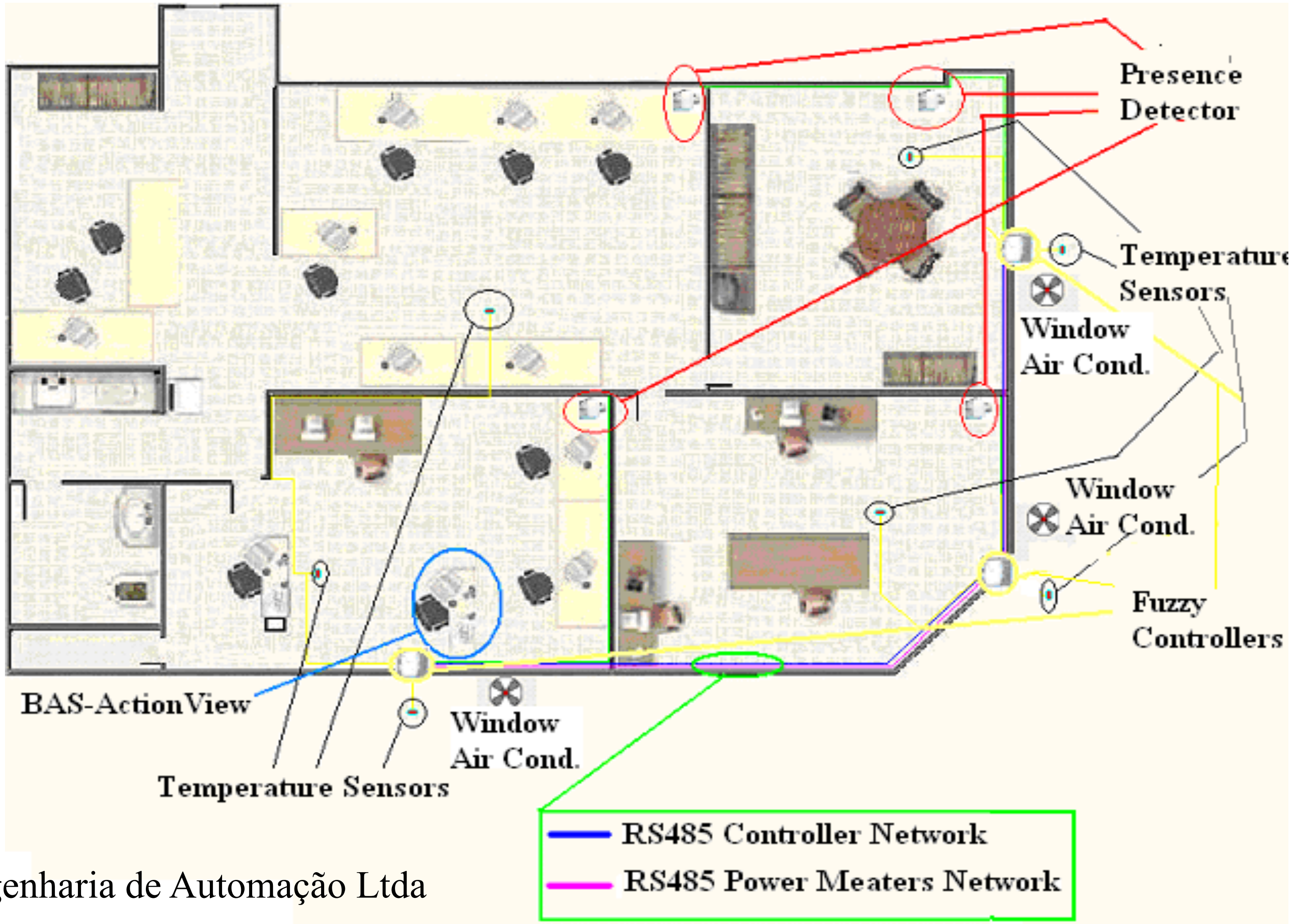




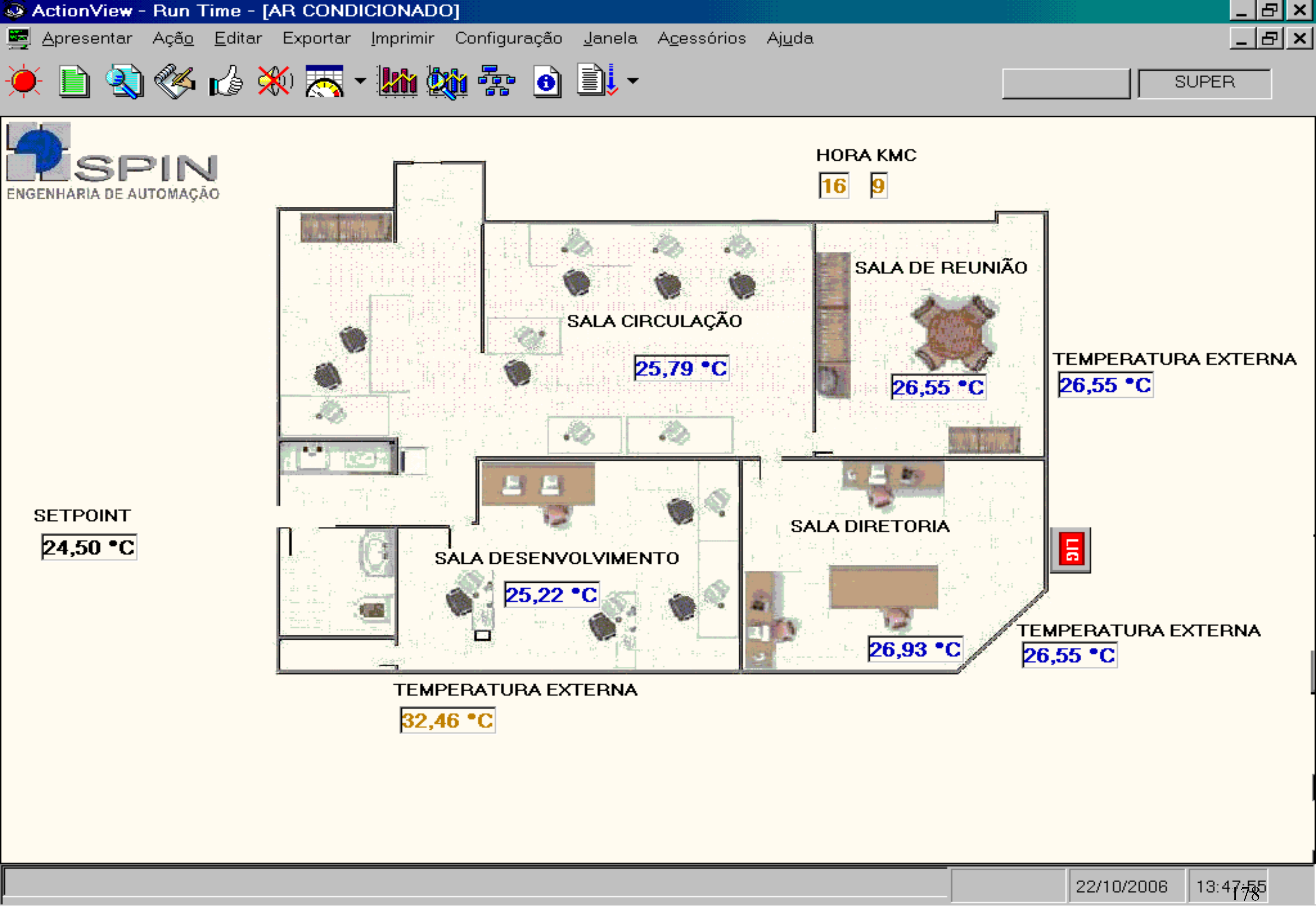
Ambient Intelligence

- Comfort and energy rationalization
  - Factors
    - temperature,
    - humidity,
    - outside temperature,
    - solar radiation,
    - neighboring rooms,
    - presence of persons,
    - furniture int the rooms,
    - heat sources (e.g., computers),
    - windows,
    - heaters,
    - air conditioners
    - etc.

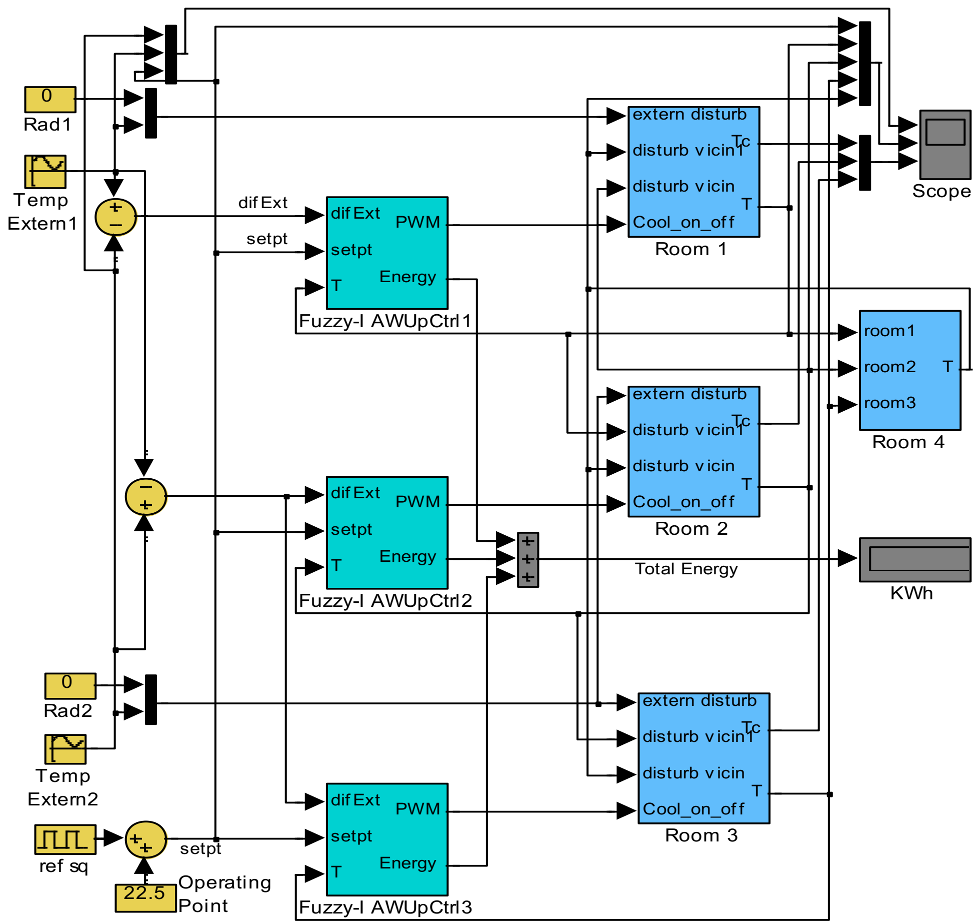
Thermal Comfort x Energy Rationalization



Spin Engenharia de Automação Ltda

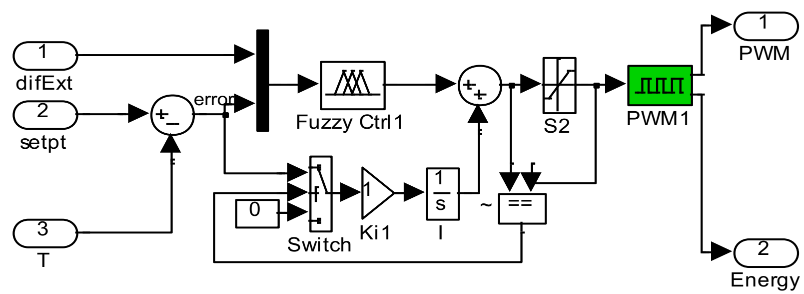

ActionView - Run Time - [AR CONDICIONADO]
Apresentar Ação Editar Exportar Imprimir Configuração Janela Acessórios Ajuda
SUPER
SPIN
ENGENHARIA DE AUTOMAÇÃO
HORA KMC
16
9
SALA DE REUNIÃO
SALA CIRCULAÇÃO
25,79 °C
26,55 °C
TEMPERATURA EXTERNA
26,55 °C
SETPOINT
24,50 °C
SALA DIRETORIA
LIG
SALA DESENVOLVIMENTO
25,22 °C
26,93 °C
TEMPERATURA EXTERNA
26,55 °C
TEMPERATURA EXTERNA
32,46 °C
22/10/2006
13:47:55

Simulation of the Energy Rationalization – Thermal Comfort

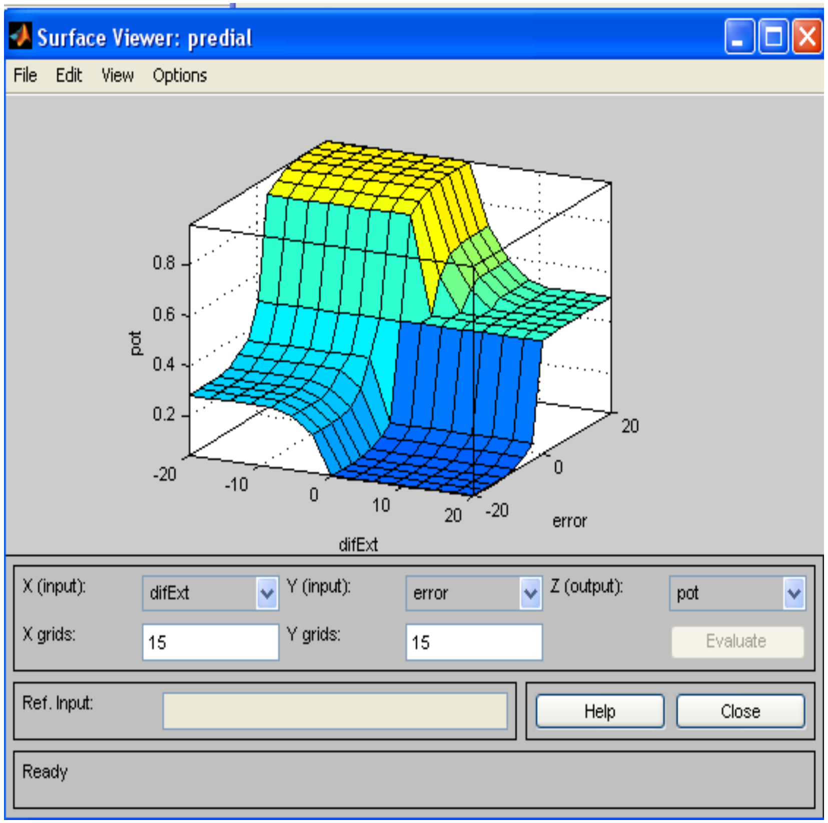


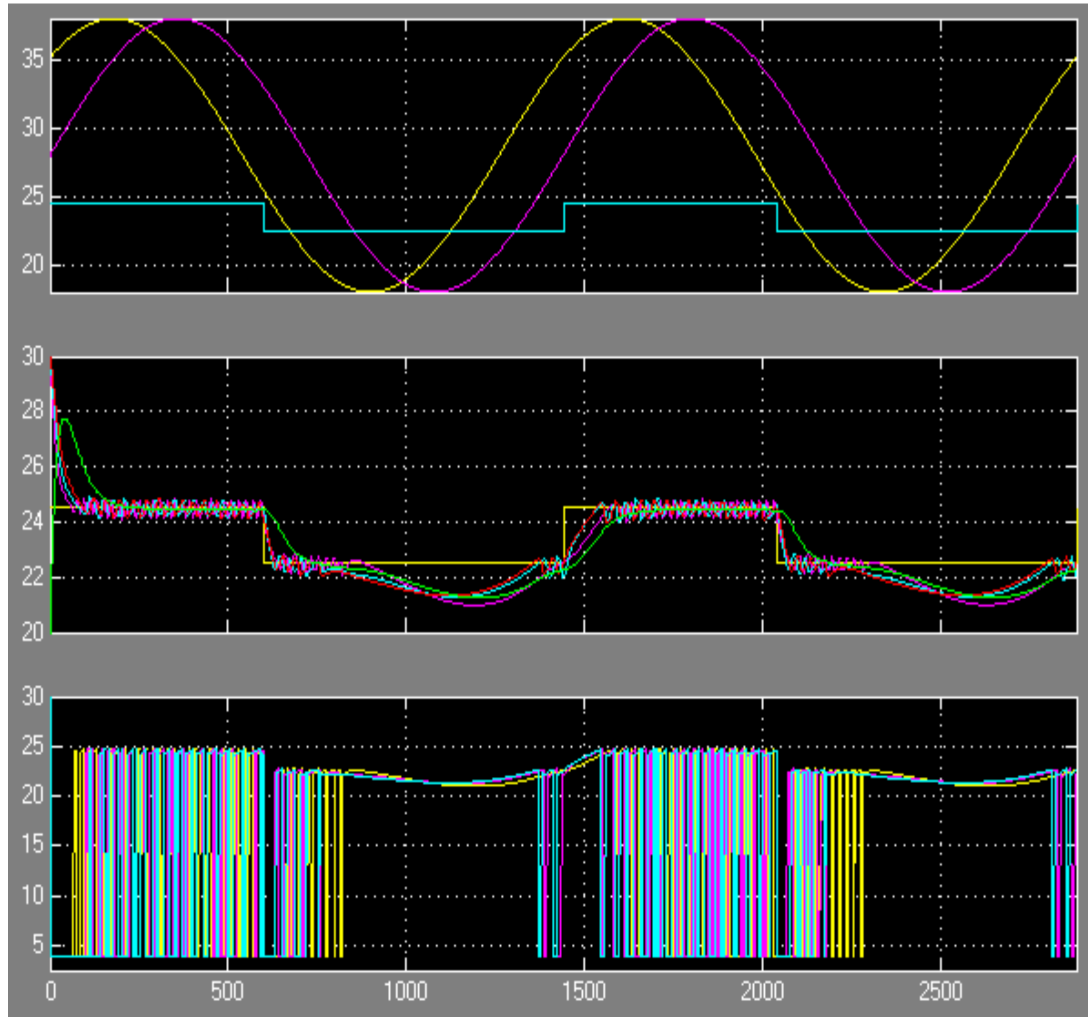
Projects:

FAP-DF
Finep – LabInov



Simulation of Fuzzy Thermal Comfort





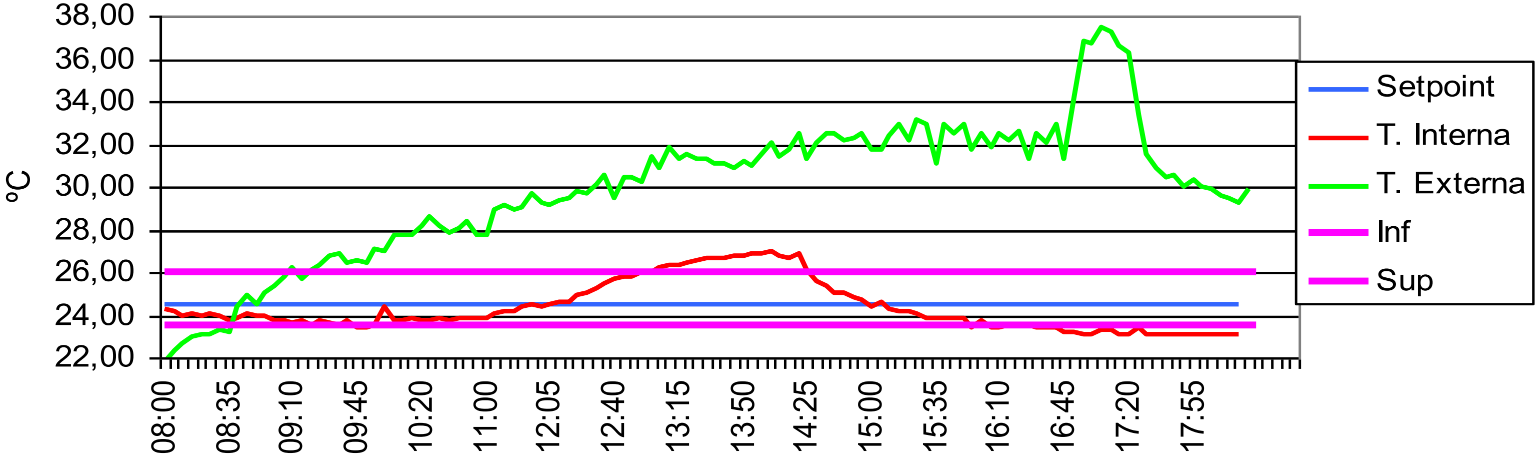
Measured Energy Consumption (kWh)

Experiment\ Room	Develop.	Directors	Meeting	Total
On-Off Dawn	19,39	11,87	12,04	43,30
Fuzzy Dawn	03,78	01,07	02,05	06,90
On-Off 8-12 14-18	35,25	17,42	19,07	71,71
Fuzzy 8-12 14-18	21,97	13,50	18,14	53,61
On-Off 8-18	35,34	17,96	19,95	73,48
Fuzzy 8-18	16,41	15,80	13,10	45,32

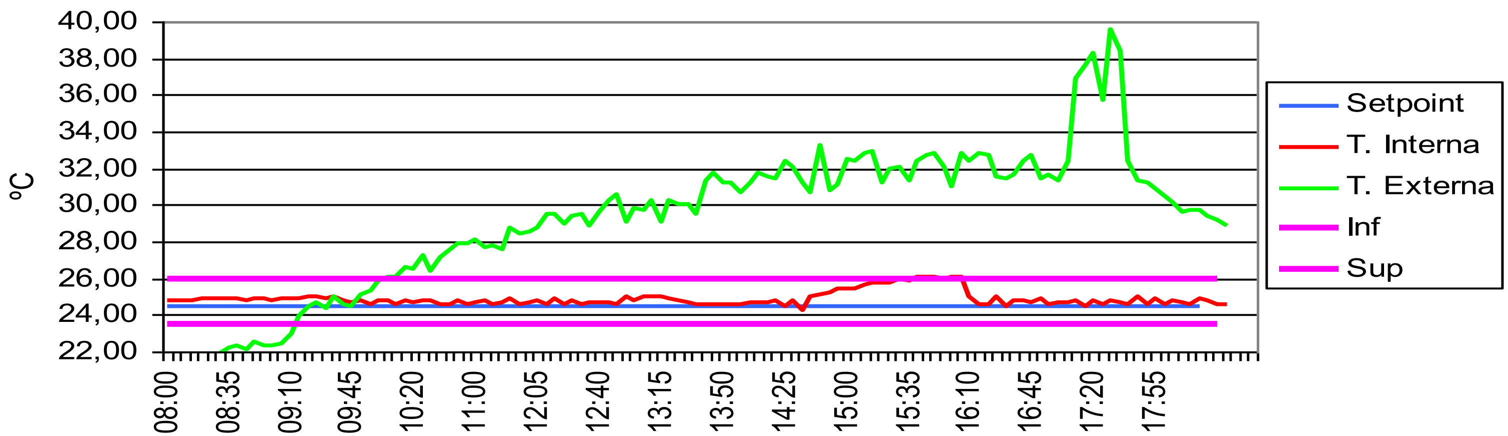
Spin Engenharia de Automação Ltda, 2006

Development Room

On-Off 16-09-2006



Fuzzy Control 14-09-2006

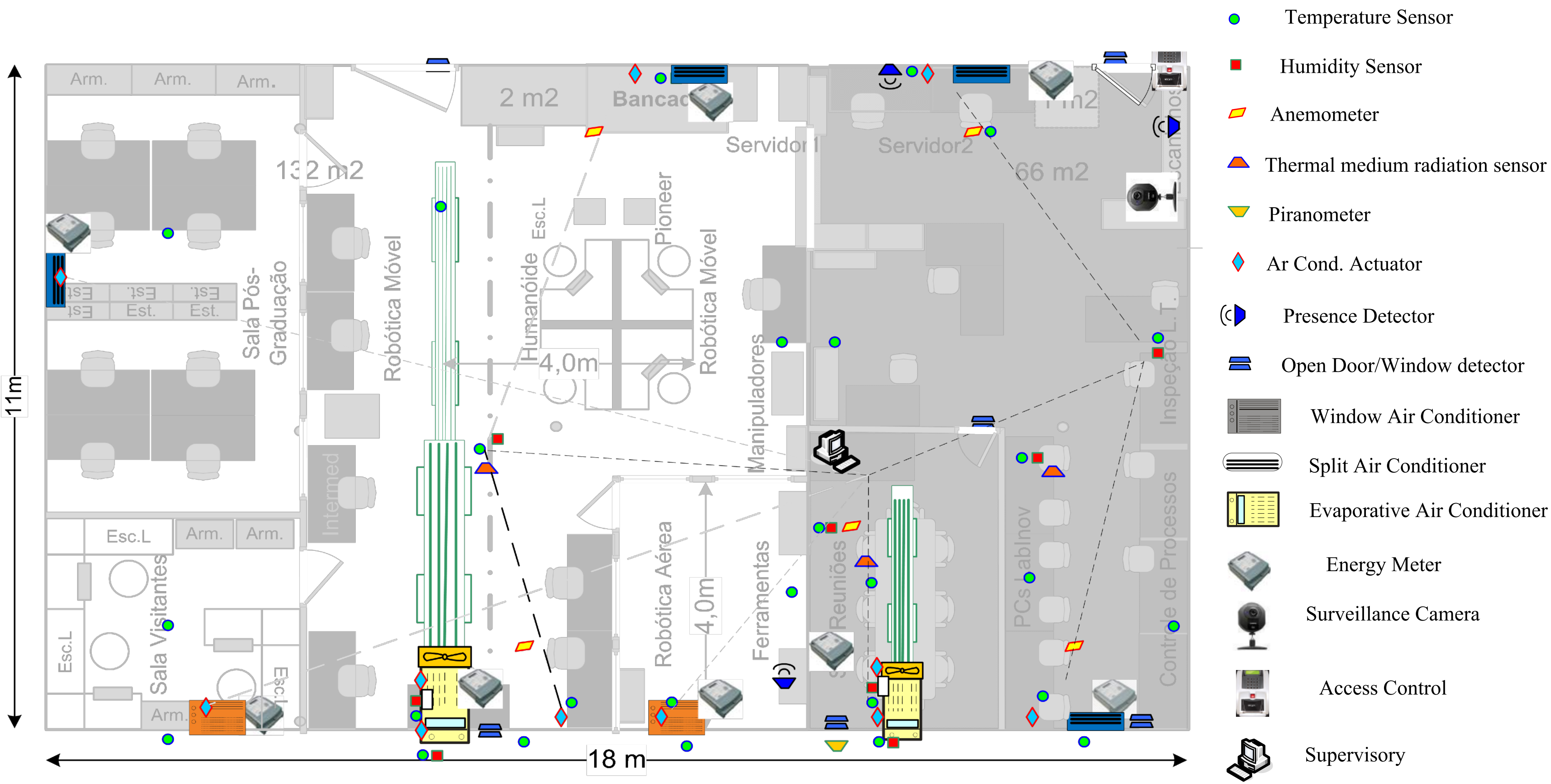


~30% saving!

The picture can't be displayed.

Finep/LabInov

Ambient Intelligence Innovation Laboratory



Energy-Saving Approach:

Model-Based HVAC Control

$$J = \underbrace{\sum_{i=1}^{h_p} \left(y(k+i) - y_R \right)^2}_{comfort\ related} + \sum_{i=0}^{h_c - 1} \Delta\mathbf{u}^T(k+i)\mathbf{Q}_{\Delta u}\Delta\mathbf{u}(k+i) + \underbrace{\mathbf{u}^T(k+i)\mathbf{Q}_u\mathbf{u}(k+i)}_{energy\ related}$$

$Where:$

$h_p - prediction\ horizon$

$h_c - control\ horizon$

$y - controlled\ variable$

$y_R - reference$

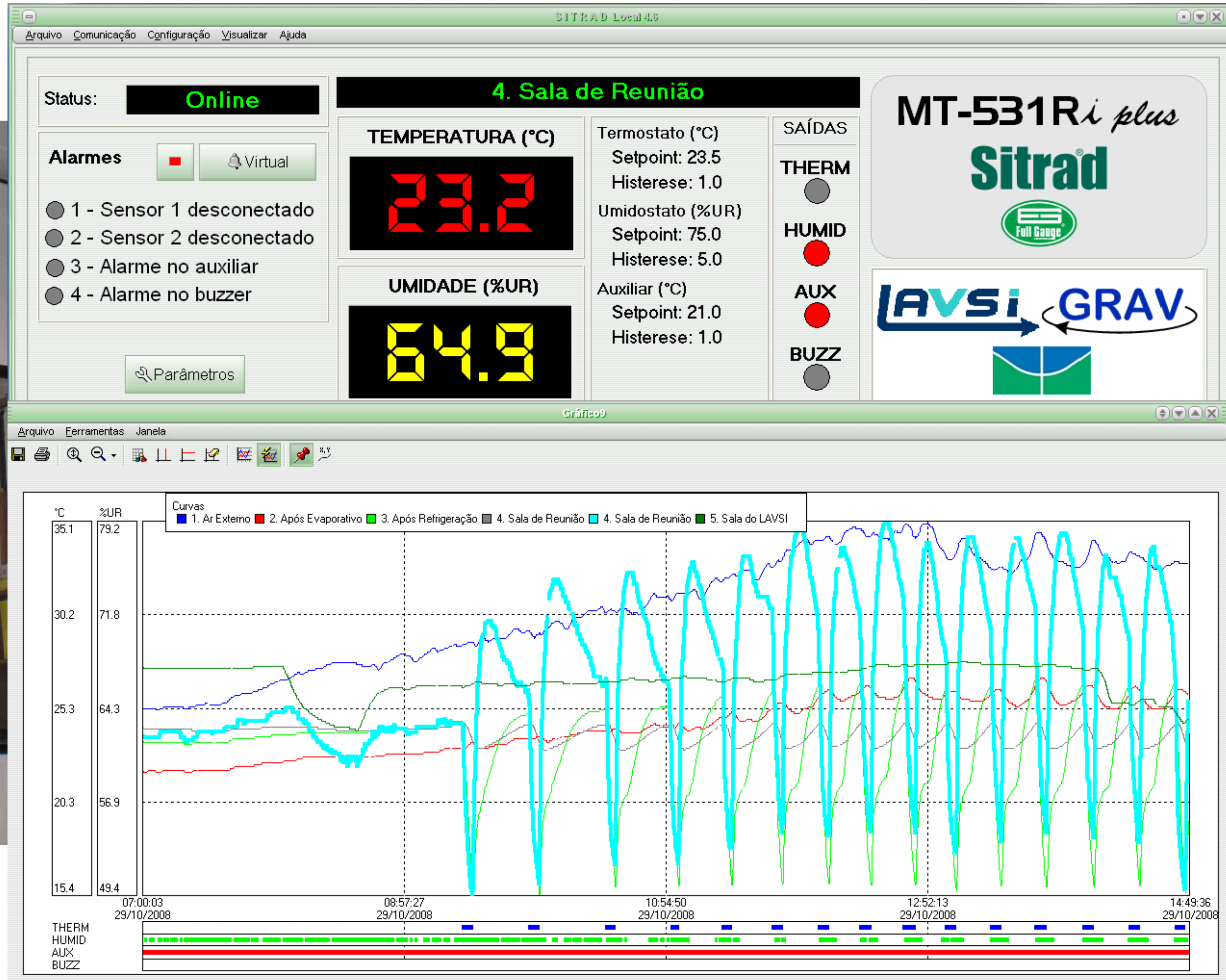
$u - manipulated\ variable$

$\mathbf{Q}_{\Delta u}, \mathbf{Q}_u - weighting\ matrices$

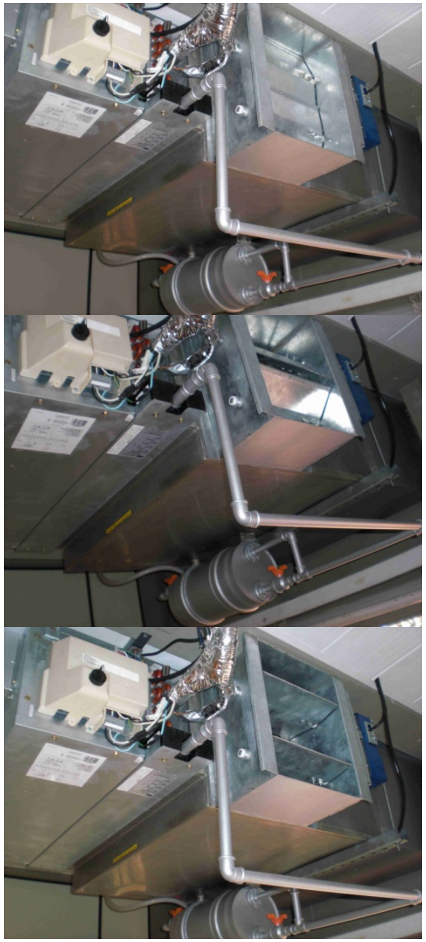
Predictive Cost Function:

Considers comfort and energy saving. Needs model!

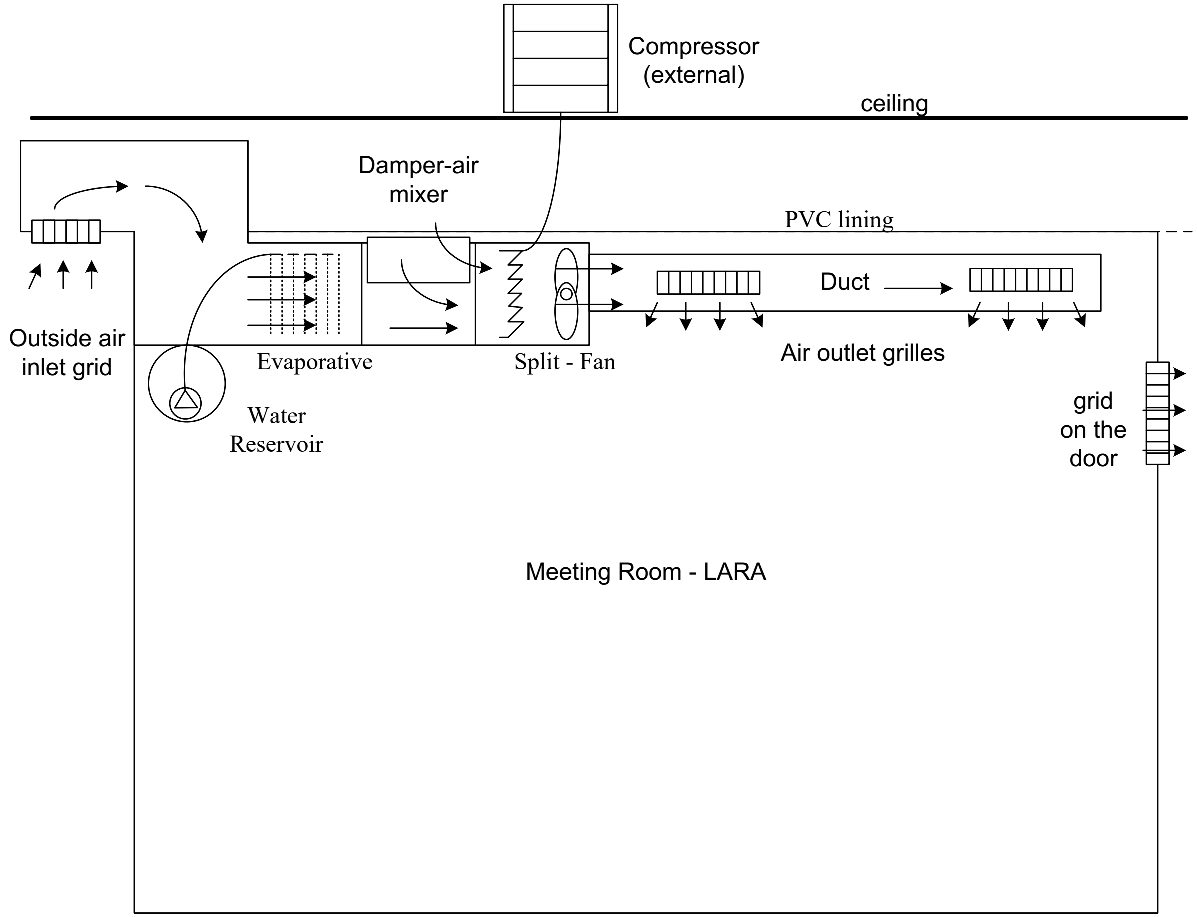
Hybrid Air Conditioning: Evaporative-Conventional



Hybrid Air Conditioning: Evaporative-Conventional

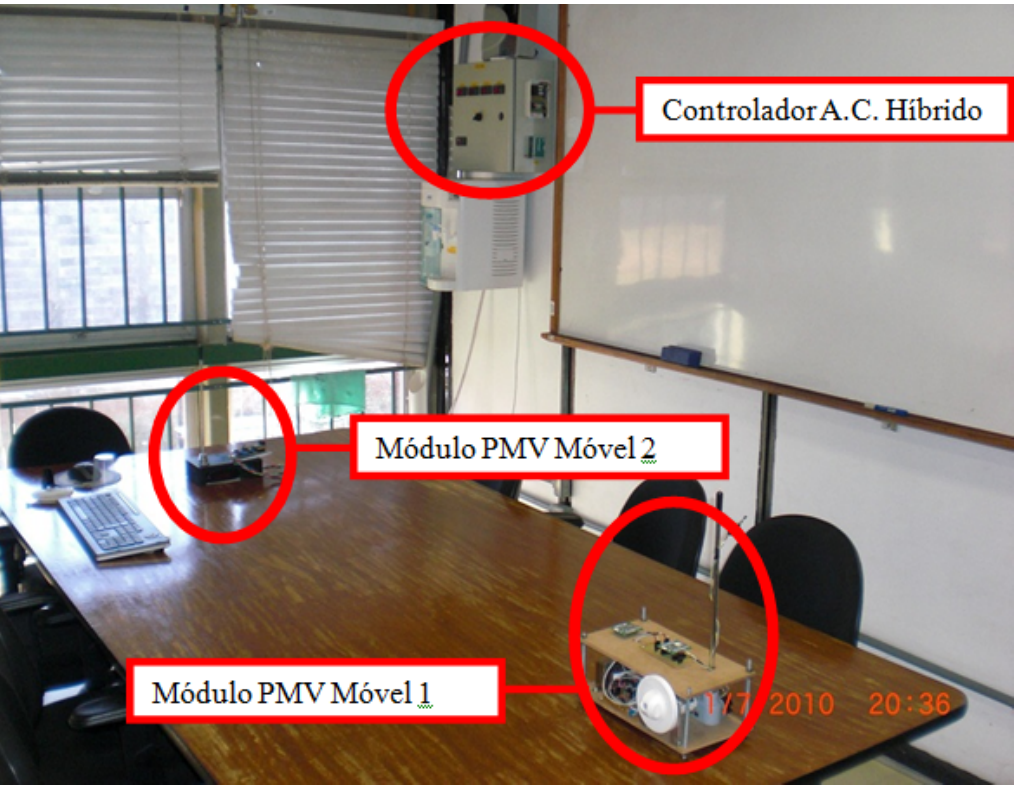


Damper
(air mixer)



Schematic Diagram

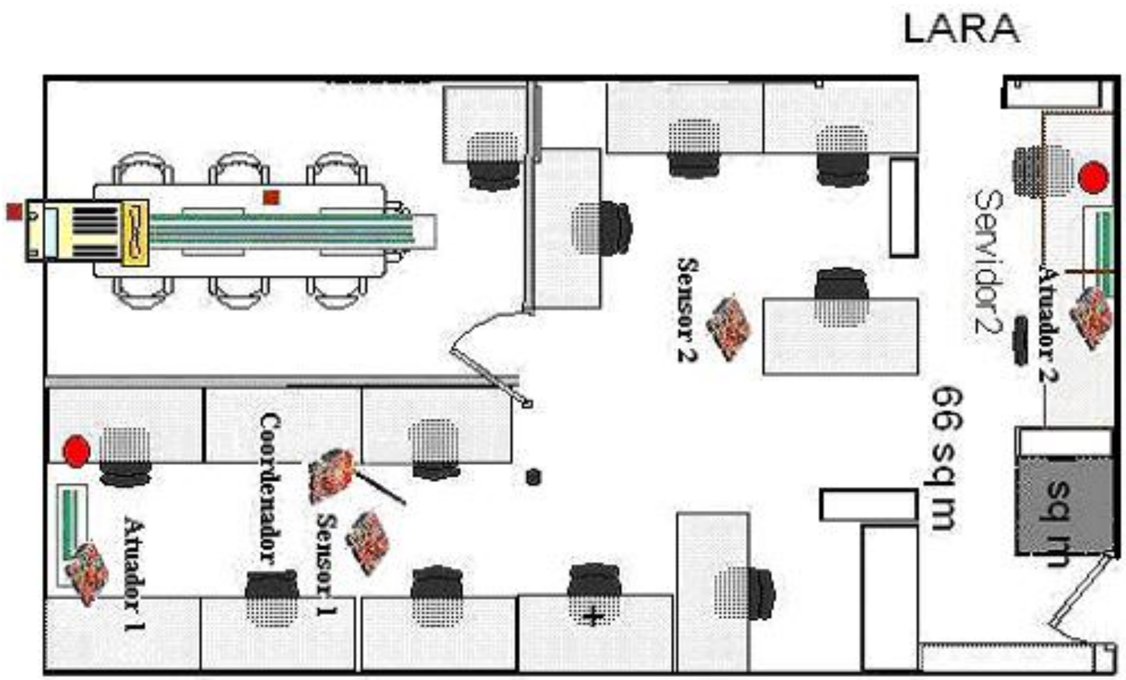
Hybrid Air Conditioner Controller

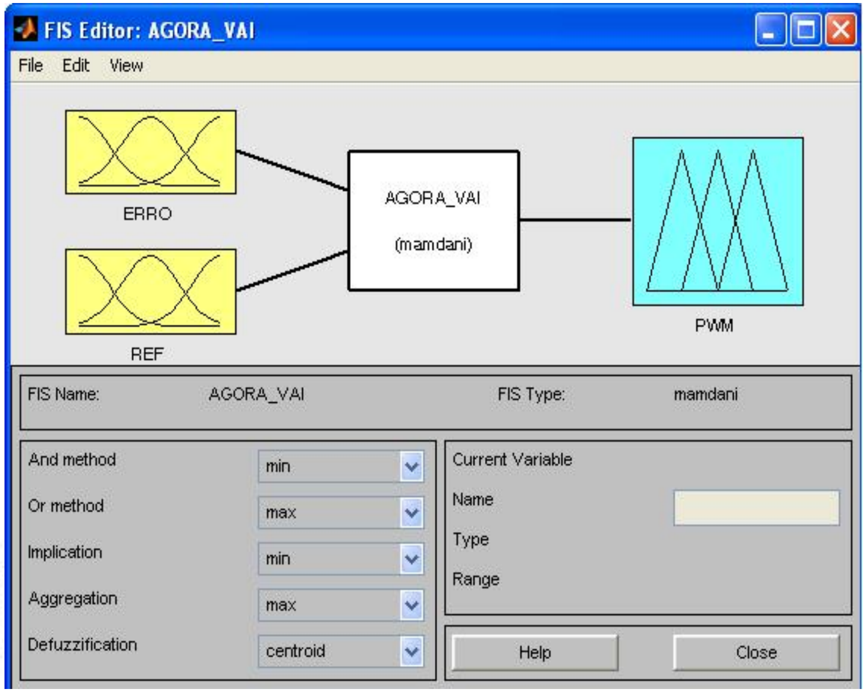


Meeting Room LARA.
Mobile modules 1 e 2. Actuator of the hybrid air conditioner híbrido

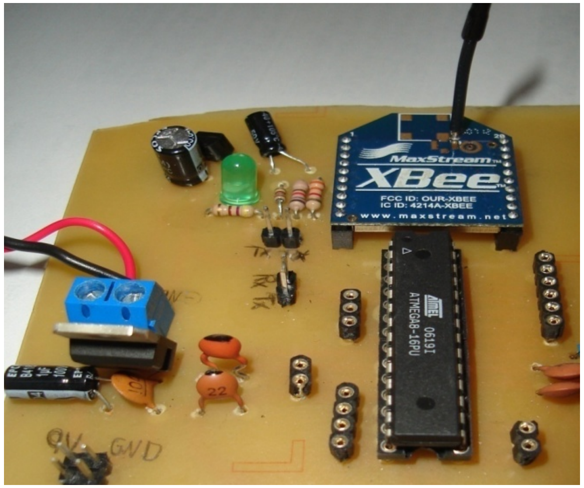
Sensors attached to the wall - Temperature, Humidity and Thermal Radiation

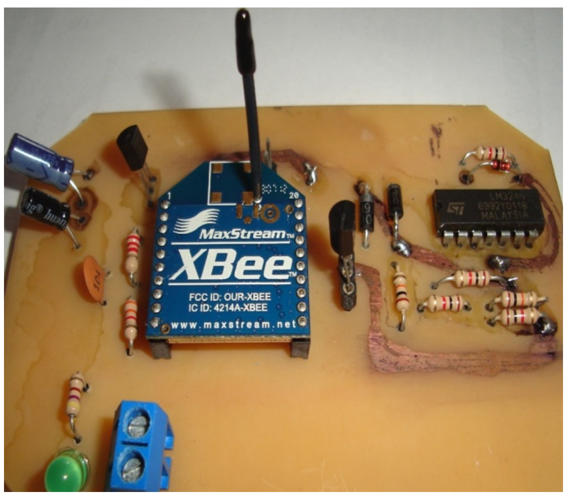
Fuzzy Control in Wireless Network

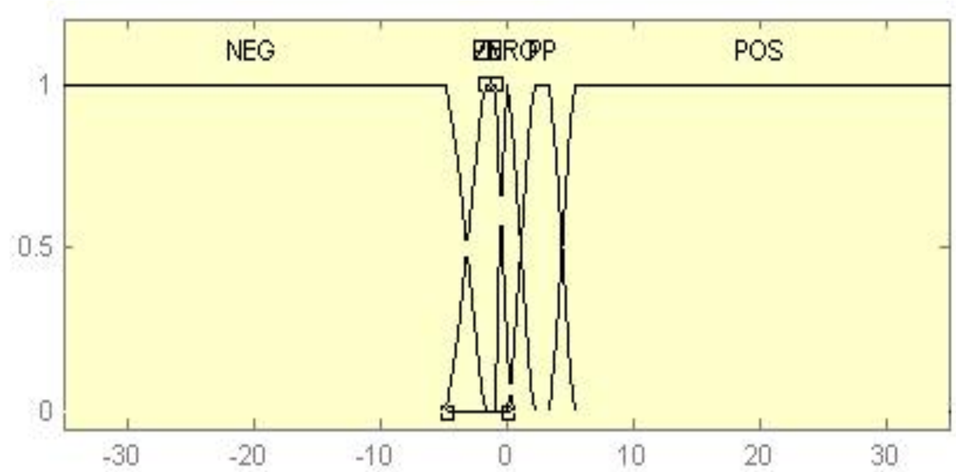




Fuzzy Control in Wireless Network





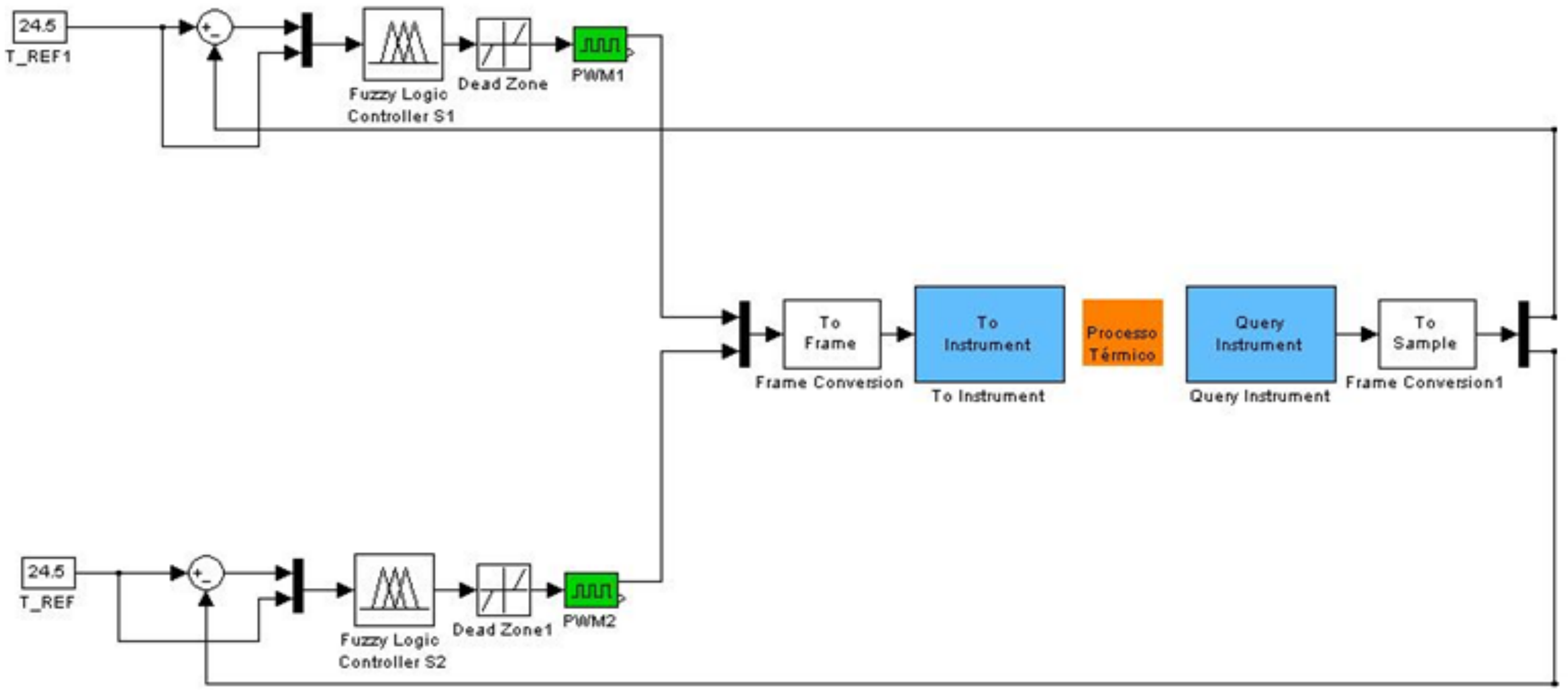


Membership functions of the input variable error

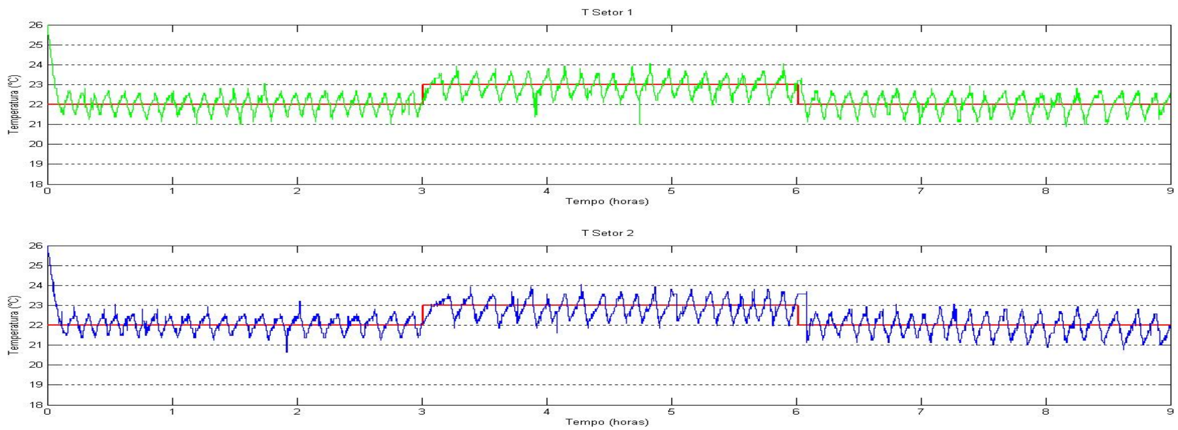
Truth Table of Fuzzy inference-LAVSI/ENE/UnB

Erro x T_ref	MB	B	M	A	MA
NEG	A	A	A	A	A
PN	M	M	M	M	M
ZERO	M	M	M	M	M
PP	MB	MB	MB	MB	MB
POS	MB	MB	MB	MB	MB

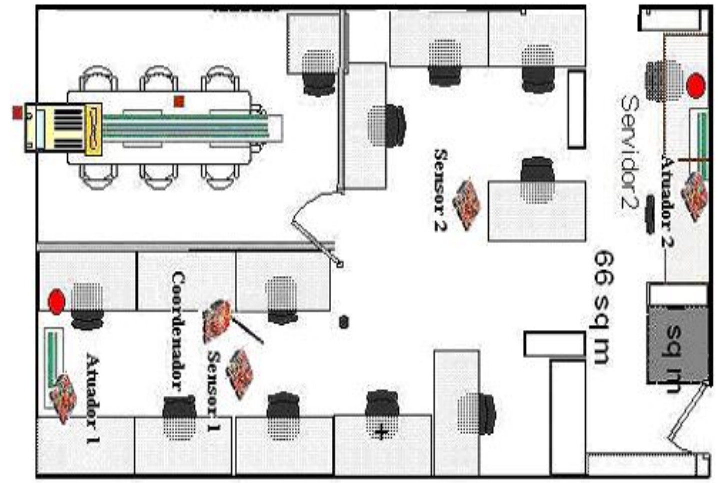
Fuzzy Control in Wireless Network

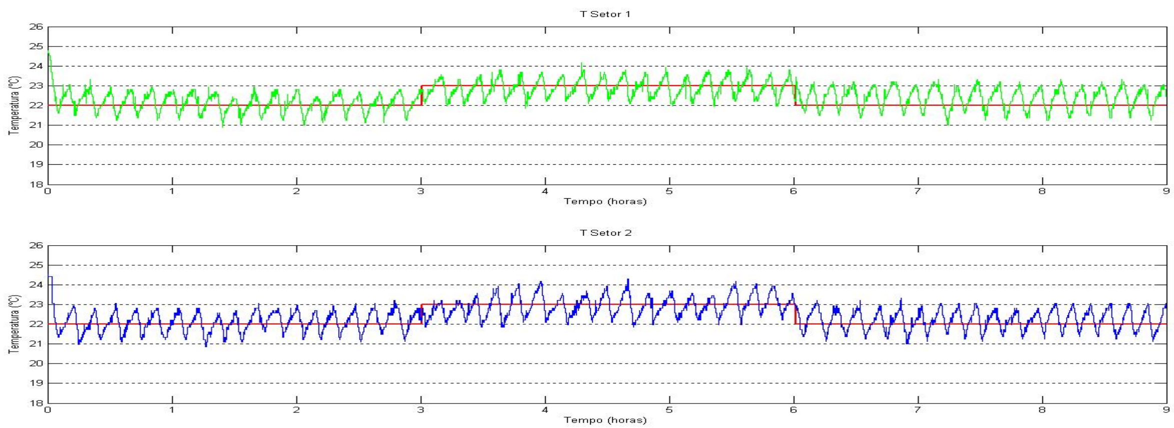


Fuzzy Control in Wireless Network



Temperaturas no setor 1 e setor 2 – Controle On-Off



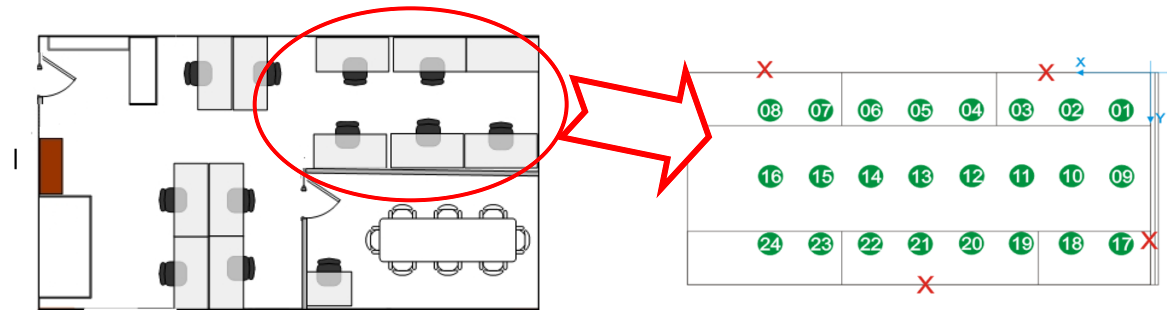


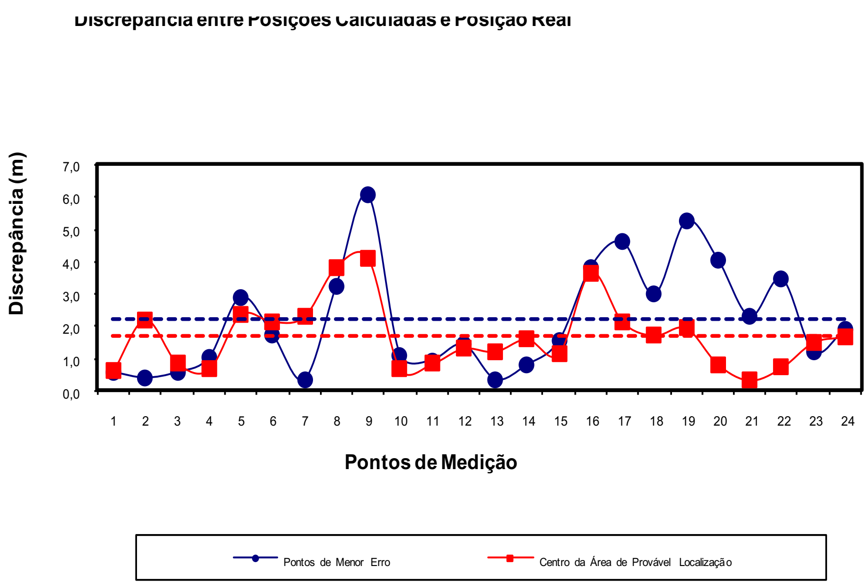
Temperaturas no setor 1 e setor 2 – Controle *Fuzzy*

Energy saving: On-Off x Fuzzy Wireless,
Ferreira Júnior, 2009.

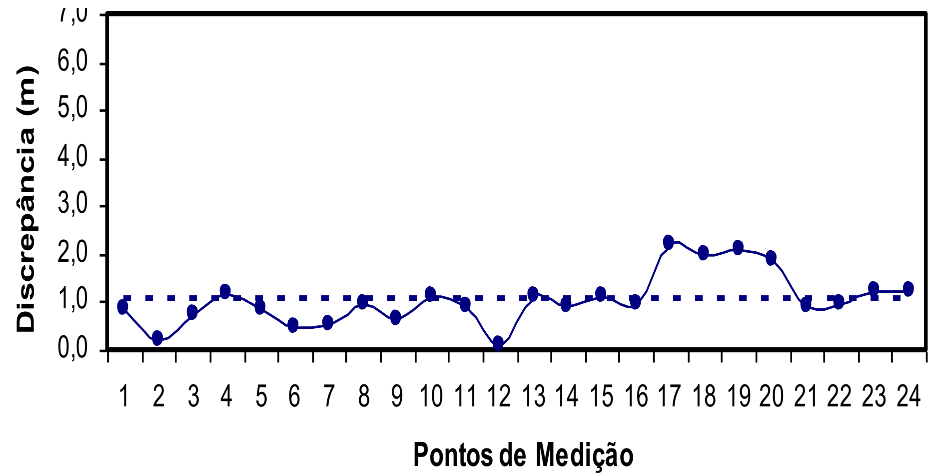
Controller	Energy (kWh)	Energy saving
On-off	15,69	17,00 %
Fuzzy	13,41	

Mobile Measurement of Thermal Comfort



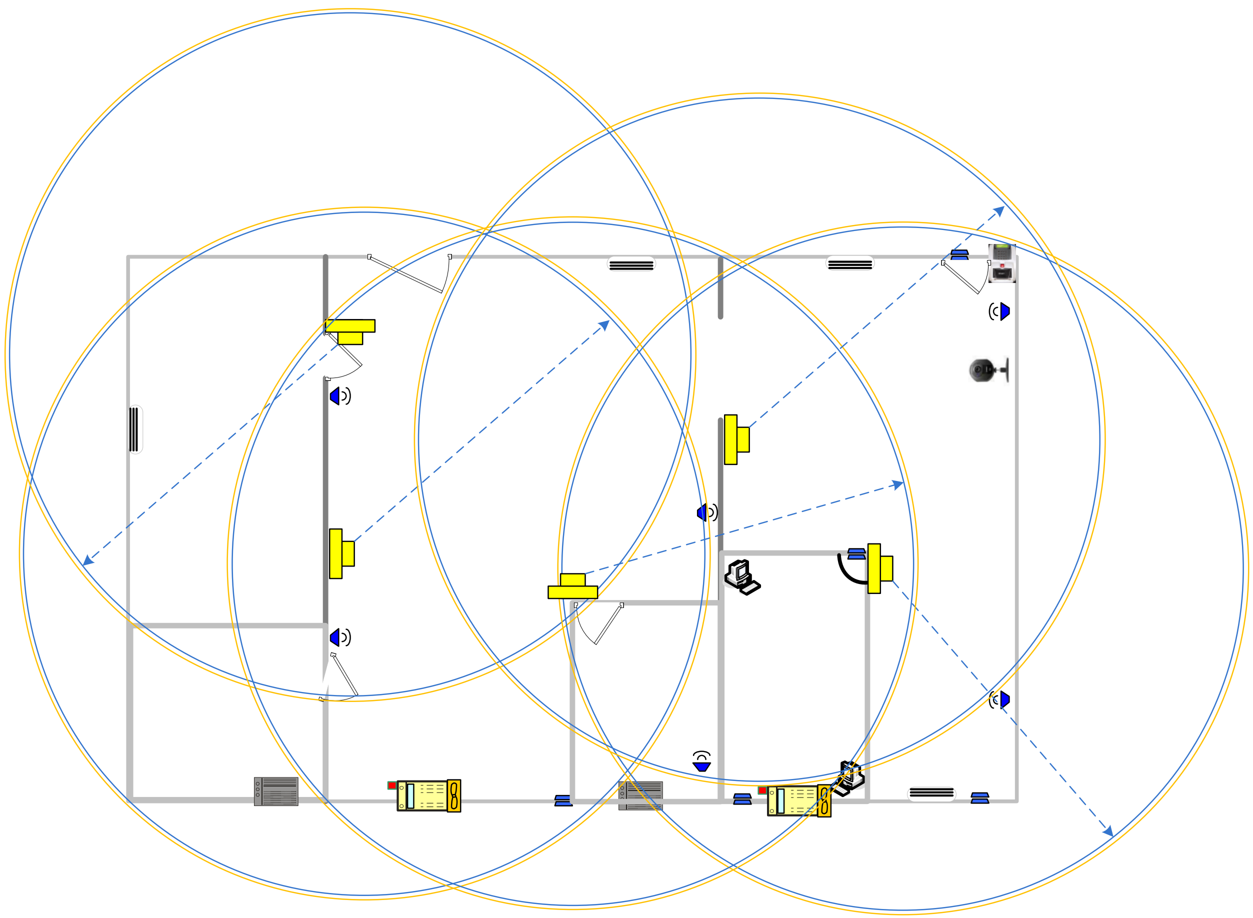


Hyperbolic triangulation

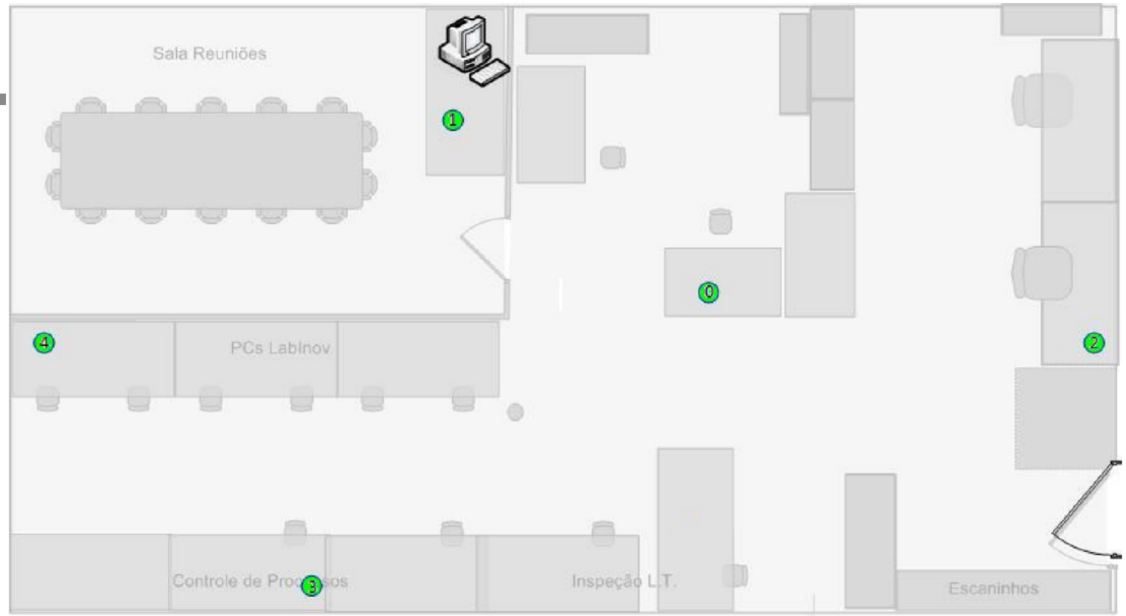


RNA

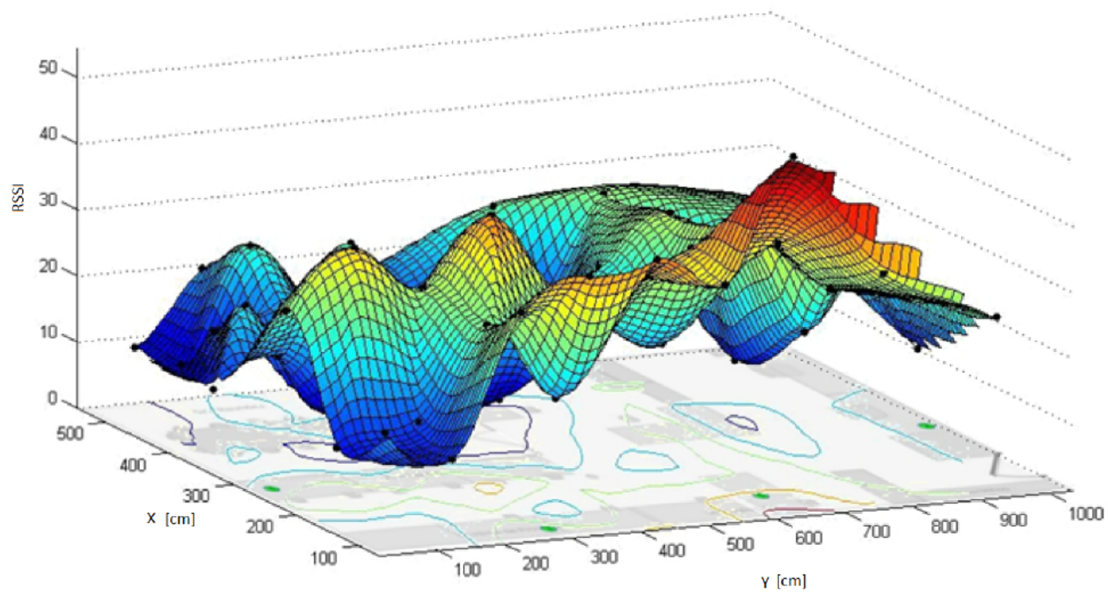
RFID occupancy identification (GPS indoor) for thermal load estimation



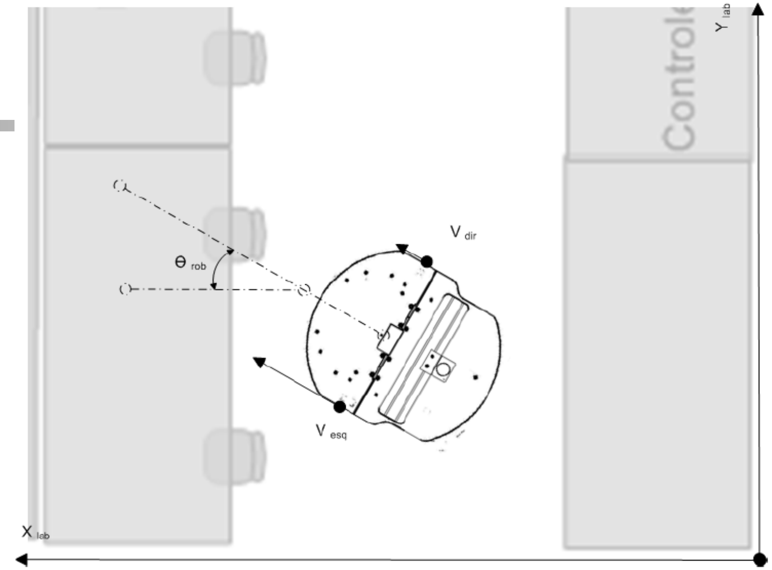
Indoor RFID Localization in the Context of Mobile Robotics with Application in Ambient Intelligence



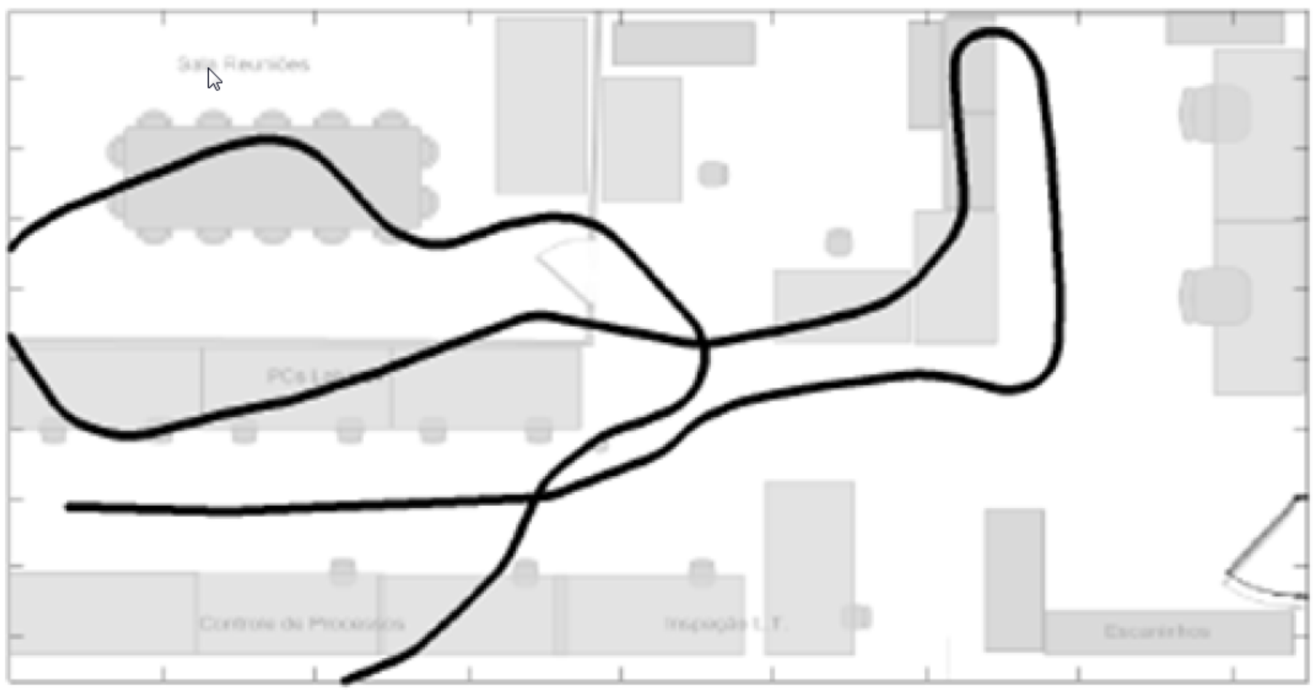
Interpolated from measured RSSI



Indoor RFID Localization in the Context of Mobile Robotics with Application in Ambient Intelligence

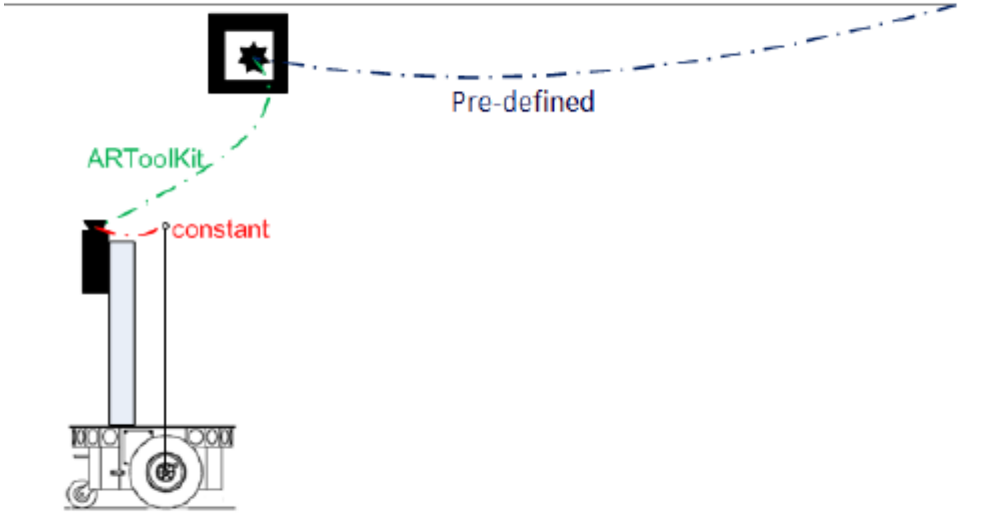


Localization results using encoders information in UKF without any update step

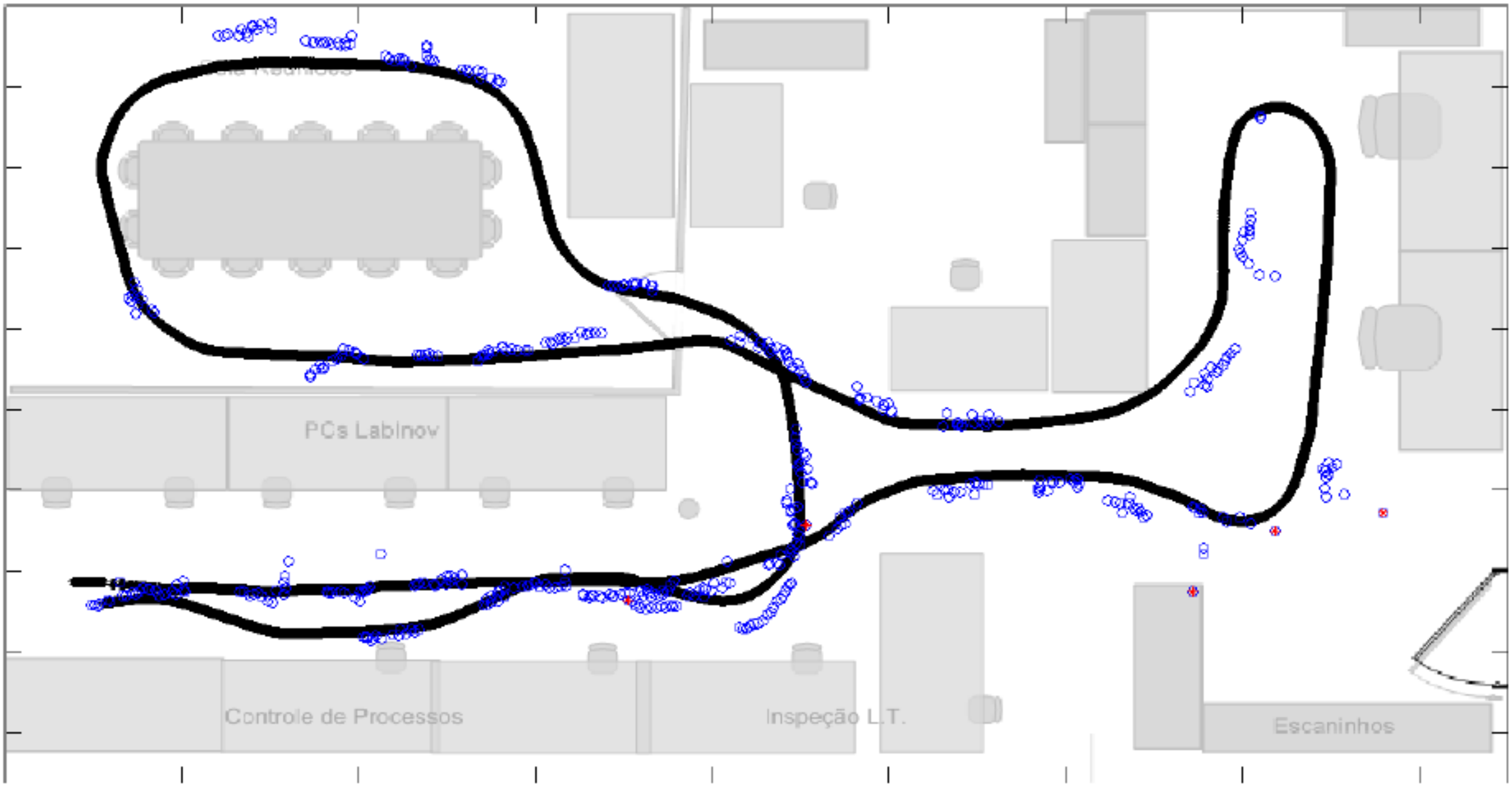


Indoor RFID Localization in the Context of Mobile Robotics with Application in Ambient Intelligence

Augmented Reality Localization System Results

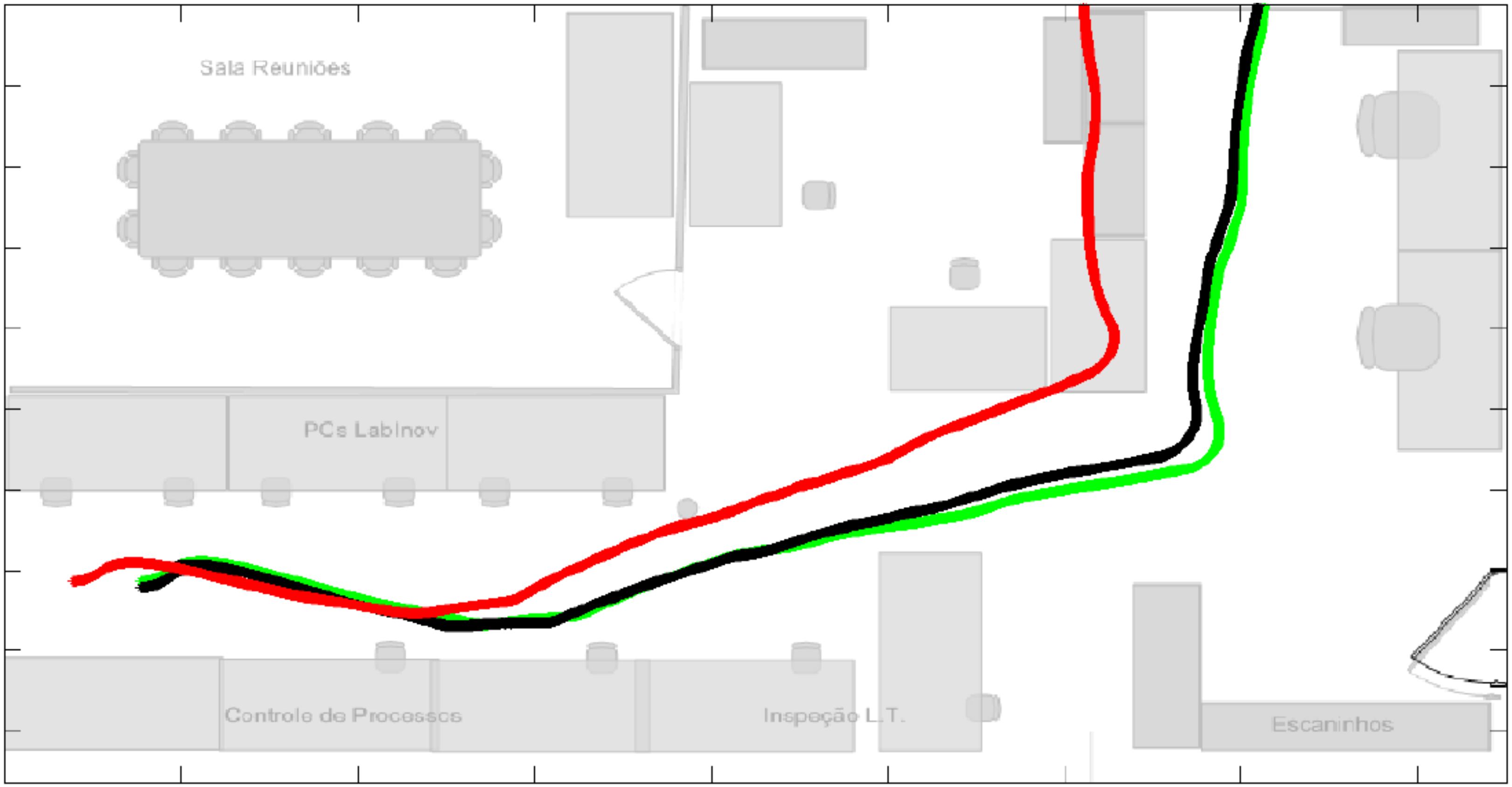


Indoor RFID Localization in the Context of Mobile Robotics with Application in Ambient Intelligence

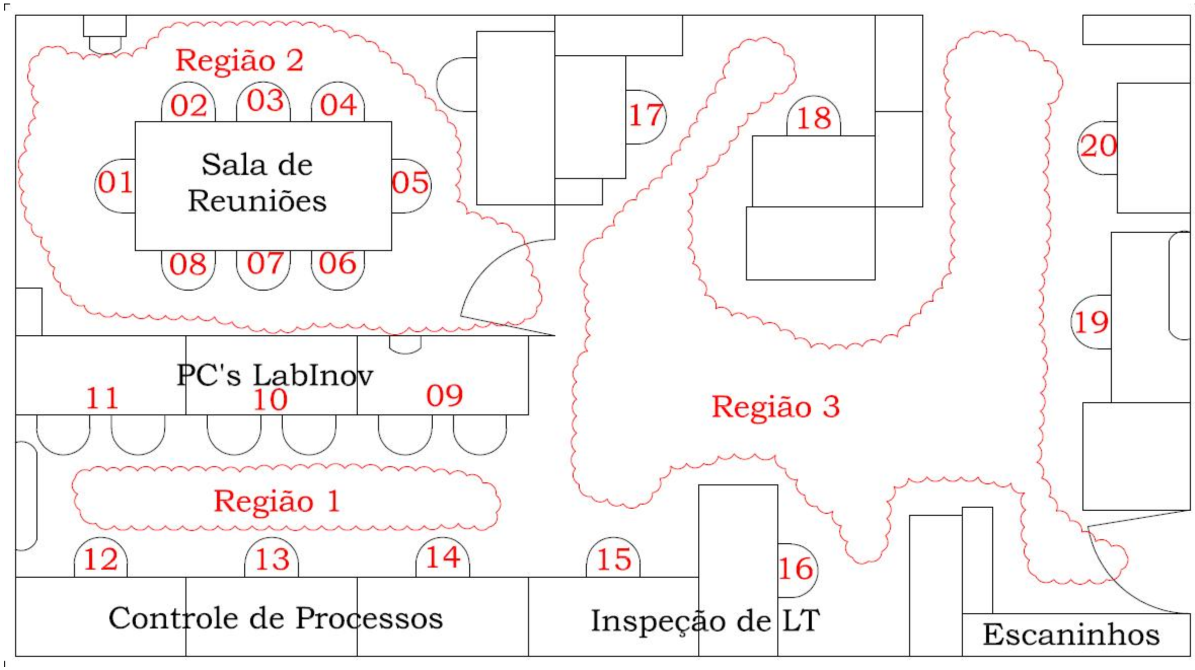


Augmented Reality Localization System Results

Comparison (Red – odo., Black – odo+vision, Green – all 3)



comparative results of the Augmented Reality-RFID RSSI system, Augmented reality system and pure odometry system



Thermal Load influence Areas

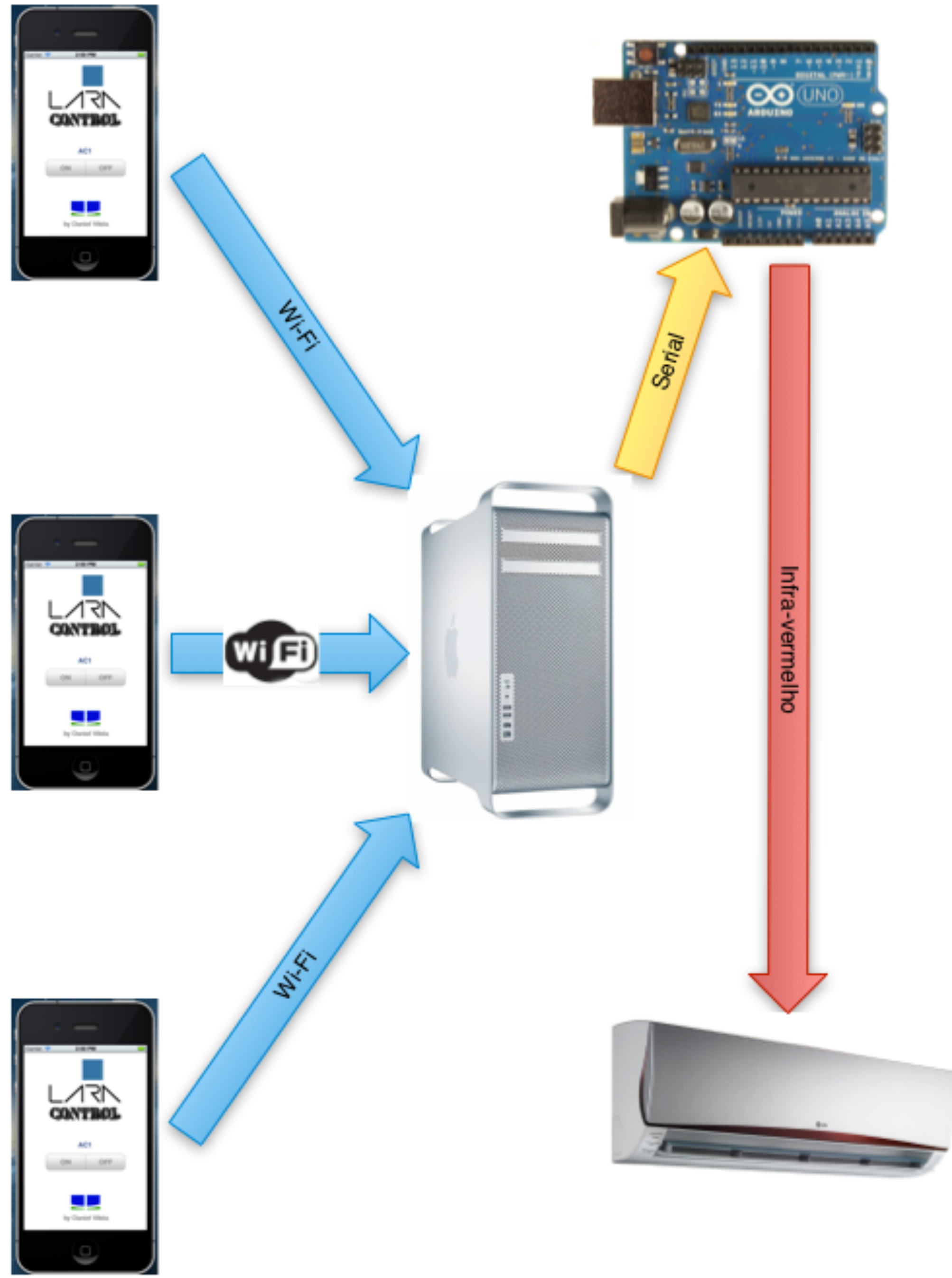
Identification of users in areas by RFID – RSSI classificators

(Cristovam Silva Jr., 2012)

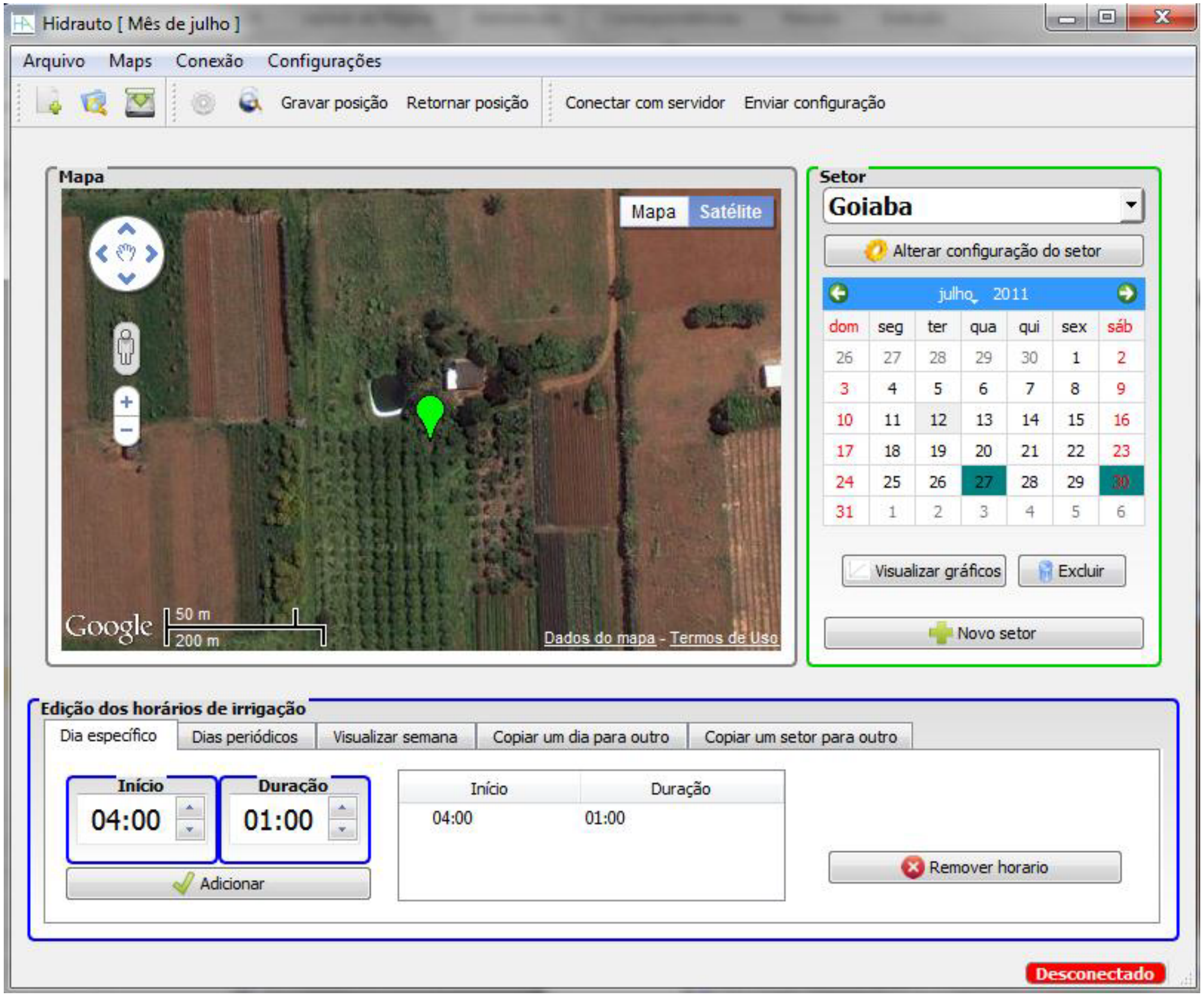
Building Automation

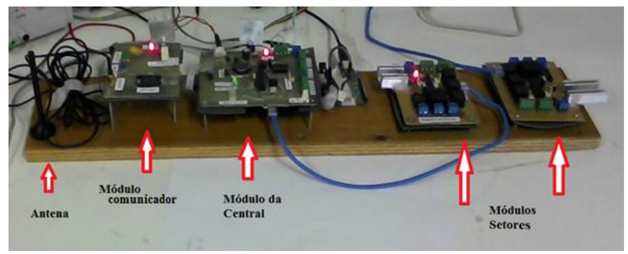
iPhone
WiFi
Arduino
ZigBee
Infra-Red
Air Conditioner

(Daniel Vilela, 2012)



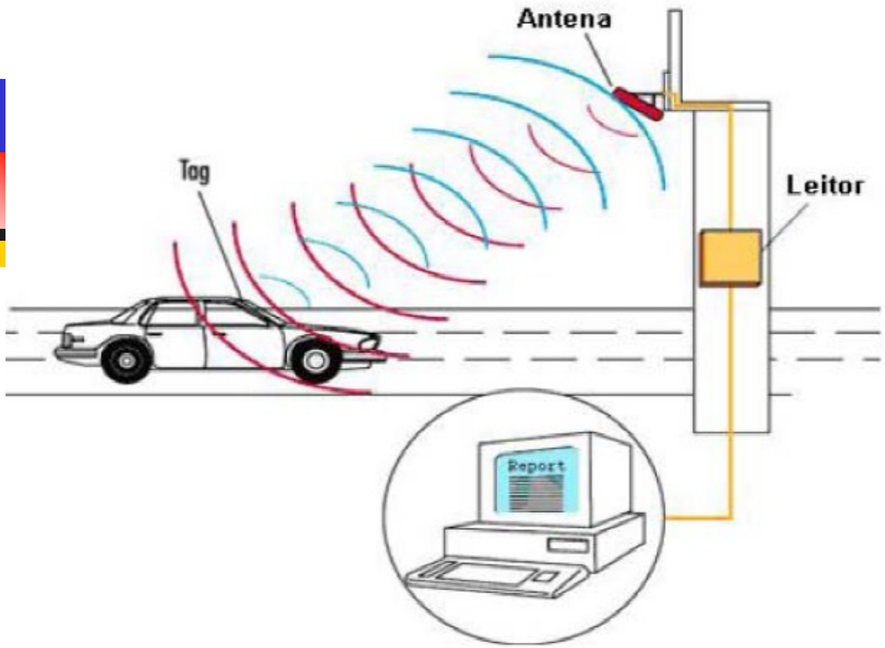
Automação e Monitoramento Remoto de Sistemas de Irrigação Visando Agricultura Familiar

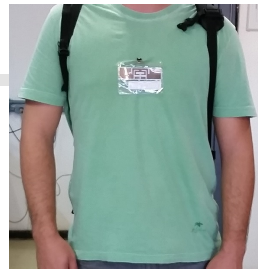


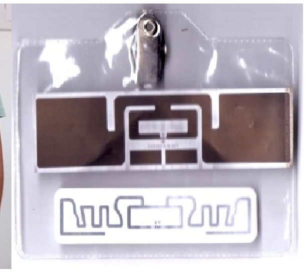


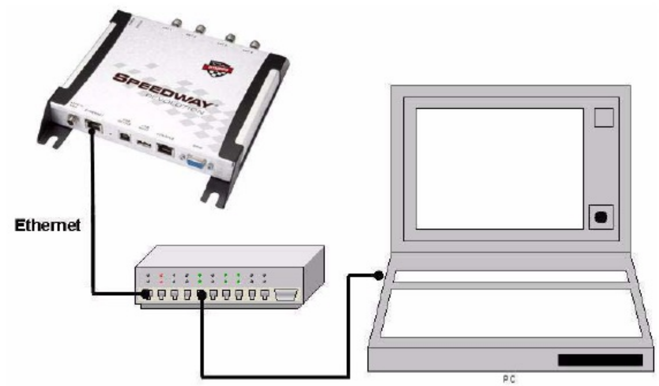
(Vinícius Guimarães, 2011)

RFID – Radio Frequency IDentification

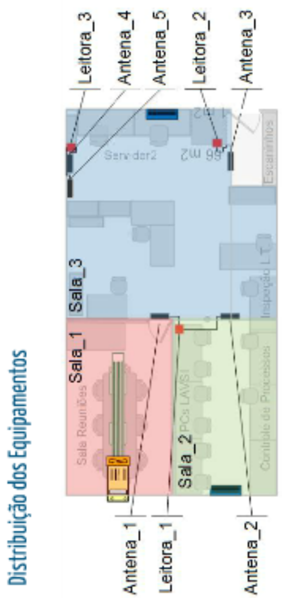


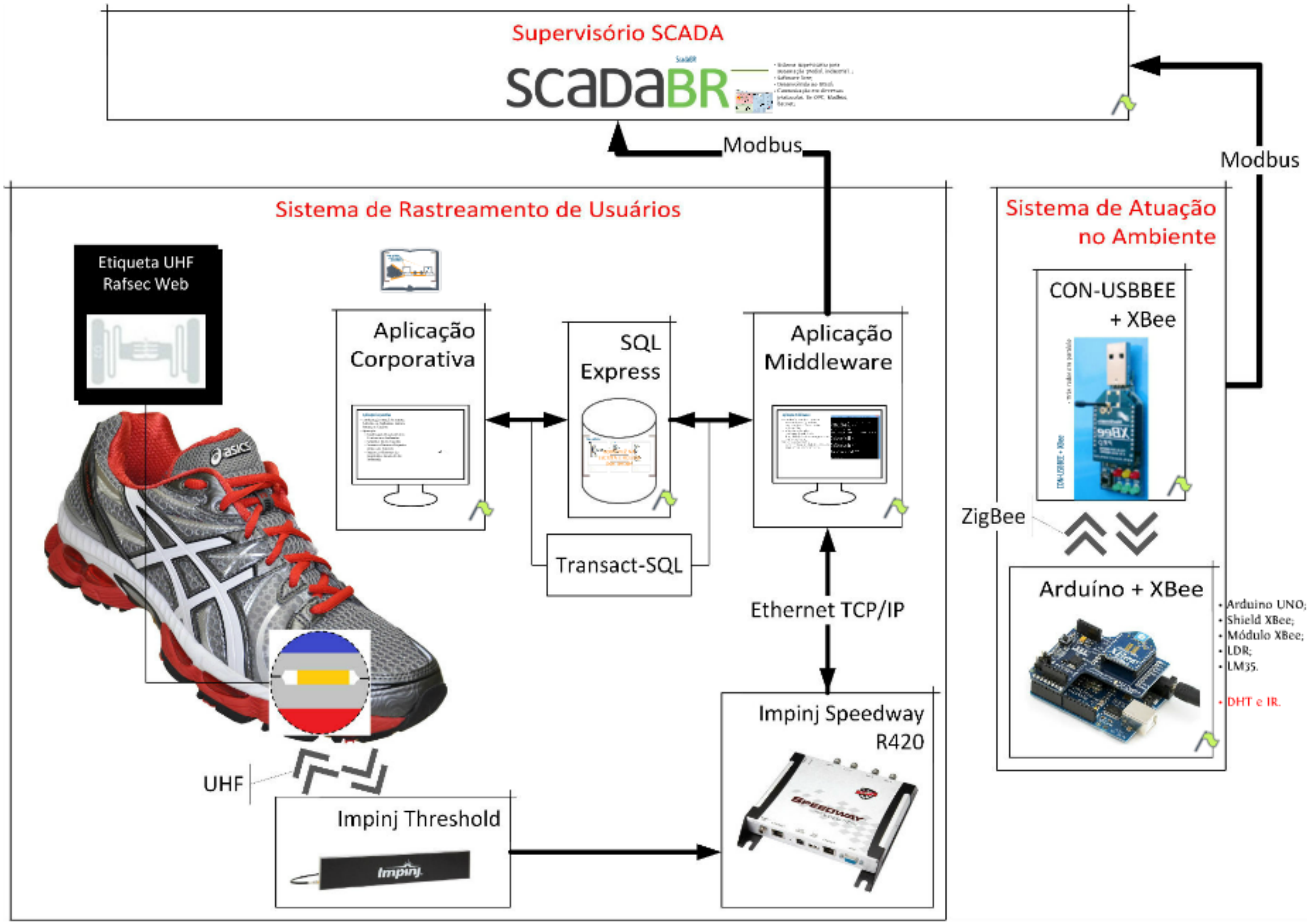




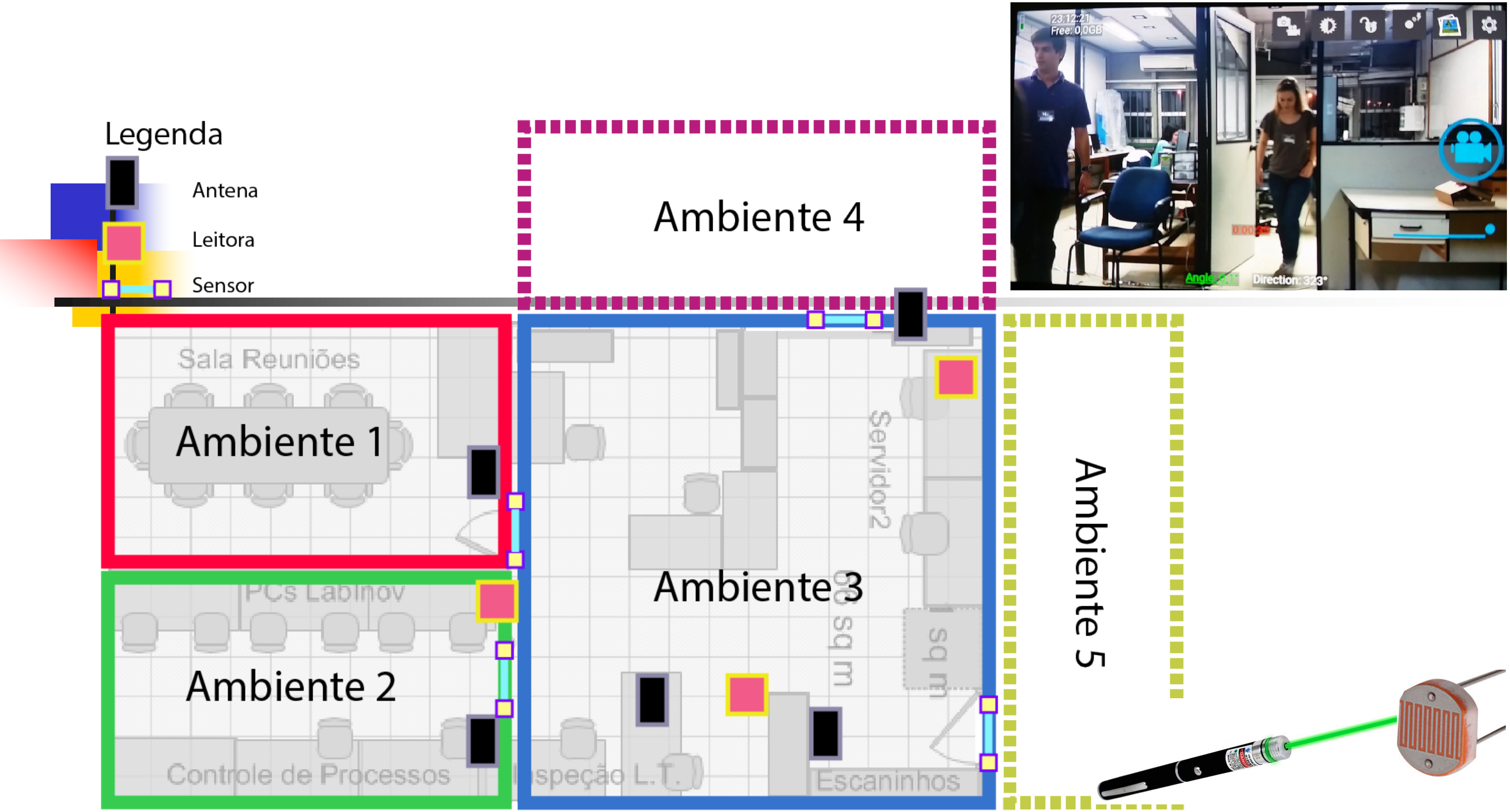


Passive RFID to track users in building automation (Frederico Rocha e Filipe Oliveira, 2013)

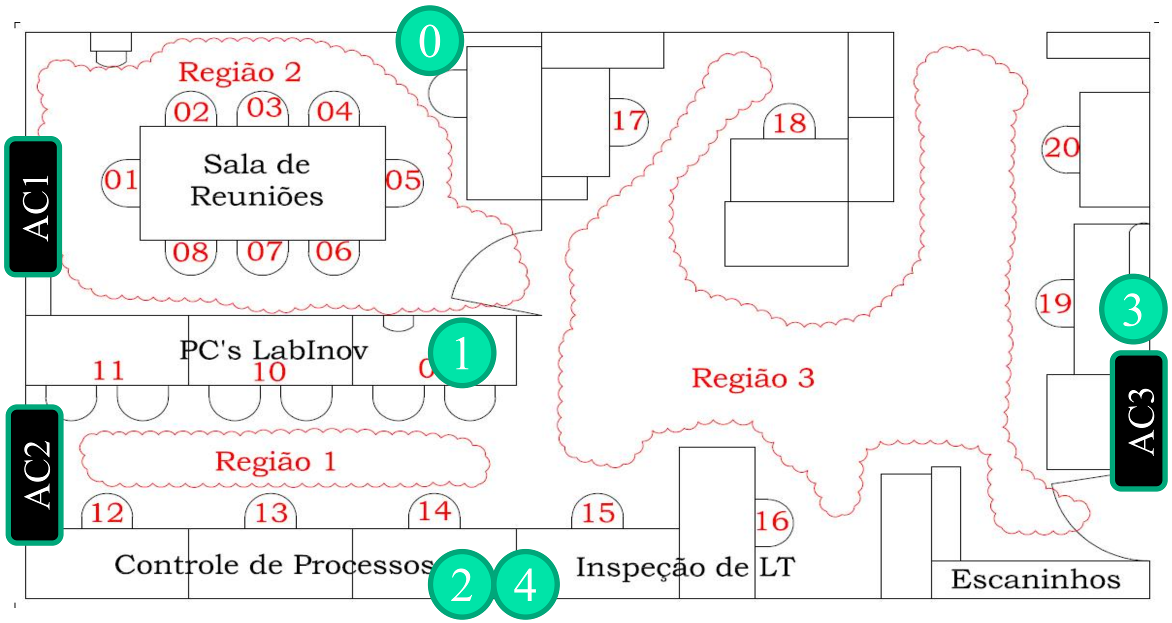




Occupancy by Passive RFID + Laser Beam



RFID occupation estimation – AC zones



	a)	b)	c)
MLP10	72,2%	76,9%	**99,3%**
MLP50	74,5%	78,9%	**99,3%**
MLP100	74,8%	78,3%	**99,2%**
MLP150	74,0%	77,9%	**99,3%**
MLP200	73,8%	76,9%	**99,1%**
LVQ10	65,5%	73,5%	**97,3%**
LVQ100	65,4%	74,8%	**96,3%**
LVQ200	65,4%	74,6%	**96,1%**
SVM - RBF	76,3%	82,1%	**97,6%**
SVM – Lin.	**44,6%**	**73,1%**	**92,5%**

LARA/SG11/UnB

01-20 Data collecting points

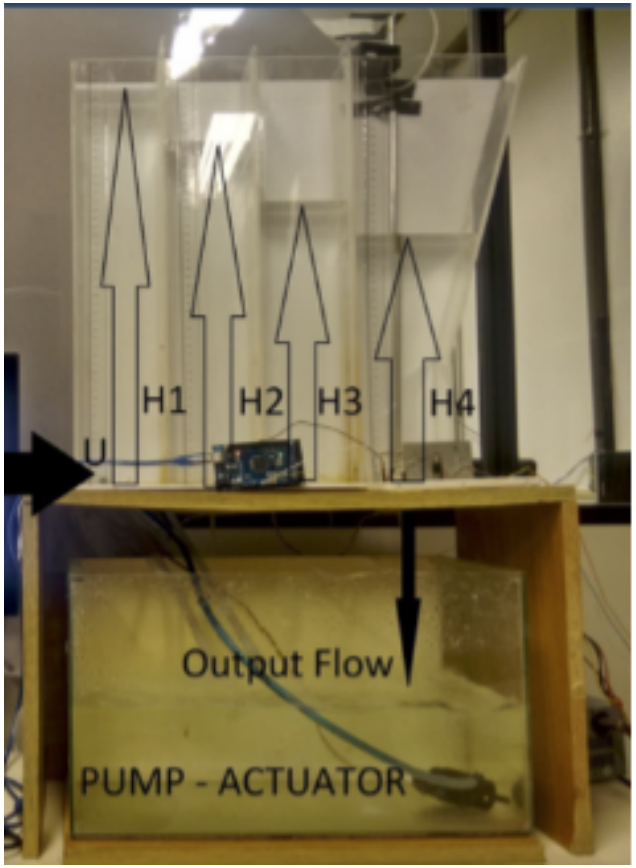
RFID Reader

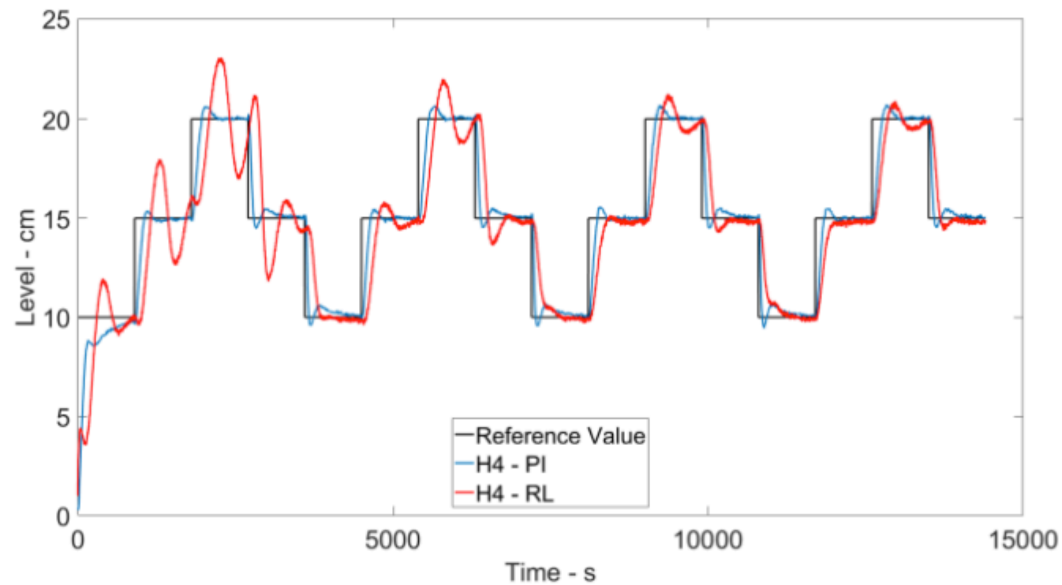
AC Air Conditioner

a) Raw RSSI (missing RSSI ->0)
b) RSSI hold last valid RSSI
c) EKF

(Cristovam Silva Jr., 2012)

RL: 4th order liquid leve Process

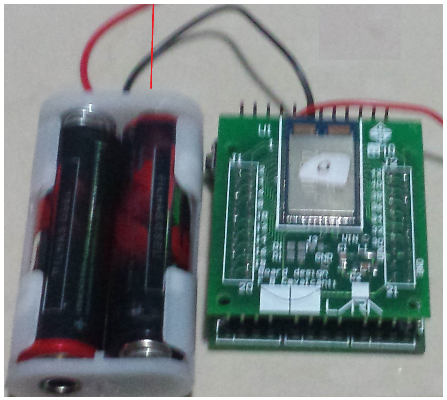




Actor-Critic Q-Learning

LARA/UnB, Lucas Matos, 2018

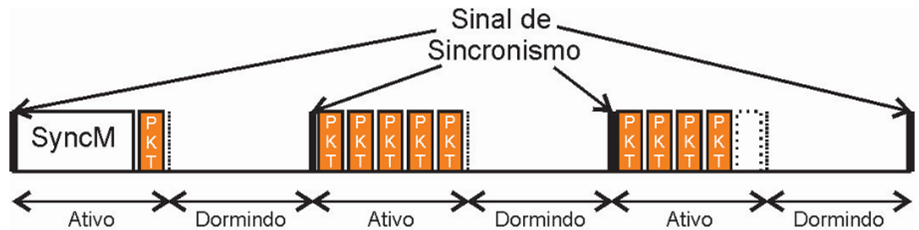
DyTEE MAC/UnB – Dynamic Timed Energy Efficient

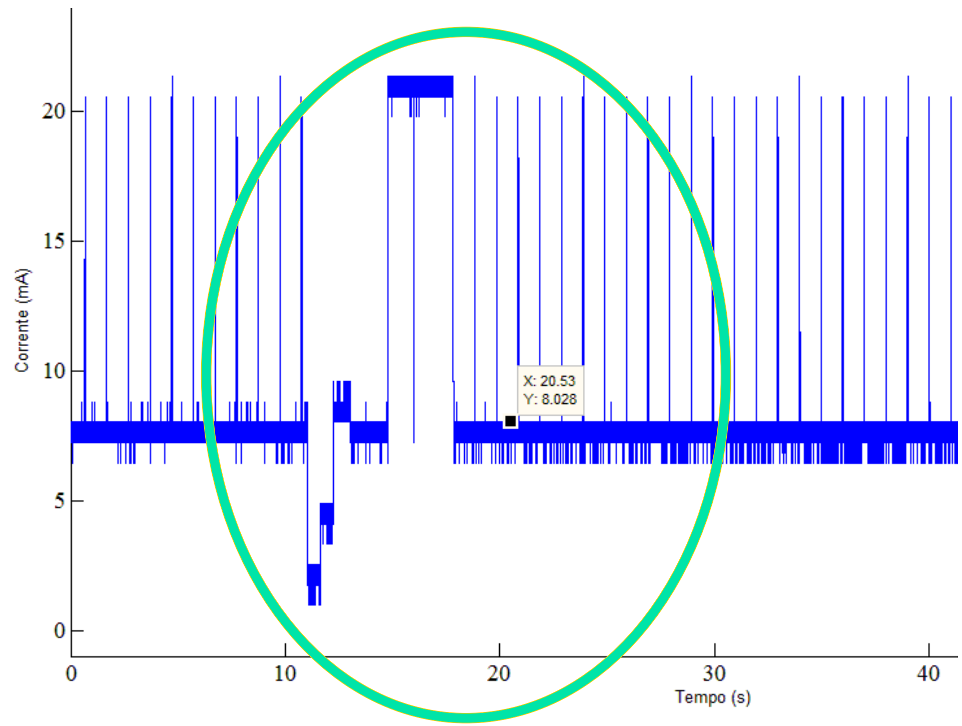


Sensor Node

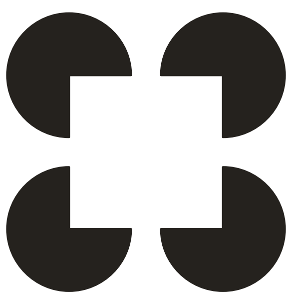
DyTEE

IEEE 802.15.4

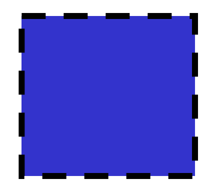




Part 5 – Conclusions



- RNA - A technique that involves learning
- Fuzzy – Demands a Human Expert
- Neuro-Fuzzy - ANFIS
- Commercial products available



Philosophical origins

René Descartes

“analytic geometry”

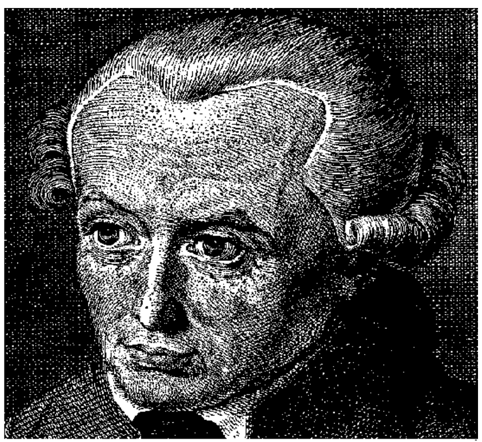
1637

Reason

Immanuel Kant

“Critique of Pure Reason”

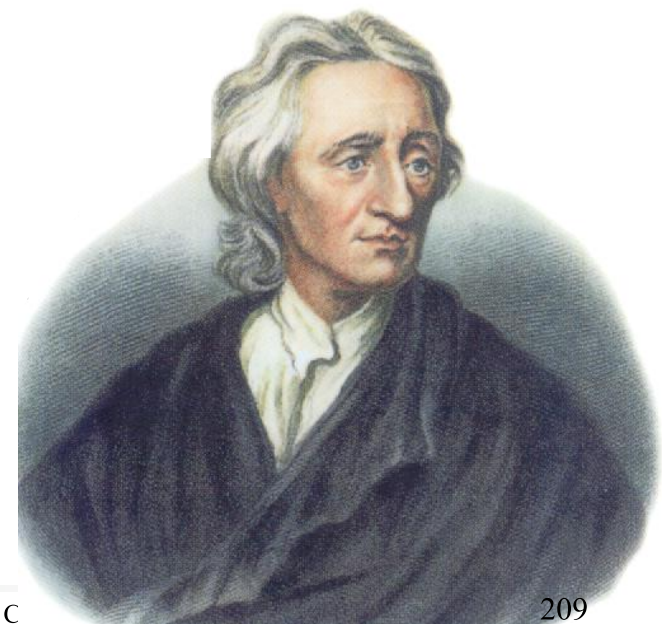
1781



Senses

John Locke

“Essay on Human Understanding”
1689



Philosophical origins

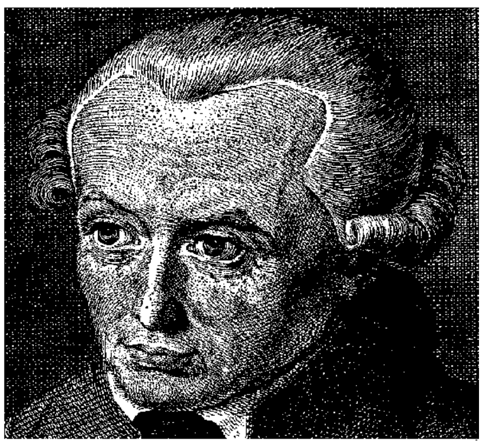
René Descartes

"I think, therefore I am"

Rationalism

Immanuel Kant

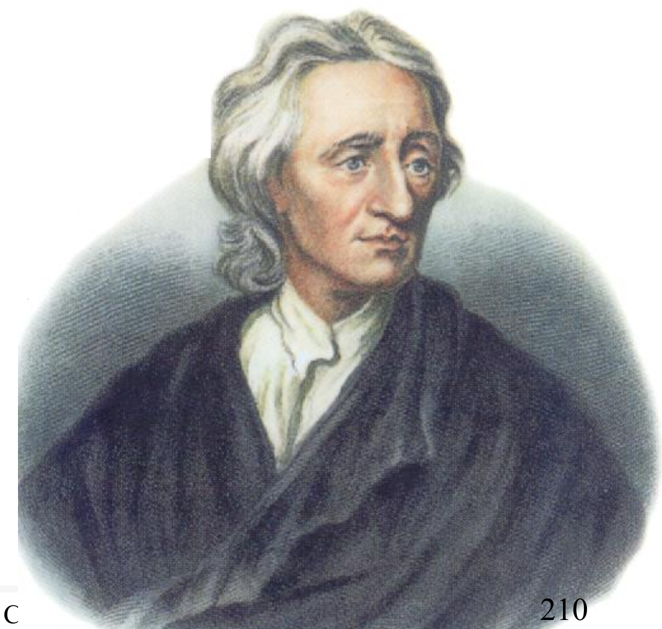
"We can only know what we perceive"



Empiricism

John Locke

"The knowing of no man can go beyond his experience"



Conclusion

Conventional engineering	→	Rationalism
Knowledge Engineering (RNA , Fuzzy , GA, ...)	→	Empiricism

R + E ~> SI

To be able to design intelligent systems that are really useful you must have a good theoretical background.

Normally, only what is already known to exist, will be found.

Prof. Adolfo Bauchpiess

Short C.V.

– SENAI/1982 – Eletricista de Dispositivos de Comandos Elétricos

– UnB/1986 - Eng. Elétrica

  – Estágios: Telebrasília (1984), Prólogo (1985), Novadata (1986)

- Engenheiro: Novadata Sistemas e Computadores Ltda (1986-1990)

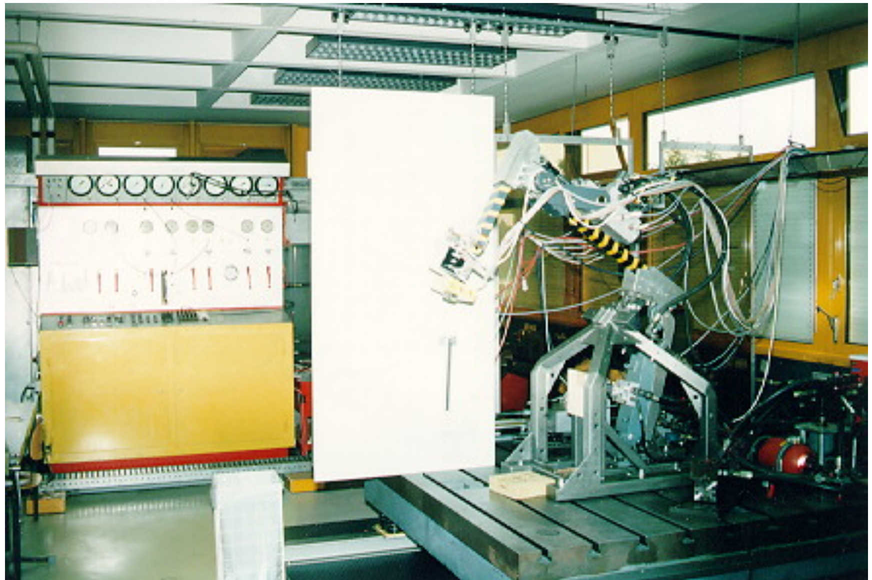
– UnB/1990 - Mestre Eng. Elétrica

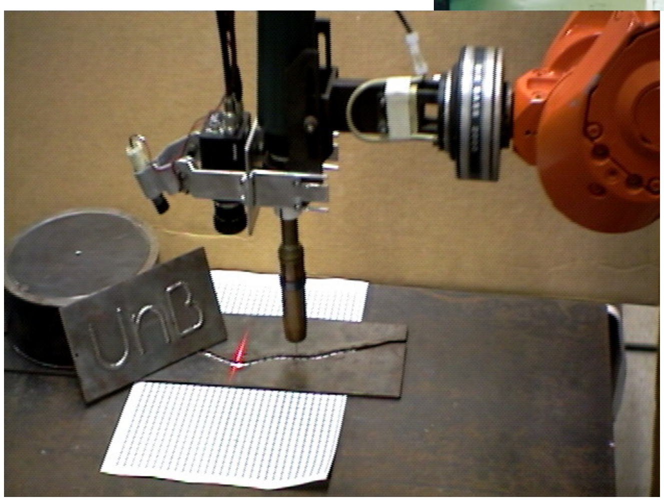
– Erlangen/1995 – Dr.-Ing.

– ENE/UnB 1995 ... atual - Prof. área de Controle

  - Pós-Doc (Aachen/1997, Kaiserslautern/2005-6, Santa Barbara/2014)
  - Projetos: FAP-DF, CNPq, CAPES, FINEP
  - Erasmus Mundus – Kaiserslautern

Projetos…





(1995-2002)

Thank You!

Adolfo Bauchspiess

lara.unb.br/~bauchspiess

bauchspiess@lara.unb.br