Advanced Topics in Computational Intelligence

Master AI

FUZZY INDUCTIVE REASONING (FIR) METHODOLOGY

Àngela Nebot

Jump to first page

Fuzzy Inductive Reasoning

A qualitative non-parametric inductive methodology for systems and signal modeling

Inductive Modeling Techniques

- Making Models from Observations of Input/Output Behavior
- Understanding Systems
- Forecasting/Classifying Systems Behavior
- Controlling Systems Behavior

Observation-based Modeling and Complexity

- ***Observation-based modeling*** is very important, especially when dealing with ***unknown*** or only ***partially understood systems***. Whenever we deal with new topics, we really have no choice, but to model them ***inductively***, i.e., by using available observations.

- The less we know about a system, the more general a modeling technique we must embrace, in order to allow for all eventualities. If we know nothing, we must be prepared for anything.

- In order to model a totally unknown system, we must thus allow a ***model structure*** that can be ***arbitrarily complex***.

Parametric *vs.* Non-parametric Models

- ***ANNs*** are ***parametric models***. The observed knowledge about the system under study is mapped on the (potentially very large) set of parameters of the ***ANN***.

- Once the ***ANN*** has been ***trained***, the original knowledge is no longer used. Instead, the learnt behavior of the ***ANN*** is used to make predictions.

- This can be dangerous. If the ***testing data***, i.e. the input patterns during the use of the already trained ***ANN***, differ significantly from the ***training data*** set, the ***ANN*** is likely to predict garbage, but since the original knowledge is no longer in use, is unlikely to be aware of this problem.

Parametric *vs.* Non-parametric Models

- ***Non-parametric models***, on the other hand, always refer back to the original training data, and therefore, can be made to reject testing data that are incompatible with the training data set.
- The ***Fuzzy Inductive Reasoning (FIR)*** engine that we shall discuss in this lecture, is of the ***non-parametric*** type.
- During the ***training*** phase, ***FIR*** organizes the observed patterns, and places them in a ***data base***.
- During the ***testing*** phase, ***FIR*** searches the data base for the most similar training data patterns. ***FIR*** then uses these data patterns to predict the new output.

Jump to first page

Quantitative *vs.* Qualitative Models

- ***Training a model*** (be it parametric or non-parametric) means ***solving an optimization problem***.
- In the ***parametric*** case, we have to solve a ***parameter identification*** problem.
- In the ***non-parametric*** case, we need to ***classify the training data***, and store them in an optimal fashion in the data base.
- Training such a model can be excruciatingly slow.
- Hence it may make sense to devise techniques that will help to speed up the training process.

Jump to first page

Quantitative *vs.* Qualitative Models

- How can the speed of the optimization be controlled? Somehow, the ***search space*** needs to be reduced.
- One way to accomplish this is to convert ***continuous variables*** to equivalent ***discrete variables*** prior to optimization.
- For example, if one of the variables to be looked at is the ambient temperature, we may consider to classify temperature values on a spectrum from ***very cold*** to ***extremely hot*** as one of the following discrete set:

***temperature* = { *freezing, cold, cool, moderate, warm, hot* }**

Qualitative Variables

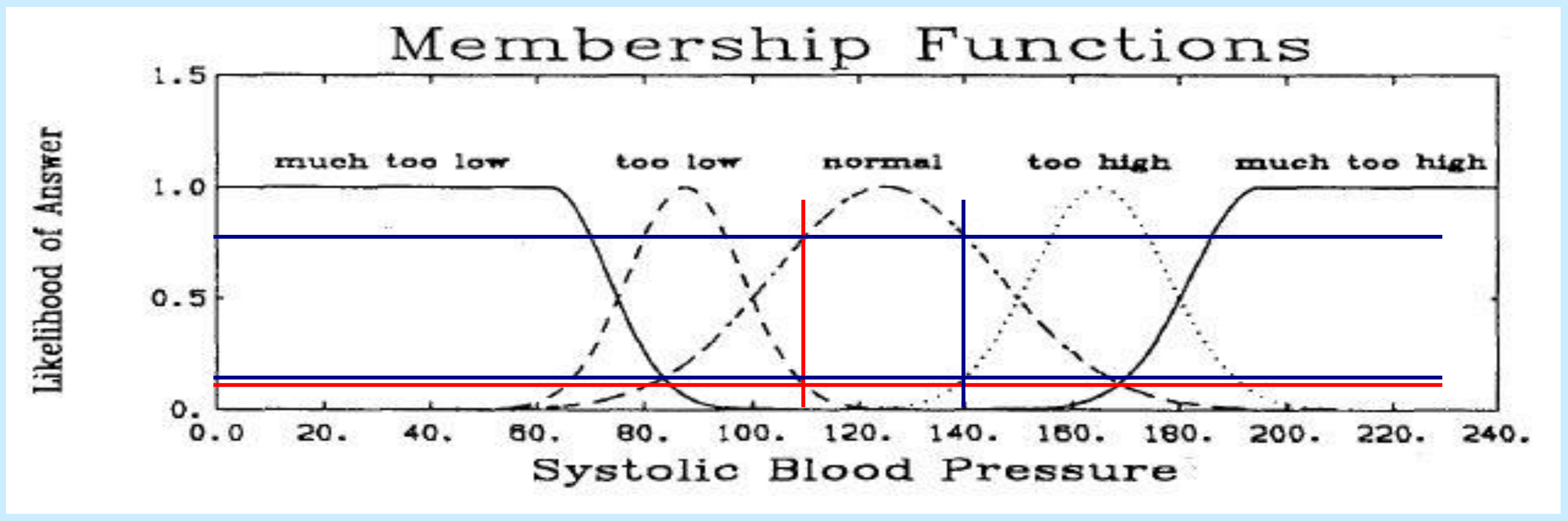
- A variable that only assumes one among a set of discrete values is called a ***discrete variable***. Sometimes, it is also called a ***qualitative variable***.

- Evidently, it must be cheaper to search through a ***discrete search space*** than through a ***continuous search space***.

- The problem with ***discretization schemes***, such as the one proposed above, is that a lot of potentially valuable detailed information is being lost in the process.

- To avoid this pitfall, the ***fuzzification*** approach (***L. Zadeh***) can be used.

Fuzzy Variables



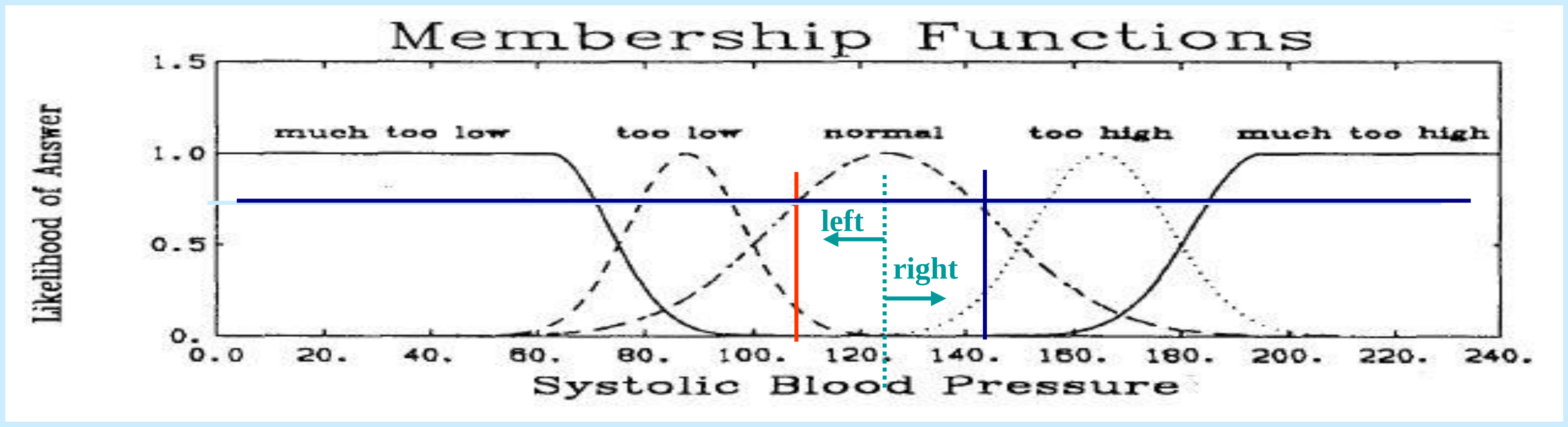
Systolic blood pressure = 110
⇒ { normal, 0.78 } ∩ { too low, 0.15 }

Systolic blood pressure = 141
⇒ { normal, 0.78 } ∩ { too high, 0.18 }

{ Class, membership } pairs of ***lower membership*** must be considered as well, because otherwise, the mapping would not be unique.

Fuzzy Variables in FIR

FIR embraces a slightly different approach to solving the uniqueness problem. Rather than mapping into ***multiple fuzzy rules***, FIR only maps into a ***single rule***, that with the largest membership. However, to avoid the aforementioned ***ambiguity problem***, FIR stores one more piece of information, the "***side value***." It indicates, whether the data point is to the left or the right of the peak of the corresponding fuzzy membership value.



Systolic blood pressure = 110
⇒ { normal, 0.78, left }

Systolic blood pressure = 141
⇒ { normal, 0.78, right }

Neural Networks vs. Inductive Reasoners

Neural Networks	Fuzzy Inductive R.
Quantitative	Qualitative
Parametric	Non-parametric
Adaptive	Limited Adaptability
Slow Training	Fast Setup
Smooth Interpolation	Decent Interpolation
Wild Extrapolation	No Extrapolation
No Error Estimate	Error Estimate
Unsafe / Gullible	Robust / Self-critical

General System Theory (GST)

Definition

- **General System Theory** is a logico-mathematical field whose task is the formulation and derivation of those general principles that are applicable to ``systems´´ in general. (L. *von Bertalanffy, 1930*)

- **General System Theory** is a name which has come into use to describe a level of theoretical model-building which lies somewhere between the highly generalized constructions of pure mathematics and the specific theories of the specialized disciplines. (*Boulding, 1956*)

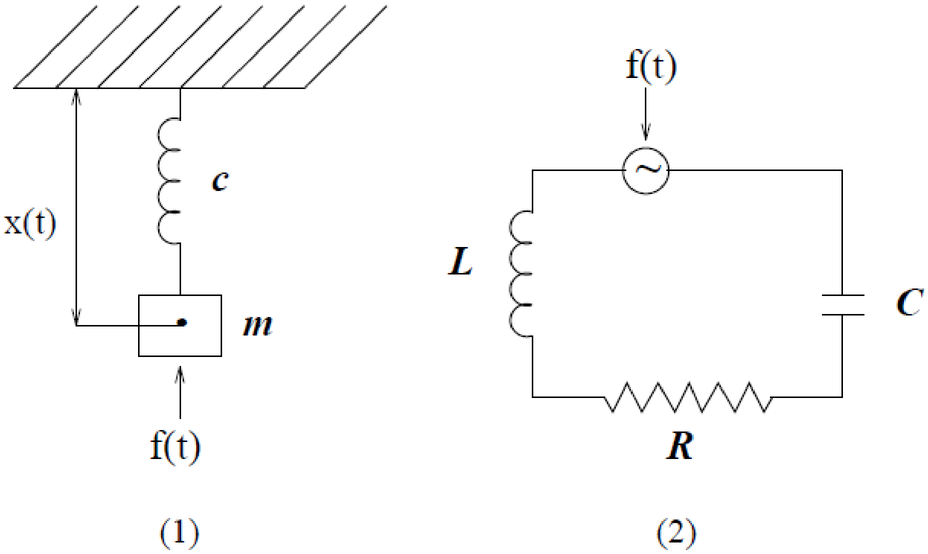
- ... Thus, there exist models, principles, and laws that apply to generalized systems or their subclasses, irrespective of their particular kind, the nature of their component elements, and the relations or ``forces´´ between them... In this way we postulate a discipline called **General System Theory**. (*von Bertalanffy*, *1930*)

Jump to first page

General System Theory (GST)

Isomorphism or structural similarities

m: mass
r: coef. of friction
c: elasticity spring



L: inductance
R: resistance
C: capacitance

FIGURE 3.1. (1) Mechanical Harmonic Oscillator and (2) Electrical Circuit

Both systems solutions are second-order differential equations:

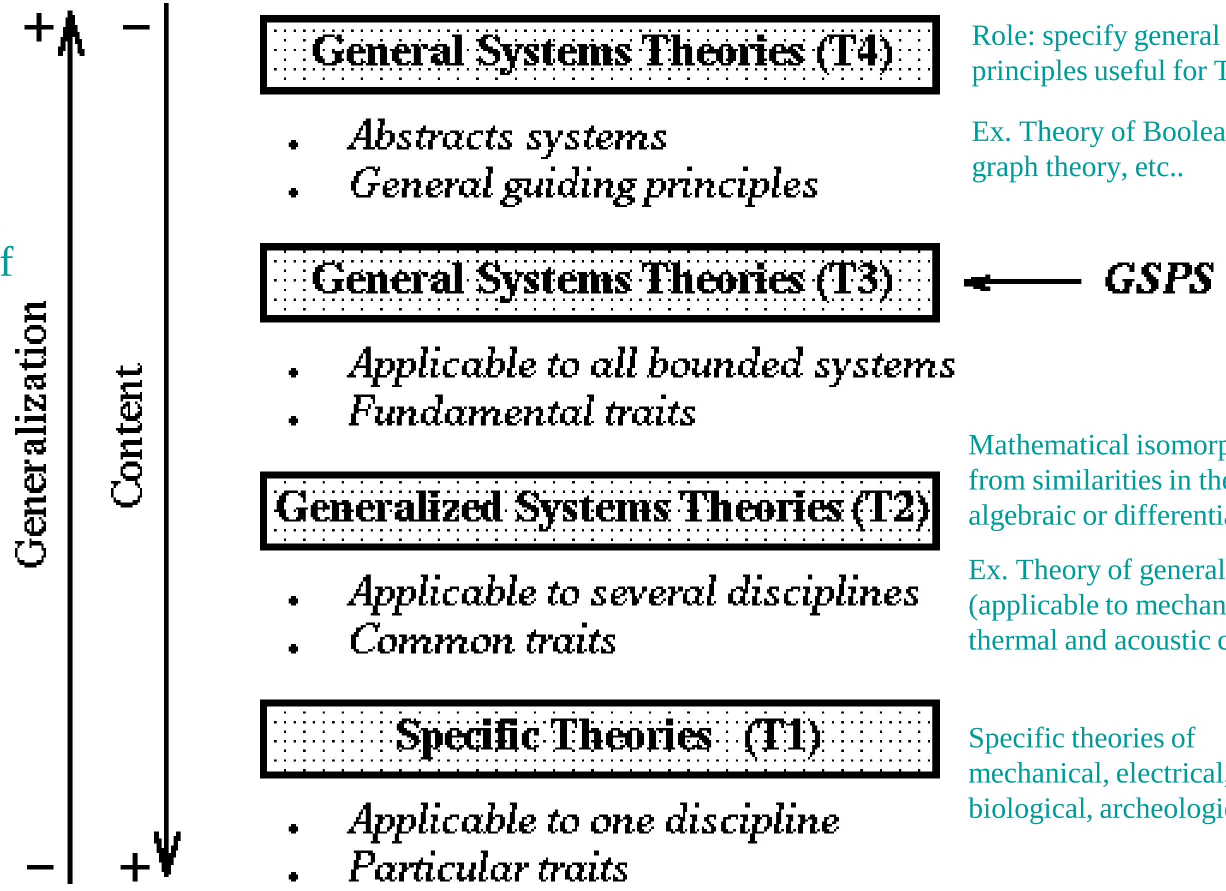
Position of the mass: $f(t) = m\ddot{x} + r\dot{x} + cx$

Quantity of charge: $f(t) = L\ddot{q} + R\dot{q} + C^{-1}q$

Jump to first page

General Systems Problem Solver (GSPS)

Def. GSPS: A conceptual framework through which types of systems problems are defined together with methodological tools for solving problems of these types.



General Systems Theories (T4)

- Abstracts systems
- General guiding principles

General Systems Theories (T3) ← *GSPS*

- Applicable to all bounded systems
- Fundamental traits

Generalized Systems Theories (T2)

- Applicable to several disciplines
- Common traits

Specific Theories (T1)

- Applicable to one discipline
- Particular traits

Role: specify general guiding principles useful for Type 3 level.

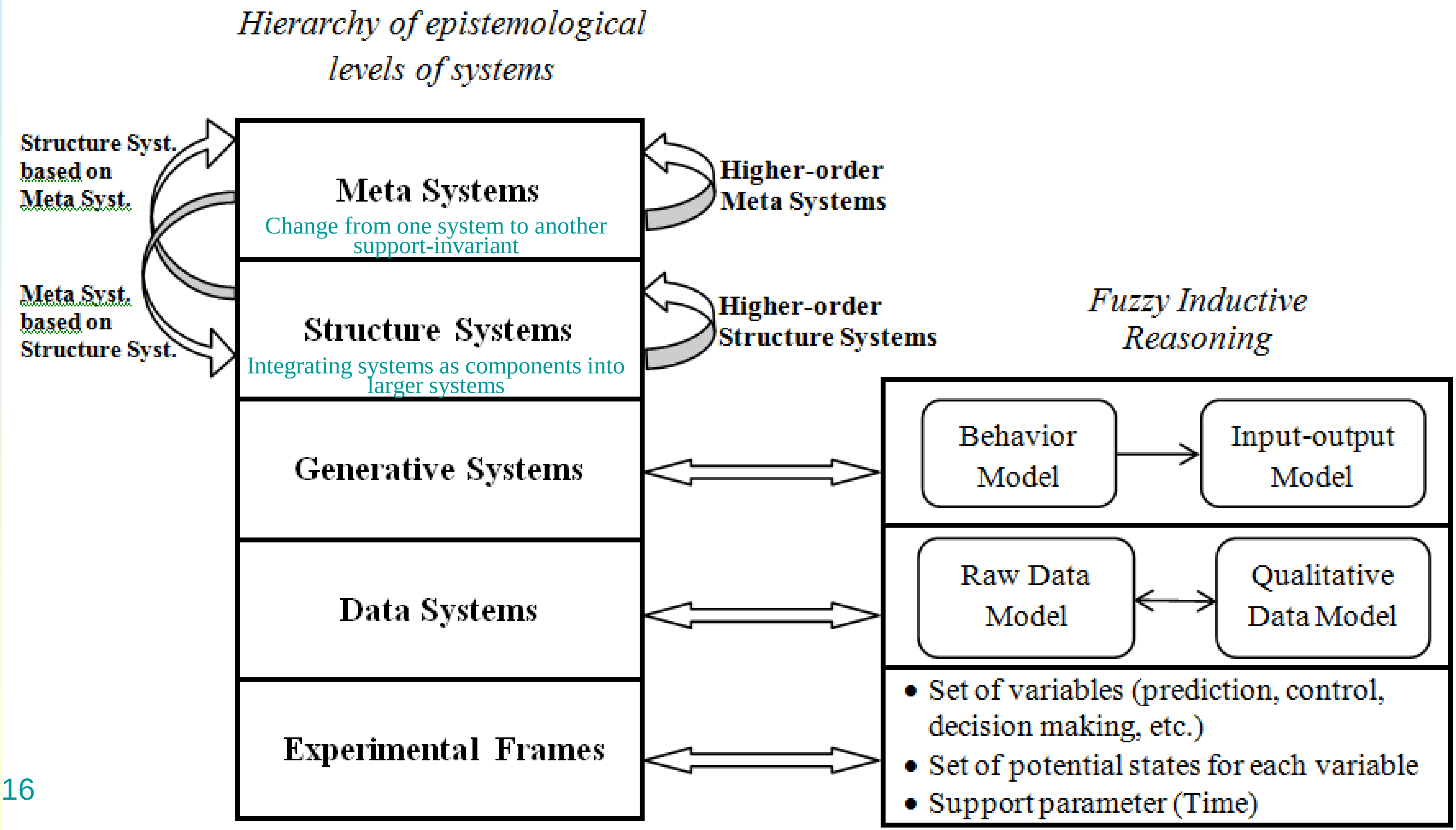
Ex. Theory of Boolean algebras, graph theory, etc..

Mathematical isomorphisms derived from similarities in the forms of algebraic or differential equations.

Ex. Theory of generalized circuits (applicable to mechanical, electrical, thermal and acoustic circuits)

Specific theories of mechanical, electrical, chemical, biological, archeological, etc.

General Systems Problem Solver (GSPS) *Structure (G. Klir)*



Hierarchy of epistemological levels of systems

Structure Syst. based on Meta Syst.

Meta Syst. based on Structure Syst.

Meta Systems

Change from one system to another support-invariant

Structure Systems

Integrating systems as components into larger systems

Generative Systems

Data Systems

Experimental Frames

Higher-order Meta Systems

Higher-order Structure Systems

Fuzzy Inductive Reasoning

Behavior Model

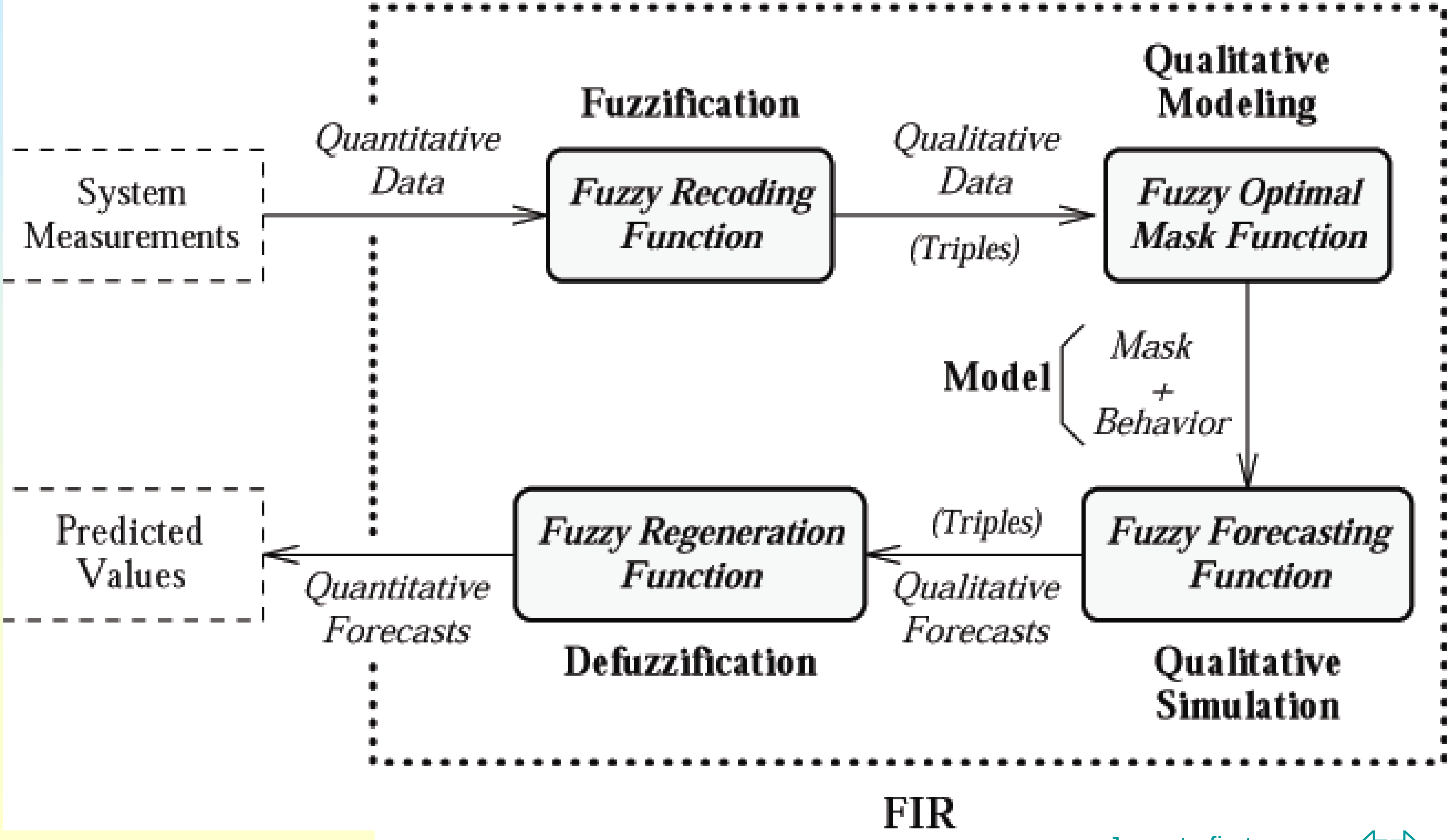
Input-output Model

Raw Data Model

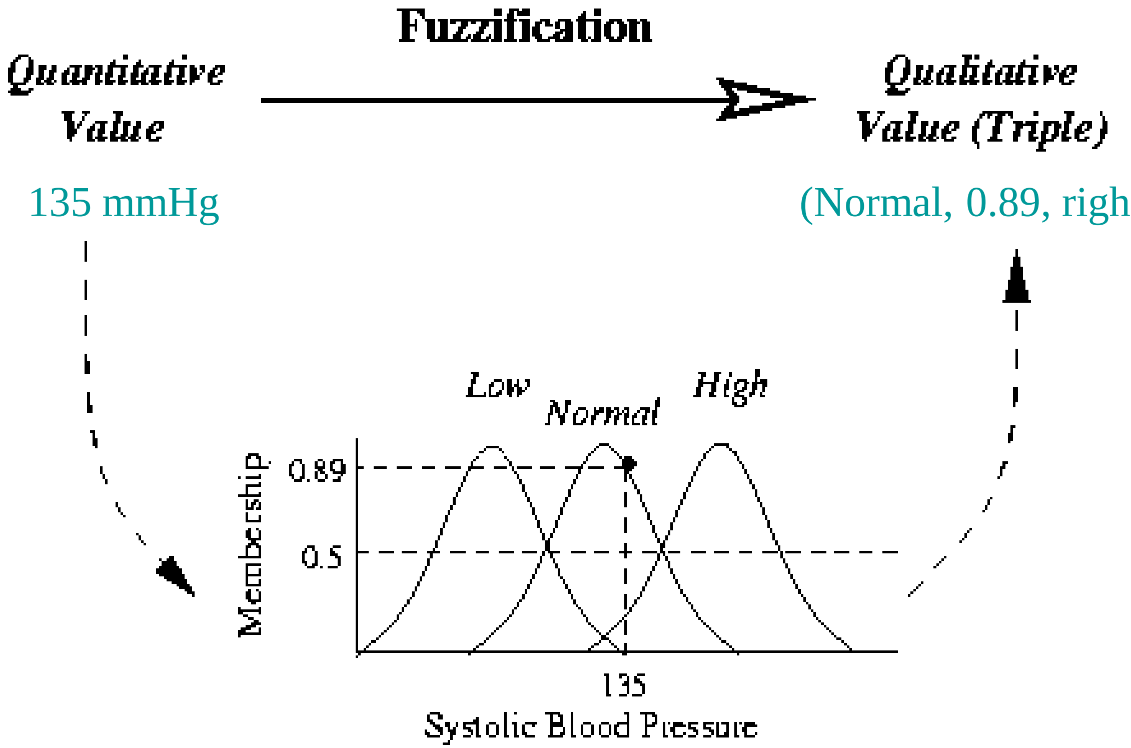
Qualitative Data Model

- Set of variables (prediction, control, decision making, etc.)
- Set of potential states for each variable
- Support parameter (Time)

Fuzzy Inductive Reasoning



Fuzzification in FIR



135 mmHg

(Normal, 0.89, right)

Jump to first page

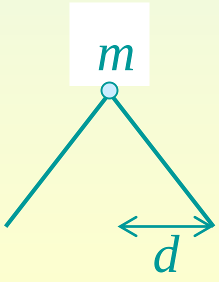
Fuzzification in FIR

- Bell-shaped membership function equation:

$$Memb_i = \exp(-\tau_i \cdot (x - \mu_i)^2)$$

x is the continuous variable to be recorded, μ_i is the algebraic mean between two neighboring landmarks, and τ_i is determined such that the membership function degrades to a value of 0.5 at both of these landmarks

- Triangular membership function equation:



$$Memb_i = \begin{cases} 1 - \left| \dfrac{m - x}{d} \right| & \text{if} \quad m - d \le x \le m + d \\ 0 & \text{if} \quad x < m - d \vee x > m + d \end{cases}$$

Fuzzification in FIR

- How many discrete levels (classes) should be selected for each variable and where the borderlines (landmarks) are to be drawn?

$$n_{rec} \geq 5 \cdot n_{leg} = 5 \cdot \prod_{\forall i} k_i$$

(Law and Kelton, 91)

n_{rec} denotes the total number of recordings, n_{leg} the total number of distinct legal states, k_i the number of levels of the i^th variable

The number of levels (classes) chosen for each variable determines the *expressiveness* of a qualitative model.

The *predictiveness* of a qualitative model is a measure of its forecasting power.

Fuzzification in FIR
(an example)

- Lets see an example of a linear system described by:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\cdot\mathbf{x}+\mathbf{b}\cdot u\\ &= \begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -2 & -3 & -4\end{pmatrix}\cdot\mathbf{x}+\begin{pmatrix}0\\0\\1\end{pmatrix}\cdot u\end{aligned}$$

$$\begin{aligned}y &= \mathbf{C}\cdot\mathbf{x}+\mathbf{d}\cdot u\\ &= \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\cdot\mathbf{x}+\begin{pmatrix}0\\0\\0\end{pmatrix}\cdot u\end{aligned}$$

Computing the step response, it can be concluded that the longest time constant is $t_l \approx 6\,\mathrm{sec}$. Time constant is a measure of how quickly the system responds.

The communication interval (Nyquist sampling rate) is set to $\delta_t \approx 3\,\mathrm{sec}$

Fuzzification in FIR
(an example)

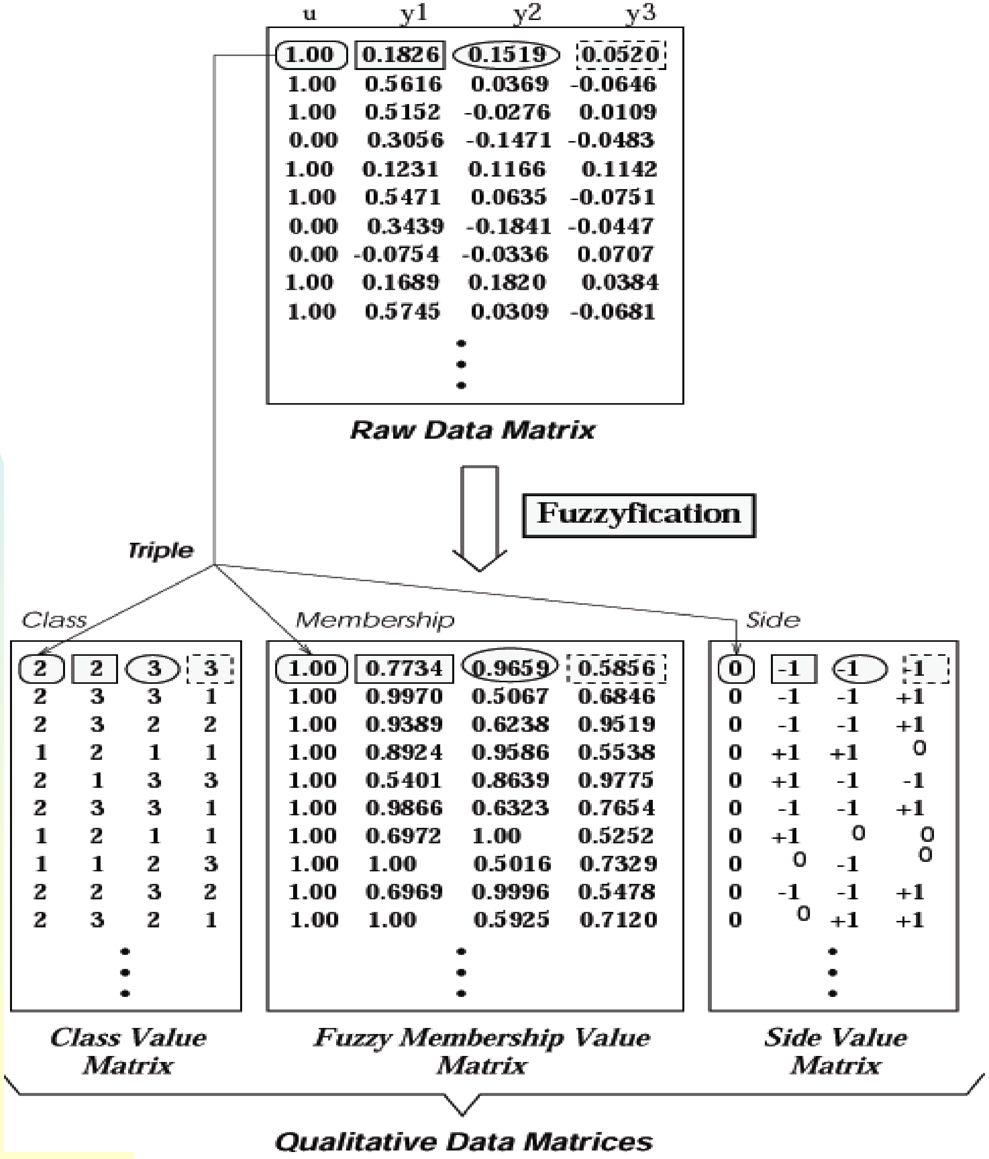
- Input signal: binary random sequence
- Number of classes (output variables): 3
- Number of legal states:

$$n_{leg} = \prod_{\forall i} k_i = 2 \cdot 3 \cdot 3 \cdot 3 = 54$$

- Required number of recordings:

$$n_{rec} = 5 \cdot n_{leg} = 270$$

Fuzzification in FIR (an example)



u	y1	y2	y3
1.00	0.1826	0.1519	0.0520
1.00	0.5616	0.0369	-0.0646
1.00	0.5152	-0.0276	0.0109
0.00	0.3056	-0.1471	-0.0483
1.00	0.1231	0.1166	0.1142
1.00	0.5471	0.0635	-0.0751
0.00	0.3439	-0.1841	-0.0447
0.00	-0.0754	-0.0336	0.0707
1.00	0.1689	0.1820	0.0384
1.00	0.5745	0.0309	-0.0681
		⋮	

Raw Data Matrix

2	2	3	3
2	3	3	1
2	3	2	2
1	2	1	1
2	1	3	3
2	3	3	1
1	2	1	1
1	1	2	3
2	2	3	2
2	3	2	1
	⋮		

Class Value Matrix

1.00	0.7734	0.9659	0.5856
1.00	0.9970	0.5067	0.6846
1.00	0.9389	0.6238	0.9519
1.00	0.8924	0.9586	0.5538
1.00	0.5401	0.8639	0.9775
1.00	0.9866	0.6323	0.7654
1.00	0.6972	1.00	0.5252
1.00	1.00	0.5016	0.7329
1.00	0.6969	0.9996	0.5478
1.00	1.00	0.5925	0.7120
		⋮	

Fuzzy Membership Value Matrix

0	-1	-1	-1
0	-1	-1	+1
0	-1	-1	+1
0	+1	+1	0
0	+1	-1	-1
0	-1	-1	+1
0	+1	0	0
0	0	-1	0
0	-1	-1	+1
0	0	+1	+1
	⋮		

Side Value Matrix

Qualitative Data Matrices

Jump to first page

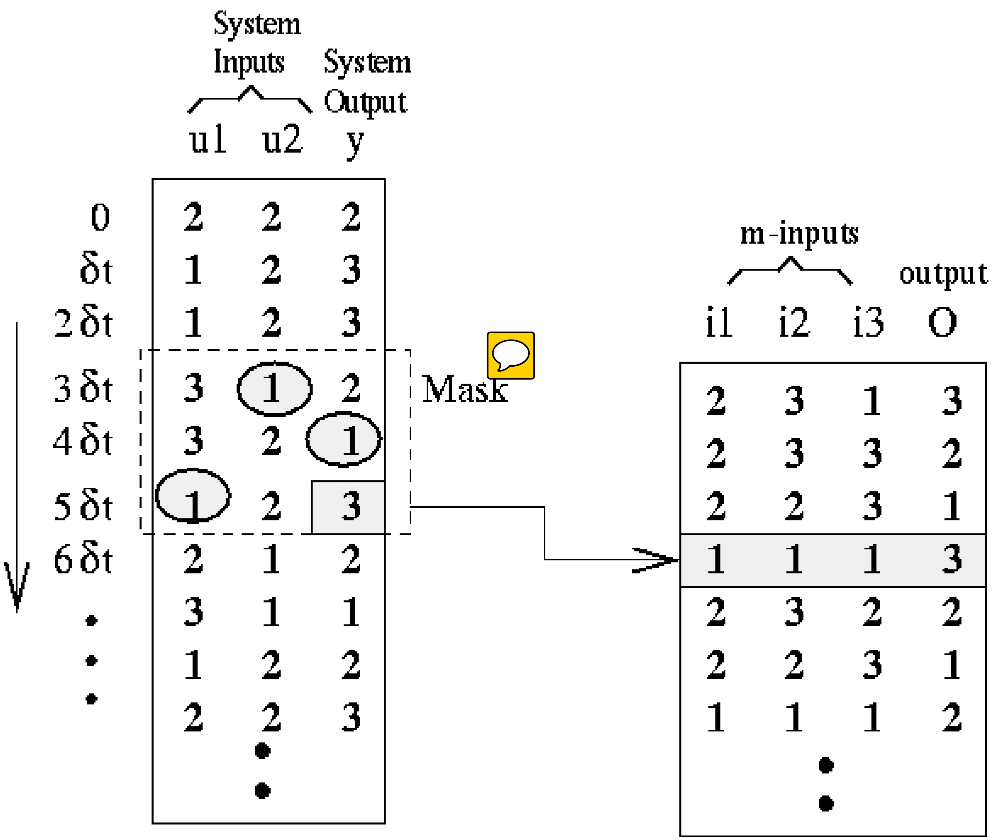
Qualitative Modeling

- A FIR model is composed by the ***mask*** (structure) and the ***behavior matrix*** (pattern rule base).

t\x	u_1	u_2	y
t-2δt	0	-1	0
t-δt	0	0	-2
t	-3	0	+1

Jump to first page

Qualitative Modeling



	u1	u2	y
0	2	2	2
δt	1	2	3
2 δt	1	2	3
3 δt	3	1	2
4 δt	3	2	1
5 δt	1	2	3
6 δt	2	1	2
•	3	1	1
•	1	2	2
•	2	2	3
		•	
		•	

Qualitative Data Matrix
(Classes)

i1	i2	i3	O
2	3	1	3
2	3	3	2
2	2	3	1
1	1	1	3
2	3	2	2
2	2	3	1
1	1	1	2
	•		
	•		

Behavior Matrix
(Pattern Rule Base)

Qualitative Modeling

Determination of the Optimal Mask

t\x	u_1	u_2	y
t-2δt	-1	-1	-1
t-δt	-1	-1	-1
t	-1	-1	+1

Each of the possible masks is compared to the others with respect to its forecasting power

Computation of the quality of a mask:

- *Shannon entropy* is used to determine the *uncertainty* associated with forecasting a particular output state given any legal input state.
- The Shannon entropy relative to one input state is calculated from the following equation:

$$H_i = -\sum_{\forall o} p(o|i).\log_2 p(o|i)$$

Jump to first page

Qualitative Modeling

Determination of the Optimal Mask

$$H_m = \sum_{\forall i} p(i).H_i \qquad H_r = 1.0 - \frac{H_m}{H_{\max}}$$

- *Observation ratio* is introduced as an additional contributor to the overall quality measure

To avoid curse of dimensionality

$$O_r = \frac{5.n_{5x} + 4.n_{4x} + 3.n_{3x} + 2.n_{2x} + n_{1x}}{5.n_{leg}}$$

n_{leg} = number of legal m–input states;

$n_{1\times}$ = number of m–input states observed only once;

$n_{2\times}$ = number of m–input states observed twice;

$n_{3\times}$ = number of m–input states observed thrice;

$n_{4\times}$ = number of m–input states observed four times;

$n_{5\times}$ = number of m–input states observed five times or more.

$$Q_m = H_r.O_r$$

Qualitative Modeling
(an example)

$$Class = \begin{pmatrix} 1 & 1 & 3 & 1 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 2 & 1 & 2 & 3 & 2 \\ 2 & 1 & 1 & 2 & 3 \\ 2 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 3 & 1 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 3 & 1 & 3 & 1 \\ 2 & 2 & 3 & 2 & 1 \end{pmatrix}$$

$$Memb = \begin{pmatrix} 1.0000 & 0.9904 & 0.5000 & 0.7200 & 0.5000 \\ 1.0000 & 0.9905 & 0.9631 & 0.6500 & 0.9709 \\ 1.0000 & 0.9904 & 0.9761 & 0.5500 & 0.9986 \\ 1.0000 & 0.9904 & 0.9992 & 0.8800 & 0.9990 \\ 1.0000 & 0.5520 & 1.0000 & 0.9900 & 1.0000 \\ 1.0000 & 0.5798 & 0.6045 & 0.8700 & 0.6711 \\ 1.0000 & 0.5172 & 0.6123 & 0.6220 & 0.5224 \\ 1.0000 & 0.6186 & 0.5236 & 0.7770 & 0.5255 \\ 1.0000 & 0.5000 & 0.5037 & 0.6550 & 0.5049 \\ 1.0000 & 0.9640 & 0.5033 & 0.5670 & 0.5000 \end{pmatrix}$$

$$\begin{array}{l} t\backslash^{x} \\ t - 2\delta t \\ t - \delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

$$Memb_{\text{joint}} = \underset{\forall i}{\mathrm{I}}\, Memb_i \overset{def}{=} \min_{\forall i}(Memb_i)$$

$$io = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 3 & 3 & 3 \\ 2 & 1 & 1 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 3 & 3 & 3 \end{pmatrix} \quad conf = \begin{pmatrix} 0.5000 \\ 0.9709 \\ 0.9904 \\ 0.5520 \\ 0.5798 \\ 0.5172 \\ 0.5037 \\ 0.5000 \end{pmatrix}$$

Qualitative Modeling

(an example)

$$beh = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 2 & 1 & 1 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 3 & 3 & 3 \end{pmatrix} \quad confbeh = \begin{pmatrix} 0.9904 \\ 1.4709 \\ 0.5172 \\ 0.5037 \\ 0.5520 \\ 1.0798 \end{pmatrix}$$

$$Memb_{\text{cumul}} = \sum_{\forall i} Memb_i$$

$$s = \begin{array}{c|ccc} in\backslash^{out} & '1' & '2' & '3' \\ \hline '121' & 0.9904 & 1.4709 & 0.0000 \\ '211' & 0.5172 & 0.0000 & 0.0000 \\ '212' & 0.5037 & 0.0000 & 0.0000 \\ '221' & 0.5520 & 0.0000 & 0.0000 \\ '233' & 0.0000 & 0.0000 & 1.0798 \end{array}$$

$$ic = \begin{array}{c|c} input & conf \\ \hline '121' & 2.4613 \\ '211' & 0.5172 \\ '212' & 0.5037 \\ '221' & 0.5520 \\ '233' & 1.0798 \end{array}$$

$$ns = \begin{array}{c|ccc} in\backslash^{out} & '1' & '2' & '3' \\ \hline '121' & 0.4024 & 0.5976 & 0.0000 \\ '211' & 1.0 & 0.0000 & 0.0000 \\ '212' & 1.0 & 0.0000 & 0.0000 \\ '221' & 1.0 & 0.0000 & 0.0000 \\ '233' & 0.0000 & 0.0000 & 1.0 \end{array}$$

$$nic = \begin{array}{c|c} input & conf \\ \hline '121' & 0.4813 \\ '211' & 0.1011 \\ '212' & 0.0985 \\ '221' & 0.1079 \\ '233' & 0.2111 \end{array}$$

Qualitative Modeling

(an example)

$$
ns = \begin{array}{c} in\backslash^{out} \\ '121' \\ '211' \\ '212' \\ '221' \\ '233' \end{array}
\begin{array}{c} \begin{array}{ccc} '1' & '2' & '3' \end{array} \\ \begin{pmatrix} 0.4024 & 0.5976 & 0.0000 \\ 1.0 & 0.0000 & 0.0000 \\ 1.0 & 0.0000 & 0.0000 \\ 1.0 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0 \end{pmatrix} \end{array}
$$

$$
\begin{aligned}
H_m &= -\ 0.4813 \cdot [0.4024 \cdot \log_2(0.4024) + 0.5976 \cdot \log_2(0.5976)] \\
&+ \ 0.1011 \cdot [1.0 \cdot \log_2(1.0)] \\
&+ \ 0.0985 \cdot [1.0 \cdot \log_2(1.0)] \\
&+ \ 0.1079 \cdot [1.0 \cdot \log_2(1.0)] \\
&+ \ 0.2111 \cdot [1.0 \cdot \log_2(1.0)] \\
&= \ 0.4680 + 0.0 + 0.0 + 0.0 + 0.0 \\
&= \ 0.4680
\end{aligned}
$$

$$
H_m = \sum_{\forall i} p(i).H_i
$$

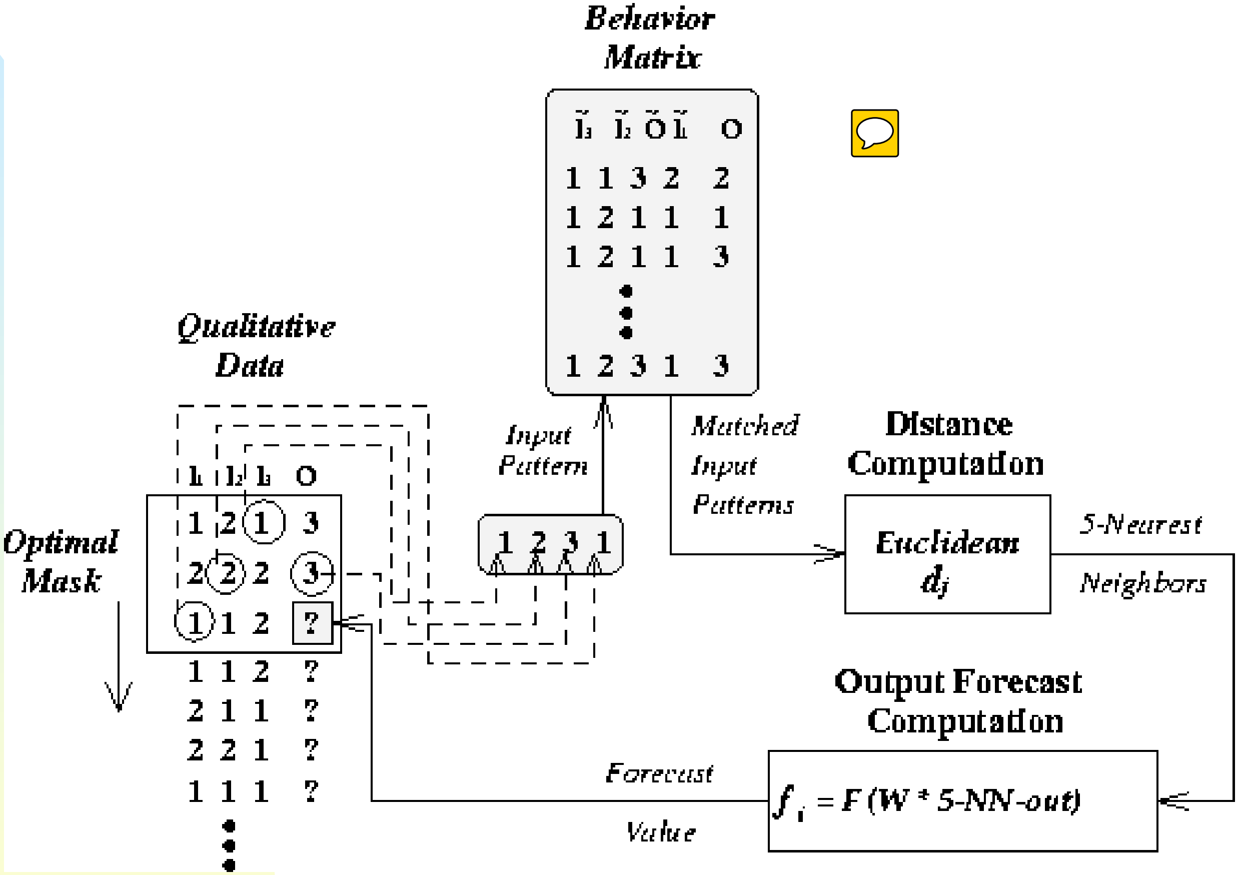
$$
H_i = -\sum_{\forall o} p(o|i).\log_2 p(o|i)
$$

$$
nic = \begin{array}{c} input \\ '121' \\ '211' \\ '212' \\ '221' \\ '233' \end{array}
\begin{array}{c} conf \\ \begin{pmatrix} 0.4813 \\ 0.1011 \\ 0.0985 \\ 0.1079 \\ 0.2111 \end{pmatrix} \end{array}
$$

$$
\begin{aligned}
H_r &= 1.0 - (H_m / H_{max}) \\
&= 1.0 - (0.4680 / 1.585) \\
&= 0.7047
\end{aligned}
$$

Jump to first page

Qualitative Simulation



Behavior Matrix

$\breve{i}_3$	$\breve{i}_2$	$\breve{o}$	$\breve{i}_1$	o
1	1	3	2	2
1	2	1	1	1
1	2	1	1	3
•				
1	2	3	1	3

Qualitative Data

i_1	i_2	i_3	o
1	2	1	3
2	2	2	3
1	1	2	?
1	1	2	?
2	1	1	?
2	2	1	?
1	1	1	?
•			

Qualitative Simulation

A *normalization function* is computed for every element of the new input state

$$p_i = Side_i \cdot B \cdot \sqrt{\ln Memb_i} + 0.5$$

where: $B = (4 \cdot \ln 0.5)^{-1/2}$. For the left class:

$$p_i = C \cdot \sqrt{\ln Memb_i}$$

with: $C = (\ln 0.5)^{-1/2}$. For the right class:

$$p_i = 1 - C \cdot \sqrt{\ln Memb_i}$$

The P_i signals ranges from 0.0 to 1.0, and are concatenated to form the vector:

$$\mathbf{p} = [p_1, p_2, \ldots, p_j]$$

Jump to first page

Qualitative Simulation

Distance computation between the new input state and the previous recordings

$$d_k = \|\mathbf{p} - \mathbf{p_k}\|_2 = \sqrt{\sum_{i=1}^{N} (P_i - P_{ik})^2}$$

The contribution of each neighbor to the estimation of the prediction of the new output state is a function of its proximity

$$w_{\mathbf{abs}k} = \frac{(d_{\mathbf{max}}^2 - d_k^2)}{d_{\mathbf{max}} \cdot d_k}$$

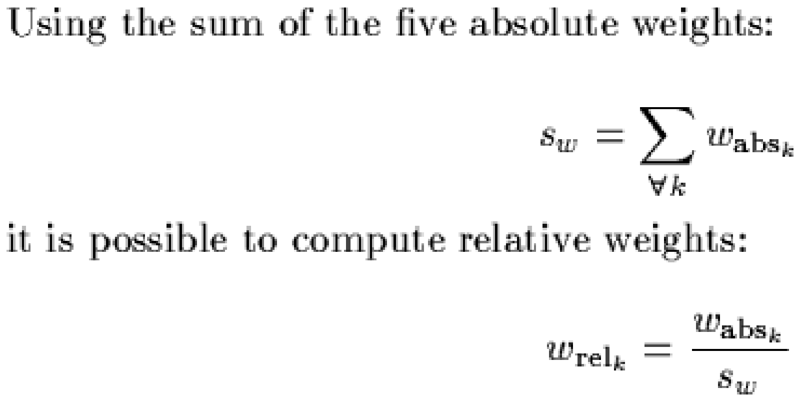
If any of the d_k values is zero:

$$w_{\mathrm{abs}_k} = \begin{cases} 0.0\ ; & d_k \neq 0.0 \\ 1.0\ ; & d_k = 0.0 \end{cases}$$

Jump to first page

Qualitative Simulation

Relative weights computing:



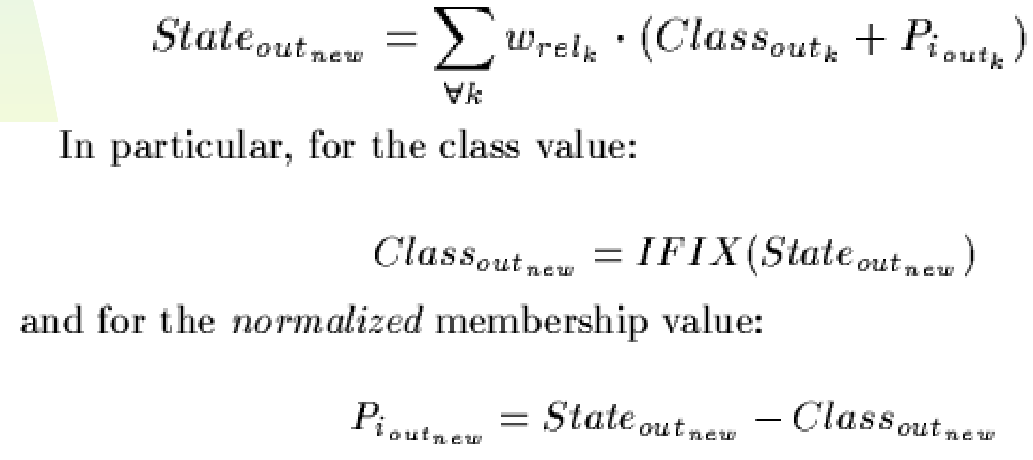
Using the sum of the five absolute weights:

$$s_w = \sum_{\forall k} w_{\mathrm{abs}_k}$$

it is possible to compute relative weights:

$$w_{\mathrm{rel}_k} = \frac{w_{\mathrm{abs}_k}}{s_w}$$

A qualitative output state can be computed as:



$$State_{out_{new}} = \sum_{\forall k} w_{rel_k} \cdot (Class_{out_k} + P_{i_{out_k}})$$

In particular, for the class value:

$$Class_{out_{new}} = IFIX(State_{out_{new}})$$

IFIX refer to the whole number

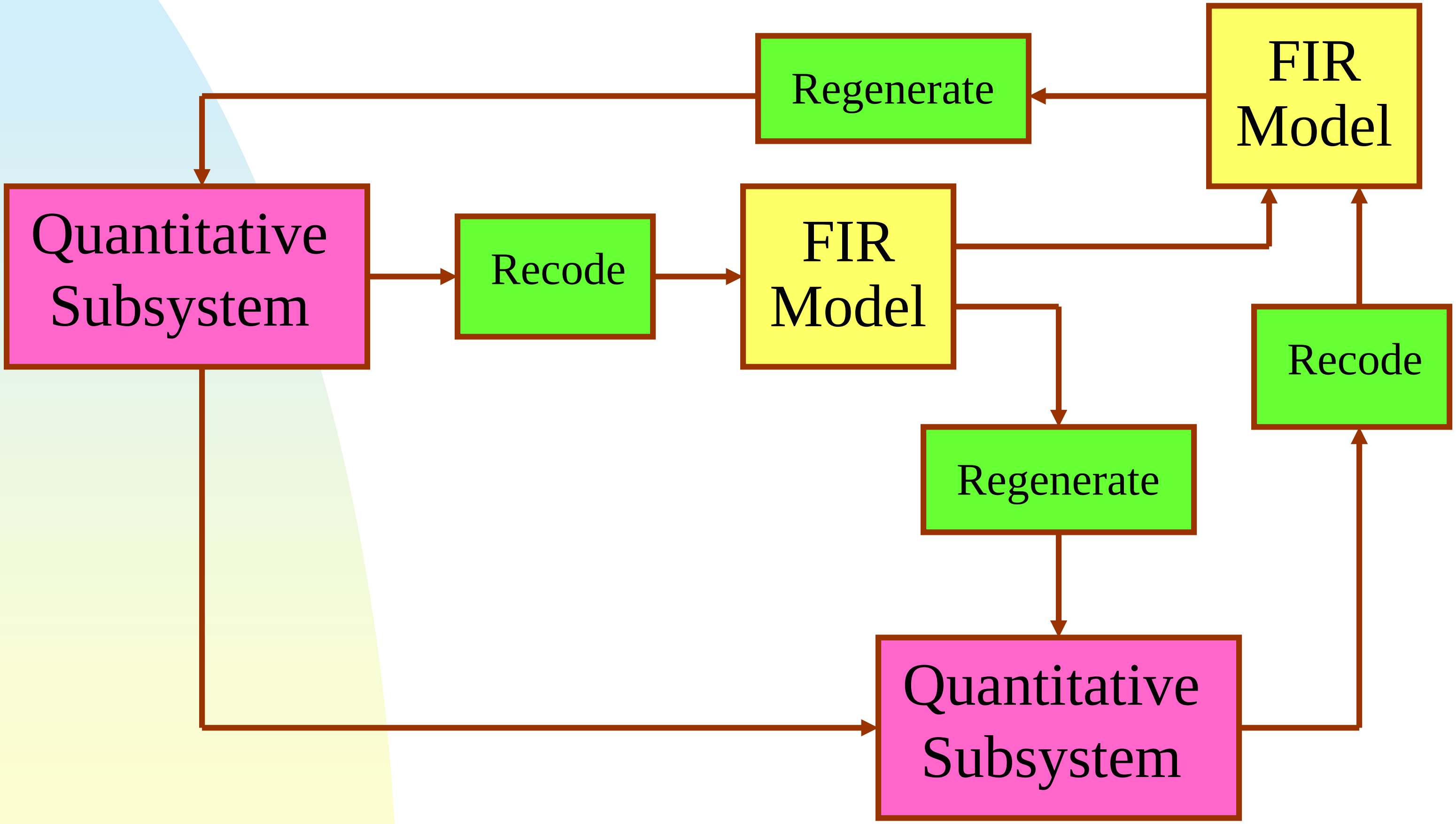
and for the *normalized* membership value:

$$P_{i_{out_{new}}} = State_{out_{new}} - Class_{out_{new}}$$

Jump to first page

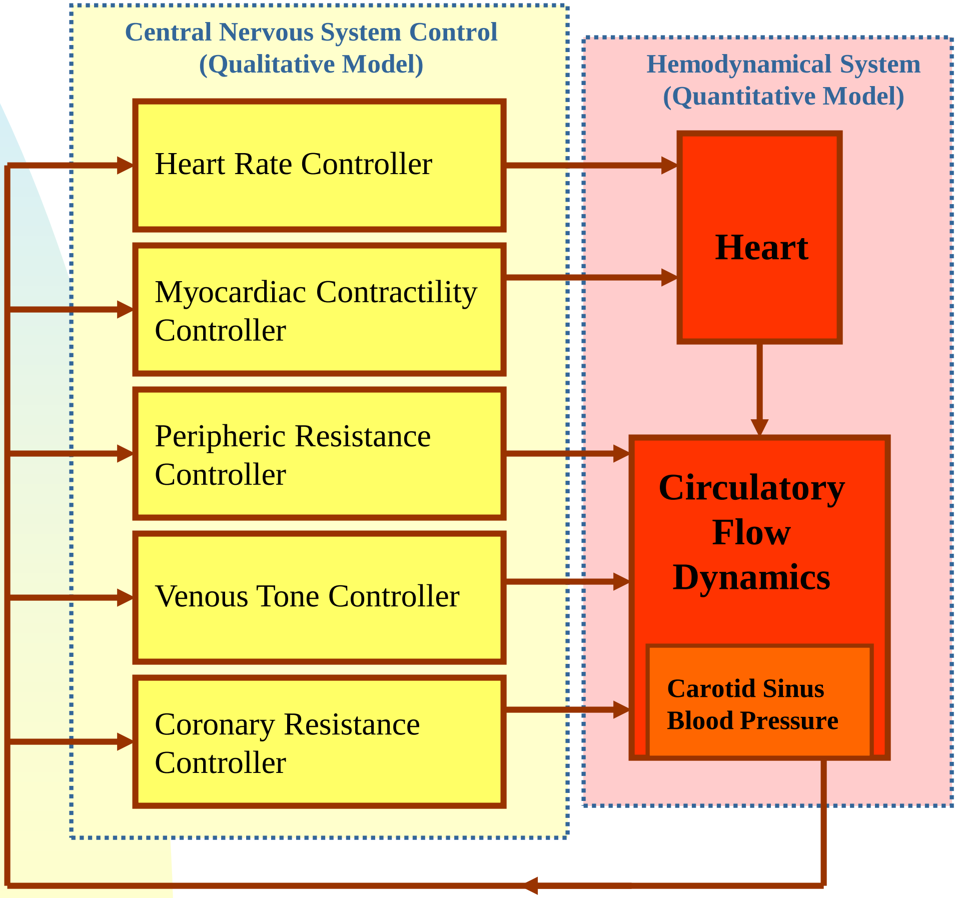
Fuzzy Inductive Reasoning

Mixed Quantitative/Qualitative Modeling



Application

Cardiovascular System



Central Nervous System Control
(Qualitative Model)

Heart Rate Controller

Myocardiac Contractility Controller

Peripheric Resistance Controller

Venous Tone Controller

Coronary Resistance Controller

Hemodynamical System
(Quantitative Model)

Heart

Circulatory Flow Dynamics

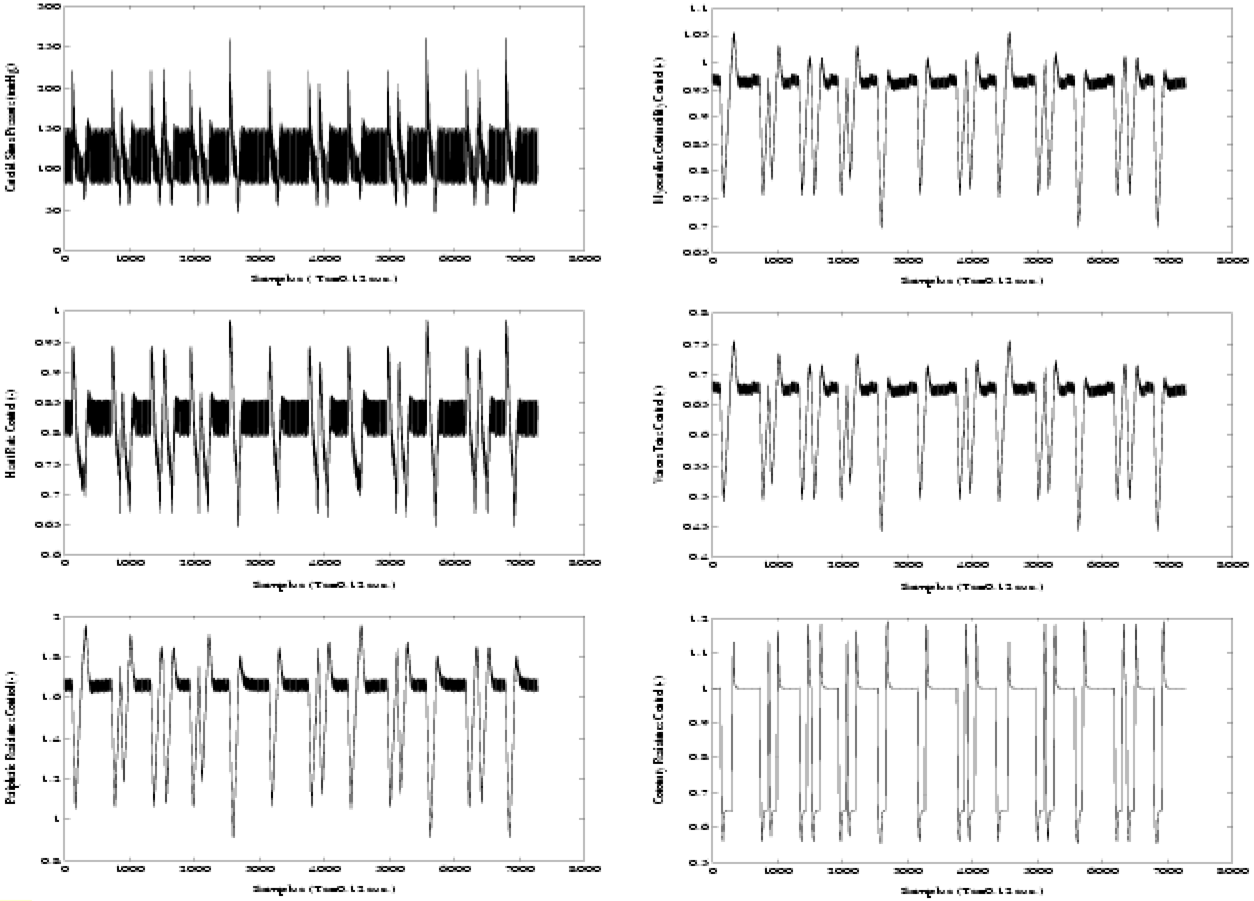
Carotid Sinus Blood Pressure

Cardiovascular System

CNS controllers training data (patient 1)

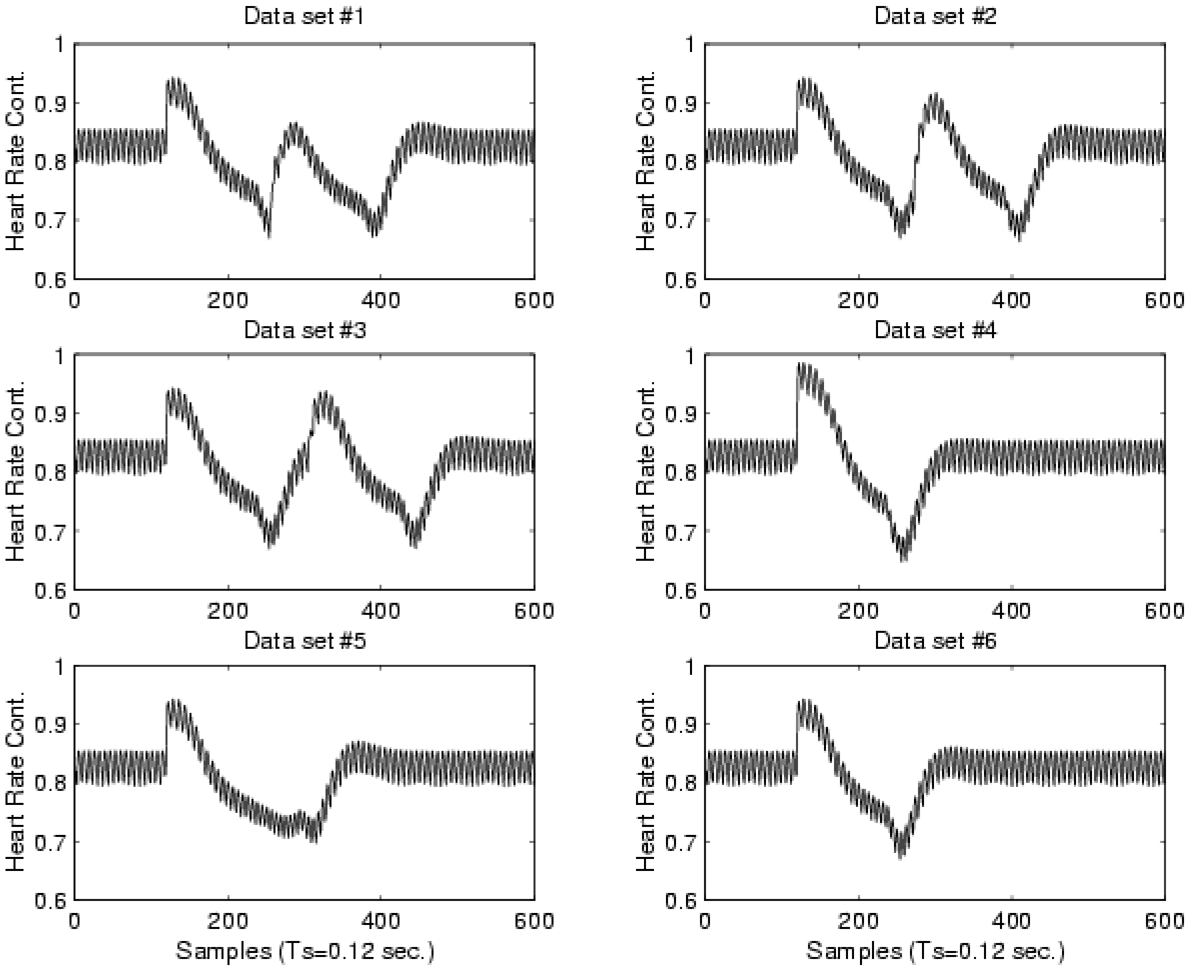
Input variable:
Carotid Sinus Blood Pressure

Output variables:
Heart rate, Peripheric resistance, Myocardiac contractility, Venous tone and Coronary resistance controllers



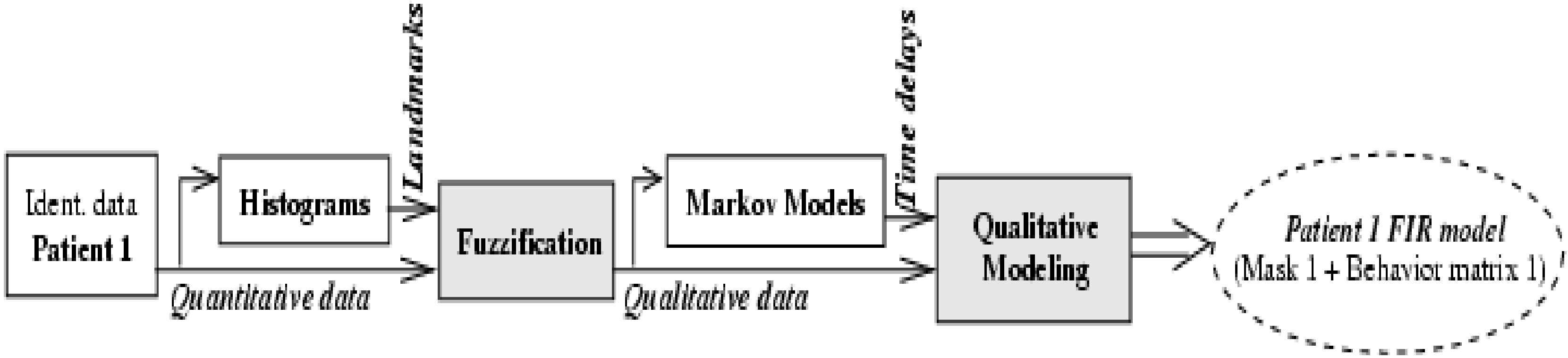
Cardiovascular System

Heart rate controller tests data sets (patient 1)



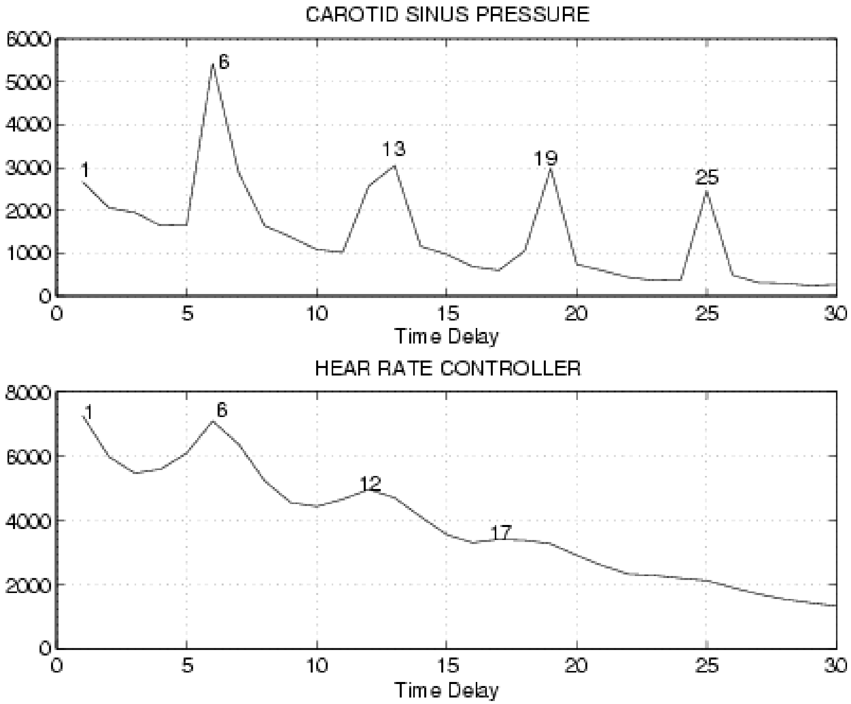
Cardiovascular System

Qualitative identification process



CAROTID SINUS PRESSURE

HEAR RATE CONTROLLER



$$
\begin{array}{c|cc}
{}_t\backslash^x & CSP & HRC \\
\hline
t-25\delta t & -1 & 0 \\
\vdots & \vdots & \vdots \\
t-19\delta t & -1 & 0 \\
\vdots & \vdots & \vdots \\
t-17\delta t & 0 & -1 \\
\vdots & \vdots & \vdots \\
t-13\delta t & -1 & 0 \\
t-12\delta t & 0 & -1 \\
\vdots & \vdots & \vdots \\
t-6\delta t & -1 & -1 \\
\vdots & \vdots & \vdots \\
t-\delta t & -1 & -1 \\
t & -1 & +1
\end{array}
$$

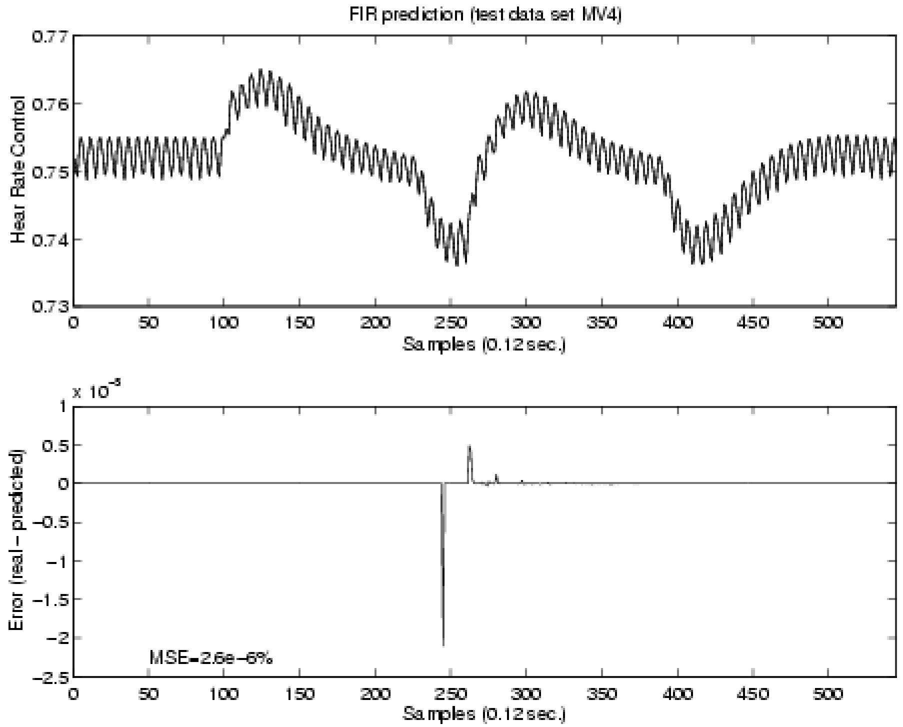
mp to first page

Cardiovascular System

Average MSE errors of the CNS controllers

$$MSE = \frac{E[(y(t) - \hat{y}(t))^2]}{y_{\text{var}}} \cdot 100\%$$

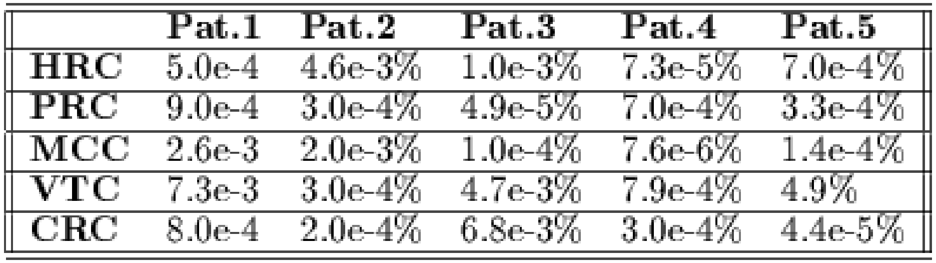
	Pat.1	Pat.2	Pat.3	Pat.4	Pat.5
HRC	5.0e-4	4.6e-3%	1.0e-3%	7.3e-5%	7.0e-4%
PRC	9.0e-4	3.0e-4%	4.9e-5%	7.0e-4%	3.3e-4%
MCC	2.6e-3	2.0e-3%	1.0e-4%	7.6e-6%	1.4e-4%
VTC	7.3e-3	3.0e-4%	4.7e-3%	7.9e-4%	4.9%
CRC	8.0e-4	2.0e-4%	6.8e-3%	3.0e-4%	4.4e-5%



Jump to first page

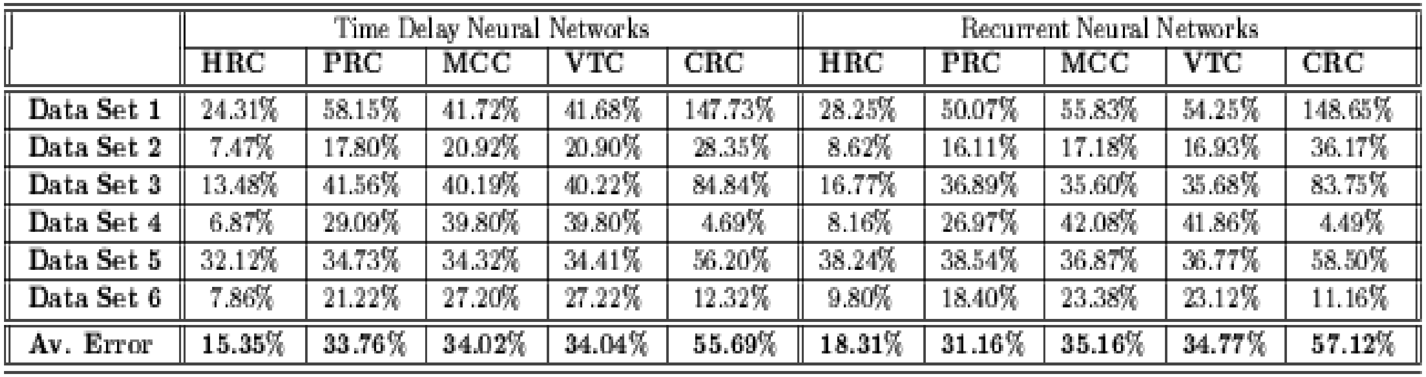
Cardiovascular System

FIR



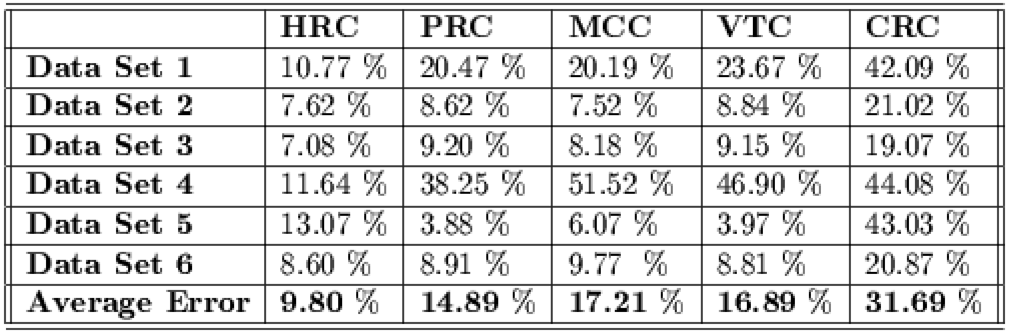
	Pat.1	Pat.2	Pat.3	Pat.4	Pat.5
HRC	5.0e-4	4.6e-3%	1.0e-3%	7.3e-5%	7.0e-4%
PRC	9.0e-4	3.0e-4%	4.9e-5%	7.0e-4%	3.3e-4%
MCC	2.6e-3	2.0e-3%	1.0e-4%	7.6e-6%	1.4e-4%
VTC	7.3e-3	3.0e-4%	4.7e-3%	7.9e-4%	4.9%
CRC	8.0e-4	2.0e-4%	6.8e-3%	3.0e-4%	4.4e-5%

NN



	Time Delay Neural Networks					Recurrent Neural Networks				
	HRC	PRC	MCC	VTC	CRC	HRC	PRC	MCC	VTC	CRC
Data Set 1	24.31%	58.15%	41.72%	41.68%	147.73%	28.25%	50.07%	55.83%	54.25%	148.65%
Data Set 2	7.47%	17.80%	20.92%	20.90%	28.35%	8.62%	16.11%	17.18%	16.93%	36.17%
Data Set 3	13.48%	41.56%	40.19%	40.22%	84.84%	16.77%	36.89%	35.60%	35.68%	83.75%
Data Set 4	6.87%	29.09%	39.80%	39.80%	4.69%	8.16%	26.97%	42.08%	41.86%	4.49%
Data Set 5	32.12%	34.73%	34.32%	34.41%	56.20%	38.24%	38.54%	36.87%	36.77%	58.50%
Data Set 6	7.86%	21.22%	27.20%	27.22%	12.32%	9.80%	18.40%	23.38%	23.12%	11.16%
Av. Error	**15.35%**	**33.76%**	**34.02%**	**34.04%**	**55.69%**	**18.31%**	**31.16%**	**35.16%**	**34.77%**	**57.12%**

Narmax

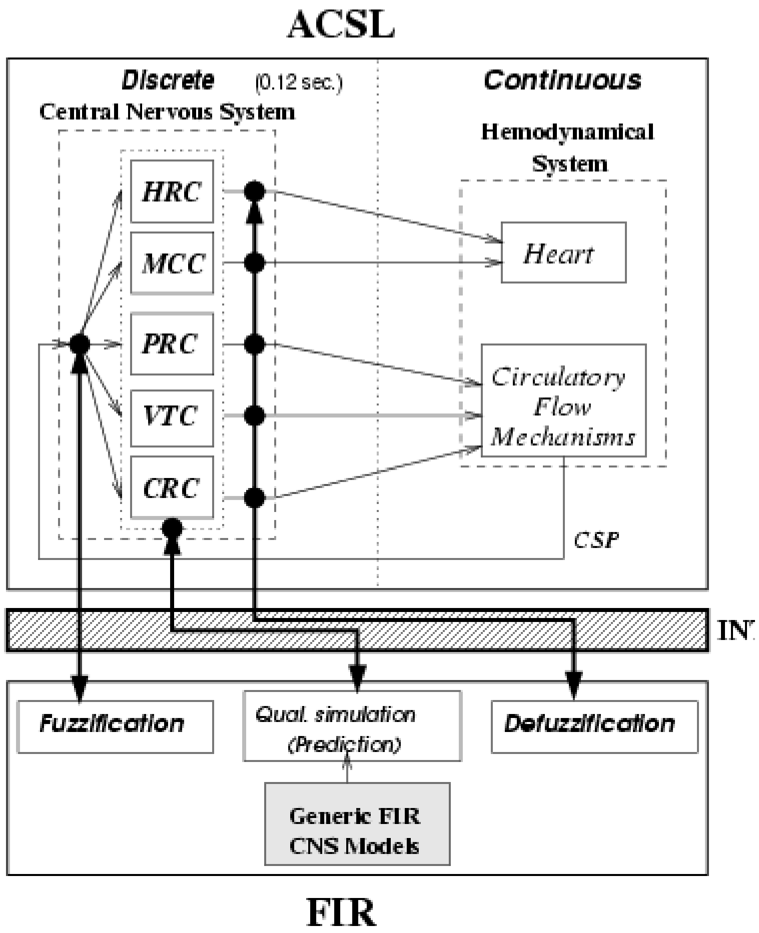


	HRC	PRC	MCC	VTC	CRC
Data Set 1	10.77 %	20.47 %	20.19 %	23.67 %	42.09 %
Data Set 2	7.62 %	8.62 %	7.52 %	8.84 %	21.02 %
Data Set 3	7.08 %	9.20 %	8.18 %	9.15 %	19.07 %
Data Set 4	11.64 %	38.25 %	51.52 %	46.90 %	44.08 %
Data Set 5	13.07 %	3.88 %	6.07 %	3.97 %	43.03 %
Data Set 6	8.60 %	8.91 %	9.77 %	8.81 %	20.87 %
Average Error	**9.80** %	**14.89** %	**17.21** %	**16.89** %	**31.69** %

Jump to first page

Cardiovascular System (Closed Loop)

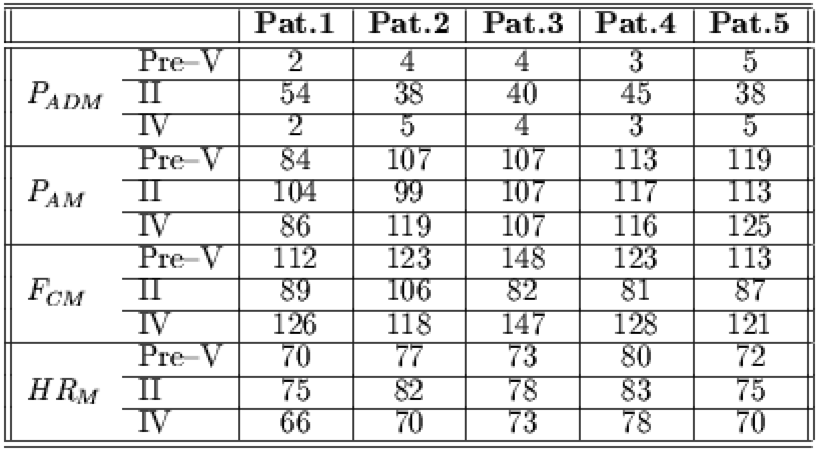
Structure of the cardiovascular system simulation program



Jump to first page

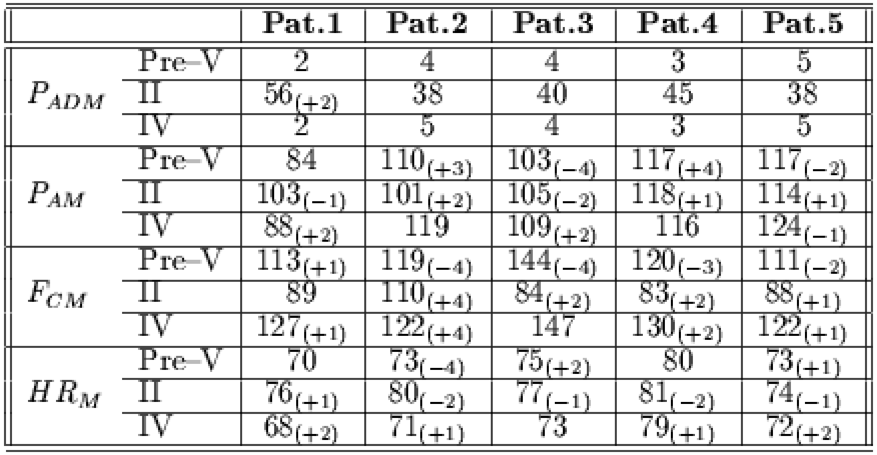
Cardiovascular System (Closed Loop)

Measurements results obtained through cardiac catherization



		Pat.1	Pat.2	Pat.3	Pat.4	Pat.5
P_{ADM}	Pre–V	2	4	4	3	5
	II	54	38	40	45	38
	IV	2	5	4	3	5
P_{AM}	Pre–V	84	107	107	113	119
	II	104	99	107	117	113
	IV	86	119	107	116	125
F_{CM}	Pre–V	112	123	148	123	113
	II	89	106	82	81	87
	IV	126	118	147	128	121
HR_M	Pre–V	70	77	73	80	72
	II	75	82	78	83	75
	IV	66	70	73	78	70

Results obtained from the mixed cardiovascular system (FIR-DE)



		Pat.1	Pat.2	Pat.3	Pat.4	Pat.5
P_{ADM}	Pre–V	2	4	4	3	5
	II	$56_{(+2)}$	38	40	45	38
	IV	2	5	4	3	5
P_{AM}	Pre–V	84	$110_{(+3)}$	$103_{(-4)}$	$117_{(+4)}$	$117_{(-2)}$
	II	$103_{(-1)}$	$101_{(+2)}$	$105_{(-2)}$	$118_{(+1)}$	$114_{(+1)}$
	IV	$88_{(+2)}$	119	$109_{(+2)}$	116	$124_{(-1)}$
F_{CM}	Pre–V	$113_{(+1)}$	$119_{(-4)}$	$144_{(-4)}$	$120_{(-3)}$	$111_{(-2)}$
	II	89	$110_{(+4)}$	$84_{(+2)}$	$83_{(+2)}$	$88_{(+1)}$
	IV	$127_{(+1)}$	$122_{(+4)}$	147	$130_{(+2)}$	$122_{(+1)}$
HR_M	Pre–V	70	$73_{(-4)}$	$75_{(+2)}$	80	$73_{(+1)}$
	II	$76_{(+1)}$	$80_{(-2)}$	$77_{(-1)}$	$81_{(-2)}$	$74_{(-1)}$
	IV	$68_{(+2)}$	$71_{(+1)}$	73	$79_{(+1)}$	$72_{(+2)}$

Right auricular pressure

Aortic pressure

Coronary blood flow

Heart rate

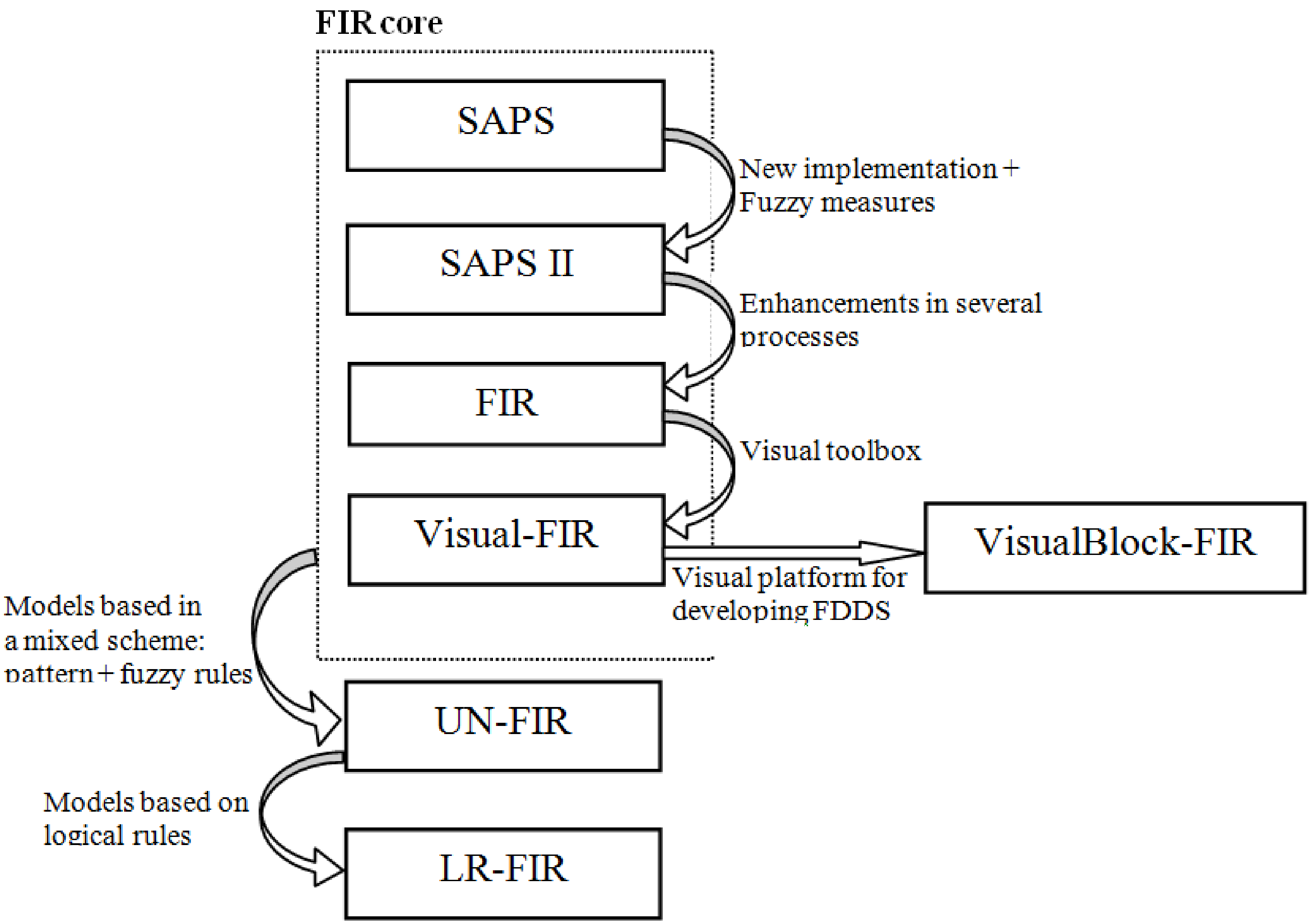
In order to the model to be accepted, non of the 4 variables must deviate from the reference values by more than ±10%

Applications

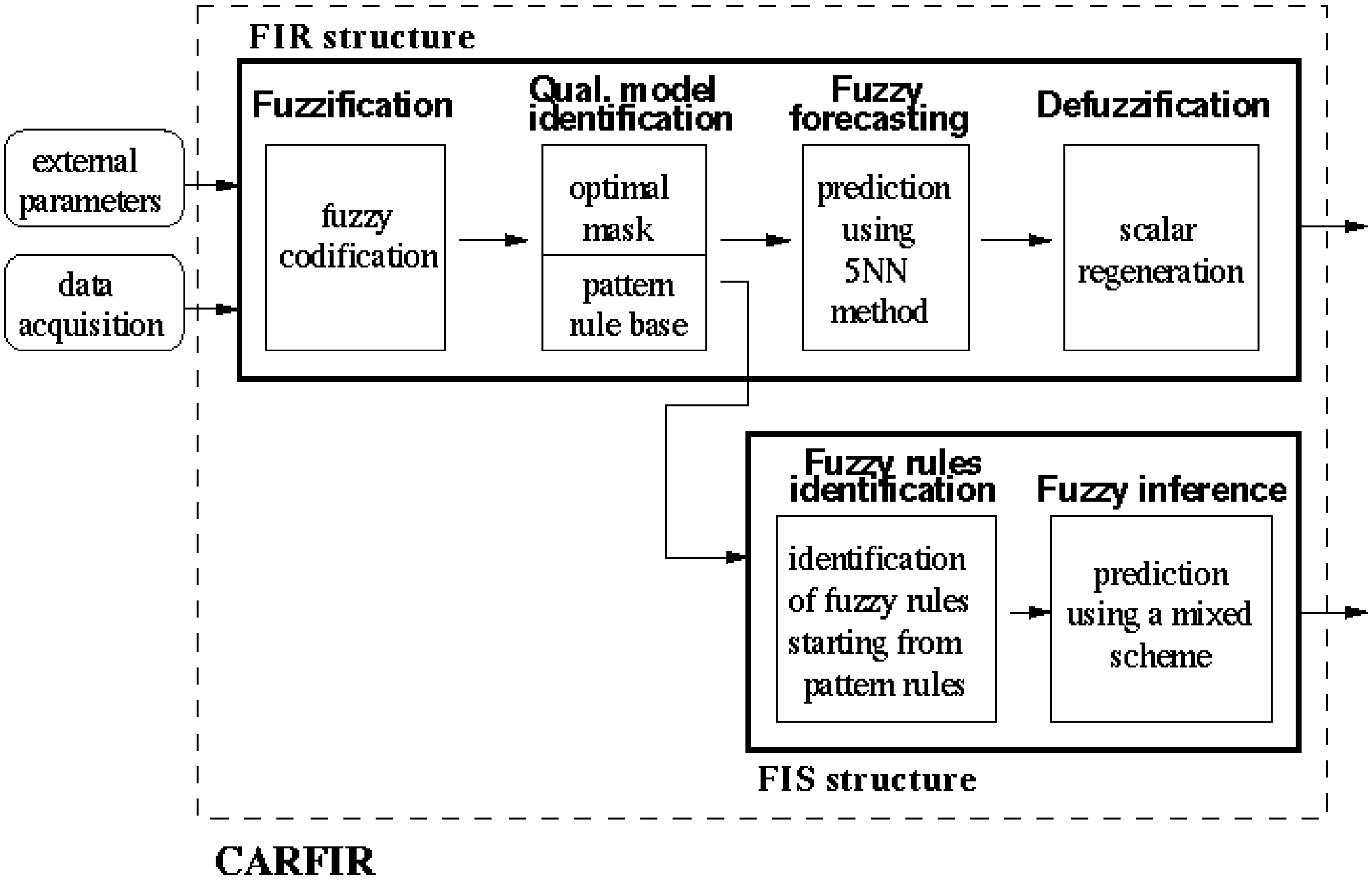
- Cardiovascular system modeling for classification of anomalies
- Anaesthesiology model for control of depth of anaesthesia during surgery
- Shrimp growth model for a shrimp farm in northern méxico
- Prediction of water demand in barcelona and rotterdam
- Prediction of ozone concentration in austria and mexico city
- Design of fuzzy controller for tanker ship steering
- Fault diagnosis on water distribution networks, nuclear power plants, etc.
- etc.

Jump to first page

Evolution of the FIR methodology


FIR core
SAPS
New implementation + Fuzzy measures
SAPS II
Enhancements in several processes
FIR
Visual toolbox
Visual-FIR
Visual platform for developing FDDS
VisualBlock-FIR
Models based in a mixed scheme: pattern + fuzzy rules
UN-FIR
Models based on logical rules
LR-FIR

UNFIR


FIR structure
Fuzzification
Qual. model identification
Fuzzy forecasting
Defuzzification
external parameters
data acquisition
fuzzy codification
optimal mask
pattern rule base
prediction using 5NN method
scalar regeneration
Fuzzy rules identification
Fuzzy inference
identification of fuzzy rules starting from pattern rules
prediction using a mixed scheme
FIS structure
CARFIR

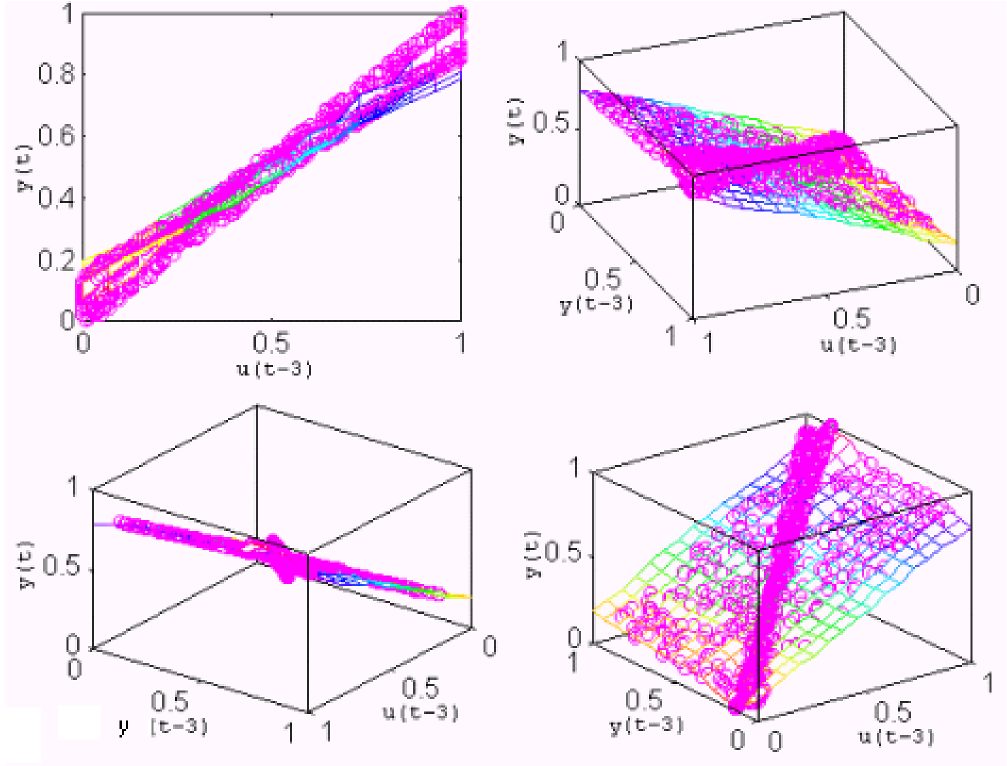
UNFIR

- Step 1: Generation of pattern rules by means of FIR
  - *Data acquisition* stemming from the system
  - Specification of the *external parameters*
  - Qualitative *model identification*
- Step 2: Identification of fuzzy rules and systems’ prediction
  - *Identification of fuzzy rules* starting from pattern rules
  - Prediction by means of a *mixed scheme*

Jump to first page

UNFIR- Identification of fuzzy rules

Tuning process: Automatically adjusting the mesh of the fuzzy rules to the surface of the pattern rules.

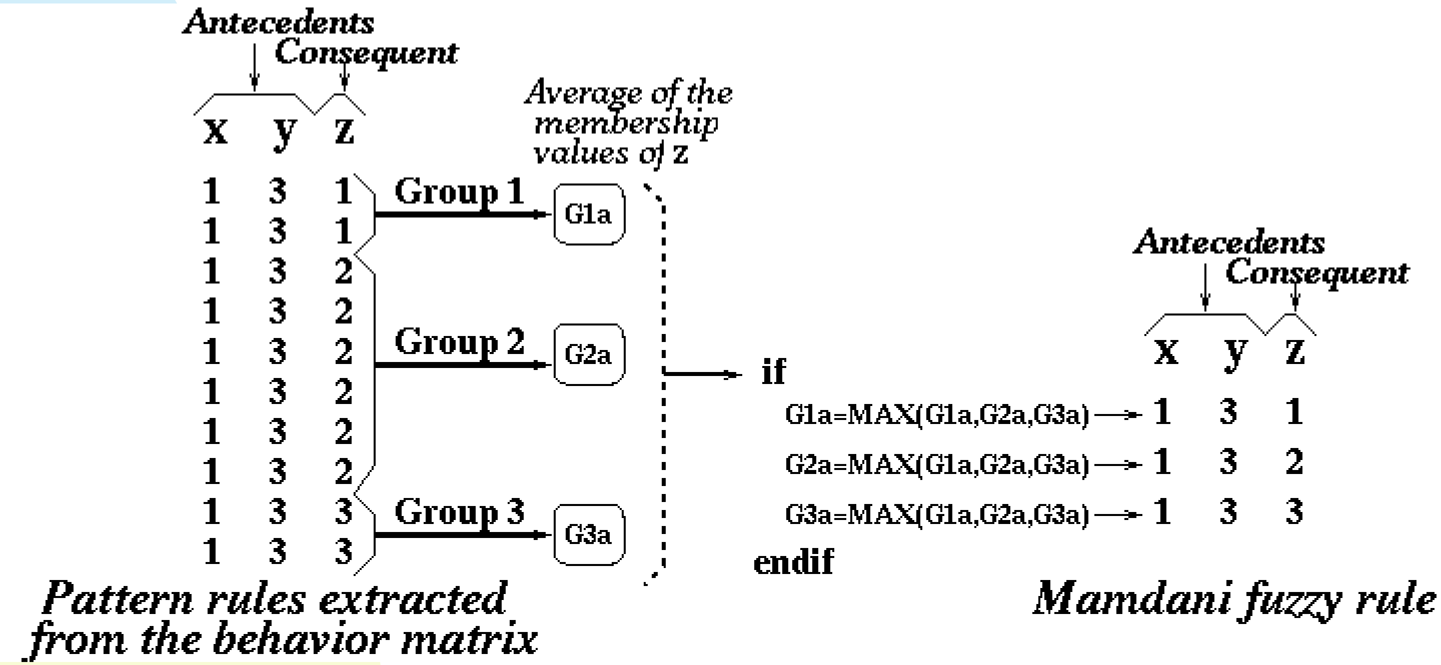


Circles: pattern rules

Squares: fuzzy rules

UNFIR- Identification of fuzzy rules

- Mamdani tuning process



x	y	z
1	3	1
1	3	1
1	3	2
1	3	2
1	3	2
1	3	2
1	3	2
1	3	2
1	3	3
1	3	3

Pattern rules extracted from the behavior matrix

if

	x	y	z
G1a=MAX(G1a,G2a,G3a) →	1	3	1
G2a=MAX(G1a,G2a,G3a) →	1	3	2
G3a=MAX(G1a,G2a,G3a) →	1	3	3

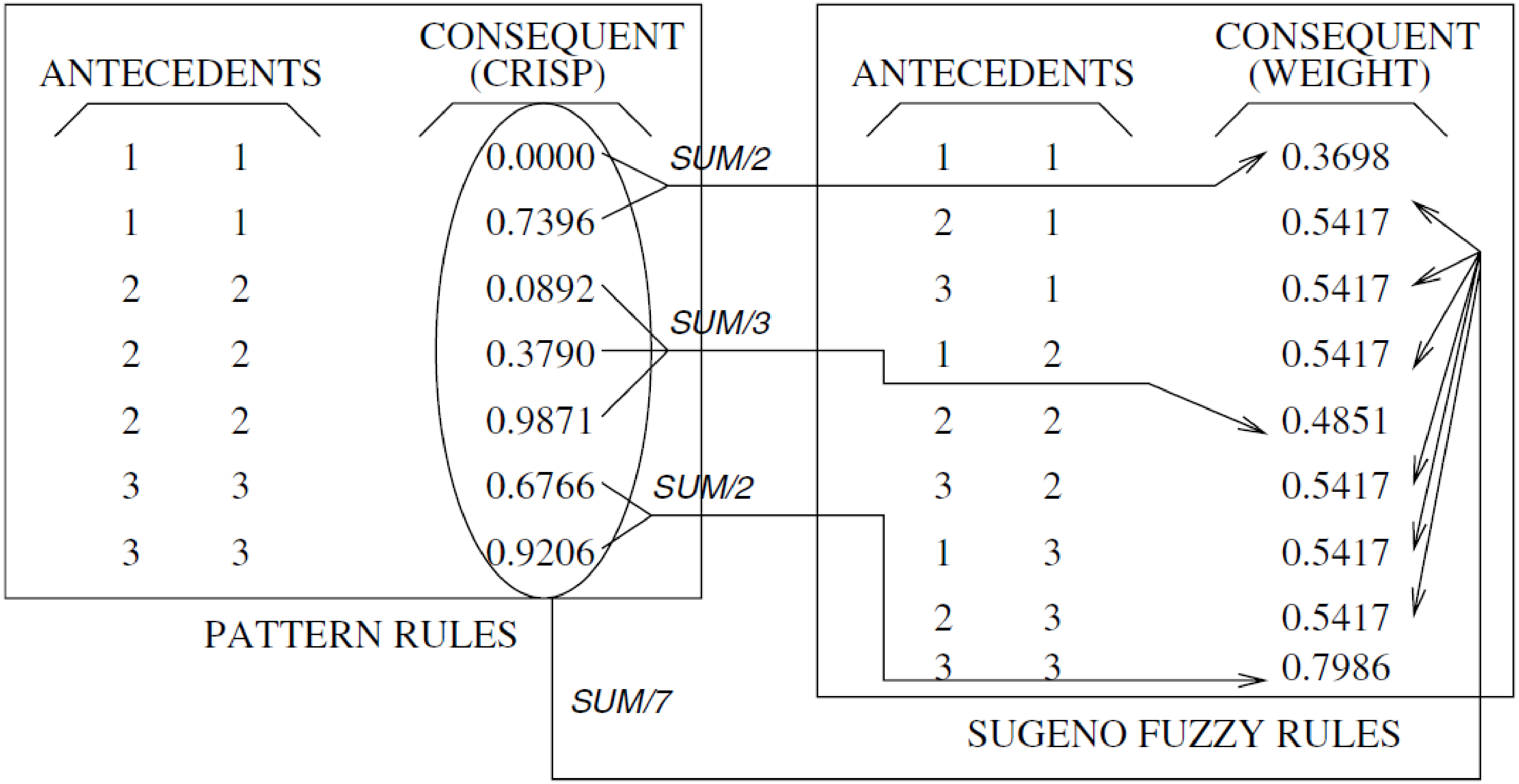
endif

Mamdani fuzzy rule

Default parameters: algebraic product (T-norm), Mean of Maximum (defuzzification process)

UNFIR- Identification of fuzzy rules

- Sugeno tuning process



ANTECEDENTS		CONSEQUENT (CRISP)
1	1	0.0000
1	1	0.7396
2	2	0.0892
2	2	0.3790
2	2	0.9871
3	3	0.6766
3	3	0.9206

PATTERN RULES

ANTECEDENTS		CONSEQUENT (WEIGHT)
1	1	0.3698
2	1	0.5417
3	1	0.5417
1	2	0.5417
2	2	0.4851
3	2	0.5417
1	3	0.5417
2	3	0.5417
3	3	0.7986

SUGENO FUZZY RULES

Jump to first page

UNFIR- Identification of fuzzy rules

- Sugeno tuning process

The consequent is obtained from the values of the antecedents:

$$y = \frac{\sum_{i=1}^{n}\left(\mu_i \bullet \omega_i\right)}{\sum_{i=1}^{n} \mu_i}$$

μ_i : Fire i^{th} rule

ω_i : weight i^{th} rule

n : total number of rules

The tuning consist on adjusting ω_i by iterating through the pattern rule base using gradient descendent method.

$$E = \frac{1}{2}\sum_{k=1}^{ND}\left(y - y^r\right)^2$$

ND : Num. pattern rules

y : Value given

y^r : Output pattern rule

Jump to first page

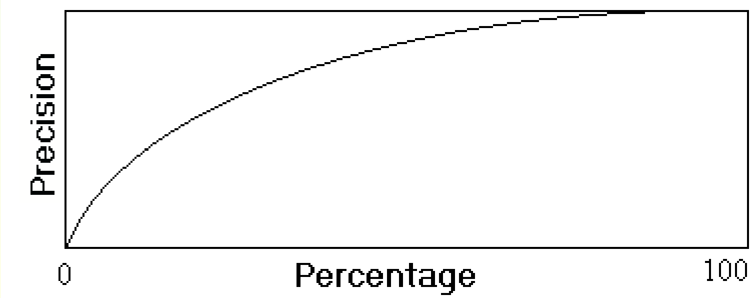
UNFIR - Mixed prediction scheme

- UNFIR methodology allows the use of 5 options to predict the future behavior:
  - Fuzzy forecasting process of FIR methodology
  - Mamdani fuzzy inference system
  - Sugeno fuzzy inference system
  - Mixed Mamdani scheme
  - Mixed Sugeno scheme

Jump to first page

UNFIR - Mixed prediction scheme

- Keeps a percentage of pattern rules that allows the prediction of system states with a high degree of uncertainty.
- Which rules? Those that are located outside of the multidimensional mesh generated by the FIS.
- A 10-25% of pattern rules. The prediction is acceptable and the computational cost is not extremely high.



UNFIR - Mixed prediction scheme

- Generation of two prediction values: (1) selected fuzzy scheme and (2) closest pattern rule.
- The prediction obtained by the mixed scheme is a weighing of both values.
- Computed with respect to the distance between the antecedents of the system state to be predicted and the antecedents of the closest pattern rule.

Jump to first page

UNFIR - Mixed prediction scheme

- Distance between the two sets of antecedents:

$$d_{norm} = \frac{d_{real}}{d_{max}}$$

d_{real}: Real Euclidean distance

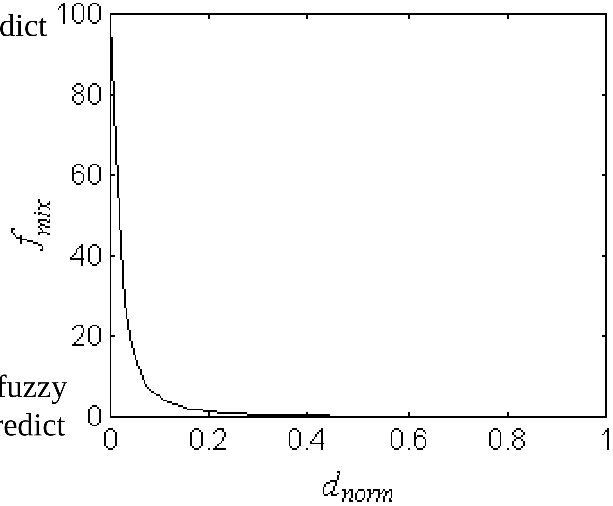
d_{max}: Maximum Euclidean distance

Range: [0.0 1.0]

Integration strategy of the pattern and fuzzy rules:

$$f_{mix} = \frac{1}{1 - e^{-d_{norm}^2}}$$

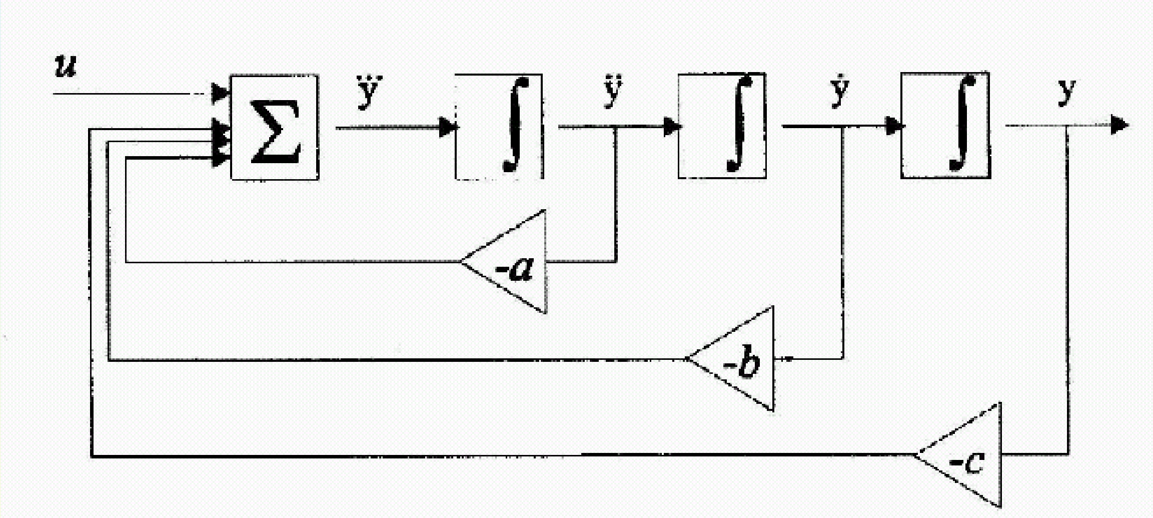
Use only pattern rules to predict



Use only fuzzy rules to predict

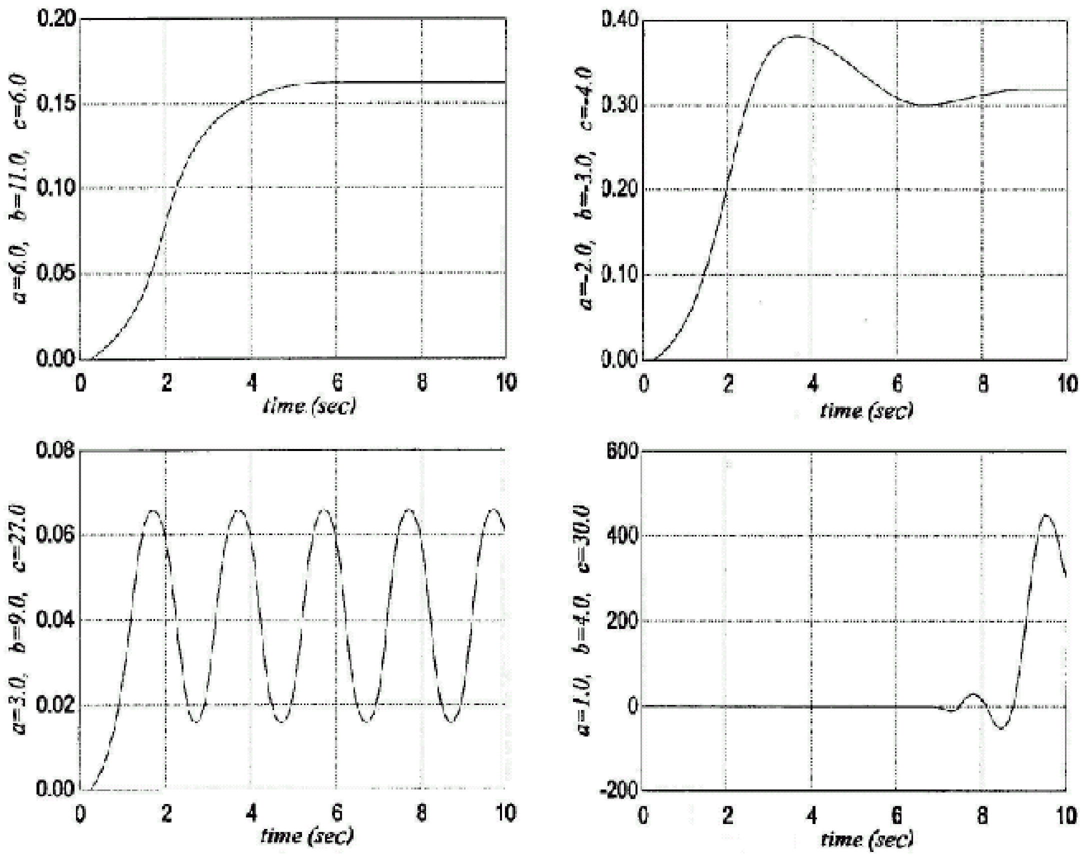
Linear System

- Third-order linear system. SISO.



Linear System

- The system has 4 trajectory behaviors resulting from selections of the parameters *a*, *b* and *c*.



a=-2; b=-3;c=-4

Stabiliz.=9sec.

Excited oscillator

Jump to first page

Linear System

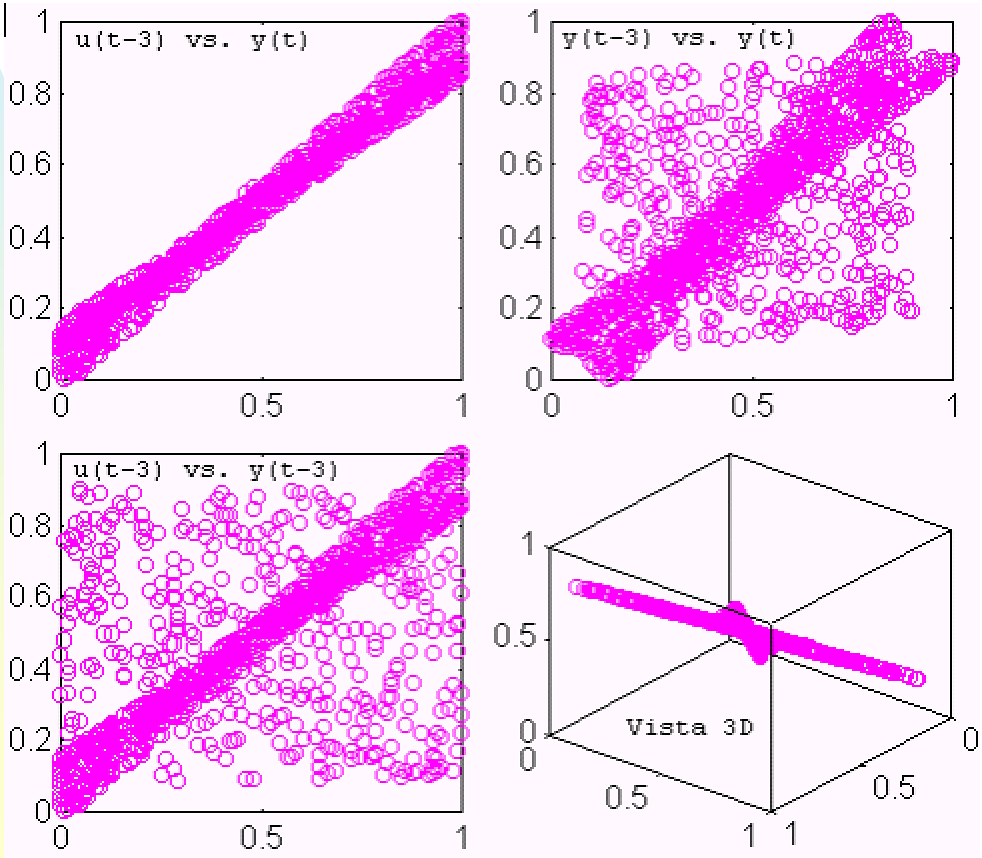
- Data acquisition:
  - Sampling rate of 3 sec.
  - Identification with 1244 samples.
- External parameters:
  - Num. Classes: 3,5,7,9
  - EFP method
  - Gaussian shape
- Qualitative model identification:

t\x	u	y
t-3δt	-1	-2
t-2δt	0	0
t-δt	0	0
t	0	+1

Jump to first page

Linear System

- Pattern rule base with 1242 rules.

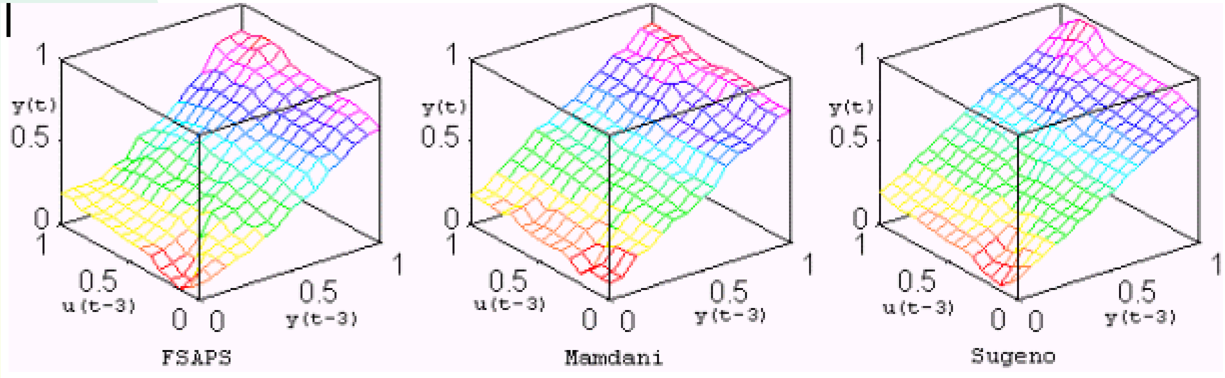


Jump to first page

Linear System

- Mamdani and Sugeno fuzzy rule bases with 81 rules (9 classes for each variable).

- Mamdani and Sugeno tuning process:



As a reference

Jump to first page

Linear System

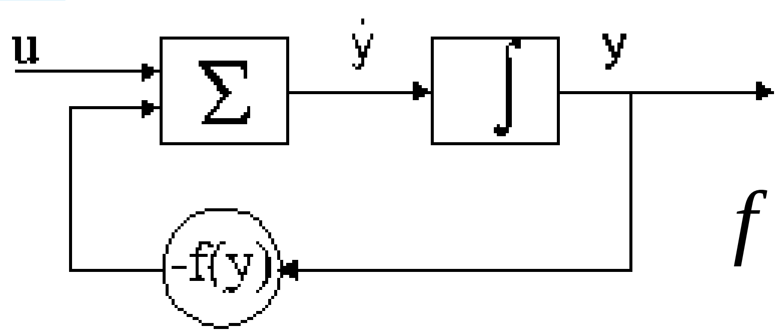
- The five prediction processes of UNFIR have been applied to the linear system.
- A set of 400 data points, not used for identification, have been used for system's prediction.

Errors

Partitions	(3,3)	(5,5)	(7,7)	(9,9)
FIR	0.0249	0.0233	0.0248	0.0373
Mamdani	1.3046	1.1297	1.0582	0.9327
Sugeno	0.3137	0.2119	0.1097	0.0975
Mixed Mamdani	0.8604	0.6277	0.5842	0.4620
Mixed Sugeno	0.6761	0.1103	0.0774	0.0515

Non-linear System

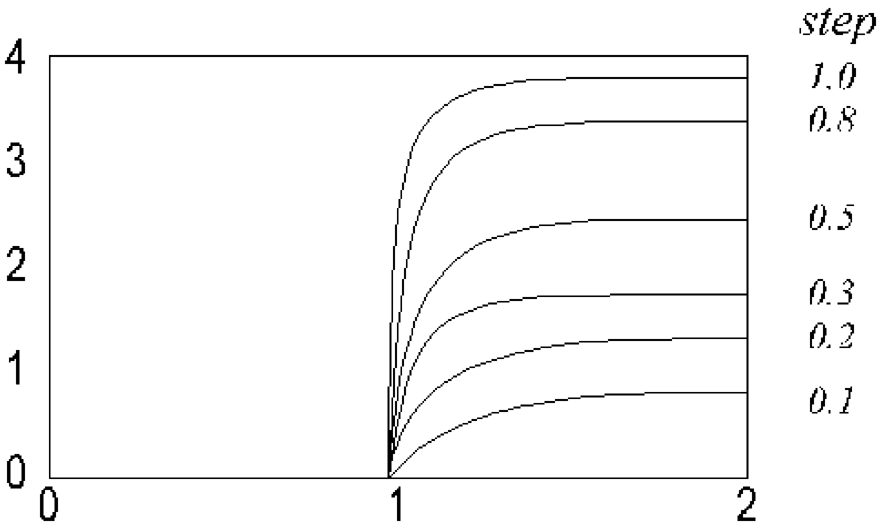
- First-order non-linear system. SISO.



$$f(y) = \alpha + k \cdot \sqrt{abs(y)}$$

α=0.1; *k*=7

System perturbed with different input steps



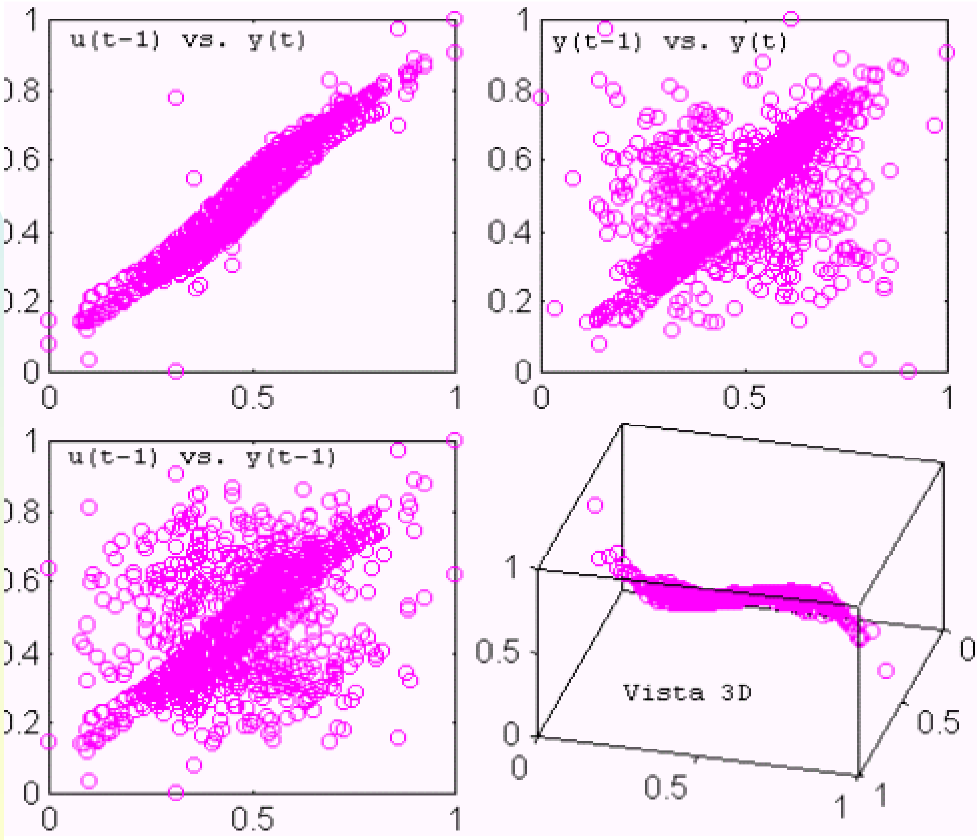
Non-linear System

- Data acquisition:
  - Sampling rate of 1 sec.
  - Identification with 1000 samples.
- External parameters:
  - Num. Classes: 3,5,7,9
  - EFP method
  - Gaussian shape
- Qualitative model identification:

t\x	u	y
t-2δt	0	0
t-δt	-1	-2
t	0	+1

Non-linear System

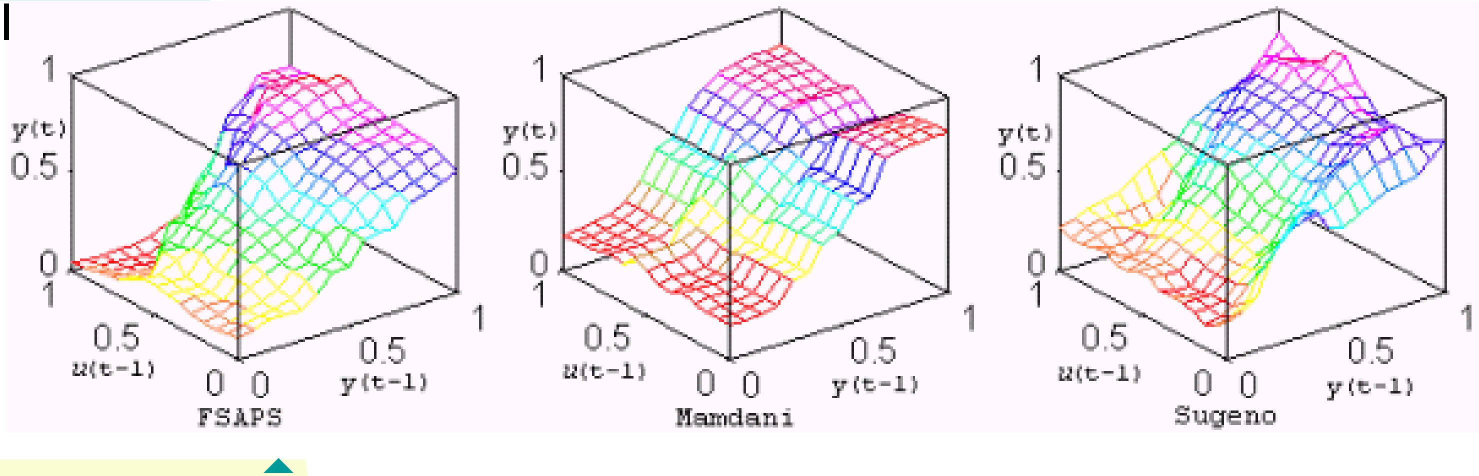
⌘ Pattern rule base with 998 rules.



Non-linear System

- Mamdani and Sugeno fuzzy rule bases with 81 rules (9 classes for each variable).

- Mamdani and Sugeno tuning process:



As a reference

Non-linear System

- The five prediction processes of UNFIR have been applied to the non-linear system.
- A set of 400 data points, not used for identification, have been used for system's prediction.

Errors

Partitions	(3,3)	(5,5)	(7,7)	(9,9)
FIR	0.0388	0.0337	0.0307	0.0293
Mamdani	1.8271	1.3143	1.1104	0.9712
Sugeno	0.6521	0.3829	0.2853	0.2591
Mixed Mamdani	1.1214	0.9713	0.8149	0.6312
Mixed Sugeno	0.4518	0.1970	0.0942	0.0591

Real complex System

- Predict how the air pollution evolves in Mexico city

- The main goal is to predict carbon monoxide from the variables:
  - Year (Y)
  - Hour of day (HRA)
  - Day of week (DW)
  - Day of month (DM)
  - Wind velocity (WV)
  - Wind direction (WD)
  - Month of year (MY)
  - Temperature (T)
  - Relative Humidity (RH)
  - Carbon monoxide (CO)

Real complex System

- Data acquisition:
  - Sampling rate of 1 hour
  - Identification data January-March from 1994-1996
- External parameters:
  - Num. Classes: 3,5,7,8,9
  - EFP method
  - Gaussian shape
- Qualitative model identification:

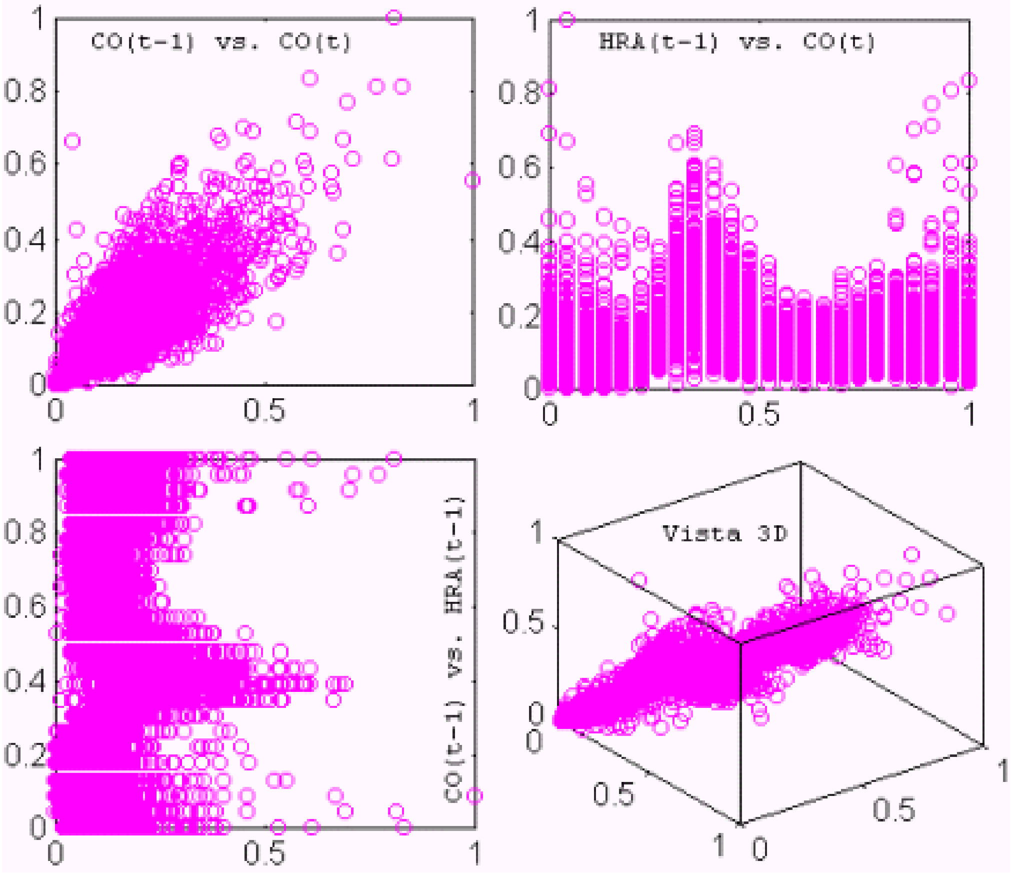
t\x	DW	HRA	DM	MY	Y	WV	WD	T	RH	CO
$t-2\delta t$	0	0	0	0	0	0	0	0	0	0
$t-\delta t$	-1	-2	0	0	0	0	0	0	0	-3
t	0	0	0	0	0	0	0	0	0	+1

-2, -3, +1: Sub-optimal mask

Jump to first page

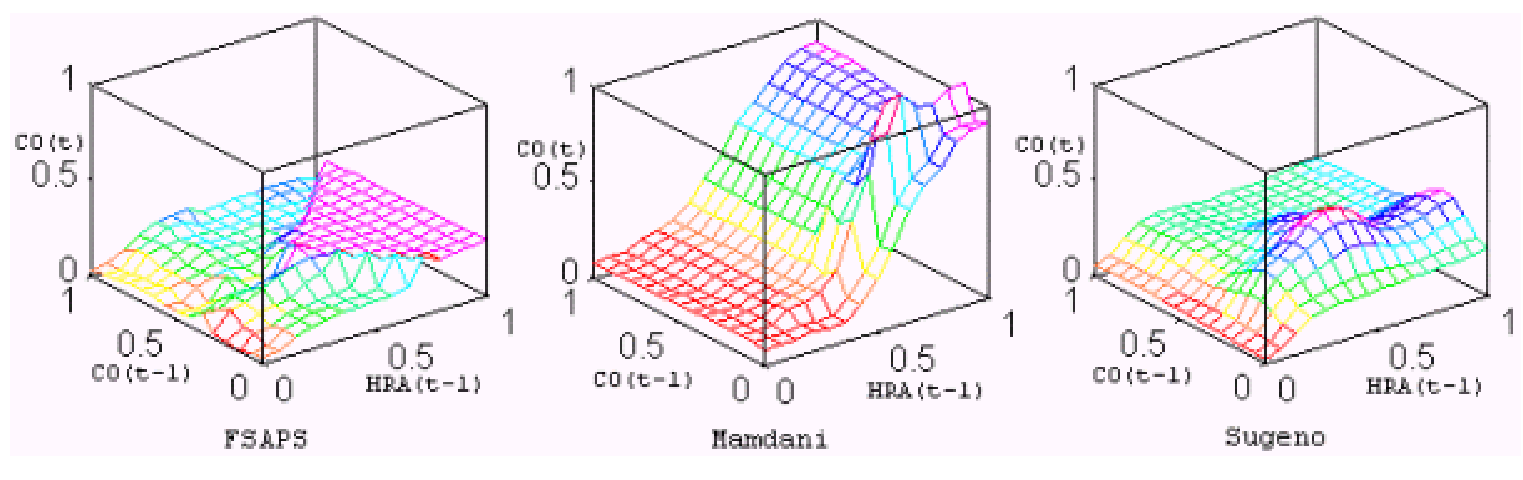
Real complex System

⌘ Pattern rule base with 6477 rules.



Real complex System

- Mamdani and Sugeno fuzzy rule bases with 72 rules (8,9,9 classes).
- Mamdani and Sugeno tuning process:



As a reference

Real complex System

Prediction errors using the sub-optimal mask

Partitions	(8,3,3)	(8,5,5)	(8,7,7)	(8,9,9)
FIR	0.3629	0.3584	0.3562	0.3553
Mamdani	2.0634	1.8470	1.6674	1.2189
Sugeno	1.7262	1.3555	1.2189	1.1795

Prediction errors using the optimal mask

Partitions	(7,8,3,3)	(7,8,5,5)	(7,8,7,7)	(7,8,9,9)
FIR	0.2493	0.2471	0.2433	0.2392
Mamdani	1.4841	1.3020	1.1635	1.0887
Sugeno	1.2902	1.0395	0.9501	0.9073
Mixed Mamdani	1.2158	1.1127	1.0465	0.9712
Mixed Sugeno	1.0624	0.9385	0.8167	0.7723

Jump to first page