Evolvable Fuzzy Systems

Plamen Angelov

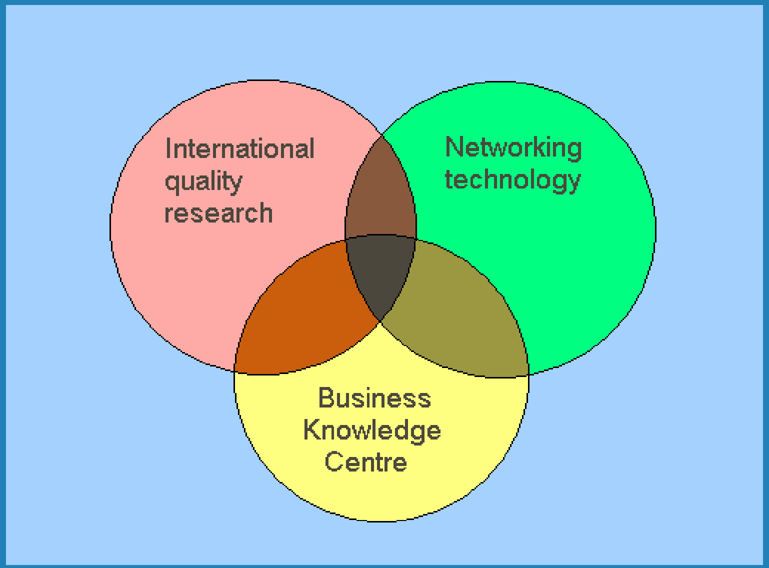
Intelligent Systems Research Laboratory
Dept of Communications Systems
InfoLab21, Lancaster University, UK

Lancaster University

One of the top 10% of the UK Universities

InfoLab21

- The Comms Systems and Computing Depts form an **ICT force of 250** researchers
- Research project income from Industry (Nokia, Philips, BAE Systems, QInetiq, Ford), Government (DTI, EPSRC), EC, Royal Society etc. £15M



DSP group

Research is focussed on the development of novel techniques in

- system modelling/identification
  fuzzy rule based systems
- Intelligent collaborative Systems
- speech/image enhancement, compression, analysis and synthesis
- information fusion
- NOKIA Lab
- Intelligent Systems Lab

Intelligent Systems Lab

Collaborative mobile robots (5 Pioneer-3DX) with evolvable intelligence using embedded eTS systems for: prediction, classification and control

Outline

- Models with Flexible Structure
- Evolving Neuro-Fuzzy Models (eTS)
- NOx emissions real-time modelling (DC)
- Quality of crude oil distillation (CESPA)
- Applications to speech processing (Nokia)
- Autonomous vehicles (BAE, Qinetiq, J&S)
- System on chip implementation (FPGA)

Outline

- Controllers with evolvable structure
- Application to EEG signals classification
- Classification of Carcinoma Kidney Tissue Status based on Protein Expression Data
- Biotech process applications

System Modeling

- predict object reactions;
- control it;
- detect faults;
- study process performance

A. *Conventional* Models

- First Principles Models

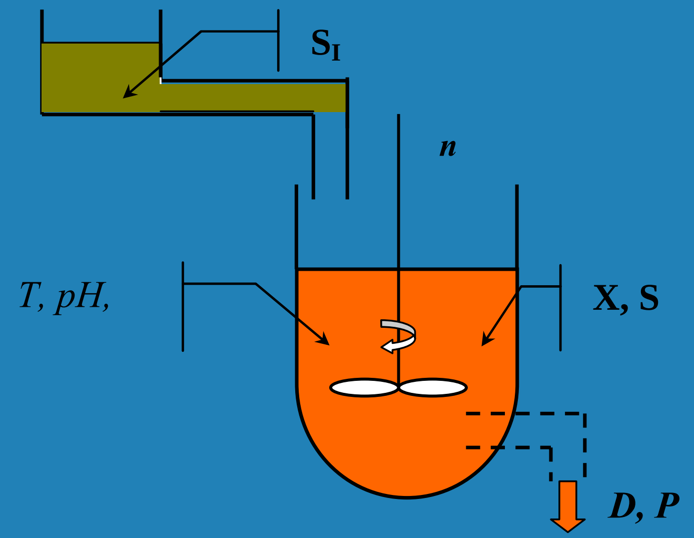
- Black-box models

B. Fuzzy Models

Fermentation Process

A. First Principle Models

- Example: Fermentation process
  - ✓ **transparent, close to nature** (mass- and energy conservation in closed systems)
  - ✗ **tedious**, even impossible, (highly) non-linear



$$\frac{dX}{dt} = \mu_X X - DX$$

$$\frac{dS}{dt} = -q_S X + D(S_I - S)$$

$$\frac{dP}{dt} = q_P X - DP$$

Black-box Models

- Linear state-space models

$$x(k+1) = Ax(k) + Bu(k) \qquad y(k) = Cx(k) + Du(k)$$

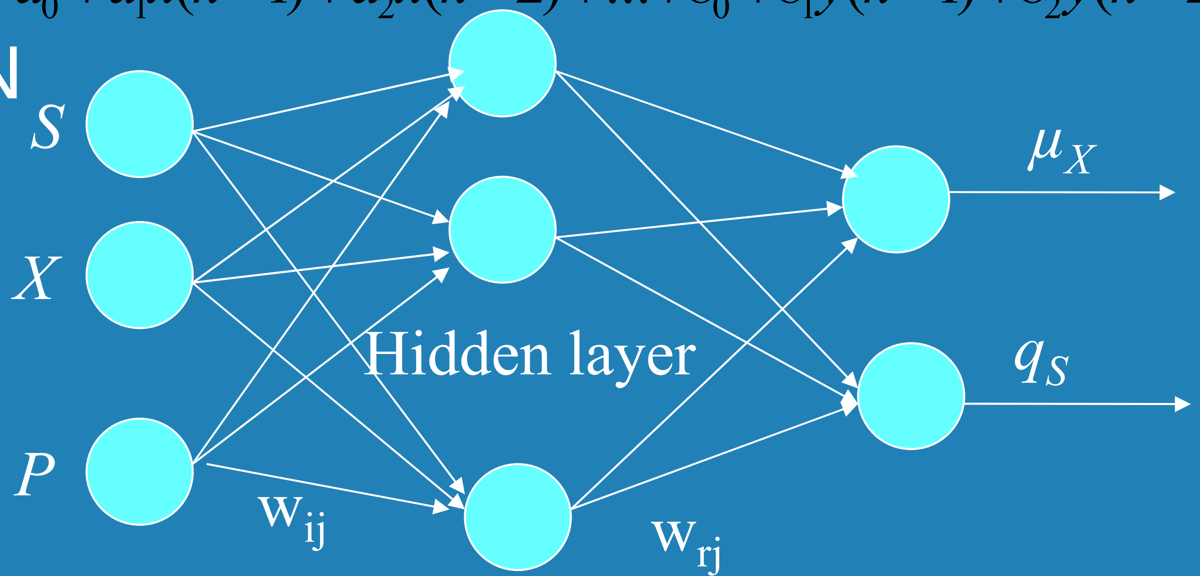
- Polynomial models

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 + a_6x_1^2x_2^2 + \dots$$

- ARMA models

$$y(k) = a_0 + a_1x(k-1) + a_2x(k-2) + \dots + b_0 + b_1y(k-1) + b_2y(k-2) + \dots$$

- ANN



Fuzzy Model Types

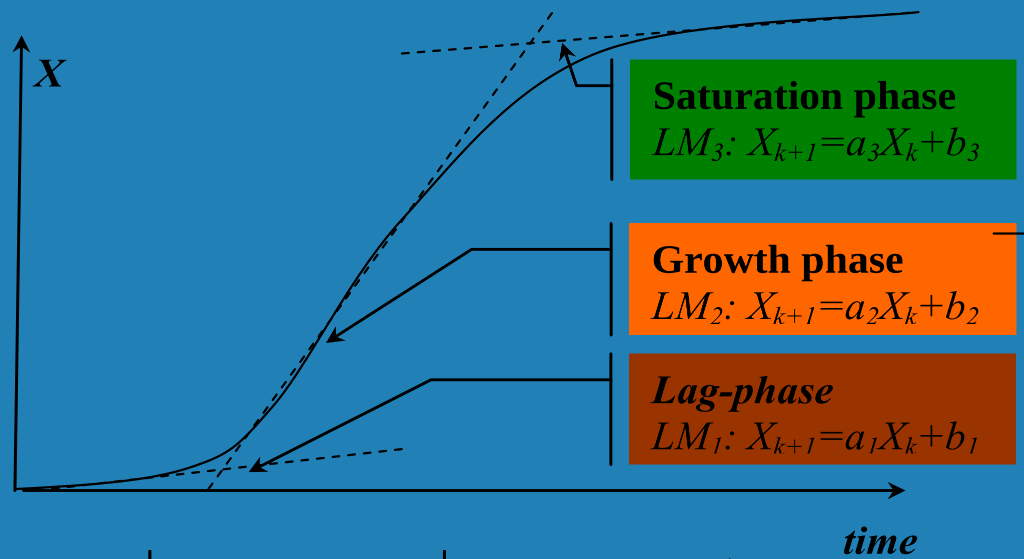
- Fuzzy parameters $\psi = \tilde{a}_0 + \tilde{a}_1 \phi + \tilde{a}_2 \phi^2 + \tilde{a}_3 \phi^3 + \tilde{a}_4 \phi^4$
- Fuzzy (in)equalities $x_{k+1} \tilde{=} f(x_k, u_k)$
- **FRB models** ***IF*** *(antecedent)* ***THEN*** *(consequence)*
  - relational $y = x \circ R$ $\quad R = \bigcup_{i=1}^{RN} R_i$
  - Mamdani $R_i : IF\,(x_1 is X_1)\,AND \ldots AND\,(x_n is X_n)$ $THEN\,(y.is.Y)$
  - **Takagi-Sugeno or TSK**

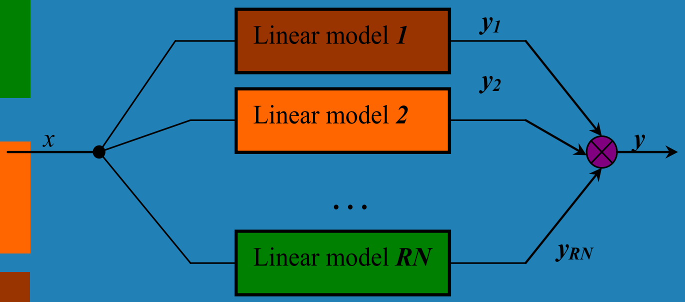
TSK Models (1985)

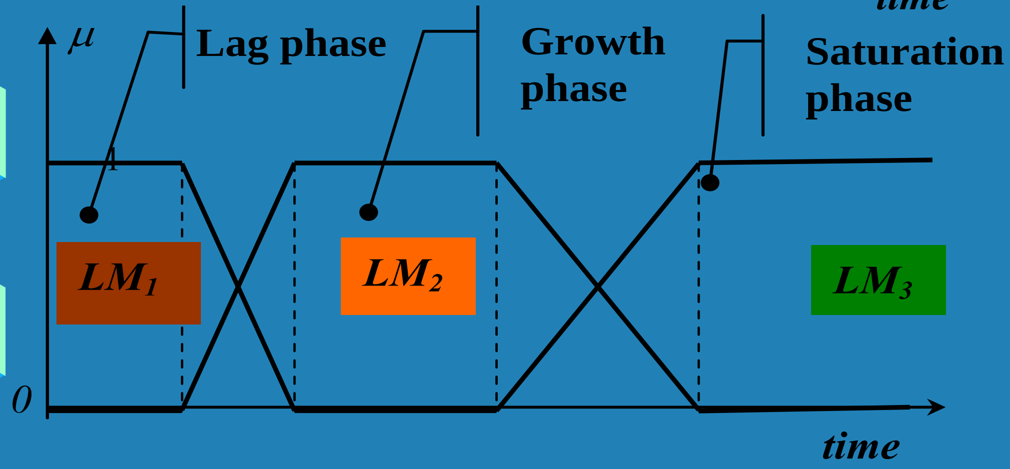
- TSK systems – important tool for system modeling and identification
  - ✓ Computational efficiency (local linearity)
  - ✓ Universal approximators
  - ✓ Good transparency
  - ✓ Convenient for data-driven design

$$R_i : IF(x_1 is X_1) AND ... AND(x_n is X_n)$$
$$THEN(y_i = a_{i1}x_1 + ... + a_{in}x_n + b_i)$$

TSK Fuzzy model of a Fermentation (concept)

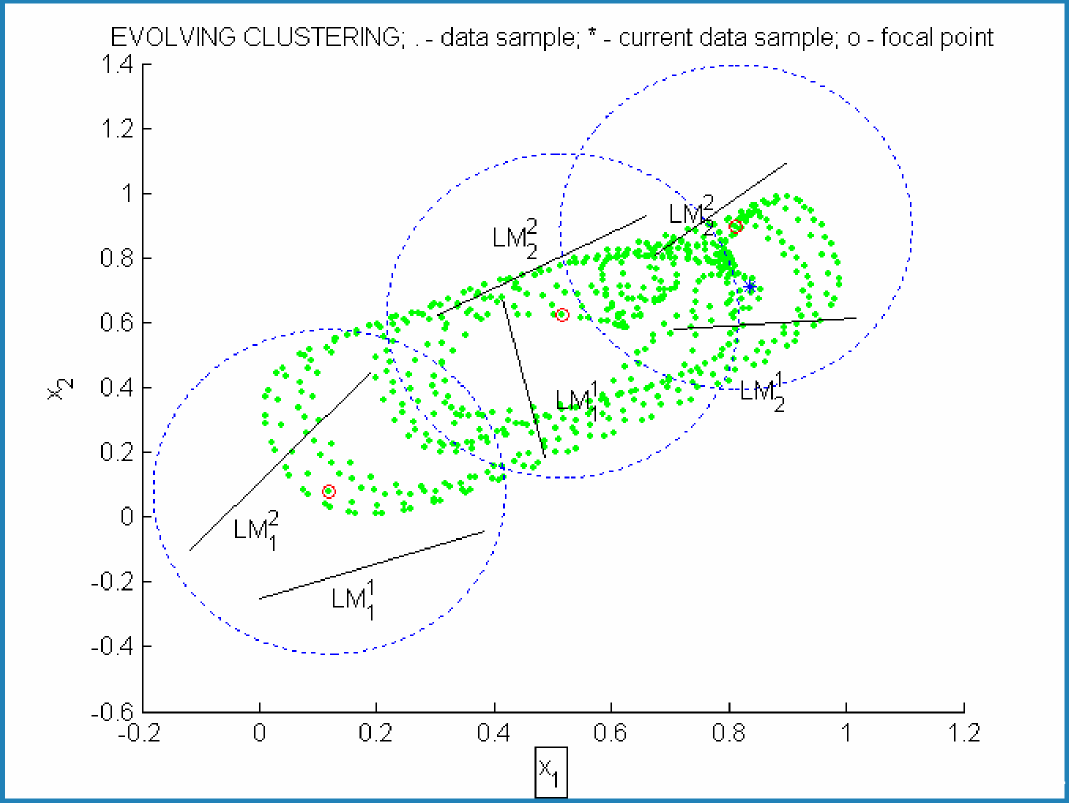






TSK in 2D Feature Space

- MIMO TSK in a 2D feature space
- eClustering



Data-driven learning

- Until 1990s fuzzy systems were designed based on ‘expert’ knowledge
- Data-driven design (’95) can include expert knowledge if it exists, but tries to extract knowledge from the data
- Recent tendency – data streams, on-line, real-time processes

The challenge

- Systems that posses *Computational Intelligence* usually rely on fixed rule-bases or NN
- Trained off-line, do not adapt to environment
- They do not develop their structure (evolve)

Example 1 current UAVs

Unmanned Aerial Vehicles (UAV):

- limited flexibility
- limited control functions
- do not learn the new environment

Herta-1A UAV

Flew 08/18/06 →

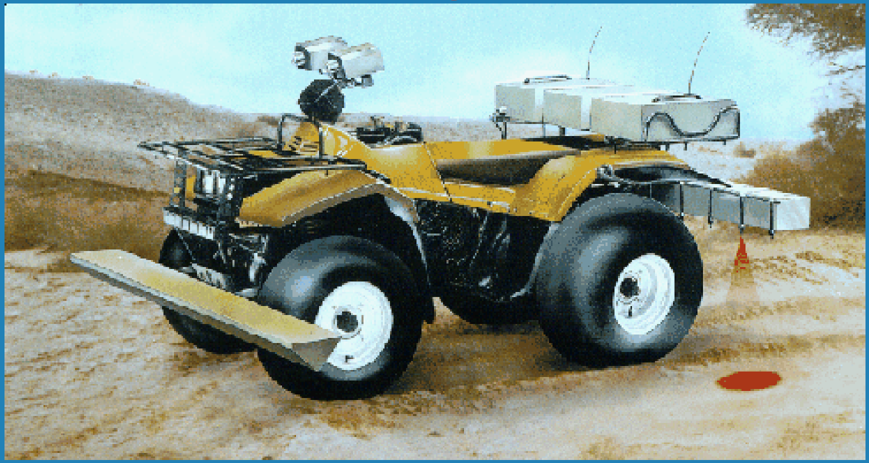
over Scotland

Example 2 Mobile robots

Mobile robots:

- Pre-programmed logic
- Remotely controlled vehicles
- Limited learning capabilities
- Do not capture new knowledge

the de-miner ELTA →



The challenge

- The environment in which real systems (technological processes, robotic systems, transport vehicles) operate is (unpredictably) changing
- The challenge - to develop systems capable of higher level adaptation to the environment and to internal changes (wearing, faults, regimes etc.)

On-line identification

- What to do when **new data do not fit** into the model with a chosen structure?
- Adaptive systems theory answer ('70s): adapt the parameters ONLY
- This may be an **outlier**
- Or it may **bring new information (knowledge)** – about a different regime, operation point, change etc.
- Thus update the **structure**

Evolving Systems

- Evolving systems – a possible solution

☑ Evolving is adaptive in terms of both structure and parameters

☑ Incremental evolution of the fuzzy rules (clusters): update, replace, add new

What to evolve?

- Consequent parameters (parameters of linear sub-models);
- Premise parameters (centers and widths of the Gaussians);
- Rule-base (rules, fuzzy sets/linguistic terms);

TSK MIMO model

$$x = [x_1, x_2, \ldots, x_n]^T \qquad y = [y_1, y_2, \ldots, y_m]^T$$

$$R^i : IF\left(x_1.is.closeto.x_1^{i*}\right) AND \ldots AND\left(x_n.is.closeto.x_n^{i*}\right)$$

$$THEN(y^i = f^i) \qquad f^i = x_e^T \pi^i \qquad x_e^T = [1, x^T]$$

$$f^i = a^i$$

$$a^i = \left[\alpha_{01}^i \quad \alpha_{02}^i \qquad \alpha_{0m}^i\right]^T$$

$$\pi^i = \begin{bmatrix} \alpha_{01}^i & \alpha_{02}^i & \ldots & \alpha_{0m}^i \\ \alpha_{11}^i & \alpha_{12}^i & \ldots & \alpha_{1m}^i \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_{n1}^i & \alpha_{n2}^i & \ldots & \alpha_{nm}^i \end{bmatrix}$$

TSK MIMO model

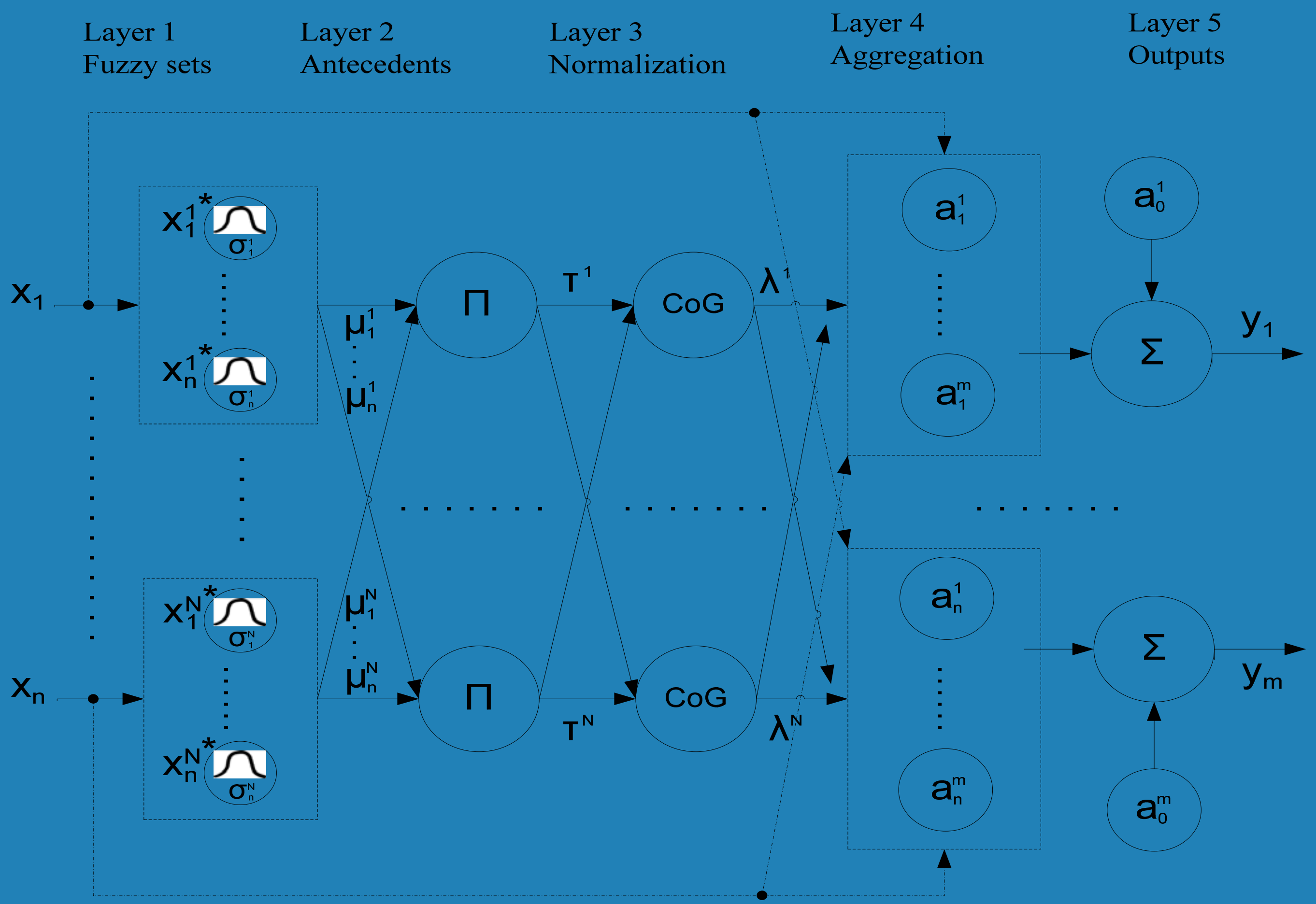
$$y = \sum_{i=1}^{N} \lambda^i y^i \qquad \lambda^i = \frac{\tau^i}{\sum_{j=1}^{N} \tau^j}$$

$$\tau^i = \prod_{j=1}^{n} \mu_j^i (x_j)$$

$$\lambda^i = \begin{cases} \tau^j & j = \arg\max_{l=1}^{N} \left\{ \tau^l \right\} \\ 0 & else \end{cases}$$

$$\mu_j^i = e^{-\frac{4 \left\| x - x^{i*} \right\|_j^2}{\left(\sigma_j^i \right)^2}}$$

TSK model as a FBFN



eTS systems (2002)

- ✓ Adapt parameters and evolve structure;
- ✓ eTS – eClustering (on-line version of the subtractive clustering) + a version of RLS
- ✓ Applied so far to control, prediction, classification, speech error recovery etc.
- ✓ Fuzzy rules and linguistic terms are not fixed and learning can start 'from scratch'

InfoLab21
Department of Communication Systems

eTS systems (2002)

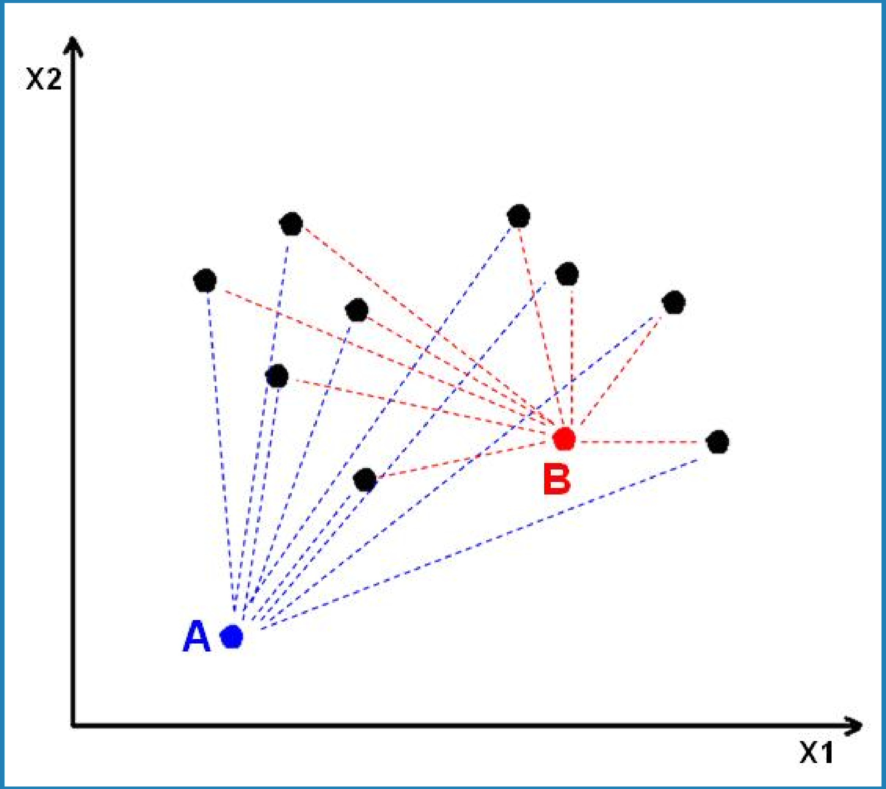
Basic Principle

The approach can be summarised as:

- ✓Decomposition of the complex data space into overlapping local regions eClustering.avi
- ✓Joint identification of local (simpler) sub-systems in real-time
- ✓Forming the output of the system as a fuzzy blending of local outputs

The concept of Potential

Key notion – spatial proximity in the input/output data space



$$P_A = \frac{1}{1 + 1/(k_A - 1)\sum_{i=1}^{k_A} d_{A,i}}$$

$$P_B = \frac{1}{1 + 1/(k_B - 1)\sum_{i=1}^{k_B} d_{B,i}}$$

$$P_A < P_B$$

Potential update

- Recursively calculated

$$\alpha_k = \sum_{j=1}^{n+m} \left(z_k^j\right)^2 \qquad \gamma_k = \sum_{j=1}^{n+m} z_k^j \Gamma_k^j$$

$$P_k(z_k) = \frac{k-1}{(k-1)(1+\alpha_k)+\beta_k-2\gamma_k}$$

$$\beta_k = \sum_{i=1}^{k-1} \sum_{j=1}^{n+m} \left(z_i^j\right)^2 \qquad \Gamma_k^j = \sum_{i=1}^{k-1} z_i^j$$

$$\beta_k = \beta_{k-1} + \alpha_{k-1}; \beta_1 = 0 \qquad \Gamma_k^j = \Gamma_{k-1}^j + z_{k-1}^j \qquad \Gamma_1^j = 0$$

- Centers’ scatter up-date:

$$P_k(z^*) = \frac{(k-1)P_{k-1}(z^*)}{k-2+P_{k-1}(z^*)+P_{k-1}(z^*)\sum_{j=1}^{n+m}\left\|z^* - z_{k-1}\right\|_j^2}$$

Rule-base evolution

- Form new rule:

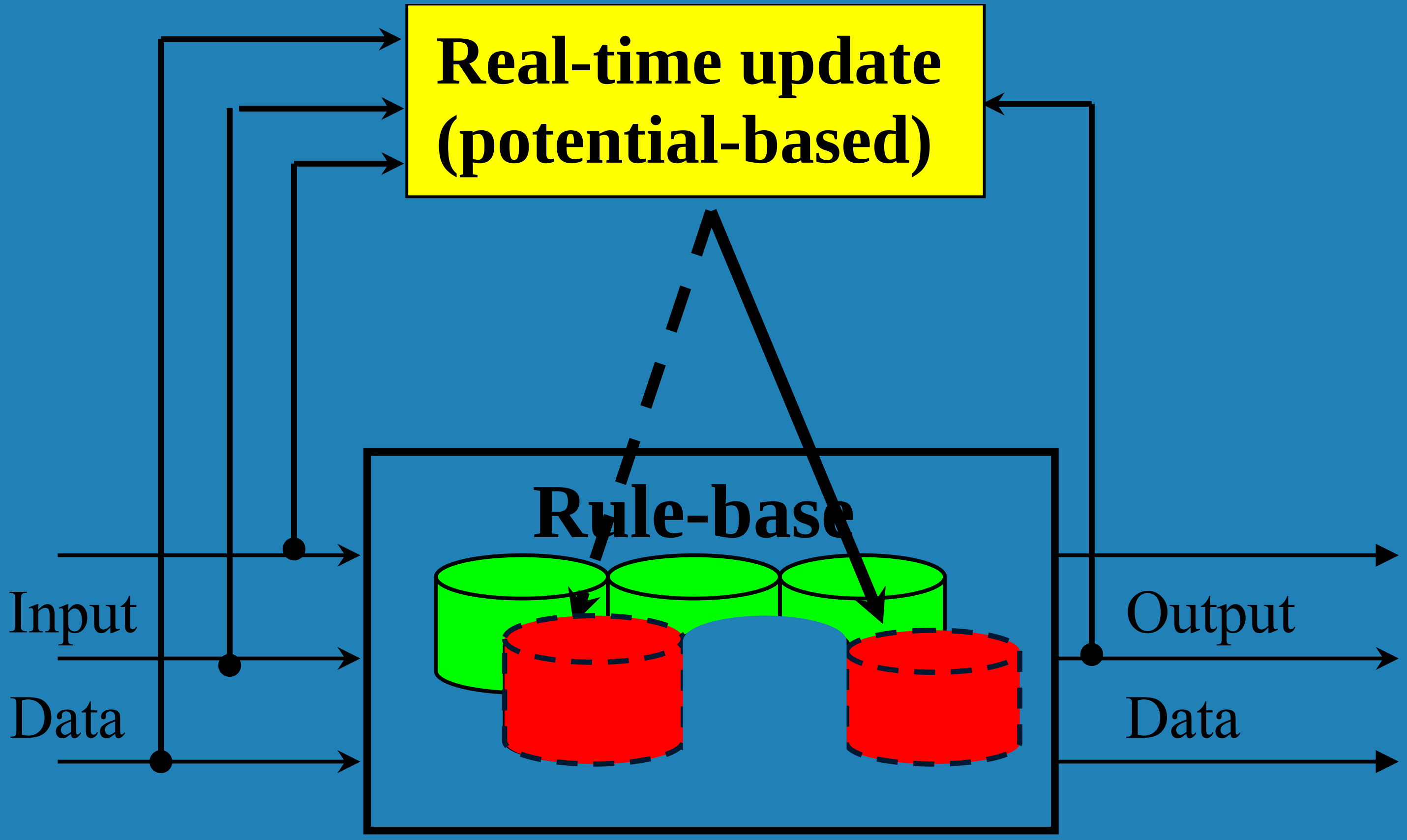
$$P_k(z_k) > \overline{P}$$

- Modify the rule:

$$\min_{i=1}^{N} \left\| x_k - x^{*i} \right\|_j < \frac{\sigma_j^i}{2}$$

- Remove – based on similarity

Rule-base evolution



Learning

$$y = \psi^T \theta$$

$$\theta = \left[\left(\pi^1\right)^T, \left(\pi^2\right)^T, \ldots, \left(\pi^N\right)^T \right]^T$$

$$\psi = [\lambda^1 x_e^T, \lambda^2 x_e^T, \ldots, \lambda^N x_e^T]^T$$

Identification criteria:

$$\left(Y - \Psi^T \theta\right)^T \left(Y - \Psi^T \theta\right) \rightarrow \min$$

Parameters learning

- Weighted RLS

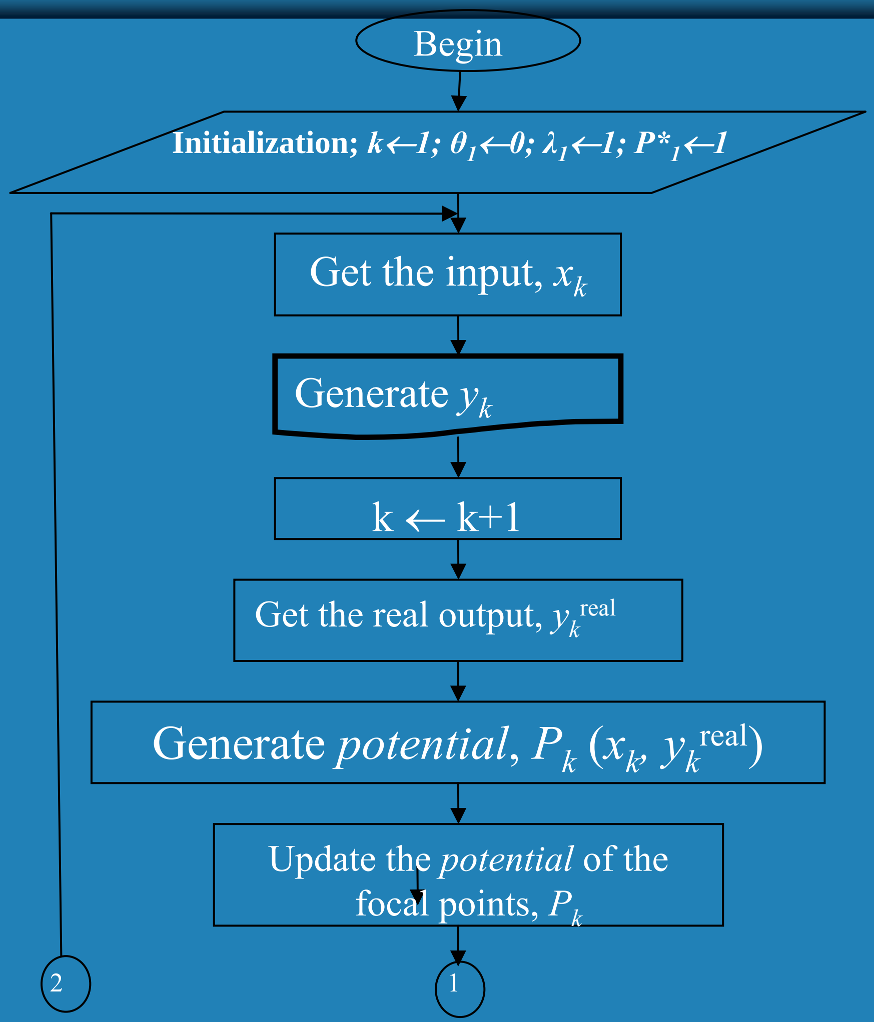
$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_k (y_k - \psi_k^T \hat{\theta}_{k-1})$$

$$C_k = C_{k-1} - \frac{C_{k-1}\psi_k \psi_k^T C_{k-1}}{1 + \psi_k^T C_{k-1} \psi_k}$$

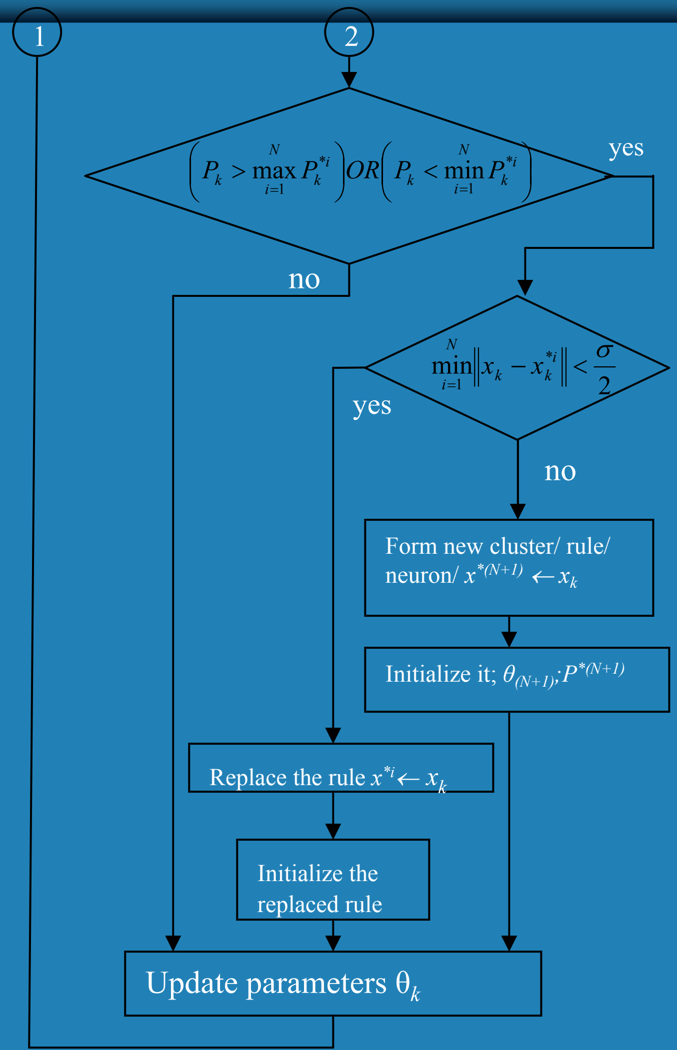
Basic procedure

1. First data - first rule center
2. Collect new data in real-time
3. Calculate S^{new}
4. Recursively up-date S^*
5. Up-grade or modify the Rule-base
6. Estimate Consequent parameters
7. Form the final output

Flow-Chart – part 1



Flow-Chart – part 2



eTS Systems Analysis

- Single algorithmic parameter – cluster radius
- evolutionary (inherits previous model structure), changes are gradual
- extracts accumulated spatial proximity information from the data;
- it is very robust and naturally excludes outliers, because their *S* is high

InfoLab21
Department of Communication Systems

Applications: Modeling

- NOx emissions real-time modelling (Daimler-Chrysler)
- Quality modelling of crude oil distillation (CESPA)
- Lost packets estimation in VoIP (Nokia)
- Autonomous systems (BAE, Qinetiq, J&S)
- Fermentation processes on-line modeling
- System on chip implementation (FPGA)

Daimler Chrysler

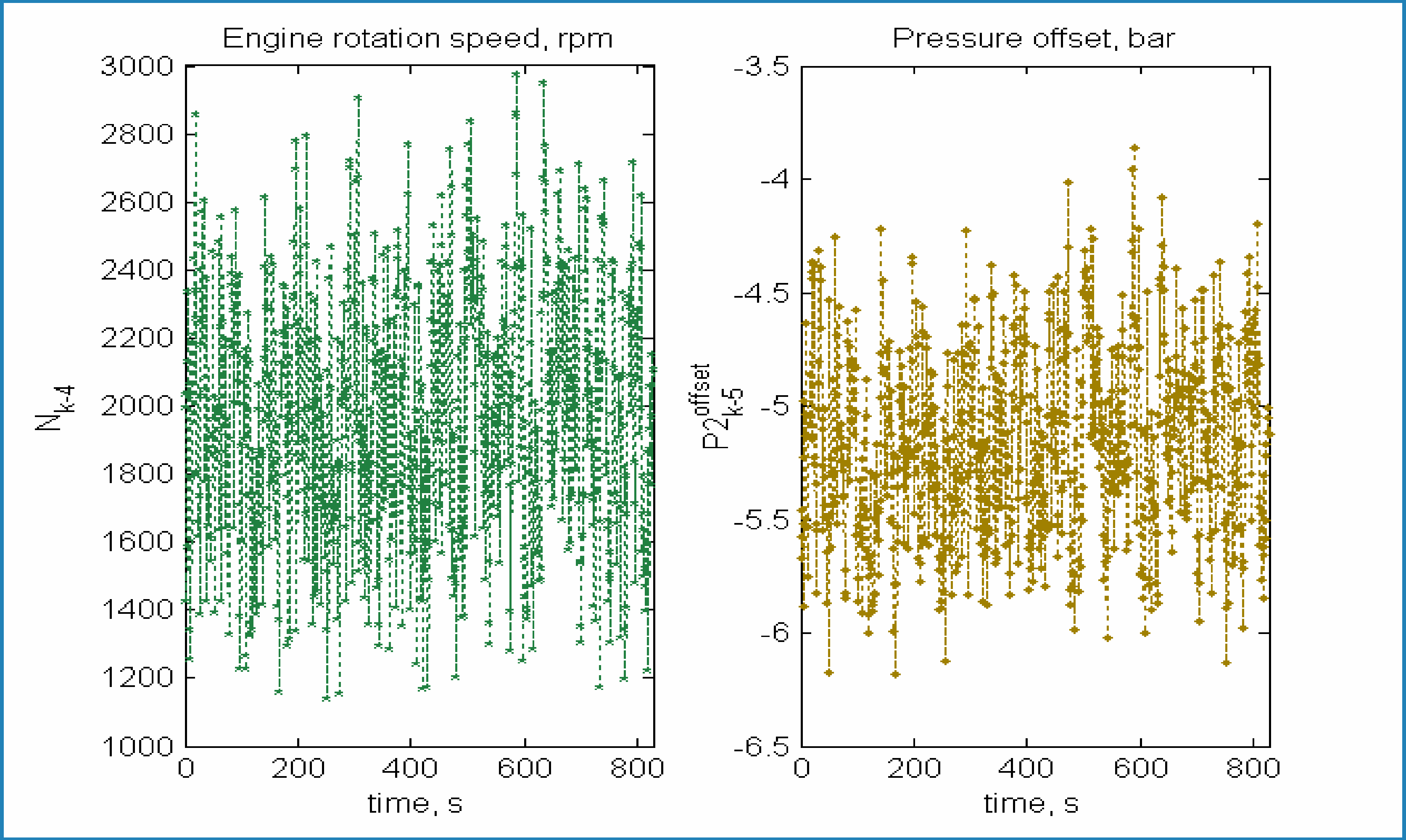
- ✓ NOx emissions modeling for a car engine (Daimler-Chrysler test engines, Dr. E. Lughofer)
- ✓ NOx – 4s ahead prediction
- ✓ 667 training+824 testing samples at 1 Hz

NOx: variables

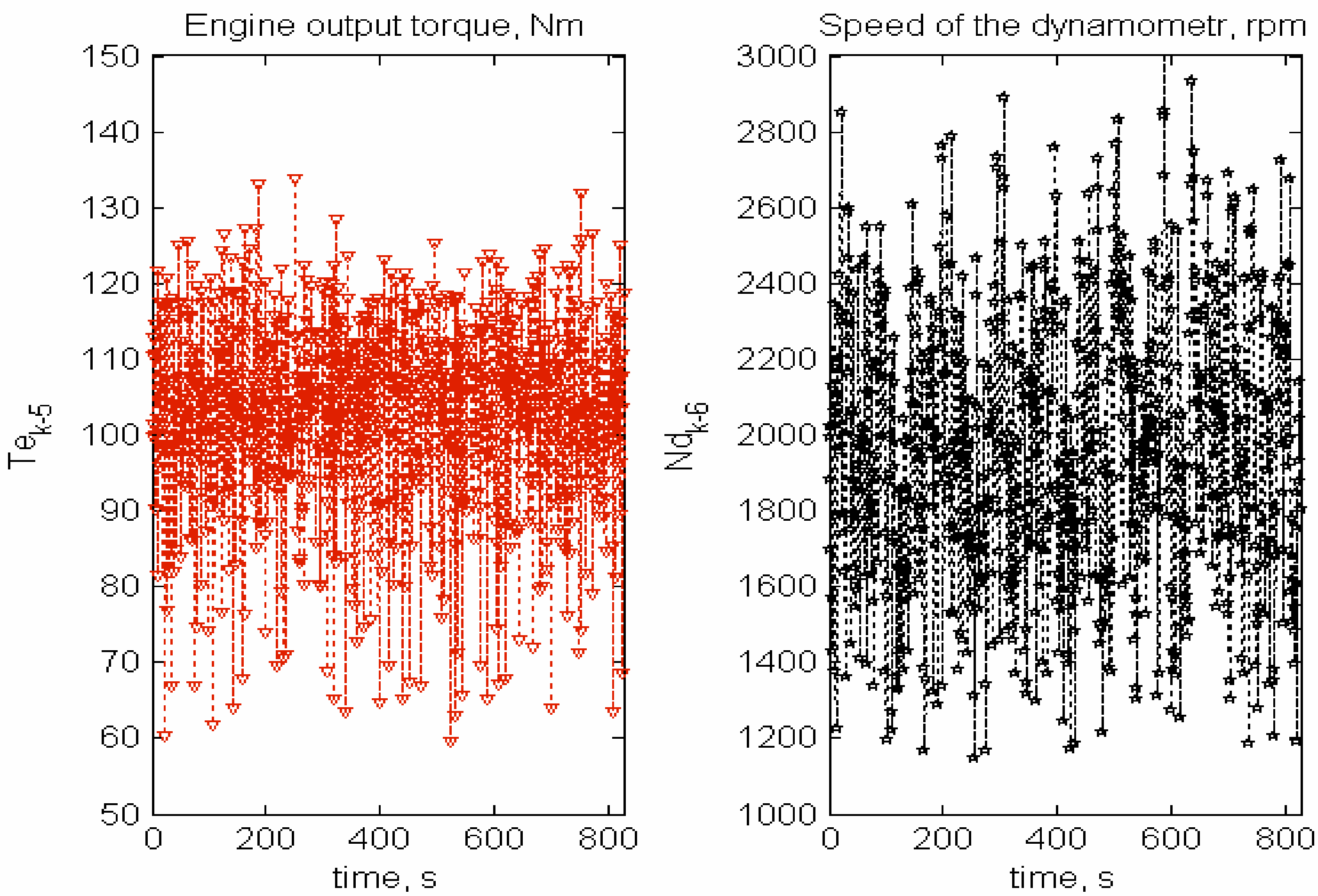
- 4 input attributes:
  - N - engine rotation speed, *rpm*
  - P_2 - pressure offset in cylinders, *bar*
  - T_e - engine output torque, *Nm*
  - N_d – speed of the dynamometer

$$NOX(k) = f(\mathrm{N}(k-4), \mathrm{P2offset}(k-5), \mathrm{Te}(k-5), \mathrm{Nd}(k-6), \mathrm{N}(k-6))$$

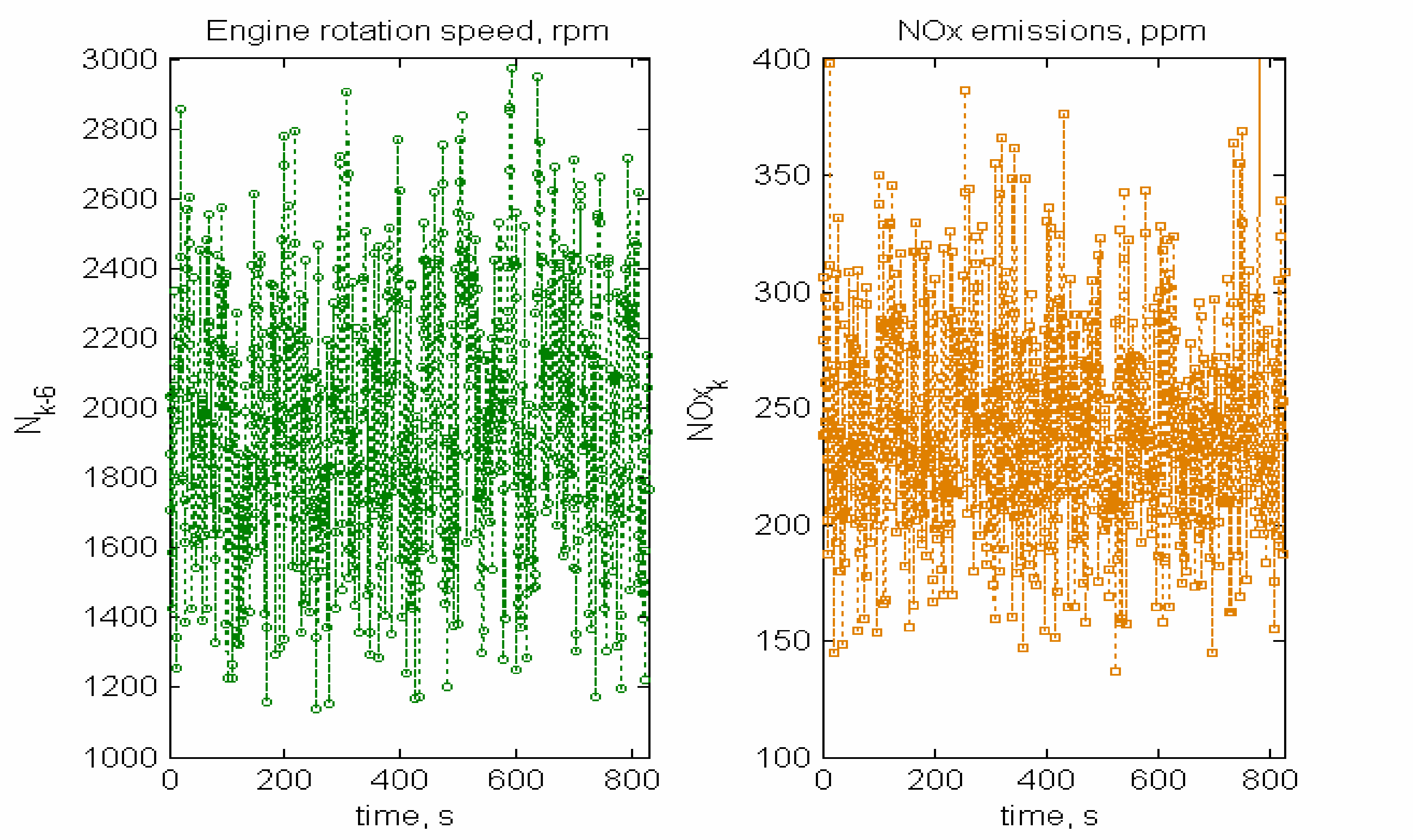
Data, N_{k-4} and $P2_{k-5}$



Data, Te_{k-5} and Nd_{k-6}

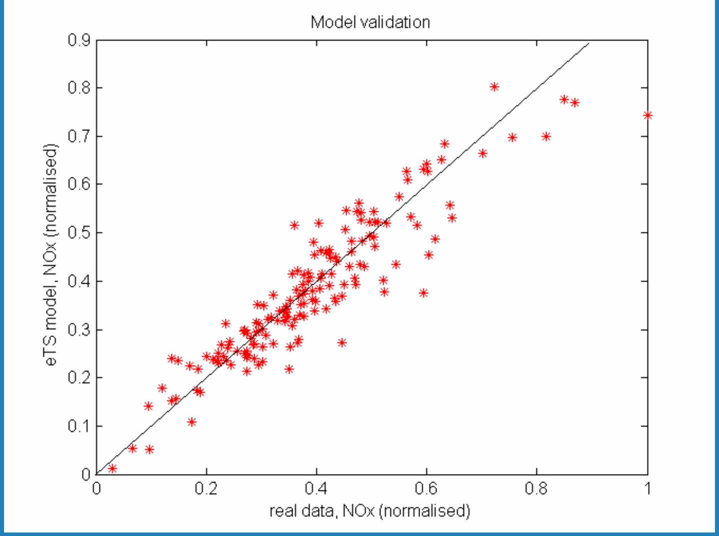


Data, N_{k-6} and NOx_k

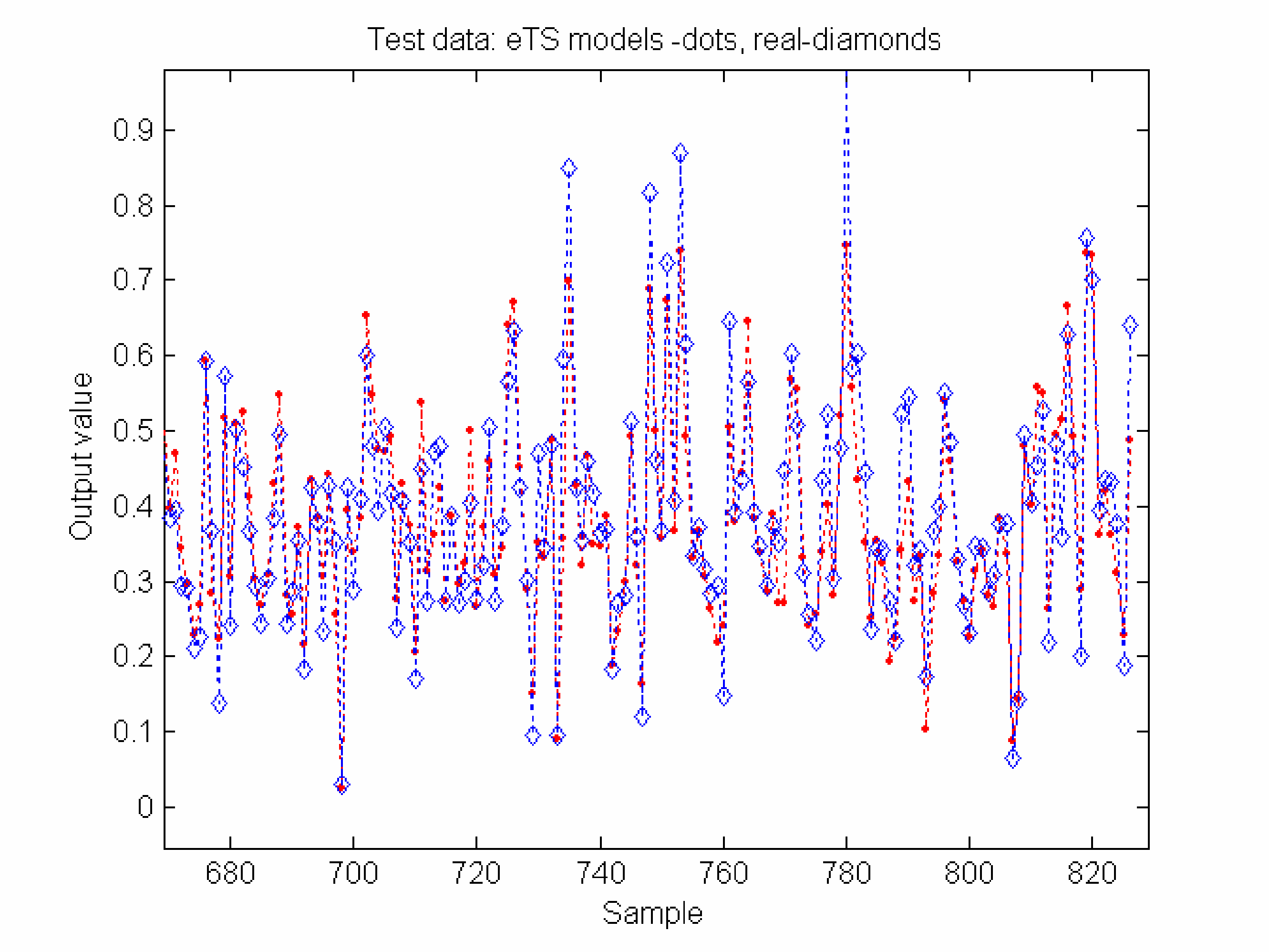


NOx accuracy (correlation)

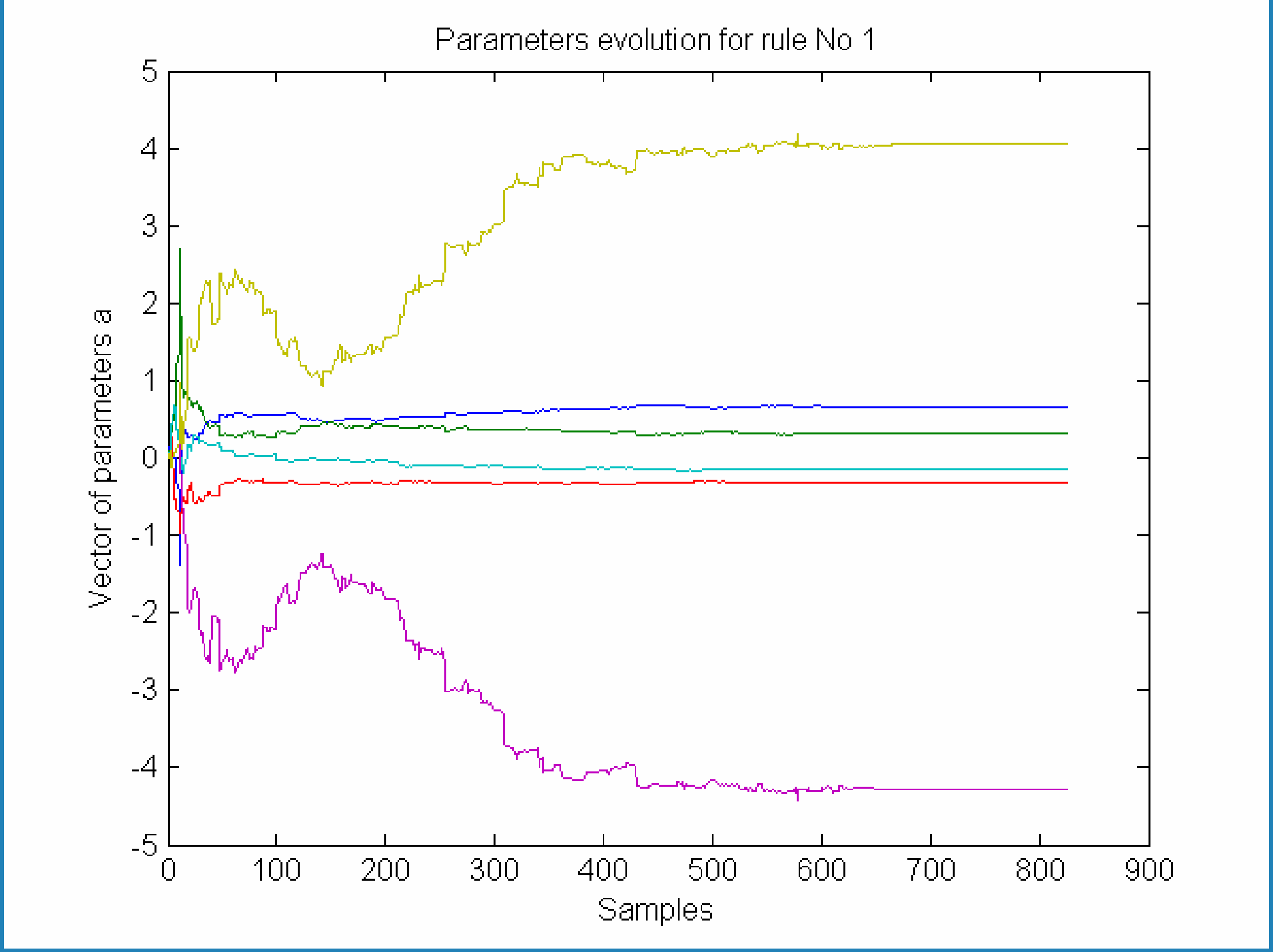
Method	Quality 3 features / No. of Rules / CPU, s	Quality 4 features / No. of Rules / CPU, s	Quality 5 features / No. of Rules / CPU, s
eTS	0.904 / 4 / 0.81	0.906 / 4 / 0.80	0.915 / 3 / 0.76
FLEXFIS	0.892 / 5 / 2.23	0.903 / 5 / 2.76	0.911 / 5 / 3.18



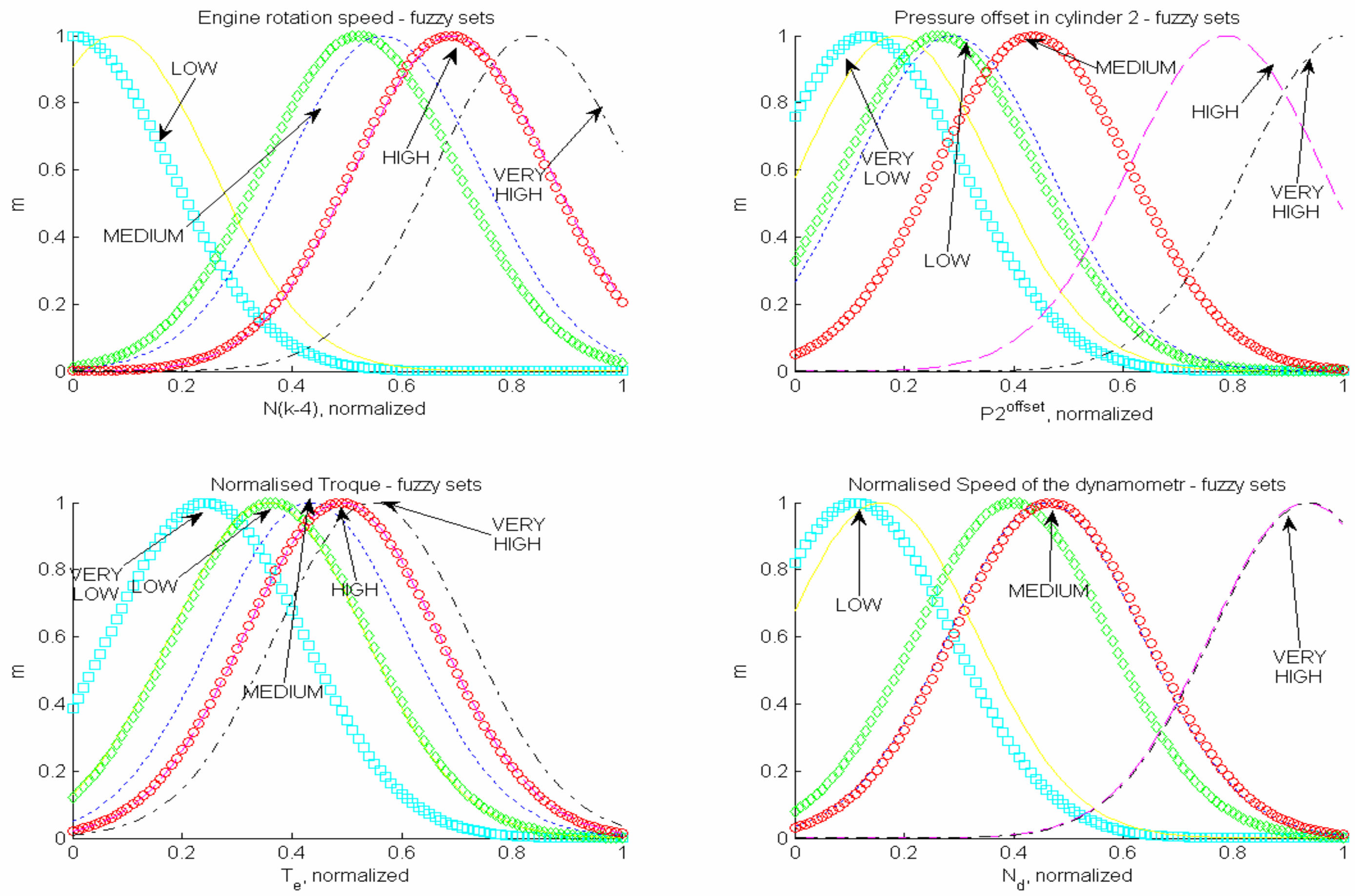
NOx Prediction



eTS parameters



Fuzzy Sets


Engine rotation speed - fuzzy sets
LOW
MEDIUM
HIGH
VERY HIGH
m
N(k-4), normalized
Pressure offset in cylinder 2 - fuzzy sets
MEDIUM
HIGH
VERY LOW
LOW
VERY HIGH
P2offset, normalized
Normalised Troque - fuzzy sets
VERY HIGH
VERY LOW
LOW
HIGH
MEDIUM
Te, normalized
Normalised Speed of the dynamometr - fuzzy sets
LOW
MEDIUM
VERY HIGH
Nd, normalized

Fuzzy Rules

R_1: **IF**(N_{k-4} is Medium) **AND** ($P_2^{offset}{}_{k-5}$ is Low) **AND** (Te_{k-5} is High) **AND** (Nd_{k-6} is ...) **AND** (N_{k-6} is Medium)

THEN $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_1: **IF**(N_{k-4} is Low) **AND** ($P_2^{offset}{}_{k-5}$ is Low) **AND** (Te_{k-5} is Very Low) **AND** (Nd_{k-6} is ...) **AND** (N_{k-6} is Medium)

THEN $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_1: **IF**(N_{k-4} is Medium) **AND** ($P_2^{offset}{}_{k-5}$ is Medium) **AND** (Te_{k-5} is High) **AND** (Nd_{k-6} is ...) **AND** (N_{k-6} is Low)

THEN $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_1: **IF**(N_{k-4} is Low) **AND** ($P_2^{offset}{}_{k-5}$ is Very Low) **AND** (Te_{k-5} is Medium) **AND** (Nd_{k-6} is ...) **AND** (N_{k-6} is Medium)

THEN $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_1: **IF**(N_{k-4} is Very High) **AND** ($P_2^{offset}{}_{k-5}$ is Very Low) **AND** (Te_{k-5} is Low) **AND** (Nd_{k-6} is ...) **AND** (N_{k-6} is Low)

THEN $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_1: **IF**(N_{k-4} is Low) **AND** ($P_2^{offset}{}_{k-5}$ is High) **AND** (Te_{k-5} is Low) **AND** (Nd_{k-6} is ...) **AND** (N_{k-6} is Very High)

THEN $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

R_1: **IF**(N_{k-4} is Low) **AND** ($P_2^{offset}{}_{k-5}$ is Very High) **AND** (Te_{k-5} is Very High) **AND** (Nd_{k-6} is ...) **AND** (N_{k-6} is Very High

THEN $NOx_k = a_0^1 + a_1^1 N_{k-4} + a_2^1 P_2^{offset}{}_{k-5} + a_3^1 Te_{k-5} + a_4^1 Nd_{k-6} + a_5^1 N_{k-6}$

Local model parameters

$R\backslash par$	a_0	a_1	a_2	a_3	a_4	a_5
R_1	0.62122	0.57316	-0.79908	-0.10607	-3.8705	3.6797
R_2	0.25019	0.24239	-0.29438	0.46455	-1.533	1.4712
R_3	0.77561	0.12709	0.37604	0.078215	0.45054	-0.2973
R_4	0.29341	0.45984	0.38647	0.27322	-0.55073	-0.30107
R_5	-0.27591	0.033233	-0.22574	0.44712	0.45803	0.46113
R_6	0.42849	0.47549	-1.9812	-1.1318	-2.7612	-0.22697
R_7	-0.43818	0.6626	1.0395	2.8411	-0.0079526	-0.071655

Rule Base Evolution

Number of ADDED rules : 7

Number of MODIFIED rules : 6

Samples that originate new rules :

1 3 5 8 12 21 588

Final position of the focal points :

1 4 5 311 12 21 588

MODIFIED rules formed around samples :

2 4 41 53 57 311

Time of calculations (CPU): **1.0469** S

Variance Accounted For (VAF): 85.034

PERFORMANCE MEASURES

Correlation	**0.92213**
MSE	0.0037546
RMSE	0.061275
NDEI	0.38991

Quality of crude oil

- CESPA oil refinery, Tenerfie, Spain
- 80000 bb/d
- Products: heavy naphta, kerosene, GOL
- Parameters: $T_{95\%}$; Pensky-Martens (inflammability analysis)
- Off-line, once a day lab test

Quality of crude oil

The aim is to predict daily:

- Temperature of the heavy Naphtha when it evaporates 95% liquid volume, ASTM D96
- Temperature of the kerosene when it evaporates 95% liquid, ASTM D96
- Pensky Martens inflammability analysis of the Kerosene
- Temperature of the GOL when it evaporates 85% liquid, ASTM D96

T of heavy naphtha

Temperature of the heavy Naphtha when it evaporates 95% liquid volume In a distillation tower, mainly depends on

- The pressure of the tower
- Amount of product taking off
- Density of the crude
- Temperature of the column overhead
- Temperature of the Naphtha Extraction

T of kerosene

Temperature of kerosene when it evaporates 95% liq. vol. depends on:

- The pressure of the tower
- Amount of product taking off, Naphtha and KNO
- Density of the crude
- Temperature of the column overhead
- Steam introduced in GOL stripper ratio to KNO
- Temperature of the Kerosene Extraction
- Temperature of the Naphtha Extraction

Pensky-Martens

Inflammability analysis of the Kerosene concern the light part of the kerosene, and therefore it depends mainly in the Naphtha above and the steam injected in the kerosene stripper.

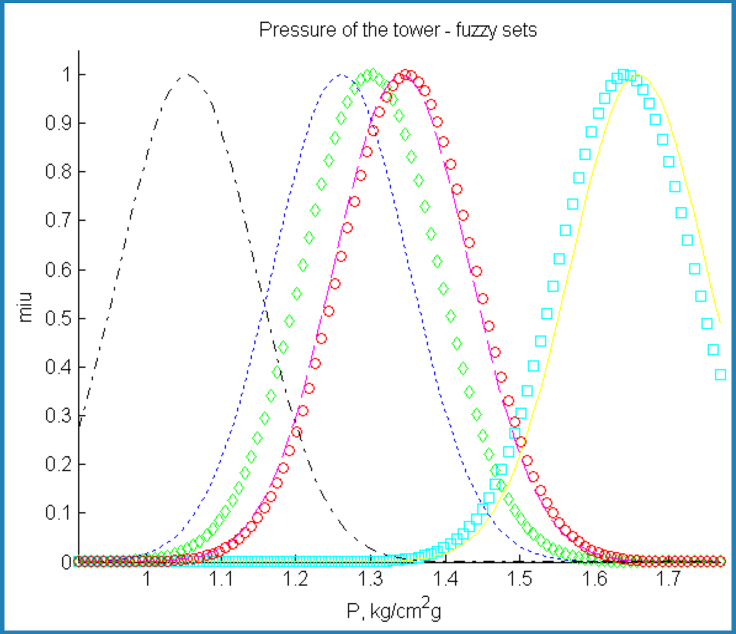
- The pressure of the tower
- Amount of product taking off
- Density of the crude
- Temperature of the column overhead
- Temperature of the Naphtha Extraction
- Steam introduced in Kerosene stripper, ratio to KNO

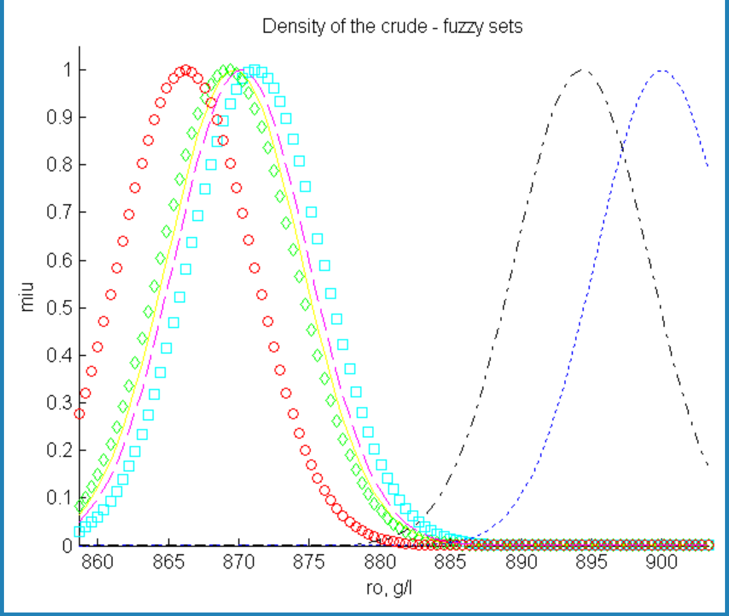
T of gas oil

Temperature of GOL when it evaporates 95% depends on:

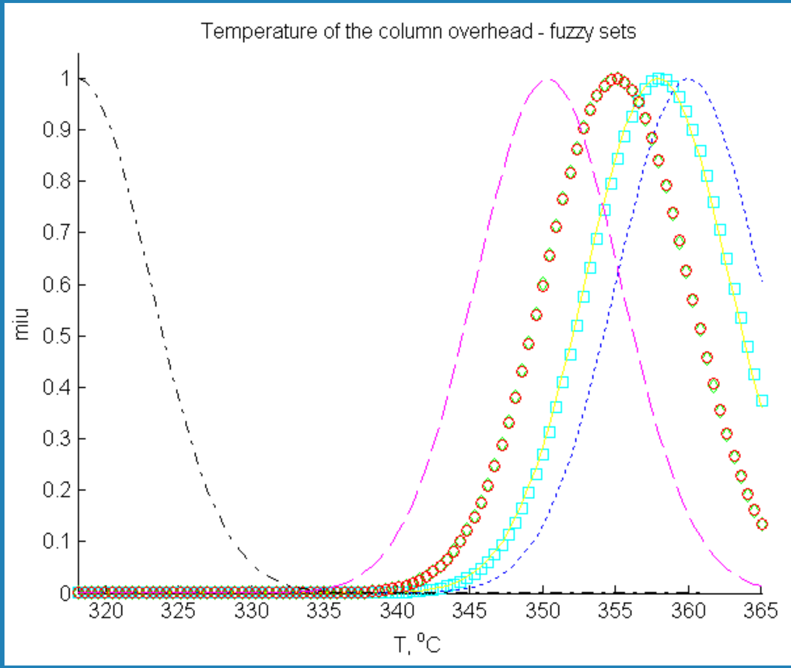
- The pressure of the tower
- Amount of product taking off, Naphtha and KNO
- Density of the crude
- Temperature of the column overhead
- Steam introduced in GOM stripper, ratio to GOM
- Temperature of the GOL Extraction
- Temperature of the Kerosene Extraction
- Temperature of the Naphtha Extraction

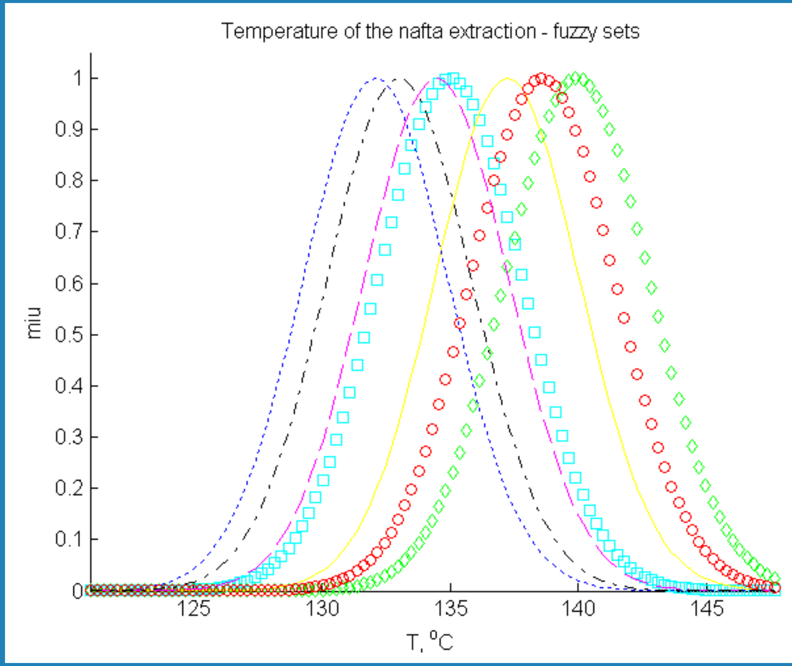
Fuzzy sets, Heavy naphtha



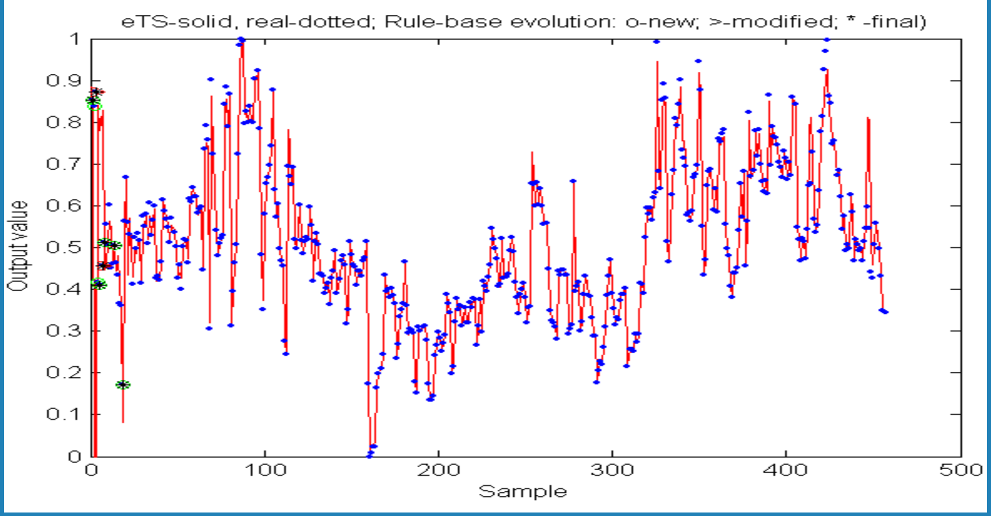


Fuzzy sets, Heavy naphtha



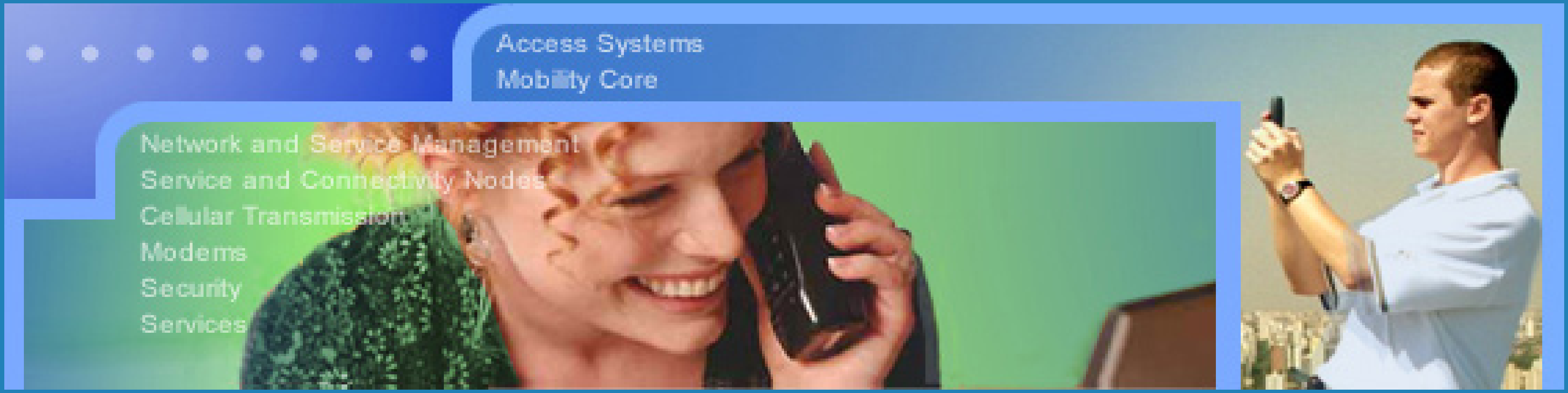


Prediction of Thn



- Error in the order of 2-3^o*C* (the T^{hn} is in the range of 100-160^o*C*.
- The precision is comparable with the precision of the laboratory **off-line** test and can be done in real-time.

QoS improvement in VoIP



- Voice communication using Packet based communication networks is becoming common place (e.g Internet)
  - ⇒ The future is Mobile IP based Comms
- GSM Air interface is hostile environment: due to urban, multi-path interference error rates/packet losses can be high

QoS improvement in VoIP

- QoS requires error concealment
- Approach – real-time novel error concealment approach using eTS models suitable for VoIP over GSM with “severe” packet losses (100 to 200 mS of Speech, getting towards word loss level) that
  - minimise transmission bandwidth
  - minimise system delays
  - maximise speech quality (QoS)

Speech processing coders

- Modern Speech commissions digitise analog speech signal & use speech coding techniques to reduce the transmission bandwidths (reduced operator fixed costs)
- speech Coding applies DSP techniques to speech segments (~20 mS) in order to
  - identify and isolate "perceptually" important characteristics of the voice signal
  - then digitize and quantize them
  - Transmit the quantized/digital values.

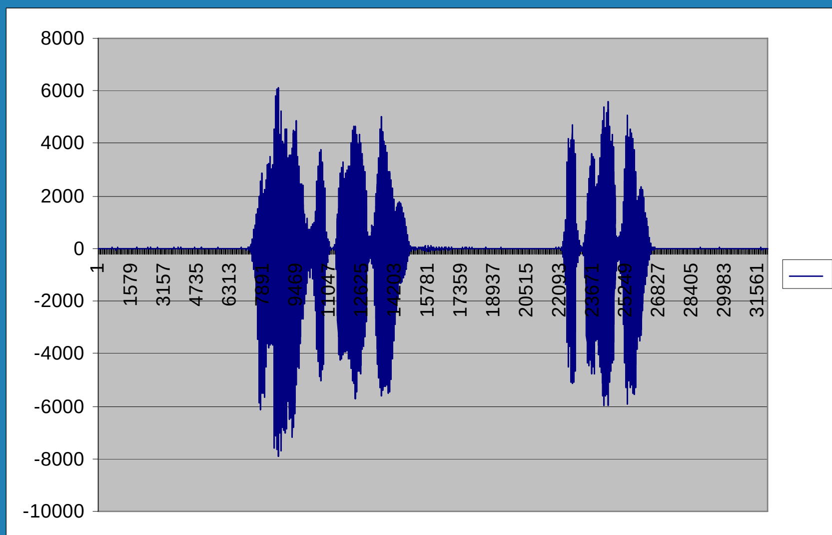
LPC parameters

- Vocoder are a class of parametric coder that use a parametric model of the vocal tract
- Linear Prediction coding is a widely used method for representing the frequency shaping attributes of the vocal tract
  - The short term spectral envelop is modeled by all pole Linear Predicting Filter
  - This yields a small number of parameters (10)

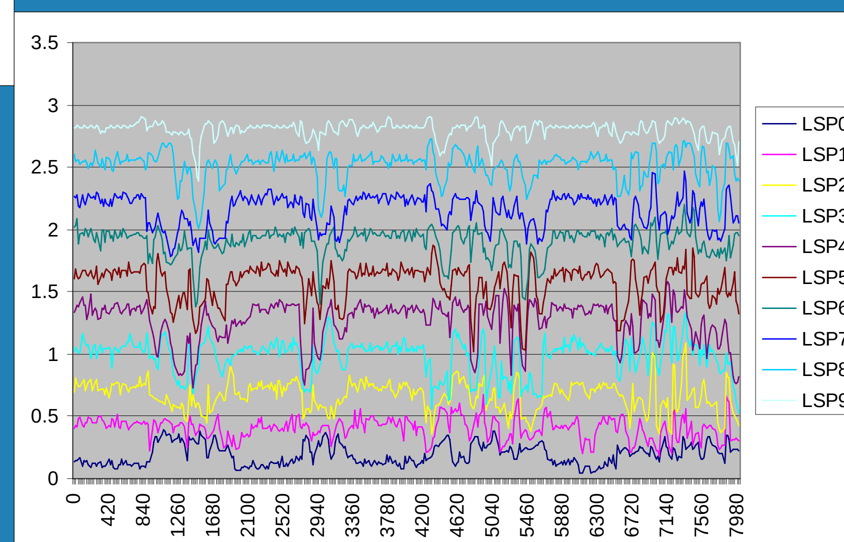
The approach

- LPC is typically 10th Order
- 10 LPC filter coefficients are generated per frame (20mS speech segments)
- LPC Filter Coefficients are usually transformed into the Line Spectral Frequencies (LSP) before quantization
- modelling different aspects of speech\ language but in the parameter domain (not necessarily the time domain)

Source speech → LSP

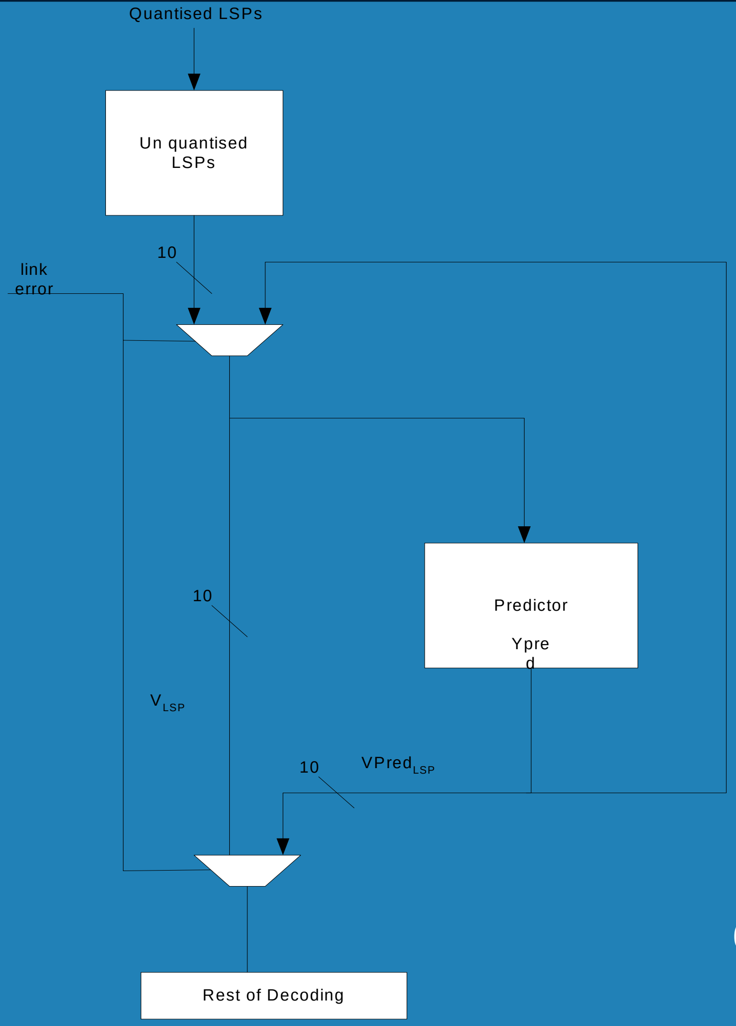


Source speech → LSP



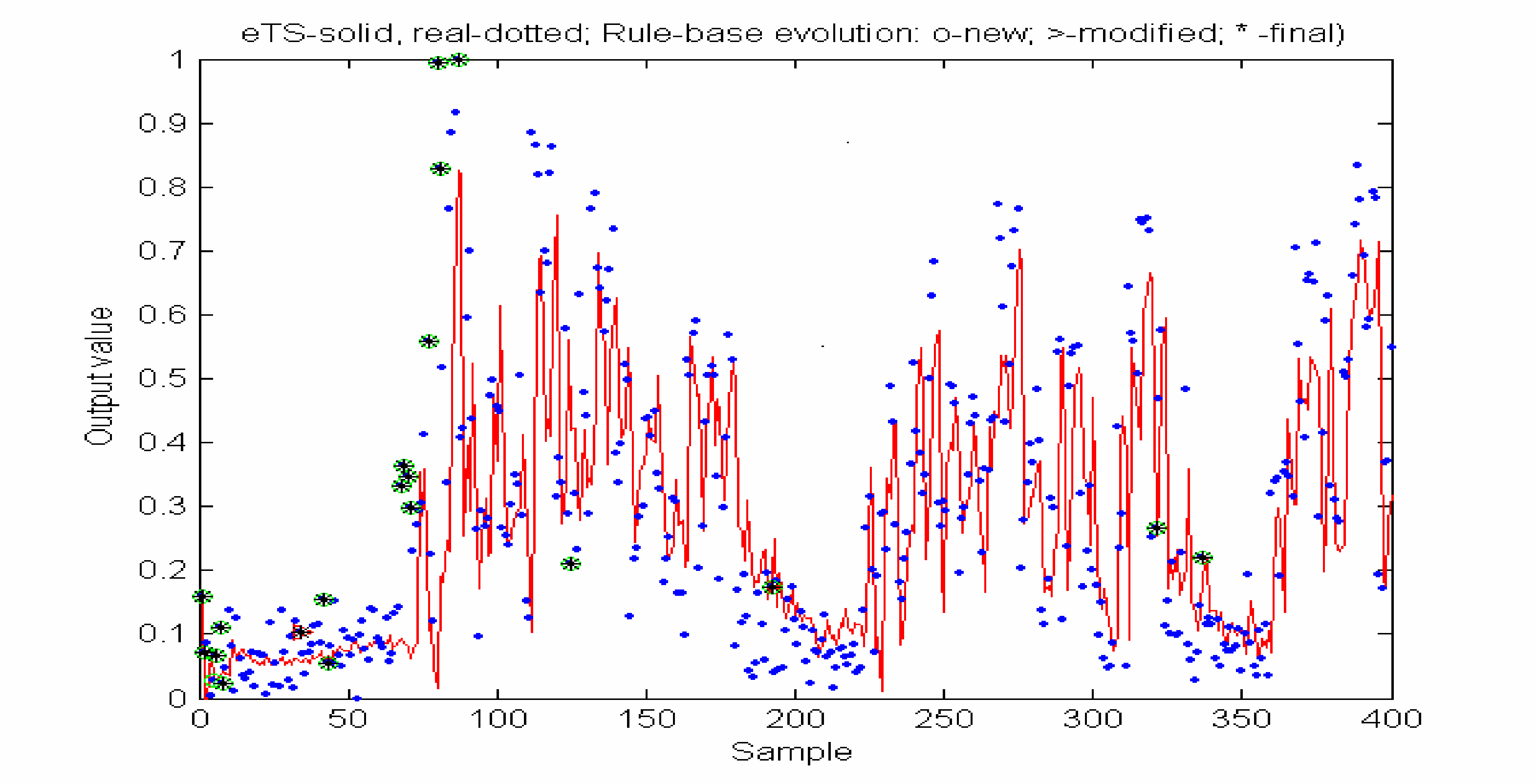
Preliminary results

- ✓ eTS is able replace around 100 mS of lost unquantized speech with small losses (state of art is currently around 40mS)
- ✓ eTS when continuously executed (always replace the LSP) can improve the overall speech quality of low bit rate codec



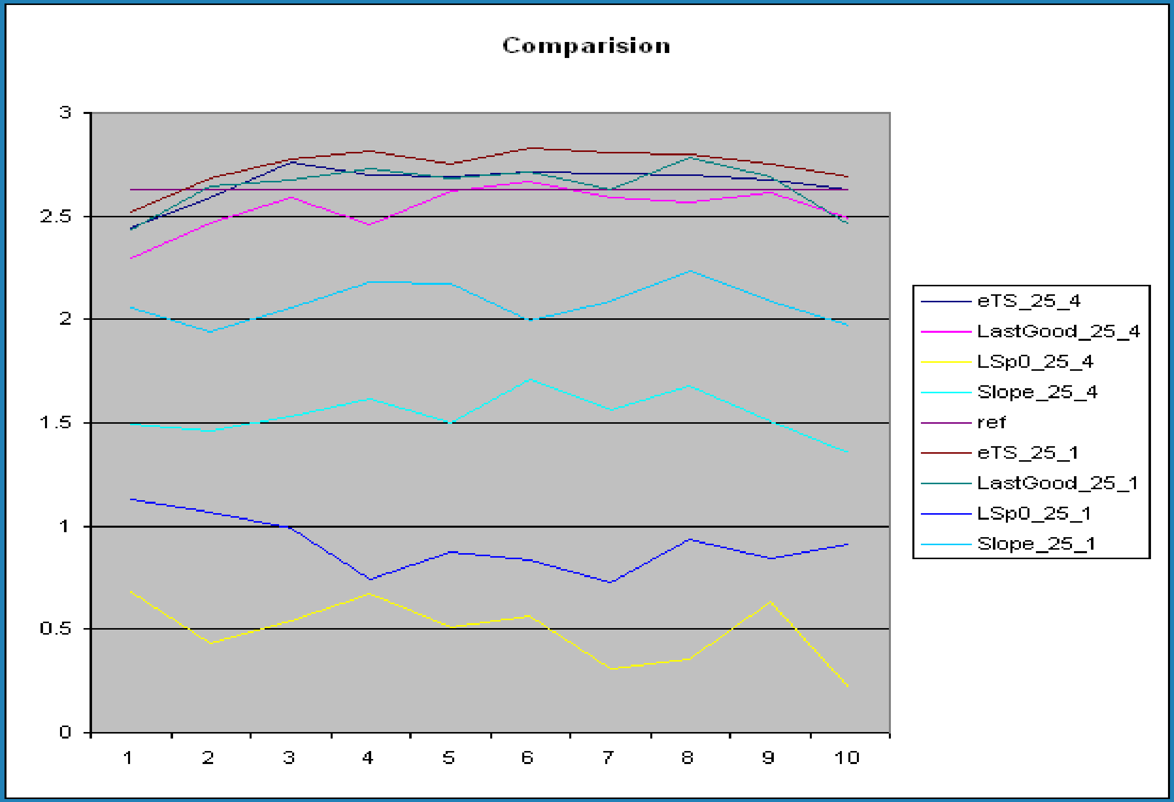
QoS for VoIP, NOKIA

- Real speech, female speaker
- MOS 2.6 insignificant drop

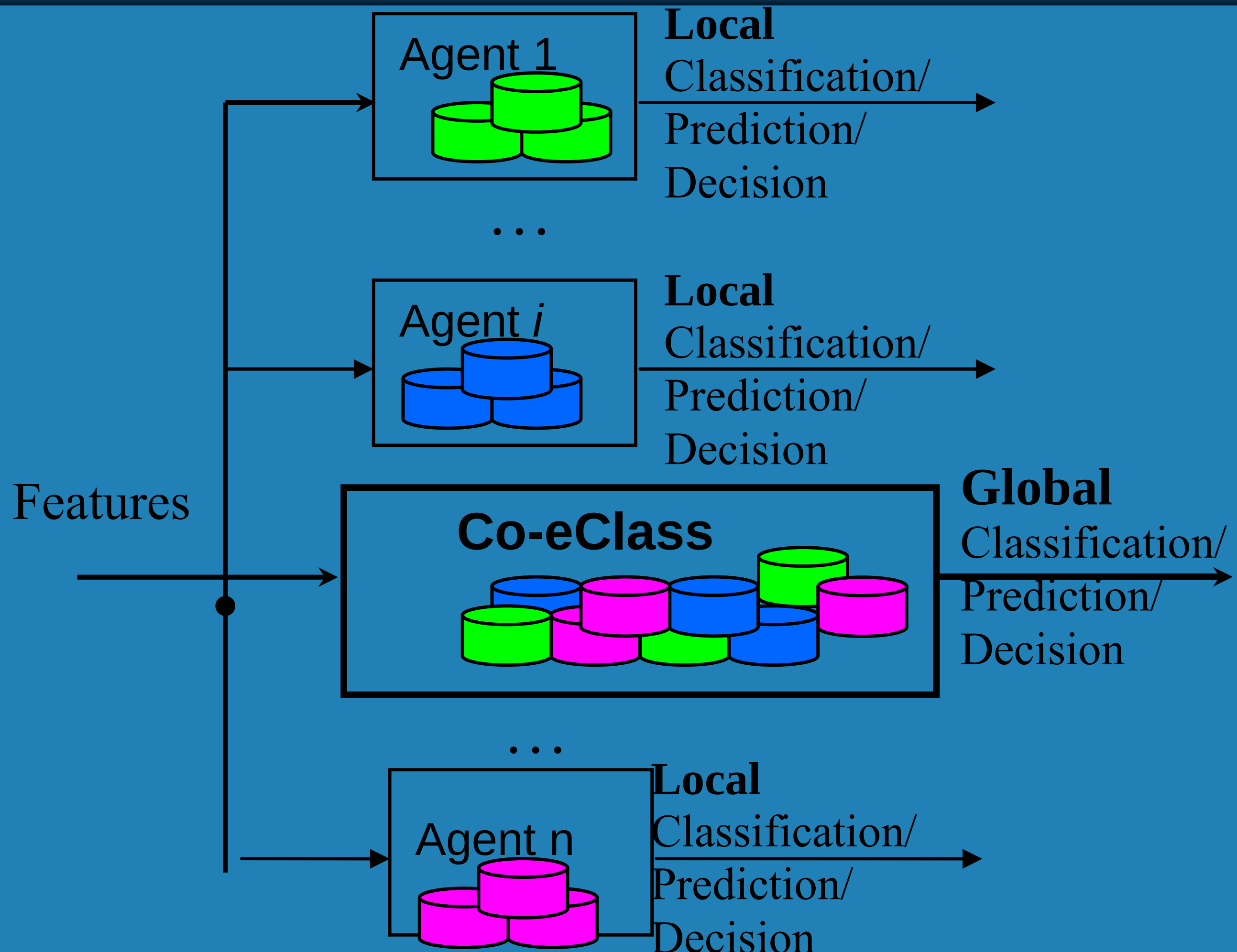


QoS for VoIP, NOKIA

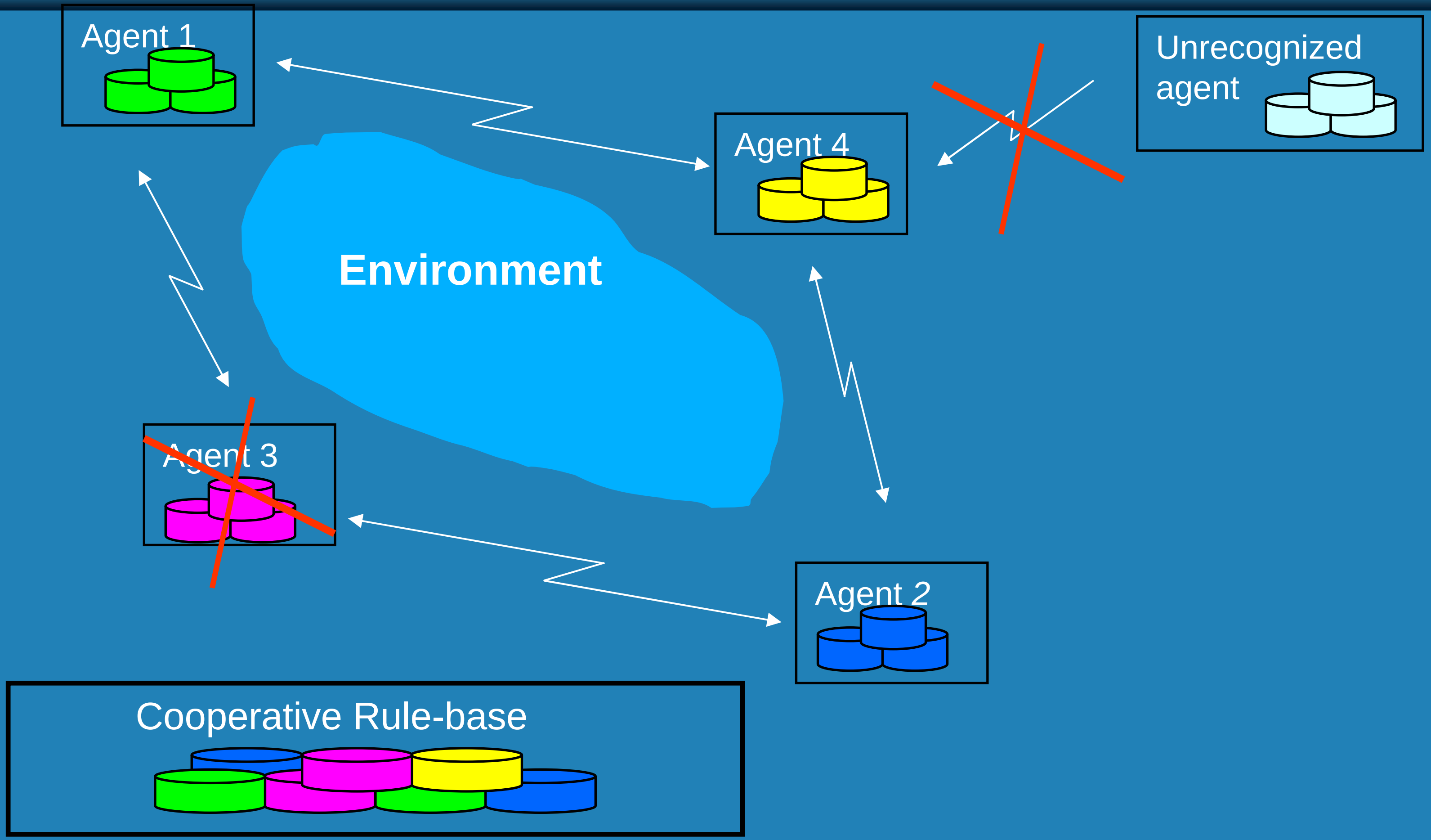
- Real speech, female speaker
- MOS 2.6 insignificant drop



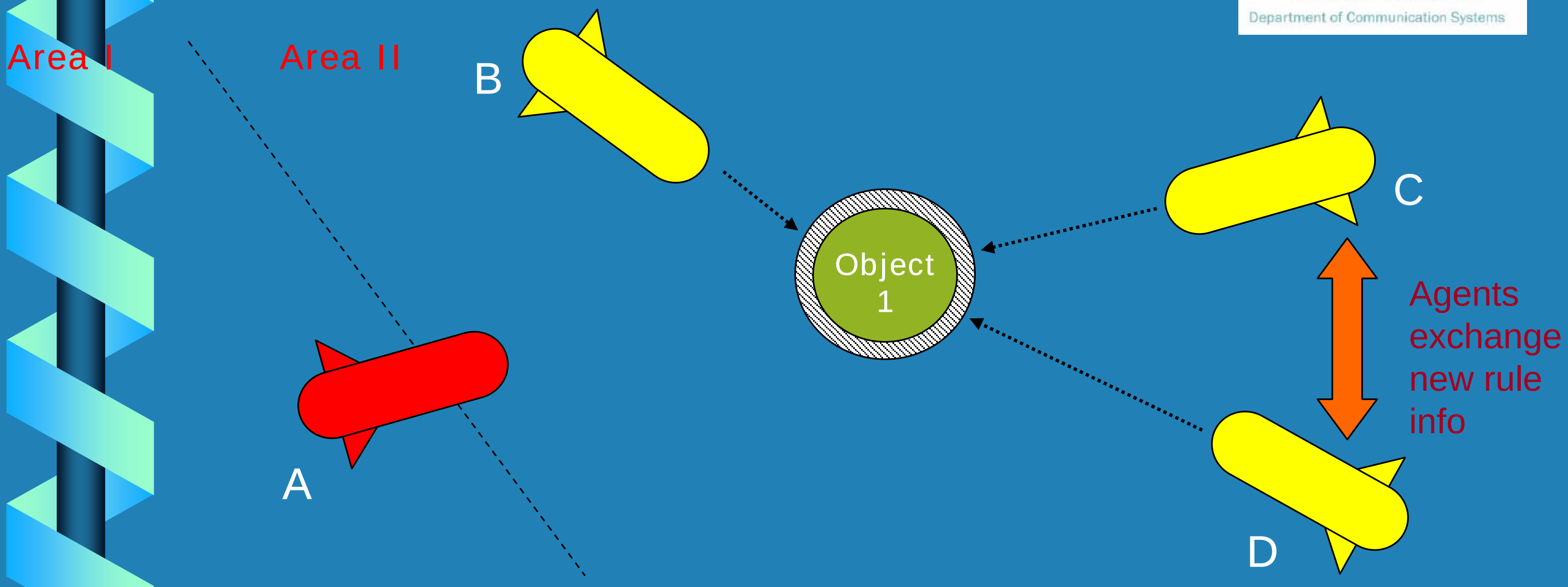
Distributed co-eClass



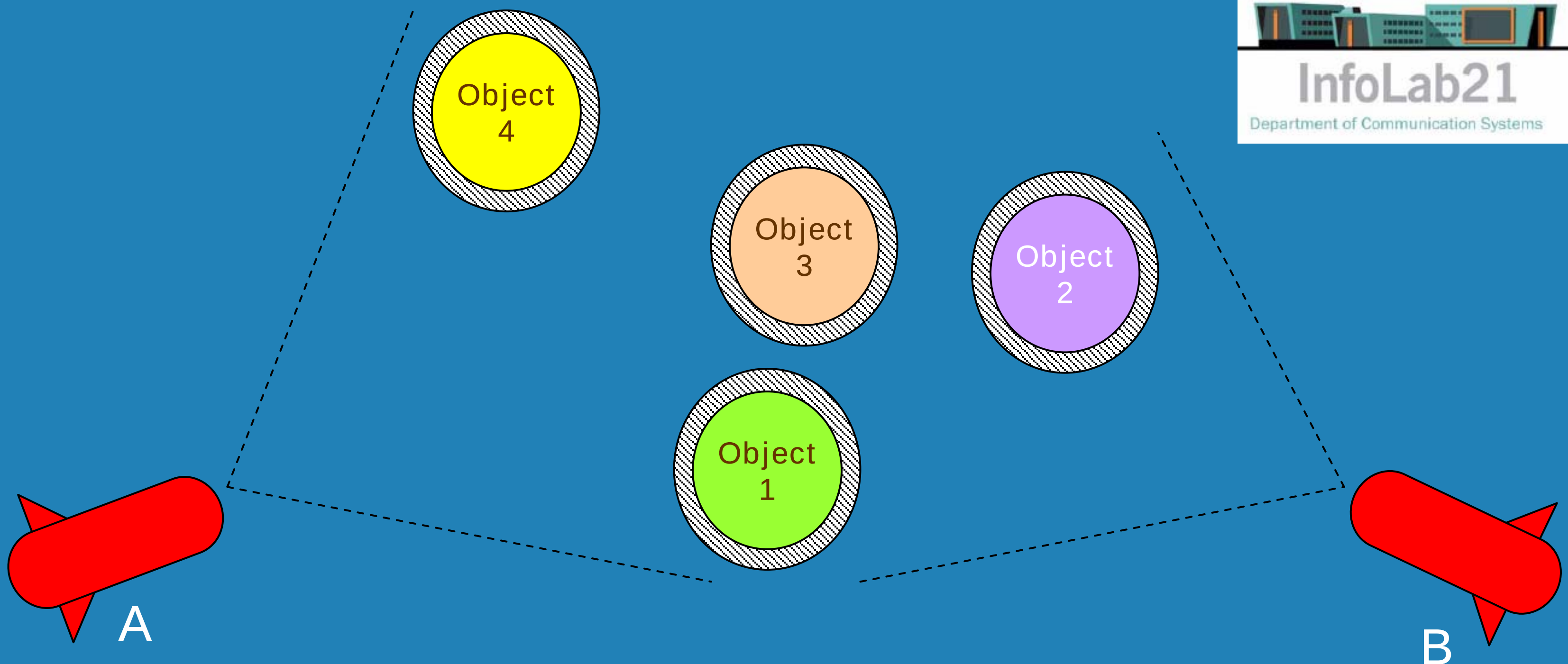
Co-operation in autonomous detection



Underwater target recognition, J&S Marine



Underwater target classification- scenario 2

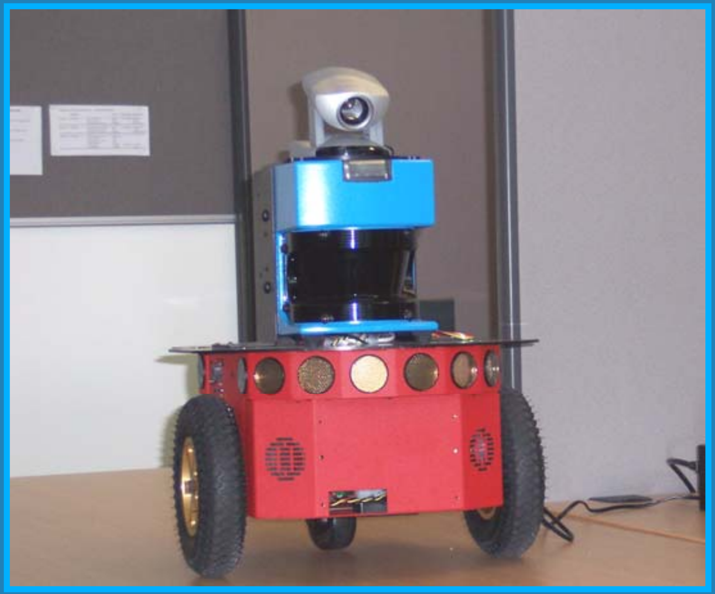


Our Pioneer3-DX Robots

- Onboard Computer (PIII CPU,256M RAM)
- Camera, Digital Compass, Sonar, bumpers, Laser
- Controller (embedded microprocessor ARCOS)
- A team of 5 robots with

  WIFI connection

  for collaborative tasks



Novelty detection and land mark recognition by eClustering

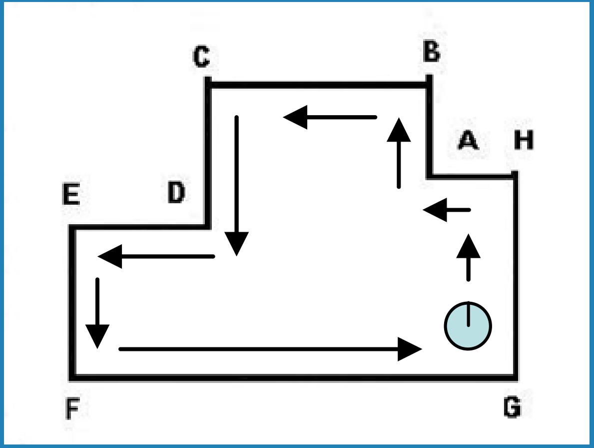
- Explore unknown environment
- No communication link (no GPS, maps...)
- Simple landmark (corner) recognition
- Fully unsupervised (no pre-training no model structure assumed)

Novelty detection and landmark recognition by eClustering

- Novelty Detection
  - The ability to differentiate between common sensory stimuli and perceptions never experienced before
- Landmark Recognition
  - With Novelty Detection, robot can select aspects of the environment that are unusual and therefore can be used as landmarks for self-localization

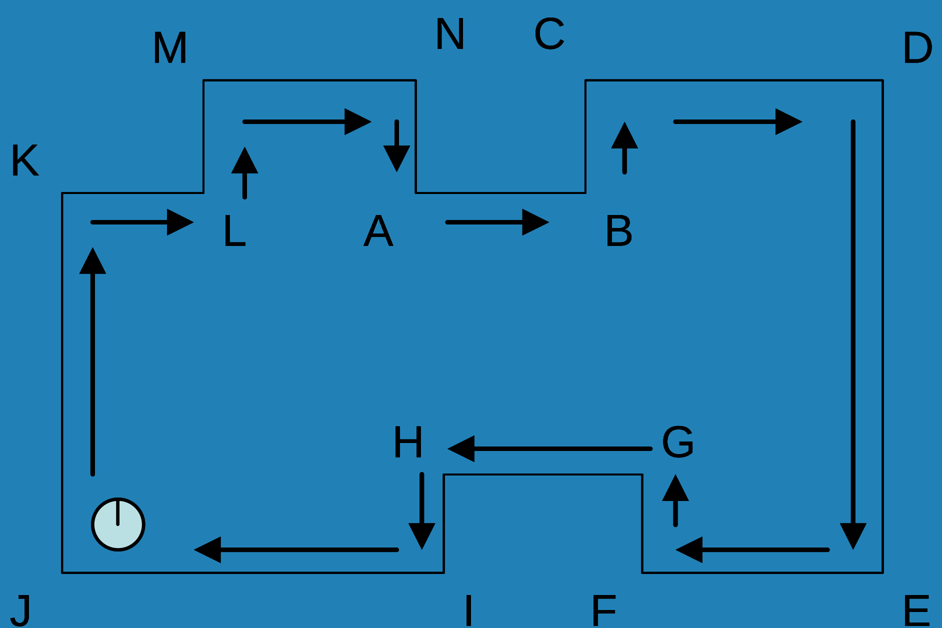
Novelty detection and landmark recognition by eClustering

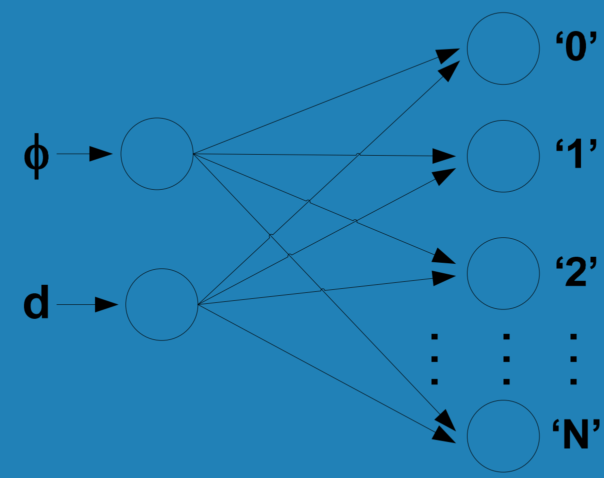
- Reference Nehmzow, 1991
- SOM (50 neurons, supervised, off-line pre-training, fixed structure)
- Off-line Pre-training
- Evolving SOM (Kasabov, 2000)– too relaxed – based on a threshold, not robust enough to avoid noise becoming cluster centres



Novelty detection and landmark recognition by eClass

InfoLab21
Department of Communication Systems





Our experiment (B-69, InfoLab21)

bb.mpg

Novelty detection and landmark recognition by eClass

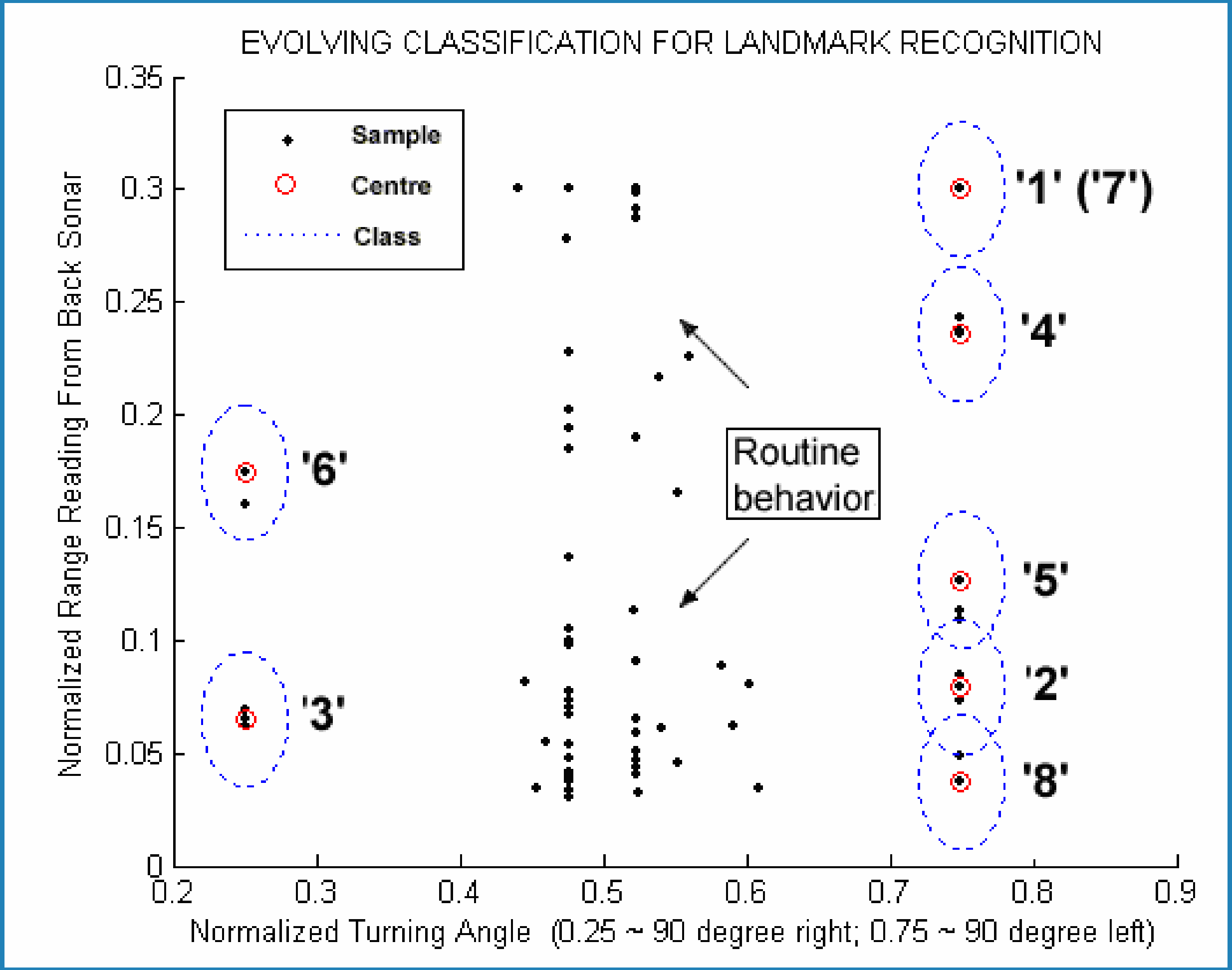
InfoLab21
Department of Communication Systems

demo corner detection

- ✓ **Recursive** calculation → less computing power required → **real-time**
- ✓ Extend to image-based (CASPIA)

R1:	**IF** (ϕ is close to ¾) **AND** (d is close to 0.3000) **THEN** (Corner is 1)
R2:	**IF** (ϕ is close to ¾) **AND** (d is close to 0.1268) **THEN** (Corner is 2)
R3:	**IF** (ϕ is close to ¼) **AND** (d is close to 0.0648) **THEN** (Corner is 3)
R4:	**IF** (ϕ is close to ¾) **AND** (d is close to 0.2357) **THEN** (Corner is 4)
R5:	**IF** (ϕ is close to ¾) **AND** (d is close to 0.0792) **THEN** (Corner is 5)
R6:	**IF** (ϕ is close to ¼) **AND** (d is close to 0.1744) **THEN** (Corner is 6)
R7:	**IF** (ϕ is close to ¾) **AND** (d is close to 0.0371) **THEN** (Corner is 8)

Results



EVOLVING CLASSIFICATION FOR LANDMARK RECOGNITION
Sample
Centre
Class
'1' ('7')
'4'
'6'
'5'
'2'
'3'
'8'
Routine behavior
Normalized Range Reading From Back Sonar
Normalized Turning Angle (0.25 ~ 90 degree right; 0.75 ~ 90 degree left)
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

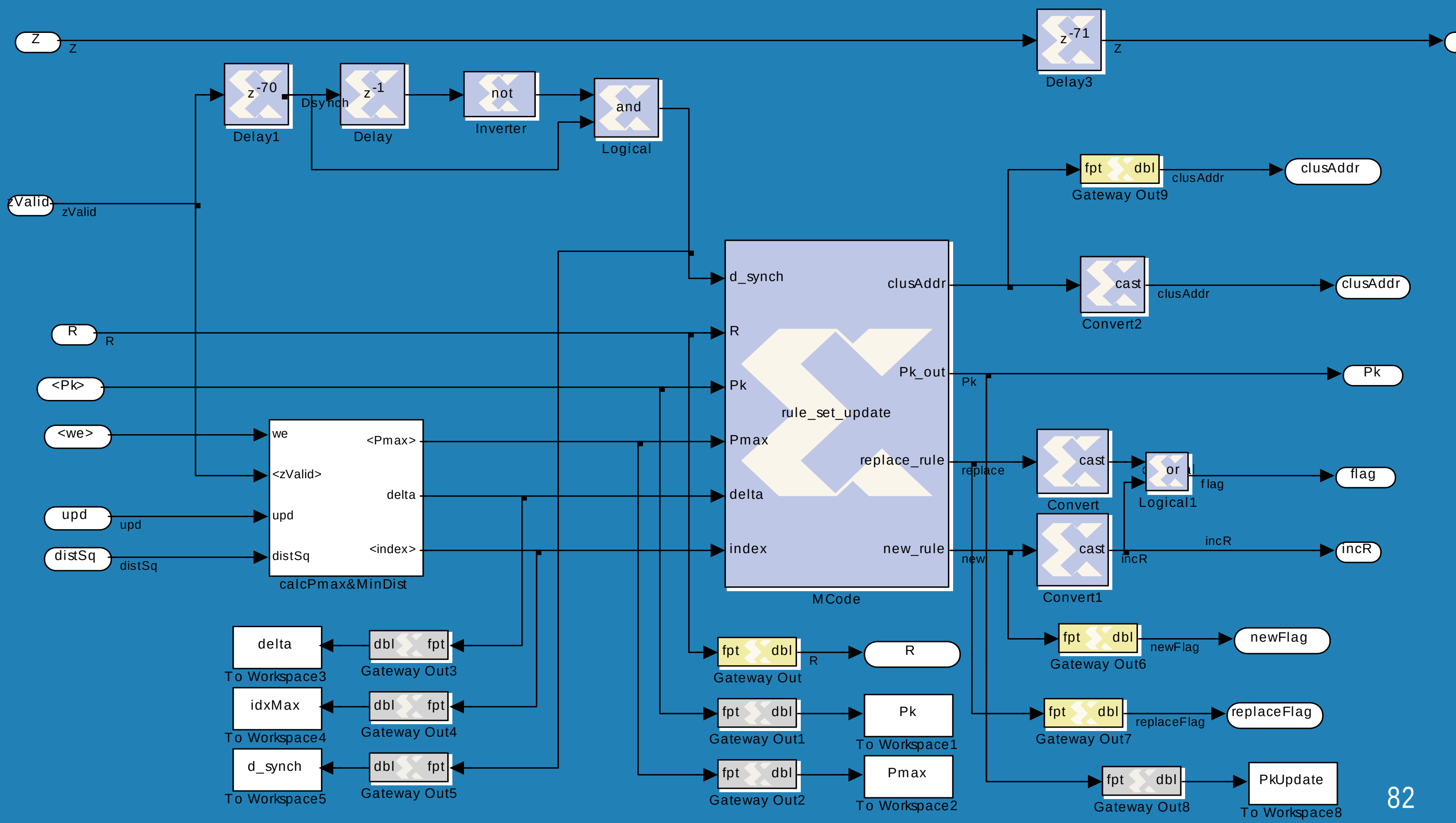
Results analysis

Vector Name	Inputs Vector	No. of Corners	**Correct**	Duplicated (Over Clustering)	Missed (lack of separation)	Uncertain (Miss or Dup)	Centre No	Description
[1]	**TDTDT**	**8**	**5**	**2**	**1**			**Same as in [1], 2 step back**
V1	**TDTDT**	**8**	**7**	**0**	**1**	**0**	**7**	**2 steps back similar to ref**
V2	TDTD	8	**6**	1	1	0	7	1 step back
V3	TD	8	**4**	1	3	0	6	no step back
V4	**TDTDTD**	**16**	**10**	**1**	**1**	**2**	**15**	**Lab Environment, 16 corners**

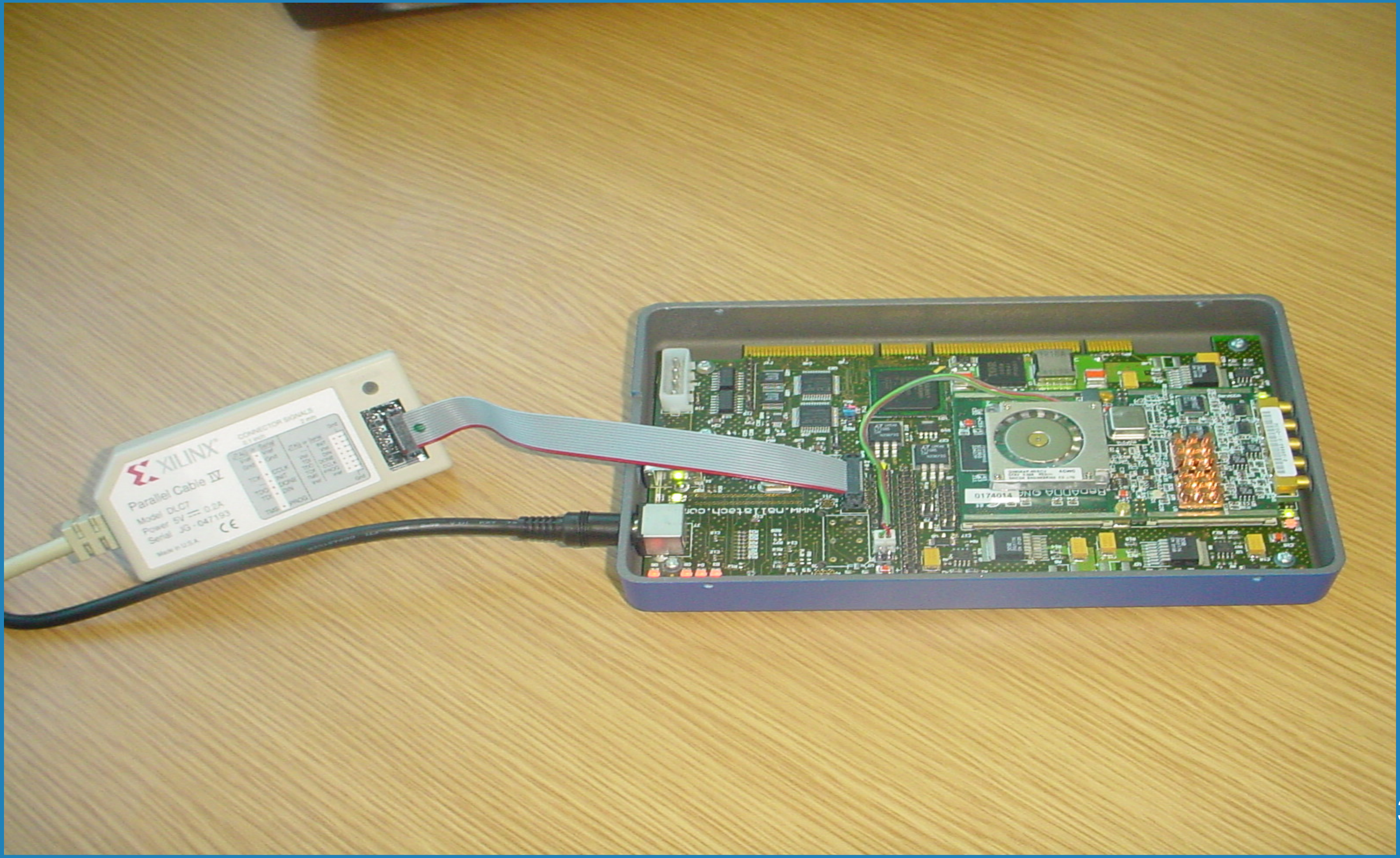
Table 1 Result Comparison (T: turning, D: distance)

- ✓ Unsupervised → fully automatic
- ✓ Learning from scratch → no pre-training
- ✓ Cluster (fuzzy rules/neurons) number is not predetermined, defined by data only → structure is flexible and evolving
- ✓ Recursive calculation → less computing power required for real-time execution

eTS on FPGA - XtremeDSP



eTS on FPGA - XtremeDSP

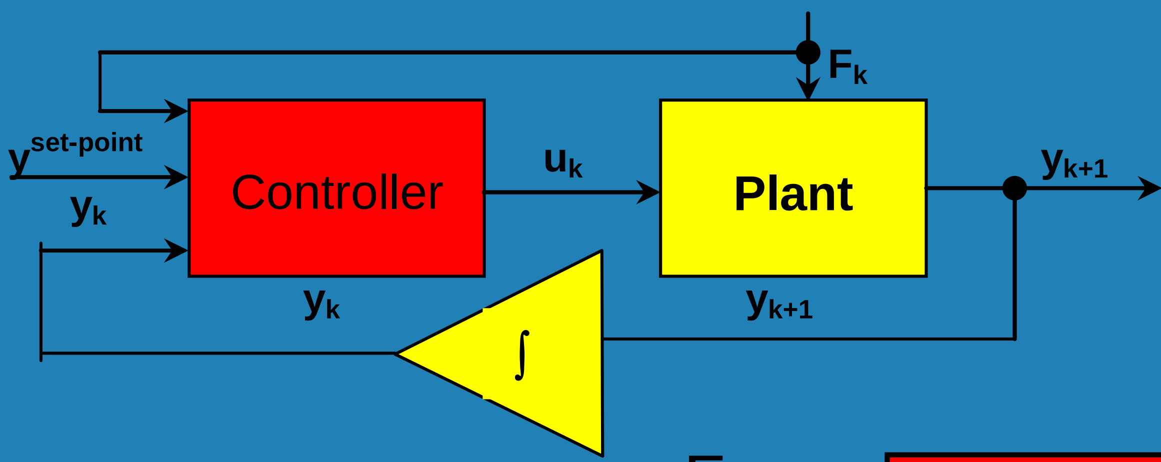


Applications: Classification and Control

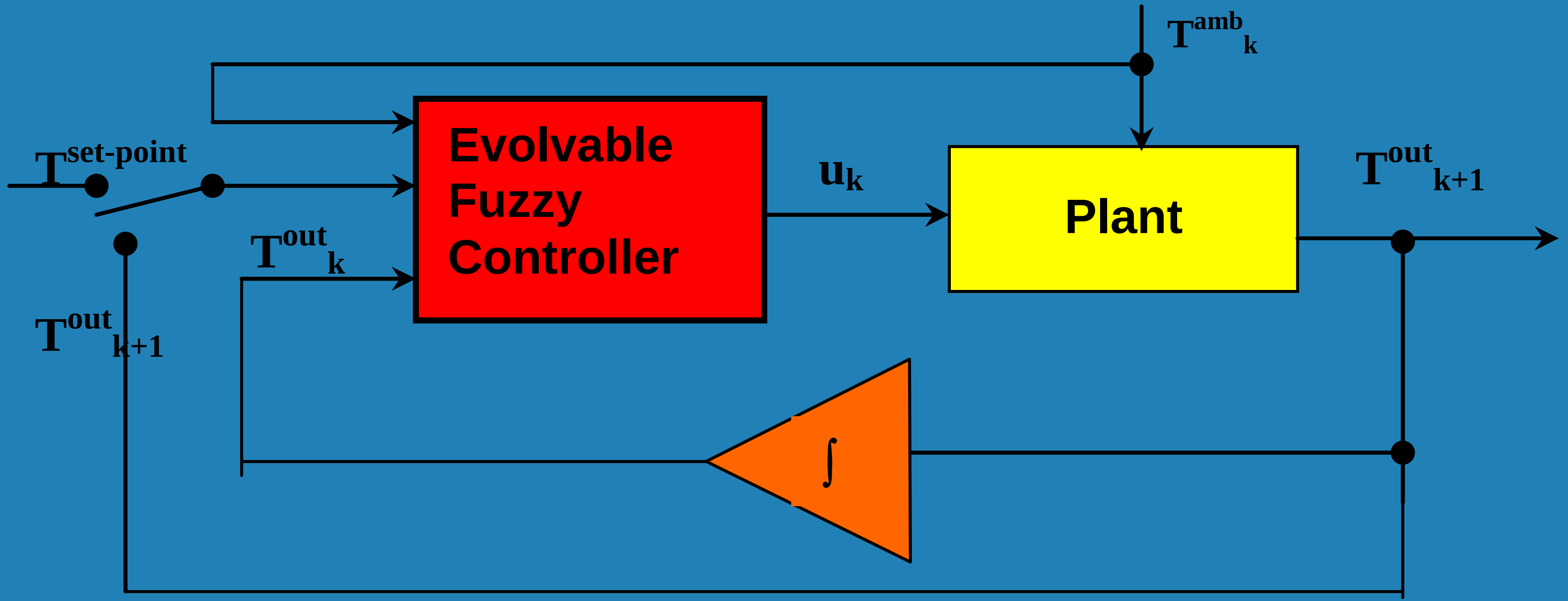
- Controllers with evolvable structure
- Application to EEG signals classification
- Classification of Carcinoma Kidney Tissue Status based on Protein Expression Data
- Biotech process applications

Evolving Controllers

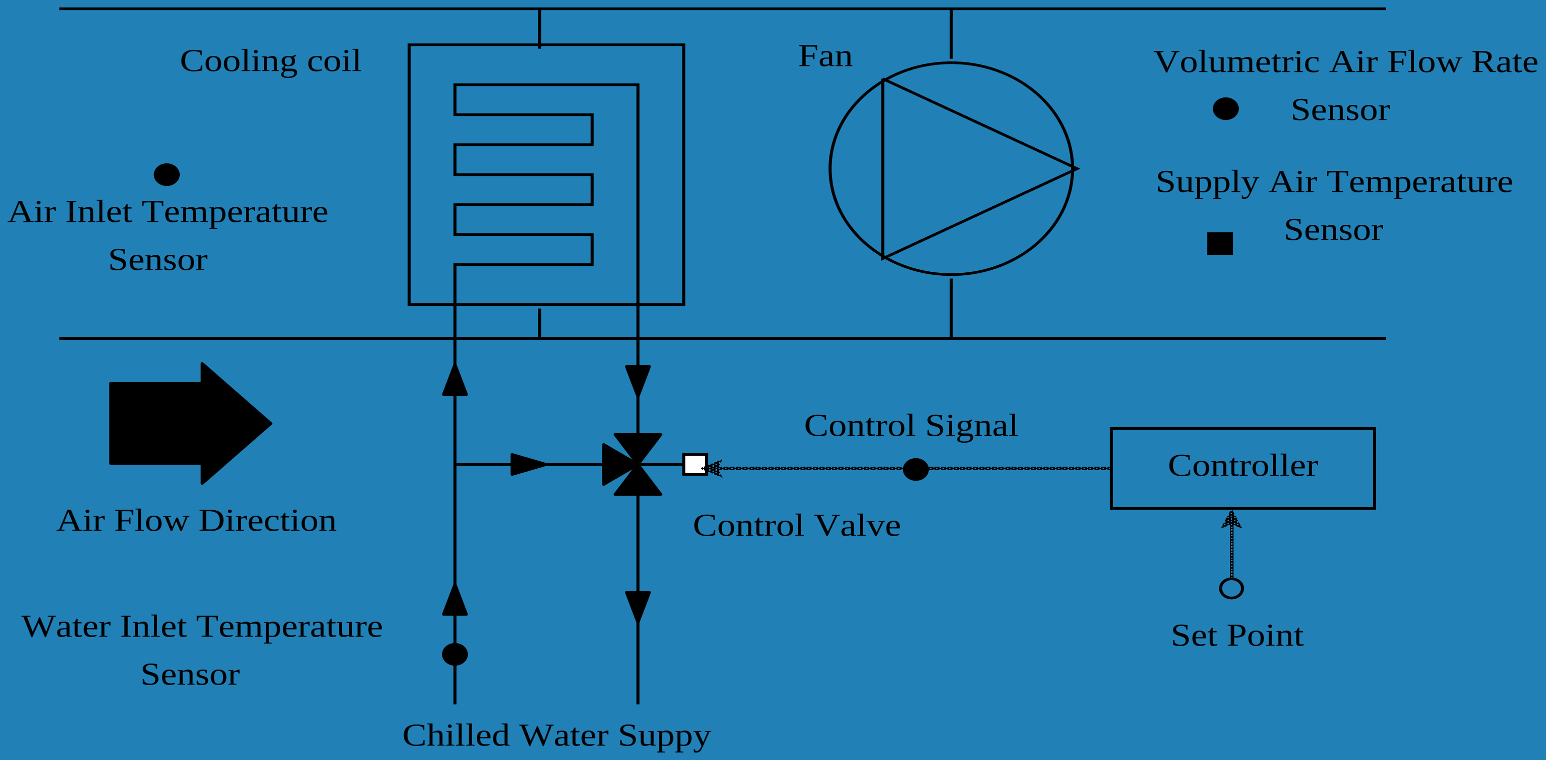
- Indirect learning (Psaltis et.al, 1988)
- Adaptive control (incl. controller structure)



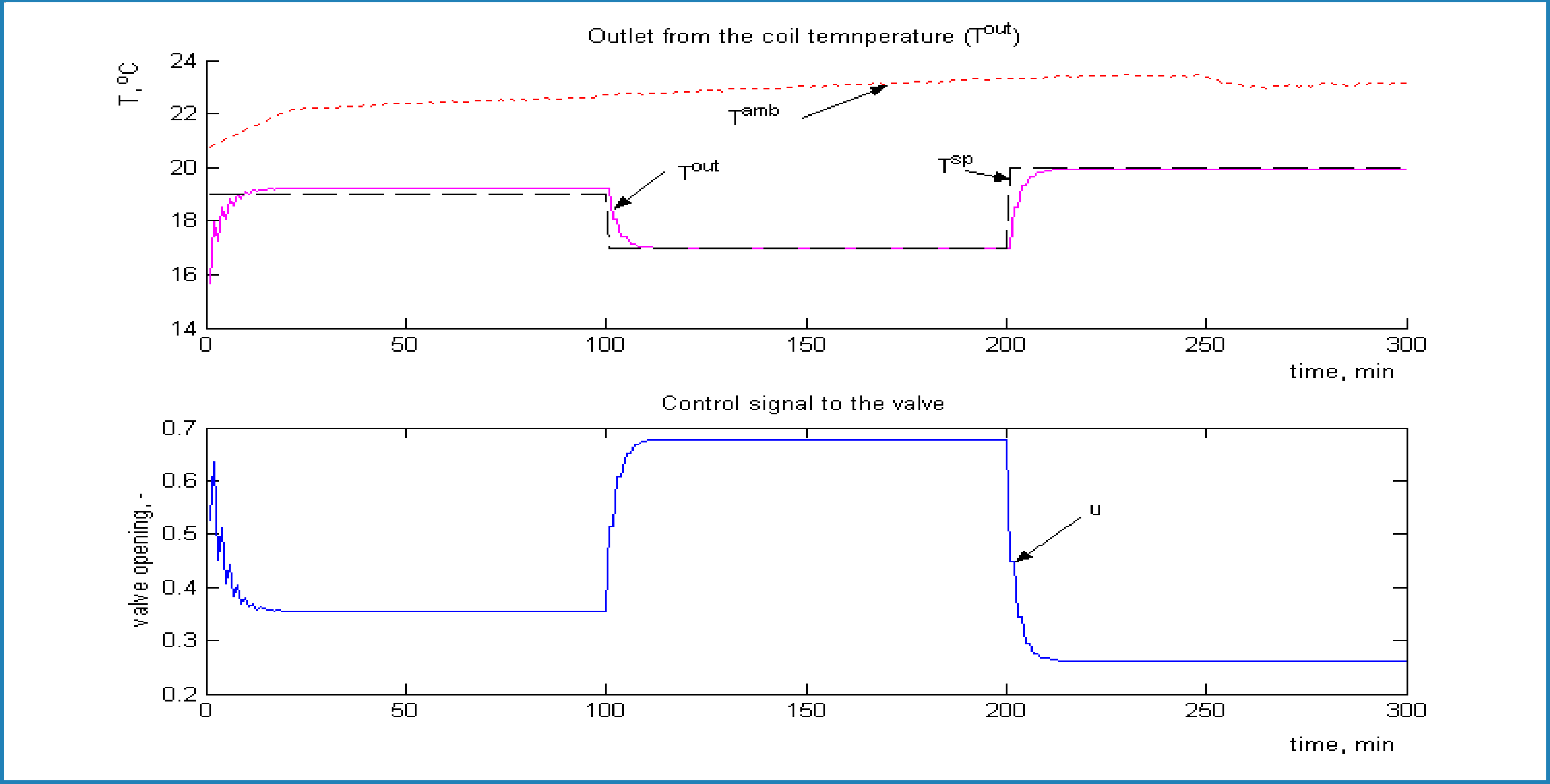
eController - example



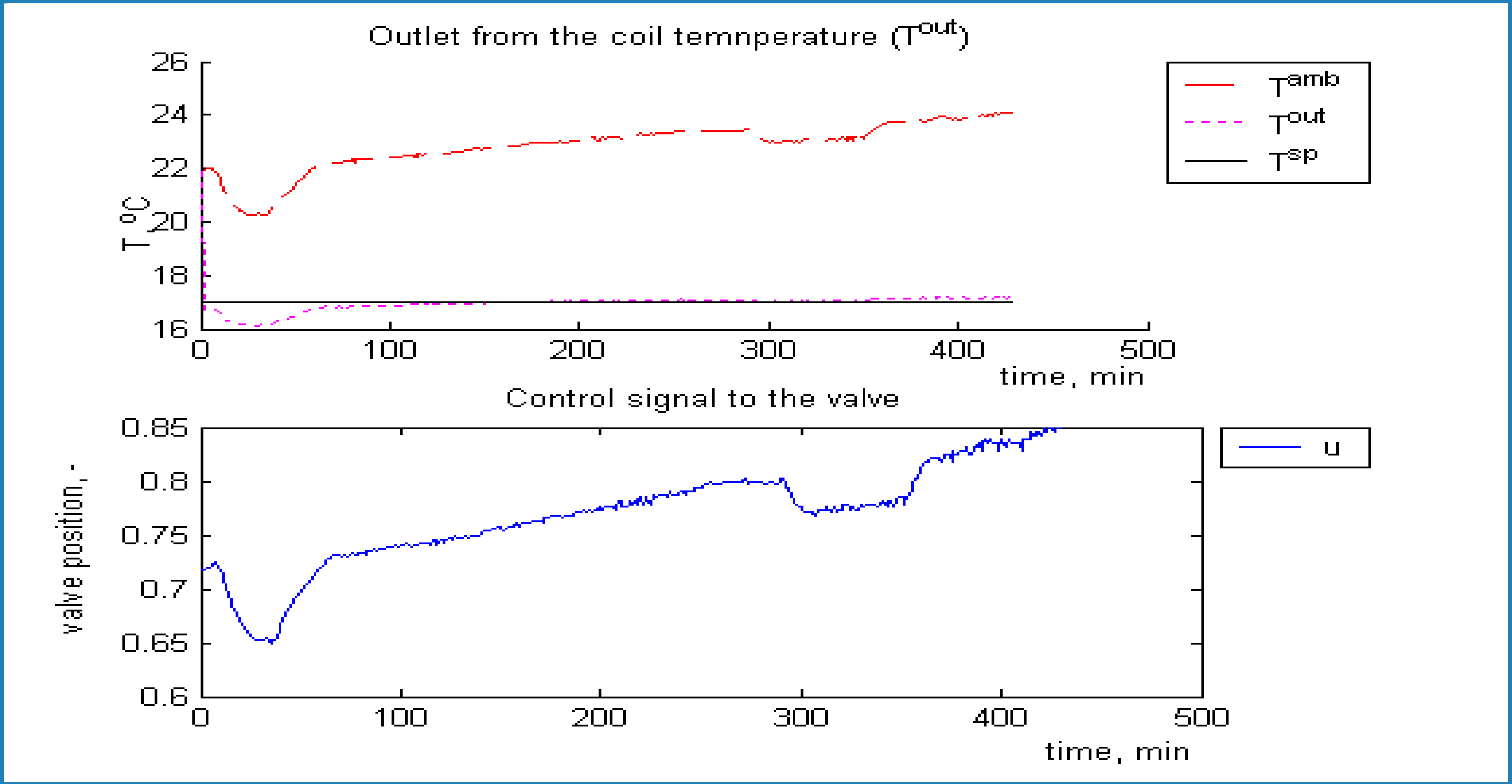
eController - example



eController - results



eController - results



eController - results

- Controller structure

R_1: **IF** ($T_k^{aml.}$ is *Low*) **AND** ($T_{k+1}^{out.}$ is *High*) **AND** ($T_k^{out.}$ is *High*) **THEN** (u_k is *Low*)

R_2: **IF** ($T_k^{aml.}$ is *High*) **AND** ($T_{k+1}^{out.}$ is *Medium*) **AND** ($T_k^{out.}$ is *Medium*) **THEN** (u_k is *High*)

R_3 **IF** ($T_k^{aml.}$ is *Very Low*) **AND** ($T_{k+1}^{out.}$ is *High*) **AND** ($T_k^{out.}$ is *High*) **THEN** (u_k is *Very Low*)

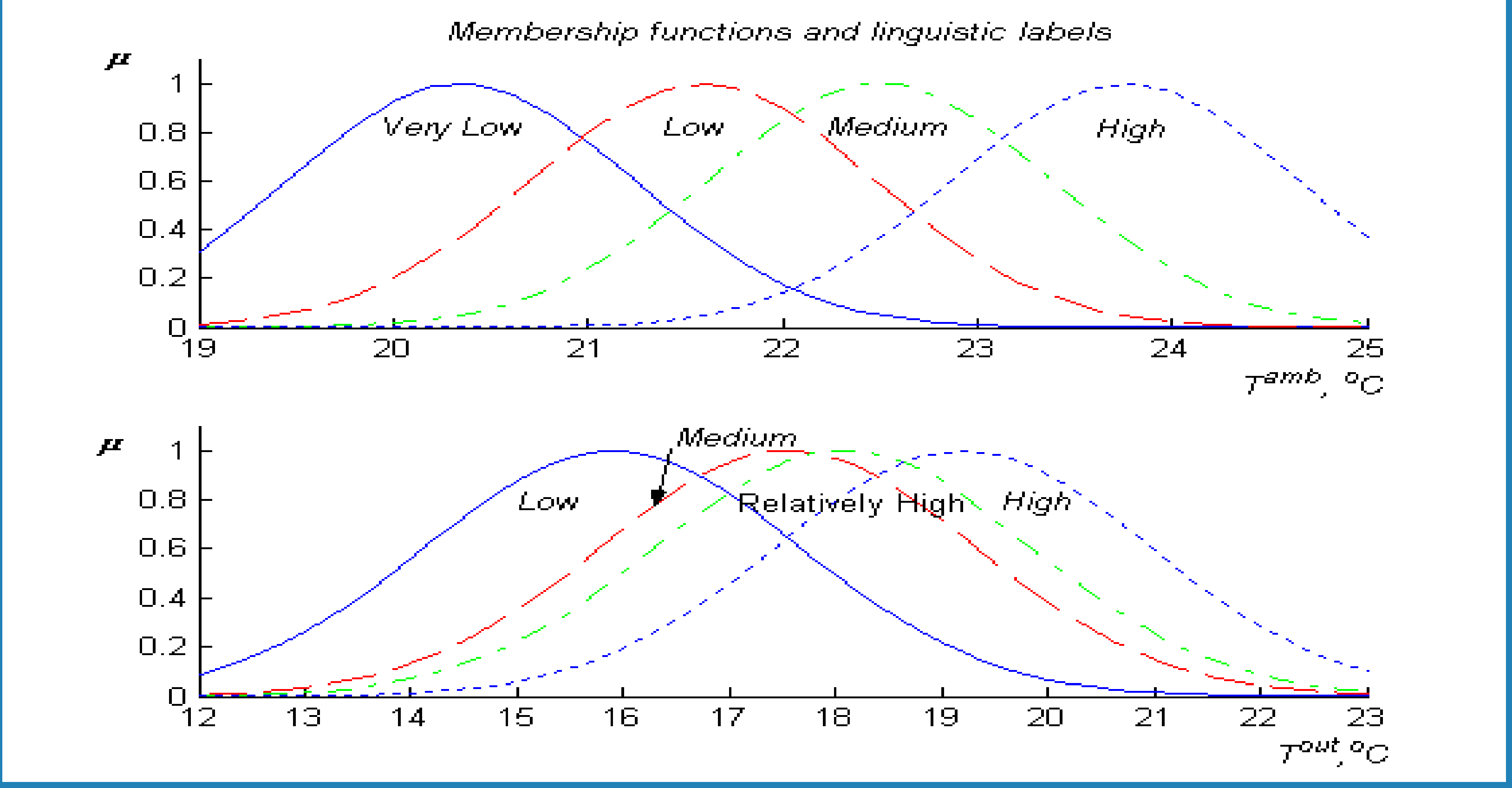
- new rule:

R_4: **IF** (T_k^{amb} is *Medium*) **AND** (T_{k+1}^{out} is *Low*) **AND** (T_k^{out} is *Low*) **THEN** (u_k is *Medium*)

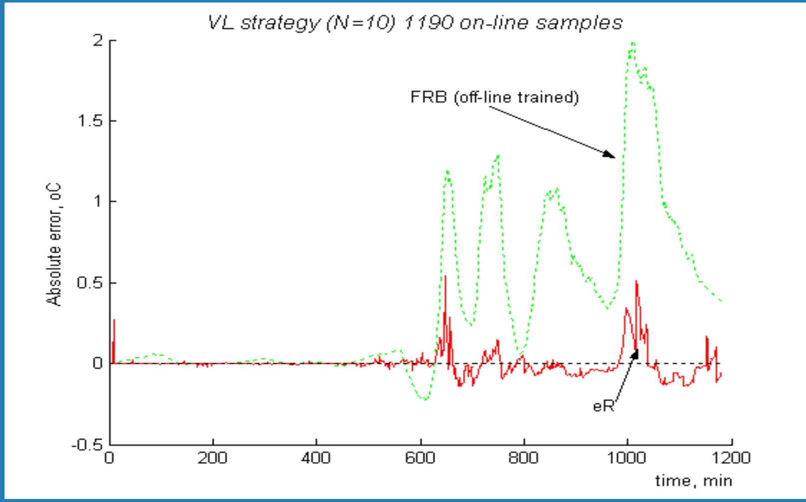
- modified **rule**:

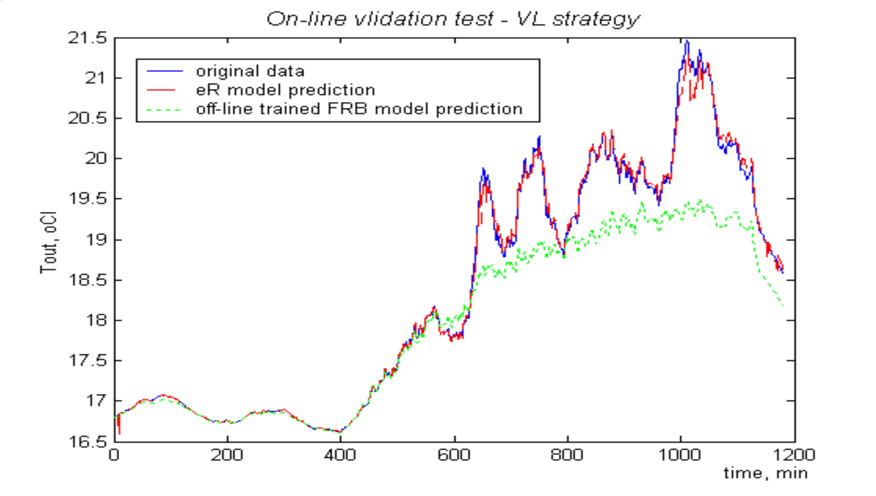
R_5: **IF** ($T_k^{aml.}$ is *Medium*) **AND** ($T_{k+1}^{out.}$ is *Relatively High*) **AND** ($T_k^{out.}$ is *Medium*) **THEN** (u_k is *Medium*)

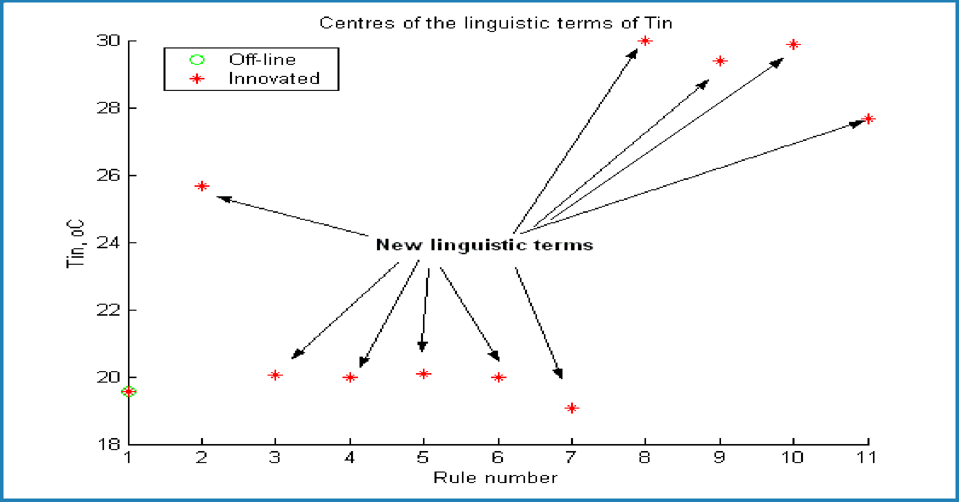
eController - results



Off-line vs on-line: real data (air-conditioning system)







InfoLab21
Department of Communication Systems

EEG signal classification procedure eClass

InfoLab21
Department of Communication Systems

1) Establish the first EEG signal as the first prototype. Its S=0
2) Starting from the next the Scatter of each new EEG signal is calculated *recursively*
3) The Scatter of the existing prototypes are *recursively* updated
4) Scatter of *new* EEG signal is compared to updated scatter of the existing prototypes

eClass: Procedure

5) If S*new* < S* then the *new* EEG signal is added as a **new prototype** and a *new rule* is formed (*R := R+1*)

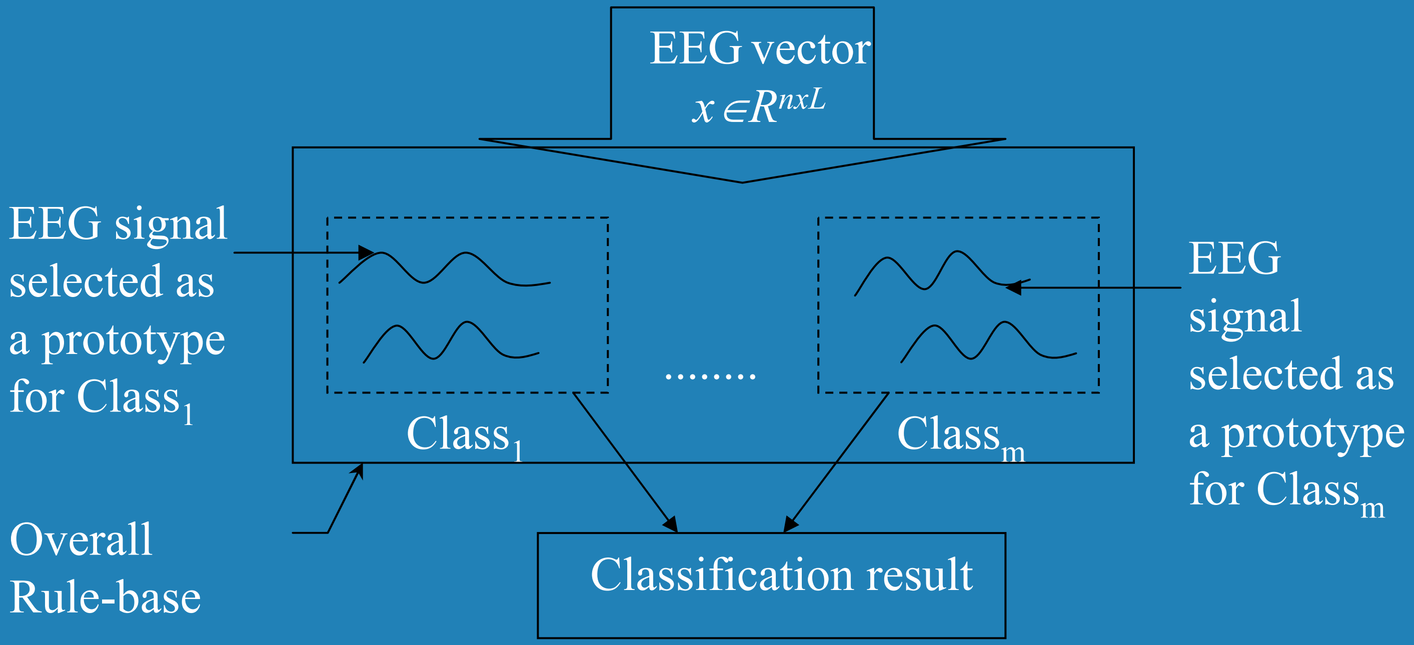
6) If in addition to 5) the new EEG signal is close to an old prototype then the new EEG signal replaces this prototype

The condition to have lower S is a very strong one, which restricts excessively large rule base to be formed

eClass

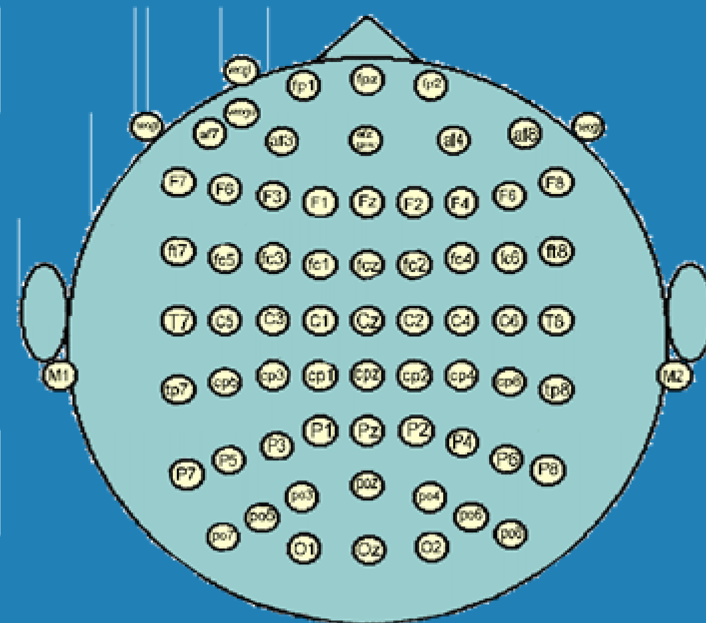
$Rule^j$: **IF** (EEG_1 is EEG_1^{j*}) **AND** …**AND** (EEG_n is EEG_n^{j*})

THEN (Class is Pain/NoPain)

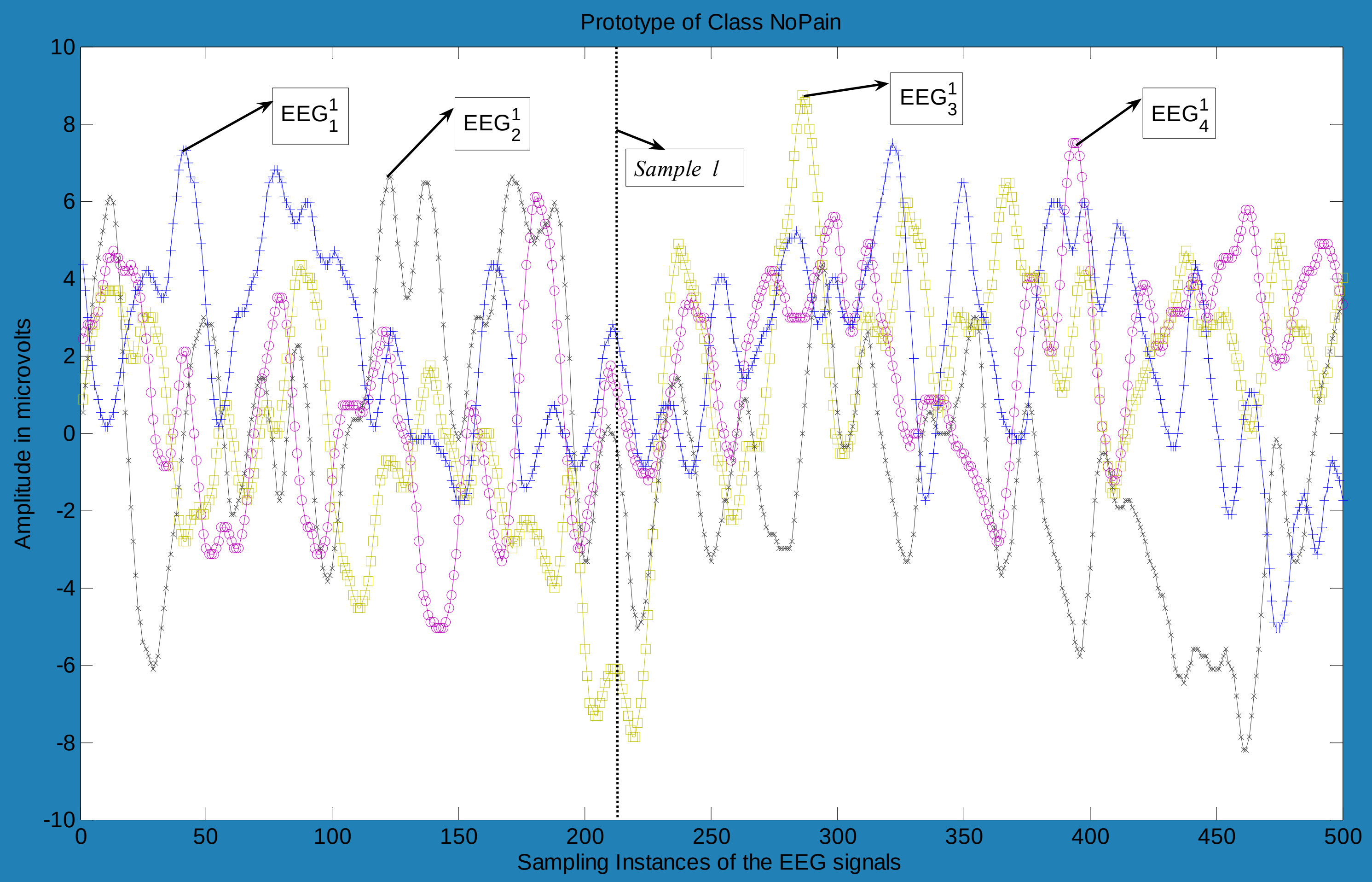


EEG, Experimental set-up

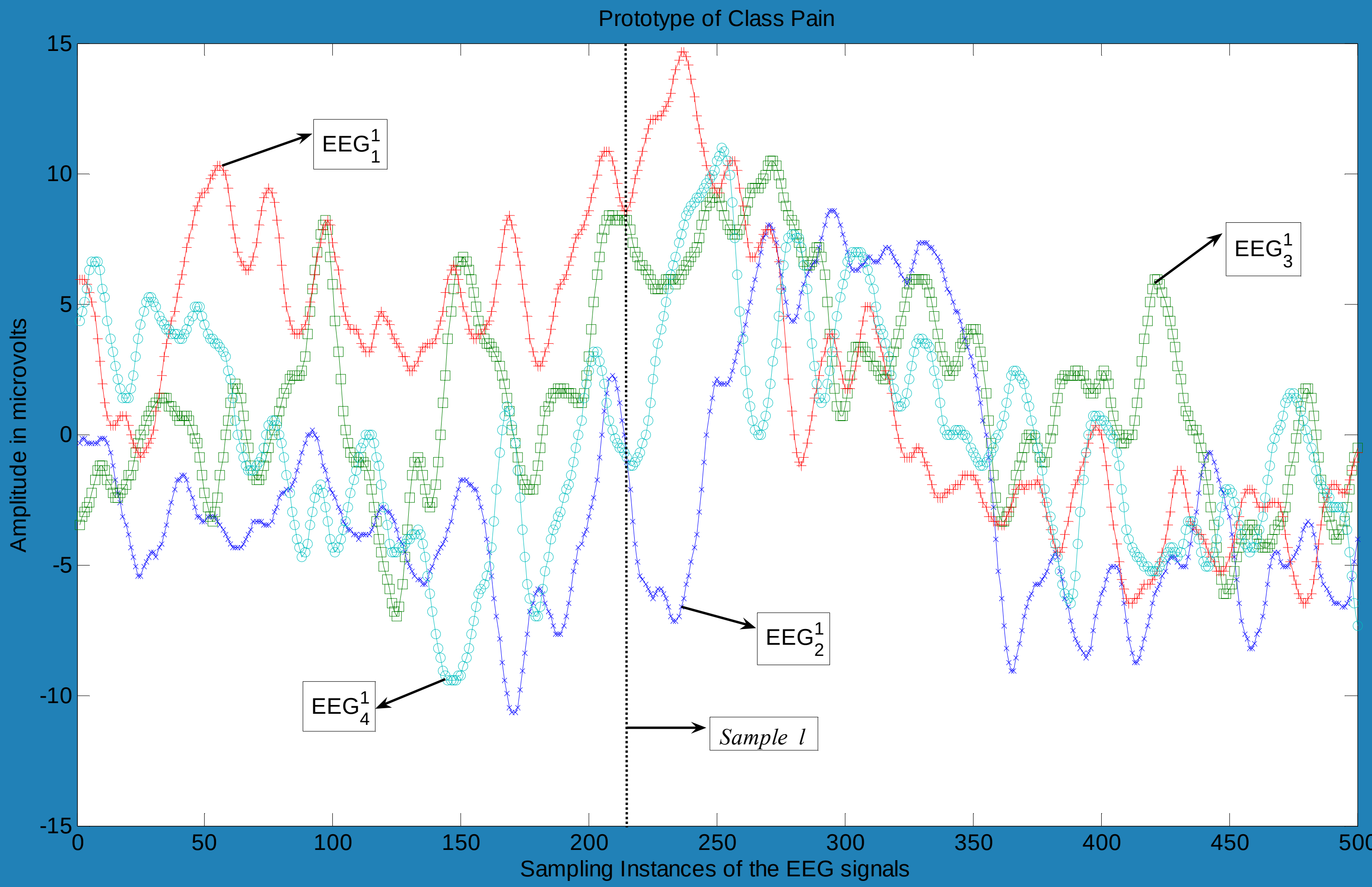
- Hope Hospital, Liverpool, UK
- Female volunteer
- 64 electrode cap



Prototype EEG signals 'No Pain' class



Prototype EEG signals ‘Pain’ class



eClass: EEG - results

- The subject was a healthy female
- EEG data were recorded on 8th and 15th June 2000, and 2nd August 2000
- On each of these days, three EEG recordings were taken during “painful” laser stimulation of the right arm
- In total, 9 continuous EEG data files
- 355 Pain epochs (Class 1) and 355 No-Pain ones (Class2)
- 168 fuzzy rules formed by eClass; 79.45% rate

Fermentation Processes On-line Prediction

InfoLab21
Department of Communication Systems

- First-principles model:

$$X_{k+1} = \mu_X(X_k, S_k, DO_k) \qquad k=0,\dots,N-1$$
$$S_{k+1} = q_S(X_k, S_k, DO_k)$$
$$DO_{k+1} = q_{DO}(X_k, S_k, DO_k)$$

- eTS model:

*R_1: **IF** (S is Very High) **AND** (DO is Very High) **THEN*** $X = a_0^1 + a_1^1 S + a_2^1 DO$

*R_2: **IF** (S is Very Low) **AND** (DO is Very Low) **THEN*** $X = a_0^2 + a_1^2 S + a_2^2 DO$

*R_3: **IF** (S is High) **AND** (DO is High) **THEN*** $X = a_0^3 + a_1^3 S + a_2^3 DO$

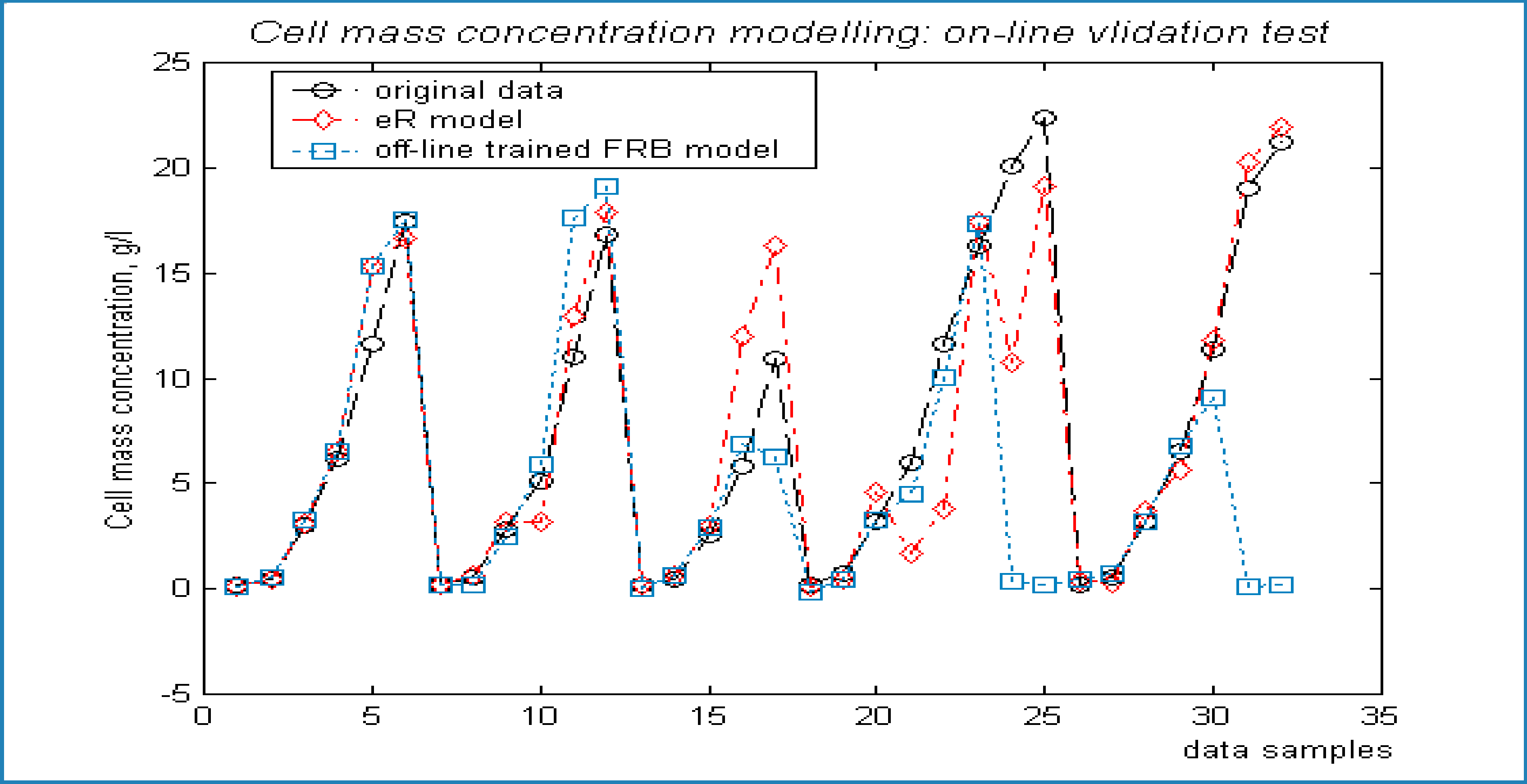
*R_4: **IF** (S is Low) **AND** (DO is Low) **THEN*** $X = a_0^4 + a_1^4 S + a_2^4 DO$

*R_5: **IF** (S is Very Medium) **AND** (DO is Medium) **THEN*** $X = a_0^5 + a_1^5 S + a_2^5 DO$

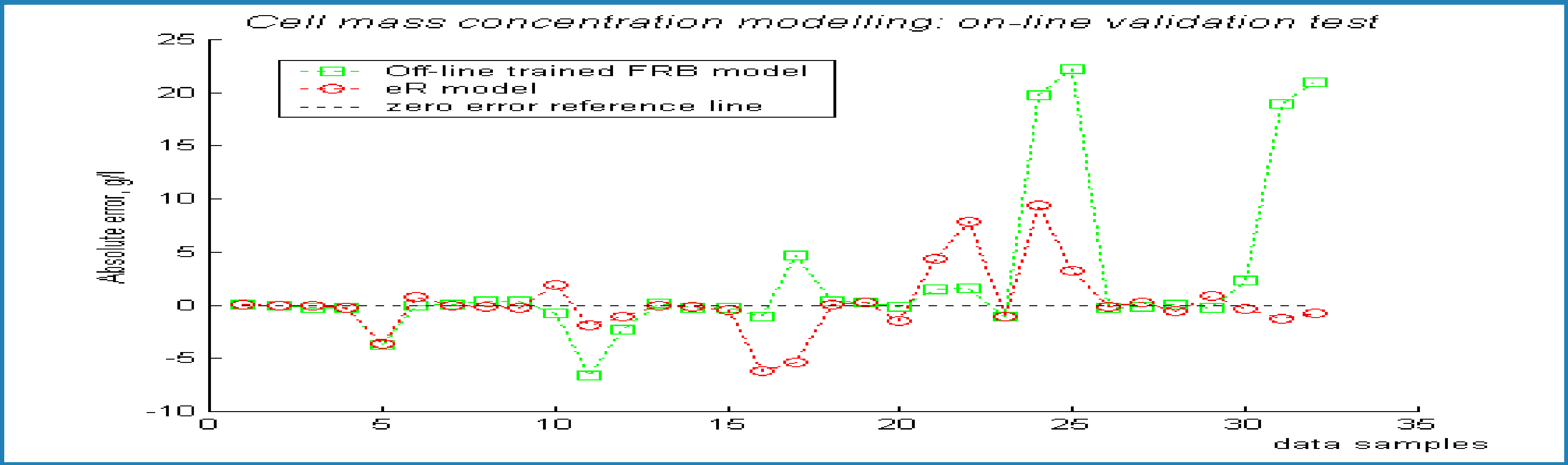
*R_6: **IF** (S is Extremely High) **AND** (DO is Extremely High) **THEN*** $X = a_0^6 + a_1^6 S + a_2^6 DO$

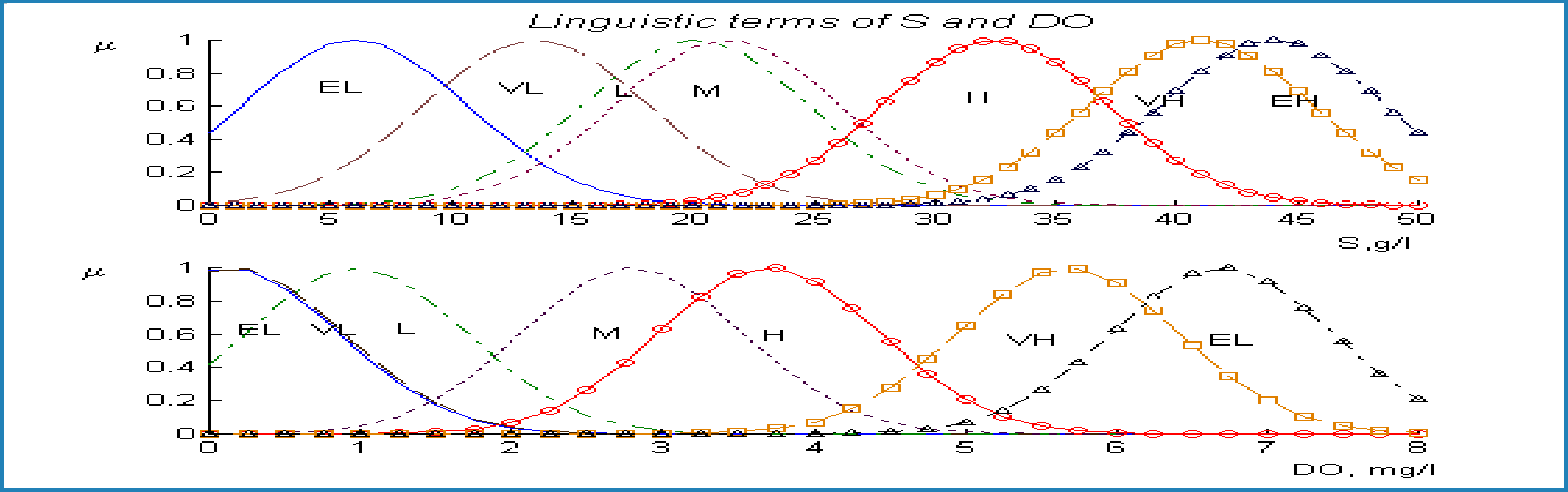
*R_7: **IF** (S is Extremely Low) **AND** (DO is Extremely Low) **THEN*** $X = a_0^7 + a_1^7 S + a_2^7 DO$)

BioProcess Modeling



eTS model of a fermentation





InfoLab21
Department of Communication Systems

Concluding remarks

- Usable in real-time, intelligent sensors
- **Non-iterative** (one-pass, incremental)
- Recursive (comp very efficient)
- adapts the structure (plus parameters)
- enrich/adds and replaces = **evolves**
- **simple** (locally linear, no search)
- **gradual** changes (inheritance, R+1)
- **starts from scratch** (no *a priori* info)
- Number of applications