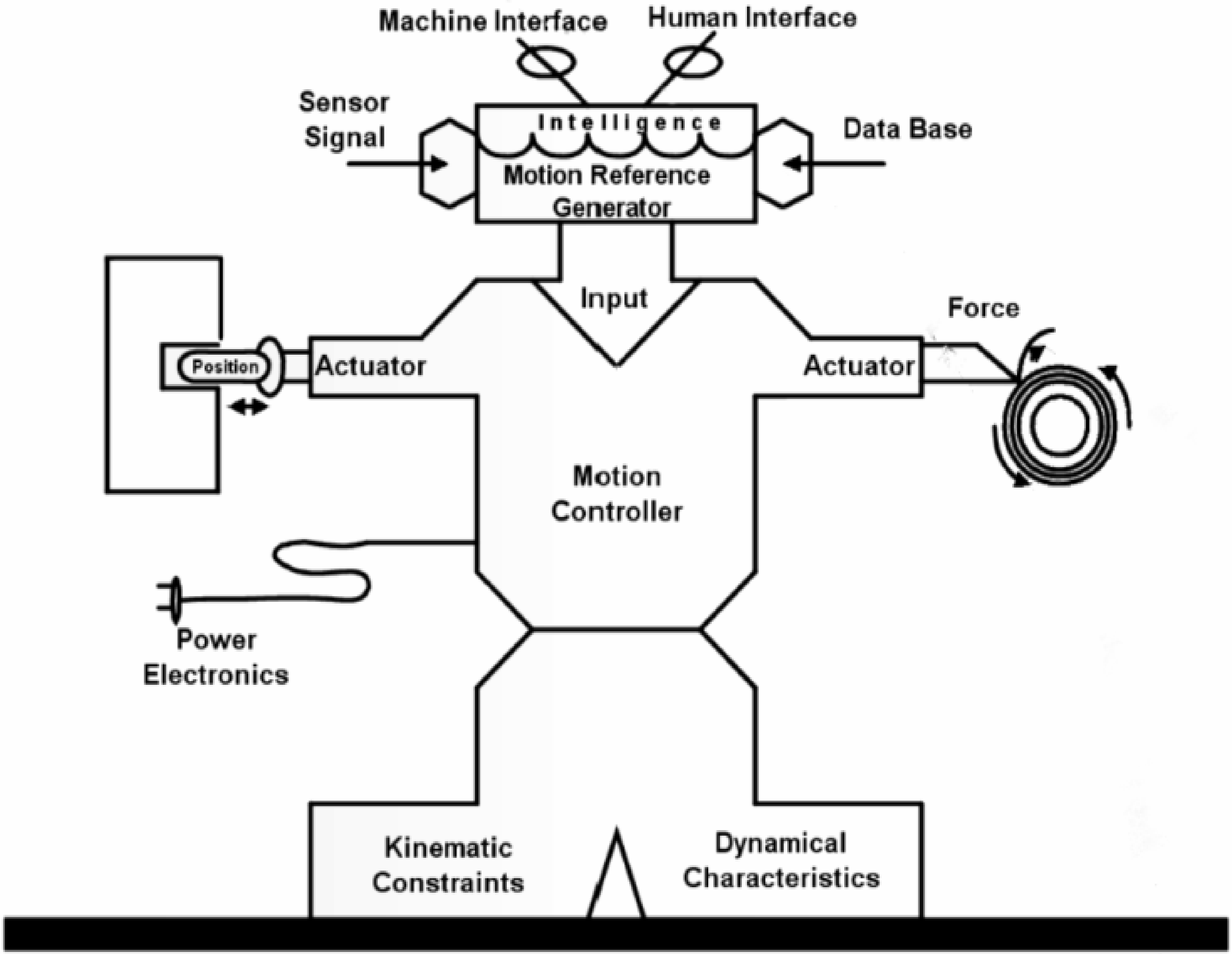

Data fusion, context awareness and fault detection in autonomous robotics

A. Vitko

Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics
Bratislava, Slovakia


Machine Interface
Human Interface
Sensor Signal
Intelligence
Data Base
Motion Reference Generator
Input
Force
Position
Actuator
Actuator
Motion Controller
Power Electronics
Kinematic Constraints
Dynamical Characteristics

Main points:

1. **What is an intelligent system?**
2. **What is context and contex awareness?**
3. **Classical versus behaviour-based AI**
4. **Why intelligent system needs fused data rather then separate single data?**
5. **Fusion hierarchy**
6. **Means and methods of data fusion - a brief survey**
7. **Detection, clustering and classification of an abnormal behaviour of the robot's leg**

WHAT IS AN INTELLIGENT SYSTEM ?

The control community is familiar with the term "intelligent control", connoting abilities that a conventional control system cannot attain.

Therefore the **Intelligent control** **has been traditionally linked with features** that were out of the scope of specialists in conventional control theory.

To perform higher level tasks, the **robot functionality** must be organized into an **appropriate architecture**, **i.e. a set of core components and organizing principles that are used to build the basis for the system.**

Core features of intelligent system

- decision-making
- adaptation to the uncertain media
- self-organization,
- planning,
- image recognition, and the more...

Intelligent control systems should not be restricted
to those based on
particularr constituents of soft computing techniques
(fuzzy logic, neural networks, genetic algorithms and probabilistic reasoning)
as it is sometimes done.

Soft computing techniques
should be considered as mere building blocks or even "bricks„
used for building up
a "large house" of an intelligent system.

WHAT IS CONTEXT ?

Definition:

Any information that characterizes the situation of an entity (person, place, object) and that is considered to be relevant to interactions between a user and applications.

Context intends to capture:

- **spatial and temporal properties** and phenomena of a real physical world which are perceptible by the senses/sensors (sensed context)
- **non-sensible data** (e.g. user's emotional state, social organization of artifacts, etc.)

Sensed context is ultimitely derivable from sensors.

The sensed context may be derived **from other contexts via transformation or interpretation**.

Sensed context is propositional in nature **and typically of the form:**

"The phenomenon P has property p"

"The device D is at location L"

"The time is T"

"The group G is meeting in room R at time T"

Being propositional, the context may be **represented by** the **first-order predicate logic**, and **associated with meta-propositional properties**, like **judgements of the quality** of the information (certainty, probability, repetability, frequency, time constantness, commpetness etc.)

Context awarenes – possibility to act on the basis of sensed context

Classical versus behaviour-based AI

CLASSICAL AI

- Classical AI uses knowledge that employs explicit symbolic representation of the environment
- Knowledge and its representation play a central role
- Control is decomposed into functional modules which process the sensed signals in serial stages.

Specific modules:
A perception module tries to reconstruct what has been sensed. This knowledge is sent to a planning module that plans a way of fulfilling robots' objectives. Finally, the commands generated by the planning module are sent to actuator module that executes actions.

BEHAVIOURAL VERSUS FUNCTIONAL DECOMPOSITION

The classic paradigm, known as **Good Old Fashioned AI (GOFAI)** uses **functional decomposition**, where perception, planning and action are **done sequentially**.

Behavioural decomposition decentralizes and divides tasks
into different **behaviour "layers".**

Every layer has the ability to overrun or subsume the behaviours of lower layers,
thus the term SUBSUMPTION ARCHITECTURE comes into play

Though sophisticated behaviours, consciousness, emotions etc.,
are tough terms to describe
the key concepts behind consumption architecture are simple and elegant.

BBC has its shortcomings,
namely **lack of predictability** and **exploding complexity**
when adding new skills.

What makes current subsumption-based artificial systems intelligent,
is just **the synergistic use of softcomputing techniques**,
which in time and space
invoke, optimize and fuse elementary behaviors into a global behavior.

As to that the fuzzy inference is a **computing framework**
which is not able to learn, it is often transformed to
a **neural network which learns from scratch.**

Comon neural network-like
architectures are **ANFIS** and **NEFCON** with the Takagi-Sugeno-Kang
and Mamdani inference mechanisms.

Intelligence of neuro-fuzzy based systems
springs from **successive generalization** of
information granules – from **crisp** through **interval granular** to **fuzzy granular** …

The strength of fuzzy-neural systems **stems from**
their **robustness** with respect to **imprecision, uncertainty, and partial truth.**

WHAT IS BEHAVIOUR ?

Mataric'definition:

A behaviour $\boxed{\text{is a control law}}$ to achieve and maintain a particular goal.

Behaviour is a basic unit for control, representation and learning.

Behaviour has emergent properties which result from interactions.

Contrary to deliberative strategy the **BBC is an extension of pure reactive** strategies in a way that
it is not based on a look-up table
and **does not employ any explicit model**.

The human or animal brain does run "programs" or "algorithms.
Its intelligence manifests itself in various behaviours and interaction with the outside word.

BBC does need any special system
to perform high-level cognitive tasks - interactions are enough

AT LEAST THREE LEVELS CAN BE IDENTIFIED:

High level behaviour (a.k.o. a **task oriented** behaviour)
- **goal seeking behaviour (a blend of low level behaviours)**

Middle level behaviours
- **obstacle avoiding behaviour**
- **inside a deadlock behaviour**
- **wall - following behaviour, etc**

Low level behaviour may be represented by
- **emergency-like behaviour** (with **highest priority**).

E.g. The sensor signal is used to directly stop the robot *if an obstacle appears dangerously close* so that avoiding is impossible.

This is done by an **atomic action** **(stop, turn by +180 degrees,etc.)**

The behaviours may be easily implemented by Neuro fuzzy learning system.

IDEA OF SUBSUMPTION ARCHITECTURE

heralded a fundamentally new approach

to get robots more intelligent.

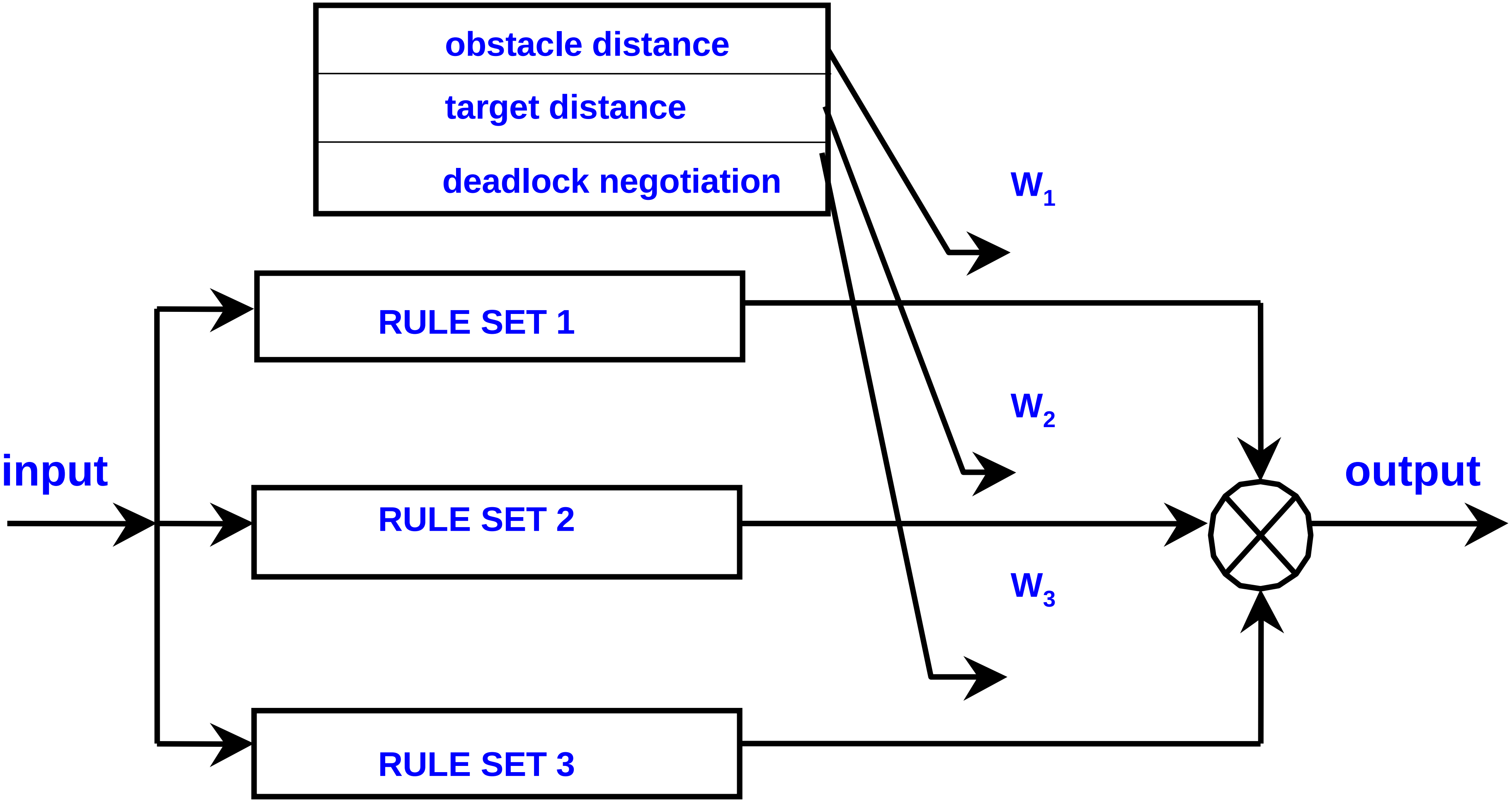
O**verall behaviour is typically broken down** into
a set of simpler behaviours, which are loosely co-ordinated.

Every behaviour either selectively **assists** (with variable intensity) or **assumes** control of the subsumed behaviours.

Layers of lower priorities can inhibit or even supersede
the ones with higher priorities.

Contrary to hierarchical control structure,
behaviours can appear **concurrently**

Fusion (blending) behaviours



Subsumption architecture
belongs to the behaviour- based strategies.

Advantage:
if implemented by fuzzy IF-THEN rules,
the transit from one behaviour to another **is very smooth.**

The transit is **exclusively controlled** by
the instantaneous sensor data
and no historic data are utilised.

On-line **learning** could be **easily included** by **conversion** of the fuzzy set
to a fuzzy-neuro structure

All these facts may be considered to be
next step towards
improvement of the system intelligence.

Why an intelligent robot needs fused data?

NEEDS FOR SENSOR INTEGRATION

Any autonomously operating system, be it a terrestrial mobile machine or UAV is required to respond to instantaneous incentives coming form the environment.

To this end the system needs to handle wide range of unexpected phenomena.

Therefore, the prior prerequisite is the system's ability to transform huge amounts of information into a more information-rich structure and use it for upgrading its knowledge about what is going on in the observed environment.

Automous robot is an instantiation of intelligent systems.

Its <u>functionality relies on</u> the fused information
coming from numerous disparate sensors through which
the <u>**robot grasps**</u>
<u>**a consistent and coherent view**</u>
of both **its own state**
and the **state of surrounding environment**.

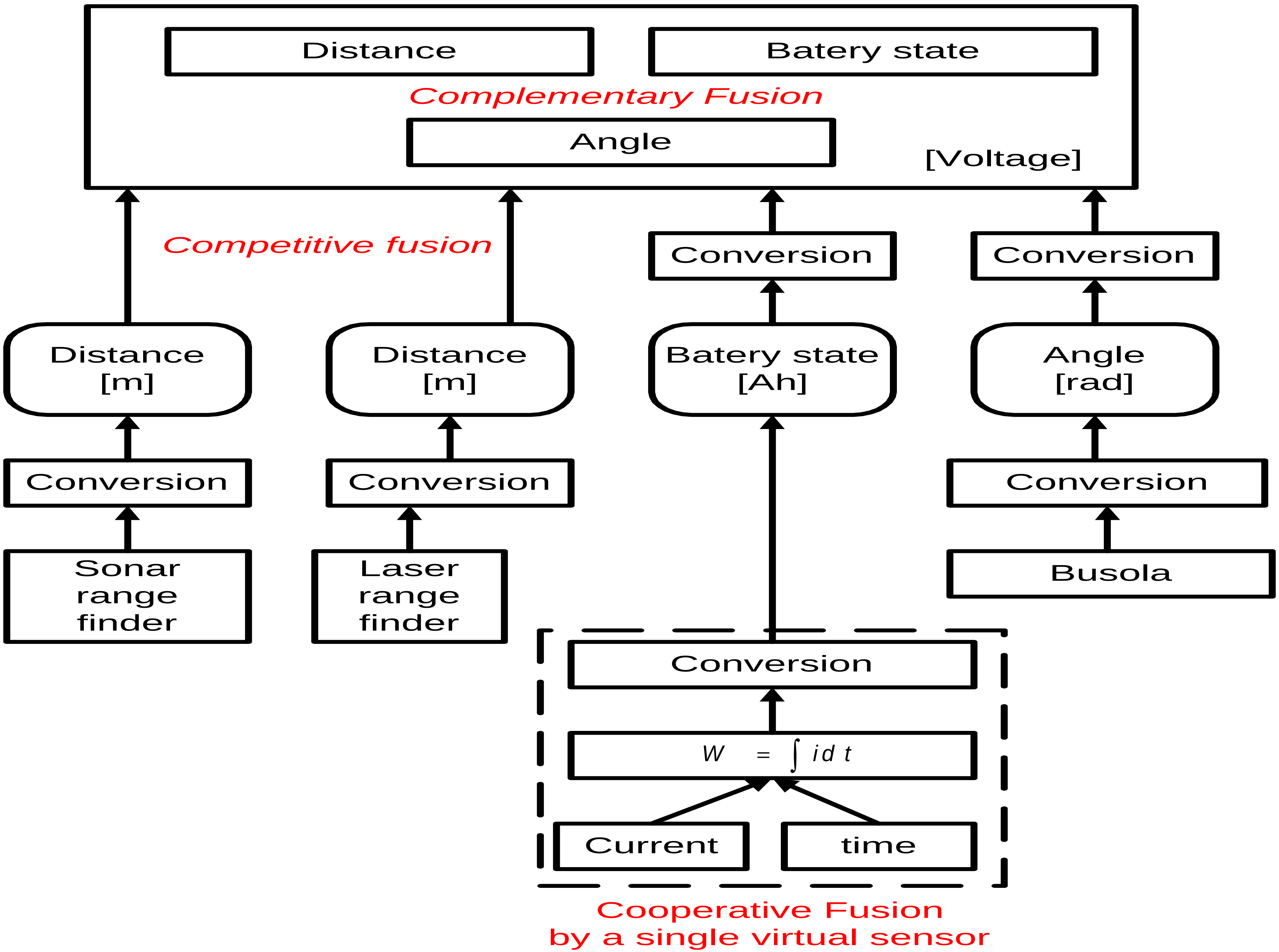
Fused information <u>is beneficial from (at least) three aspects:</u>

noise reduction - overall uncertainty is reduced
novelty extraction - makes the hidden data patterns more obvious
improvement of overall robustness - because <u>solitary</u> sensors operate under certain conditions and their performance <u>could be abruptly degraded</u> by environment changes (e.g. illumination conditions).

TWO SPECIFIC REASONS

First, the system is to obtain more complex information. This is done by fusing ***complementary*** information i.e. **pieces of information which are mutually complemented.**
For instance images from two cameras looking in different directions

Second, the system is to obtain more accurate information. This is done **by multiple disparate sensors**, which **sense the same quantity**.
For instance, sonar and laser range sensors can should sense the distance. In this case the **sensors "compete" in a sense**, therefore one can speak about ***competitive fusion***.


Distance
Batery state
Complementary Fusion
Angle
[Voltage]
Competitive fusion
Conversion
Conversion
Distance [m]
Distance [m]
Batery state [Ah]
Angle [rad]
Conversion
Conversion
Conversion
Sonar range finder
Laser range finder
Busola
Conversion
$W = \int i\,dt$
Current
time
Cooperative Fusion
by a single virtual sensor

FUSION IN TIME AND SPACE

Robot commonly operates in a changing environment.
On that account the information fusion must take place
not only in space but also in time.

Hence, the sensors of different modalities
should be distributed in space
and sense data in a time multiplex.

Such structure allows fusing high-level information
(**e.g. statements**) and even **to grasp a context**.

IMPORTANT! A measure of certainty of the sensed information must be fused as well !

ROLE OF META-INFORMATION

Outputs of the fusion process are not only estimates of the sensed quantities but also their certainties

In case of the **Kalman filter** the **result is an estimate of the mean** while **variance of the mean** is a **measure of its certainty.**

In case of the **Dempster-Shafer fusion the result is** a **statement,** while its **believe** is a **measure of its certainty.**

In case of the **fuzzy fusion**, **the result is a consequent statement** while the **degree of rule fulfilment is its measure of certainty**.

Certainty is a kind of meta-information

Hierarchy of the data fusion

FUSION LAYERS

Low levels are delineated for fusing pixels and features.
For instance, **if grey values** of neighbouring pixels are above a given threshold the AND filter is true.

Features are patterns occurring in data for instance means, correlations, variances, kurtosis, co-variances etc.

The low-level fusion uses Kalman filter as a typical representative while **declarations** about the **current context** are fused **at higher levels**.

Brief survey of means and methods

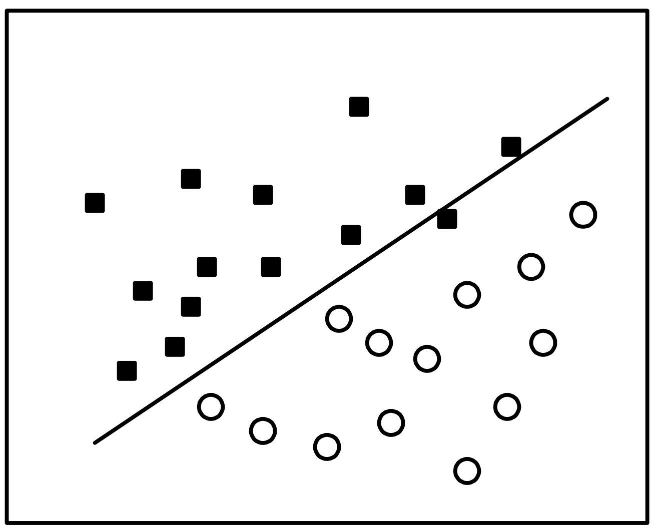
PATTERN

is an ordering that contains some underlying structure.

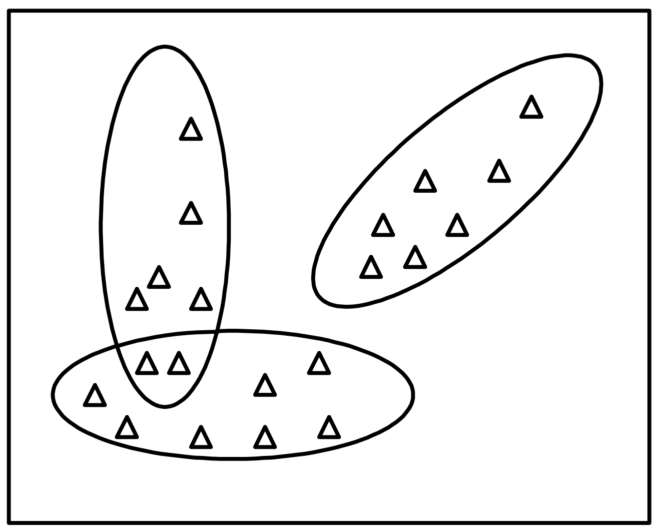
Tasks of pattern processing:

Classification **- maps data into several <u>predefined classes</u>**

Clustering **- identifies clusters of <u>similar data</u>**



Classification



Clustering

Bayesian fusion

The system searches for **posterior probability $p(Z/R_A)$** of the **proposition Z** given particular sensor reading "R_A" of the sensor A.

Certainty of the proposition Z is then given by **$p(Z/R_A)$** using **Bayes rule:**

$$p(Z|R_A) = \frac{p(R_A|Z)}{p(R_A)} p(Z)$$

Fusion of two readings R_A, R_B is performed by taking the **joint probability**, which is given **by the product of two conditional probabilities.** (for competitive fusion) **and by the sum** (for complementary fusion)

That is, for competitive fusion we have

$$p_{joint}(Z|R_A, R_B) = \frac{p(Z|R_A)}{p(Z)} p(Z|R_B)$$

Signal level fusion - Kalman filter (a case of direct measurements)

Let us suppose that the normally distributed random **signal** ***X*** is directly measured by two different **sensors S1 and S2.**

Two **estimates of the X are x_1, x_2,**

and **corresponding (un)certainties** are given by **standard deviations σ_1, σ_2.**

The LMS optimal estimate X is then **fused in accordance with the rule**

$$X = \left[\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right] x_1 + \left[\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right] x_2$$

and the **variance σ^2** of the **fused estimate X** fulfils the relation

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

The **LMS optimal estimate** can be rewrited as

$$X = x_2 + \left[\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right](x_1 - x_2)$$

EXAMPLE

Let data x_a, x_b , exhibiting Normal distribution

$$x = N(\bar{x}, \sigma)$$

are delivered by two different sensors and weighted

$$x = W_a x_a + W_b x_b$$

Then the **LMS optimal estimate X** is given by

$$X = \left[\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right] x_1 + \left[\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right] x_2$$

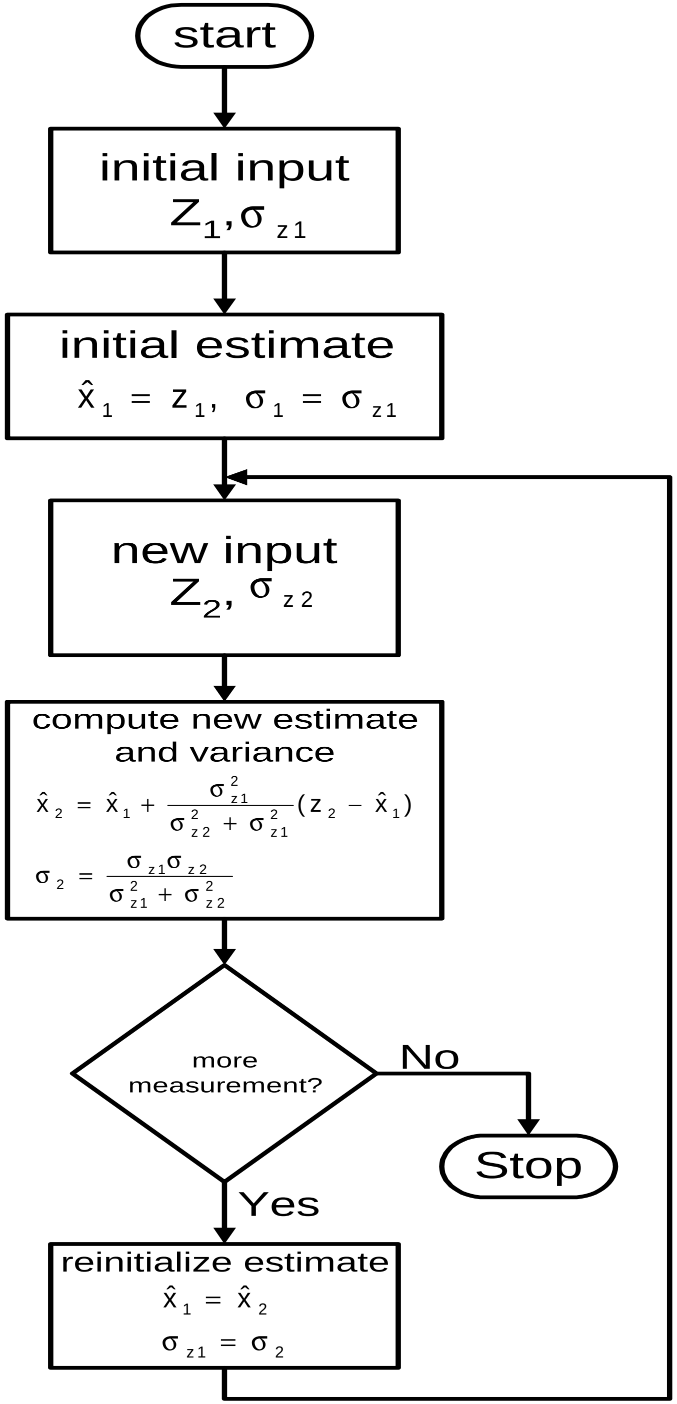
and variance of the estimate X, will be given

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

The estimate X can be rewrite as

$$X = x_2 + \left[\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right](x_1 - x_2)$$

$$x = x_2 + K(x_1 - x_2)$$



Weakness of Kalman filtering

The Kalman filter works well
under some certain conditions.

It does not allow adding or removing sensors.

Hence, once the Kalman filter has been defined
to accommodate certain sensors,
all those sensors and no others
must give information in all iterations.

Kalman filter cannot handle sensor intermittence,
that is ,
if one of sensors does not provide
new information for the current iteration,
the output of the filter can be corrupted.

HIGH LEVEL FUSION – a.k.o. DECISION MAKING

High level fusion **combines propositions**.
Each proposition is accompanied with its **certainty value (score)**,
expressing a **measure of truthfulness**.

Examples of the propositions forom the field of robotics

$z_{i.e}$ = there is a cube "i" in the robot's environment "e"
$z_{i.c}$ = object "i" belongs to the cluster "c"
$z_{d,\alpha}$= at the angle "α" there as an obstacle at the distance "d"

High-level fusion uses:

- **Probabilistic means (Bayesian statistics)**
- **Dempster-Shafer theory of evidence**
- **Theory of fuzzy sets and fuzzy inferrence**

Higher levels are occupied by more sophisticated procedures of
notion identification and notion interpretation

i.e. intend to identify **what was observed ?**
and interpret **what it means to have observed that ?**

That is why higher levels are domains where the
Statistical means, Dempster –Shafer theory of evidence,
Fuzzy logic (and other symbolic logics)
are commonly used.

DEMPSTER–SHAFER THEORY OF EVIDENCE

(as a means of data fusion)

All independent declarations about the environment are enumerated in a set $\Theta = \{E_1, E_2 \ldots E_n\}$ - frame of discernment.

A certainty - m (*believe mass* or *basic probability assignment-bpa*) is assigned not only to a single event (E) but also to every element of the power set - Λ (a set of all subsets of the set Θ).

Bpa-s are finally combined

by the Dempster's rule of combination

as will be exemplified next

The inference engine then searches for the *total evidence* denoted "*believe (E)*" or *Bel (E)* that supports an occurence of the event E.

Bel (E) is given by the sum of the masses „m“ assigned to (not necesary all) subsets B of the power set, which form a part of E

$$Bel(E) = \sum_{\forall B, B \subseteq A} m(B)$$

It is also possible that the certainty is left uncommitted.

For instance

Object O belongs to C1 with m(C1) = 0.3, or else I do not know

The final belief in existence of the (fault) event E, is then obtained by a (recursive) application of the Dempster's rule of combination

Dempster's rule of combination

Let the two sensors S_1 and S_2 with respective bpa-s m_1 and m_2

sense the elements of the power set Λ

Then, in accordance with the Dempster's rule of combination

the total belief

(i.e. the fused evidence supporting existence of an event E) will be

$$Bel(E) = \frac{\sum_{\forall \{B,C\} \in \Lambda : B \cap C = E} m_1(B) \cdot m_2(C)}{1 - \sum_{\forall \{B,C\} \in \Lambda : B \cap C = \varnothing} m_1(B) \cdot m_2(C)}$$

Example

Robot divides its surroundings into a grid. Each cell has assigned a mass – i.e. a measure of confidence in each of 3 possibilities **"occupied", "empty" and "unknown".**

Because "**unknown" equals to "occupied or empty",** these tree possibilities form a whole power set together with the **empty Φ, where m(Φ)=0 .**

Hence, the **power set is (Φ, Occupied, Empty, Unknown) = (Φ, O, E, O v E) = (Φ, O, E, U)**

Let the measures of confidence are m_s for sensors and m_o **for old evidence (the mass *m* from previous iteration of DS rule)**

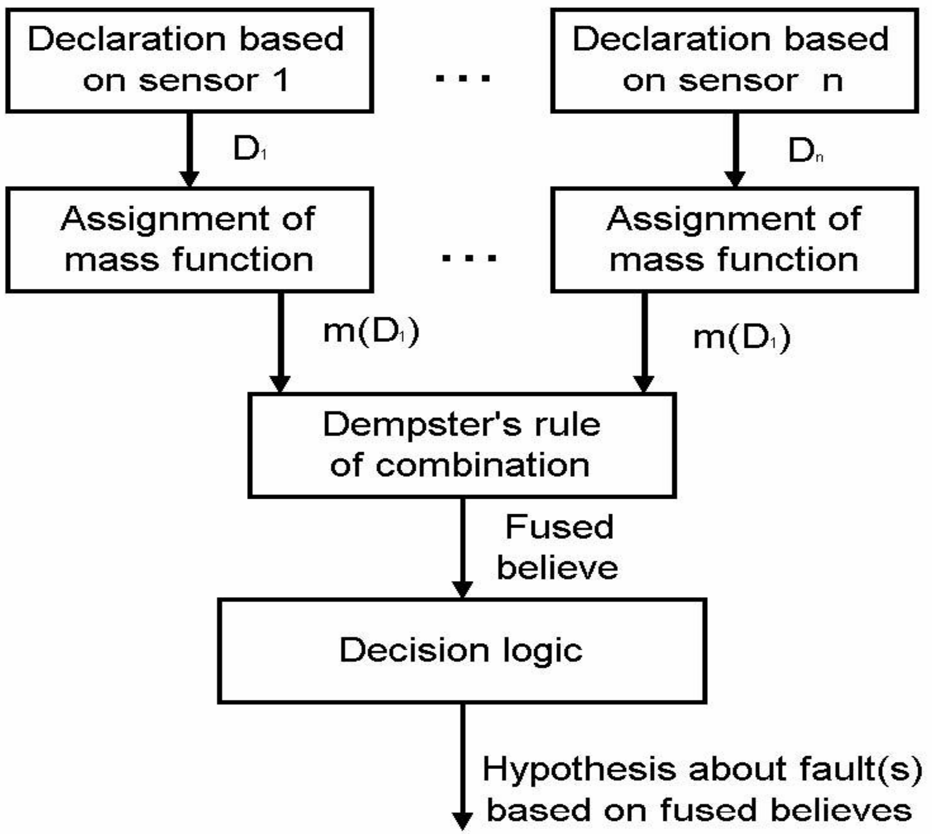
When exploring the surroundings the **robot calculates m(O**) for each cell in accordance with the above expression:

$$m(O) = \frac{m_s(O)m_o(O) + m_s(O)m_o(U) + m_s(U)m_o(O)}{1 - m_s(O)m_o(E) - m_s(E)m_o(O)}$$

If m(O) is greater then a given confidence level the robot makes a point...
In this way the robot generates a free path

Quality (of the results) depends on the choice of the sensor masses m_s

Structure of the fusion based on Dempster-Shafer evidence theory



The DS' theory is a generalization of Bayesian statistic in the sense that it **allows for expression of partial beliefs and ignorance.**

The belief in a proposition **does not imply any belief in its complement**, thus allowing an expression for **ignorance**. **Ignorance** is an intuitive **way of dealing with missing data**.

As withb Bayes and Kalman assesments the DS theory **allows to change prior opinion in the light of new evidence**, but at a higher level.

Some remarks:

It is also defined a dual function called *plausibility* Pl(E) of a given event E,

Pl(E) expesses a measure of evidence, <u>which does not contradict</u> E

Pl(E) = 1- Bel(non E)

The Bel(non E) is called the <u>dubiety</u> and represents the <u>degree to which the evidence impugns</u> (or undermines) the declaration about existence of the event E.

If particular evidence does not impugns E,
it does not mean
that the same evidence supports (nonE) and vice versa !!!

The following inequalities are valid

Bel(E) ≤ Pl(E)

Bel(E) + uncertainty = Pl(E)

Bel(E) + uncertainty + Bel(non E) = 1

Uncertainty interval is a measure the remaining ignorance.

FUZZY FUSION OF TWO SIGNALS

Sensed values are first granulated - fuzzification.

Certainties are expressed by values of **membership functions**.

The fusion is done by the generalized modus ponens

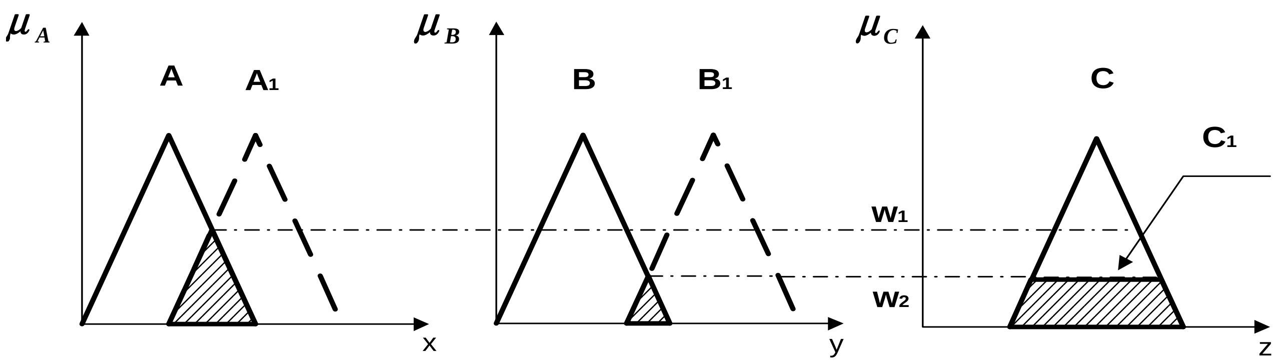
Declaration: *x is A_1 and y is B_1*

Fusion rule: *if x is A and y is B then z is C*

Conclusion: *z is C_1*

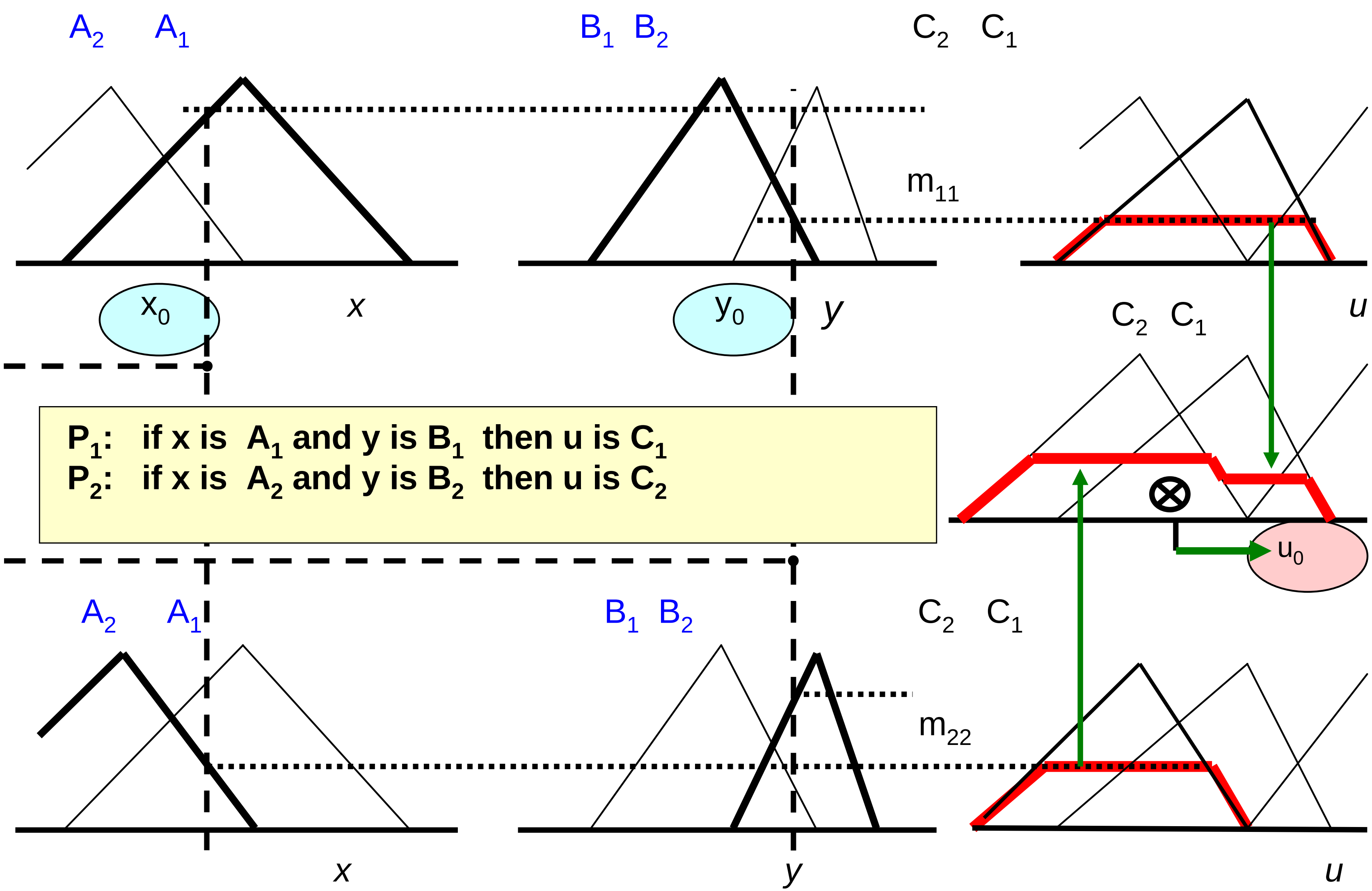
Let w_1, w_2 are **degrees of match** between **A** and **A_1**, and **B** and **B_1** respectively. Then $w_1 \wedge w_2$ is a ***degree of the rule fulfilment***.

Finally the **output membership** μ_{c1} is clipped as shown below.



$$\mu_{C1} = w_1 \wedge w_2 \wedge \mu_c$$

FUZZY FUSION OF TWO DECLARATIONS



Fuzzy logic offers the same advantages as DS theory,
but it does not suffer from exponential complexity**.**

The <u>strength of fuzzy logic</u> follows from the fact that it makes possible to capture knowledge <u>from human expert in a very intuitive manner</u>.

Competitive fusion is implemented through AND-rules
while
complementary fusion is implemented through OR-rules.

Fault detection
(traditional model- based approach)

Model – based fault detection

The idea rests in the on-line identification and continual comparison of the parameters with nominal ones.

A mismatch between them is used for detection of a fault.

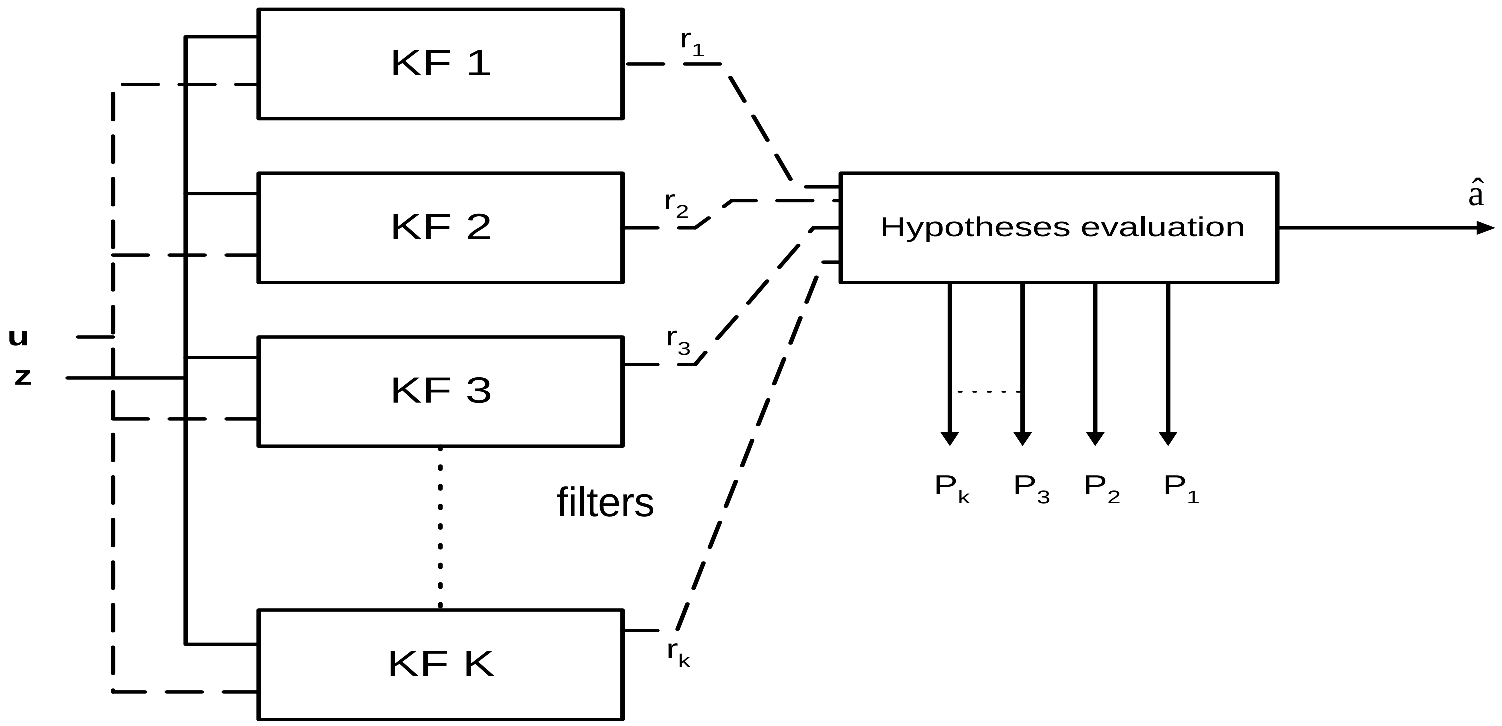
Main difficulty of this approach is that there is **no one-to-one relationship** between physical and model parameters.

An alternative to this approach compares nominal and somewhat “pathologic” **root loci.**

Traditional model- based approach uses
the bank of Kalman one-step predictors
in which the sensor signals are fused and the optimal state is predicted.

Predictors are "tuned" to particular fault dynamics
and connected to the plant in parallel
as seen in the following figure

A bank of the Kalman filters / predictors

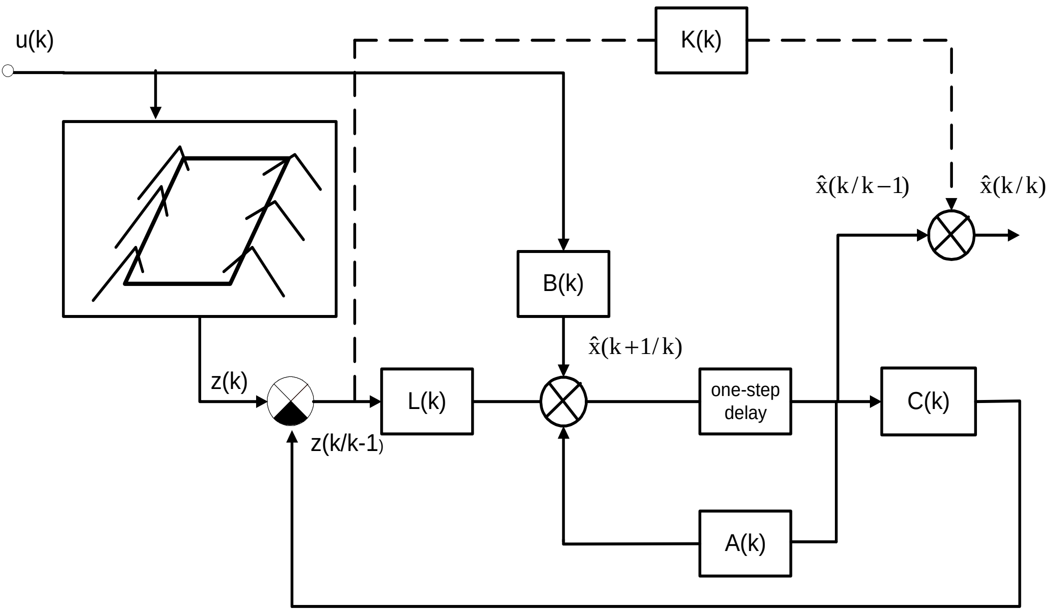


Outputs of Kalman filters **are continually compared** with that of tuned to the normal (non-faulty) dynamics and the **residuals are evaluated**.

The one with maximum residum supports hypotheses about an imminent fault.

Drawback: process is ***computationally intensive***.

A way of designing models of the fault dynamics



$$x(k+1) = Ax(k) + Bu(k) + Dw(k)$$

$$z(k) = Cx(k) + v(k)$$

The main idea behind detection of the **actuators and/or sensors <u>miss-behaviours</u>** is based on the fact that both are <u>related to the changes in the columns/raws</u> of the **control and measurement matrices B and C** respectively.

In particular, the joint actuator exerting an action through the k-th input channel of the **control vector „u“** and thus influences the system state through the k-th column of the control matrix B.

Therefore, **the total inactivity** of the k-th actuator could be modelled by zeroing the k-th column of the matrix B.

Similarly, the **total inactivity** of the j-th sensor causes that the j-th component of the <u>measurement vector z</u> is zero.

Therefore this fact may be **modelled by zeroing the j-th row** of the measurement matrix C.

A combined actuator+sensor faults can be detected by fusion of the both fault models.

Partial inactivities may be modelled by modification of the corresponding raws or colums

Fault detection
(neural network - based approach)

Main reason for using NN instead of **statistical means** is that the NN algorithms **make weaker assumptions concerning the shape of statistical distribution** of the input patterns.

In essence, any neural network can be used for fault classification.
Once learned the network can be used to classify **also unseen** fault patterns

Another major motivation is request for **detection new and unexpected faults**.

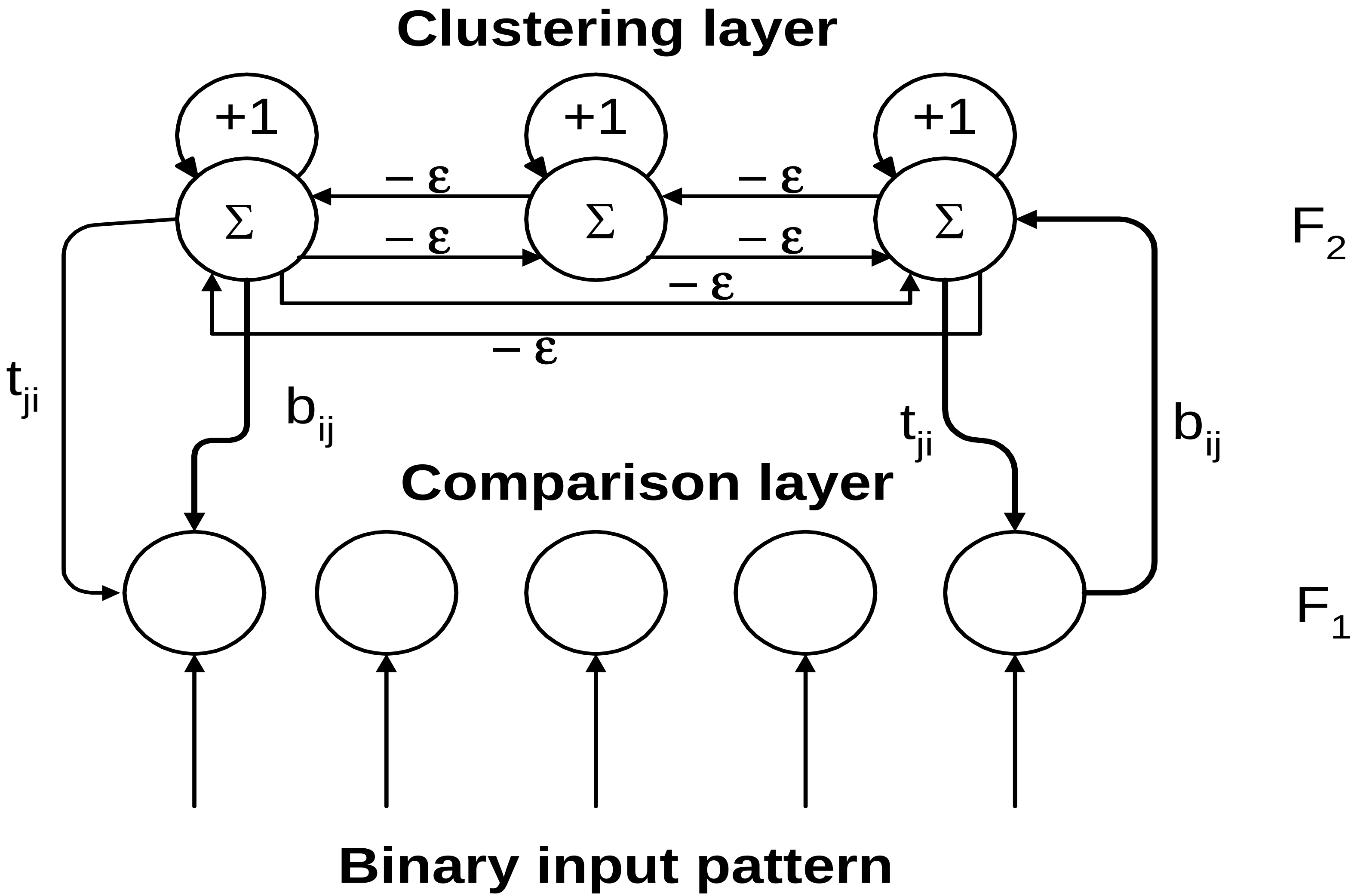
The NN should be **adaptive (plastic)** in response to any novelty,
yet remain **stable (i.e. preserve already learned patterns)**

This phenomenon is known as
stability-plasticity dilemma.

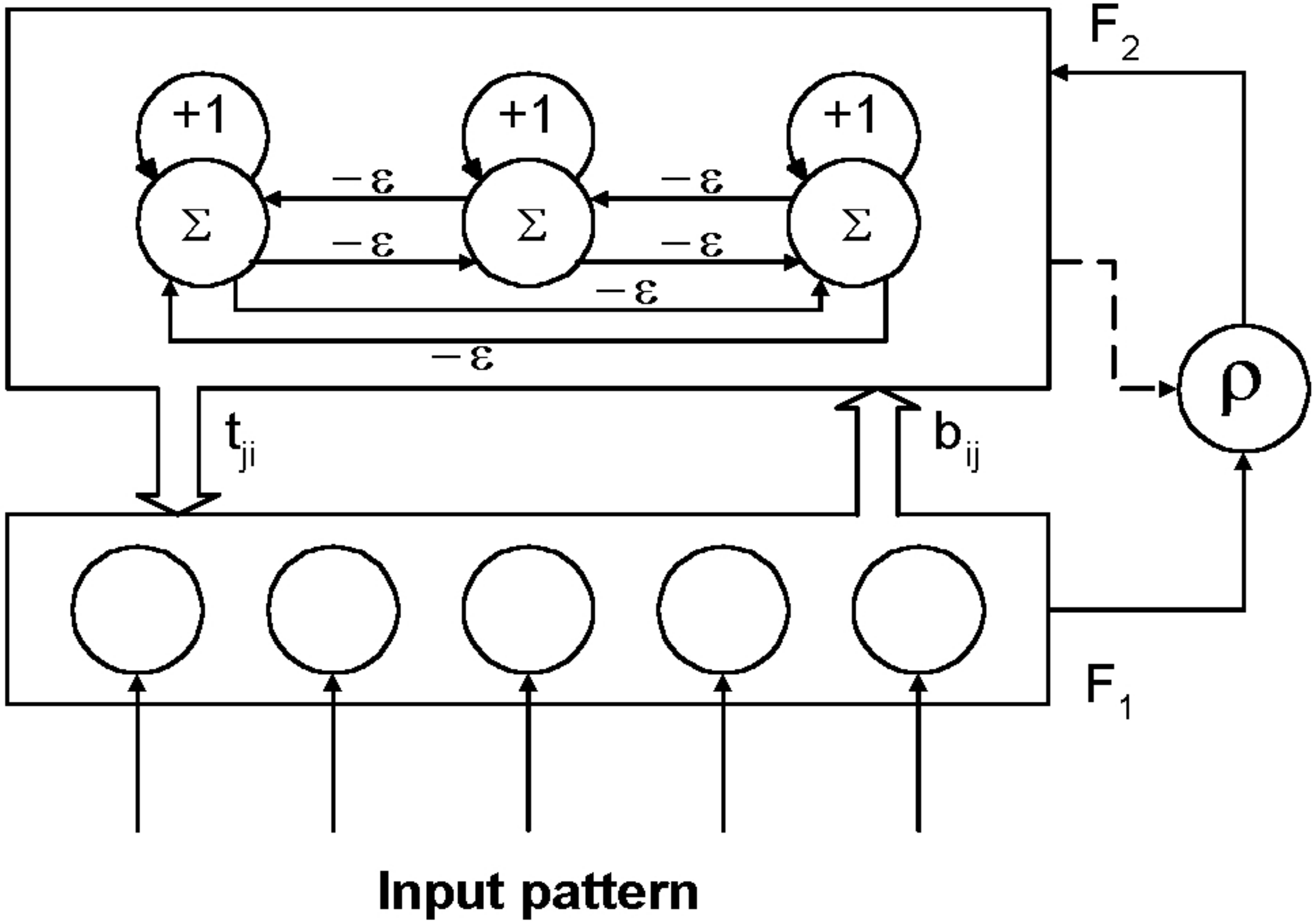
The ART neural network was developed
to resolve the stability-plasticity dilemma.

It allows the **incremental updating the clusters' prototypes**

Unsupervised learning of ART neural network



Unsupervised learning of the ART neural network



Principle

The ART consists of two fully connected single layer networks. The pattern-representation layer F1 accepts the input pattern I and through button-up weighted connections bij (initially set to one) sends it to cluster-representation layer F2.

The ART enforces a hard competitive learning mechanism, otherwise called Winner-Takes-All strategy

A binary coded **input vector I** is a concatenation of any differences between normal and faulty patterns.

Thus the **vector I** represents **a certain behavior pattern**

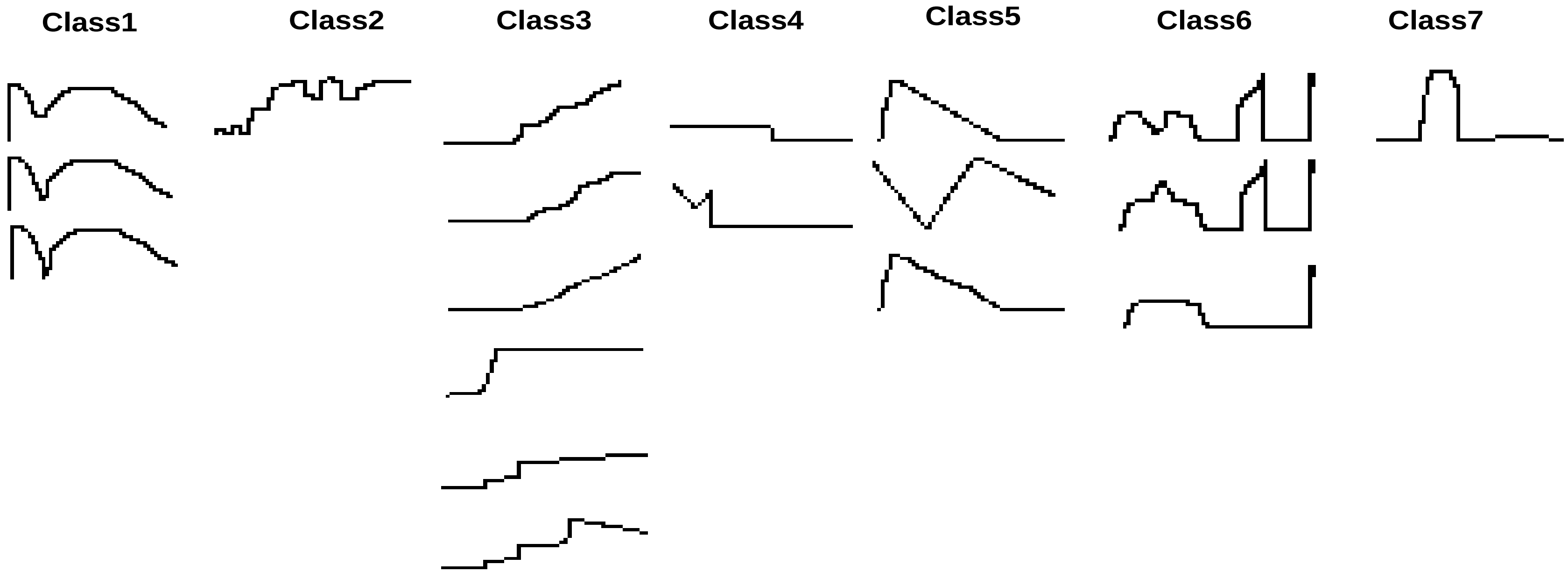
1. The first input pattern is selected to be the receptive center of the first cluster
2. Every next input pattern is compared with each of already existing receptive centers .
3. **Due to the competition caused by the negative lateral** ($-\varepsilon$) and **positive own** feedback ($+1$), the neuron in $\mathbf{F_2}$ which received highest button-up activity:

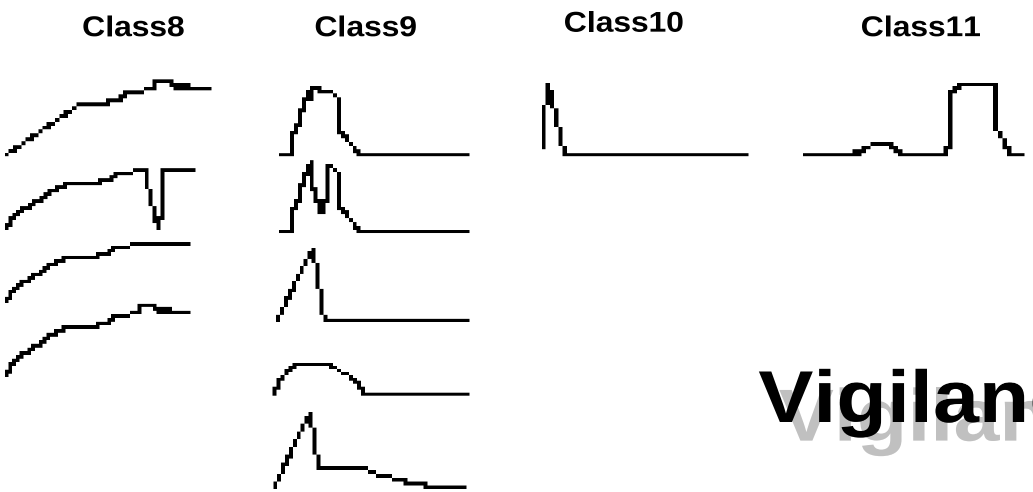
$$b_{ij} = \sum_{i=1}^{n} b_{ij} I_i \qquad\qquad j = 1 \ldots m$$

is declared a winner.

4. **Output of the winner** is set to one and projected through the top-down links back to F_1.. If the similarity between the projected winner and input pattern is greater then the user defined **vigilance** ρ, so called **resonance** occurs (hypothesis was accepted), and buttom-up weihts $\mathbf{b_{ij}}$ **are moved closer to input pattern - winner clusters the input pattern**

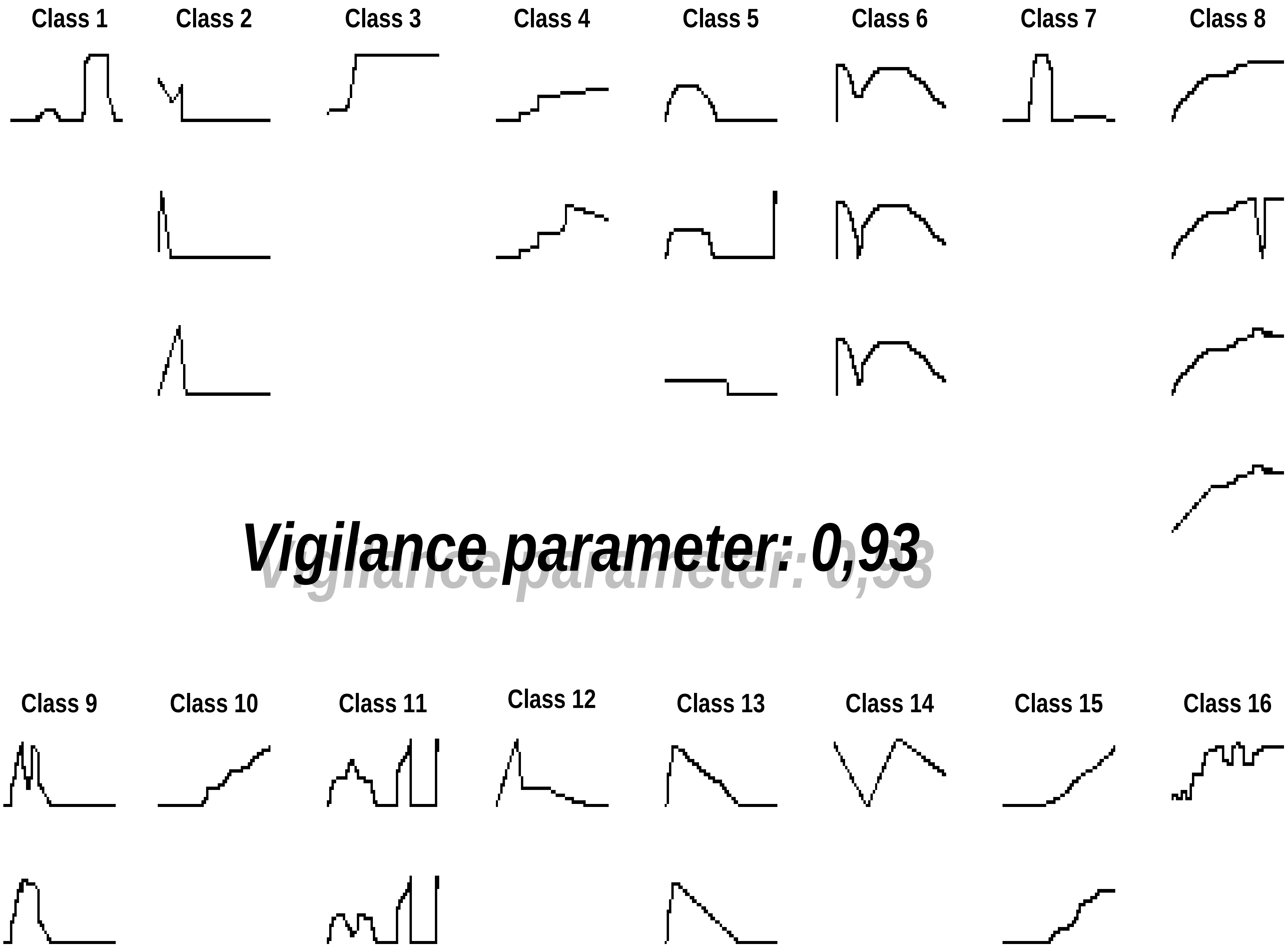
EXAMPLES : The results of the clustering and classification of 30 input patterns





Vigilance parameter: 0,9

Vigilance parameter: 0,9


Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Vigilance parameter: 0,93
Class 9
Class 10
Class 11
Class 12
Class 13
Class 14
Class 15
Class 16

Robotic applications

DETECTION, CLUSTERING AND CLASSIFICATION of the faults in robotics

Due to the extensive use of complex mechanical components like arms, legs, actuators, gears, grippers etc., the robot mechanical construction suffers from significantly higher fault rates then the pure electric and (micro) electronic circuitry.

A fault is an undesired change (tolerable malfunction) in system that degrades system performance.
A failure is a catastrophic or complete breakdown of a system.

Imminent failures are often manifestated through the declined values of system parameters or their fused complexes.

Numerous approaches to the fault management have been developed.

They range from checking the limit values, state and/or parameter estimation, **and those using the means of artificial intelligence like expert systems, case-based reasoning, and the fuzzy and neural learning approaches.**

MALFUNCTIONS OF MECHANICAL PARTS

Faults should be detected <u>sufficiently soon</u> so as to allow the system to **anticipate** **possible faults** **on the basis of classification of an abnormal behaviour.**

In other words, **novelty detection** becomes necessary.

Imminent failures are often manifested through the **novelties in system parameters**

Therefore, idea is to identify any novel behaviour.

ABNORMAL MOTION OF THE ROBOT LEG

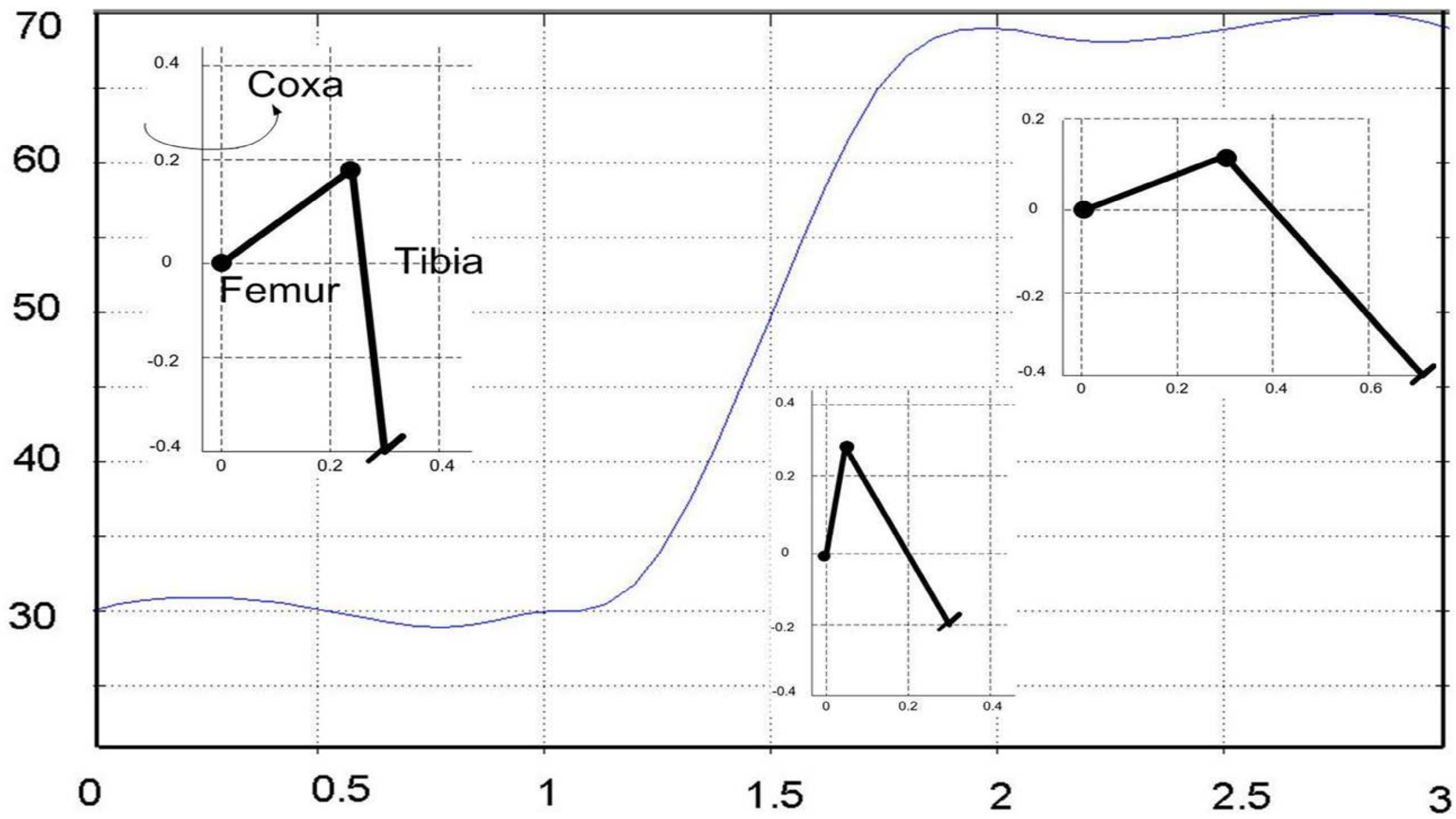
Abnormal leg motion of the leg may be induced by **increased friction in bearings, slipping or dragging clutches**, **partial loss of energy delivered to joints etc.**.

Other faults may be caused by **incorrect coordination of the legs** due to improper timing e.g. **fall out of the phase**.

Malfunctions of this kind may remain hidden for long time and may gradually lead to serious failures jeopardizing the robot stability.

To detect such novelties, a neural detector & classifier was designed and used for detection and classification of abnormal joint torques.

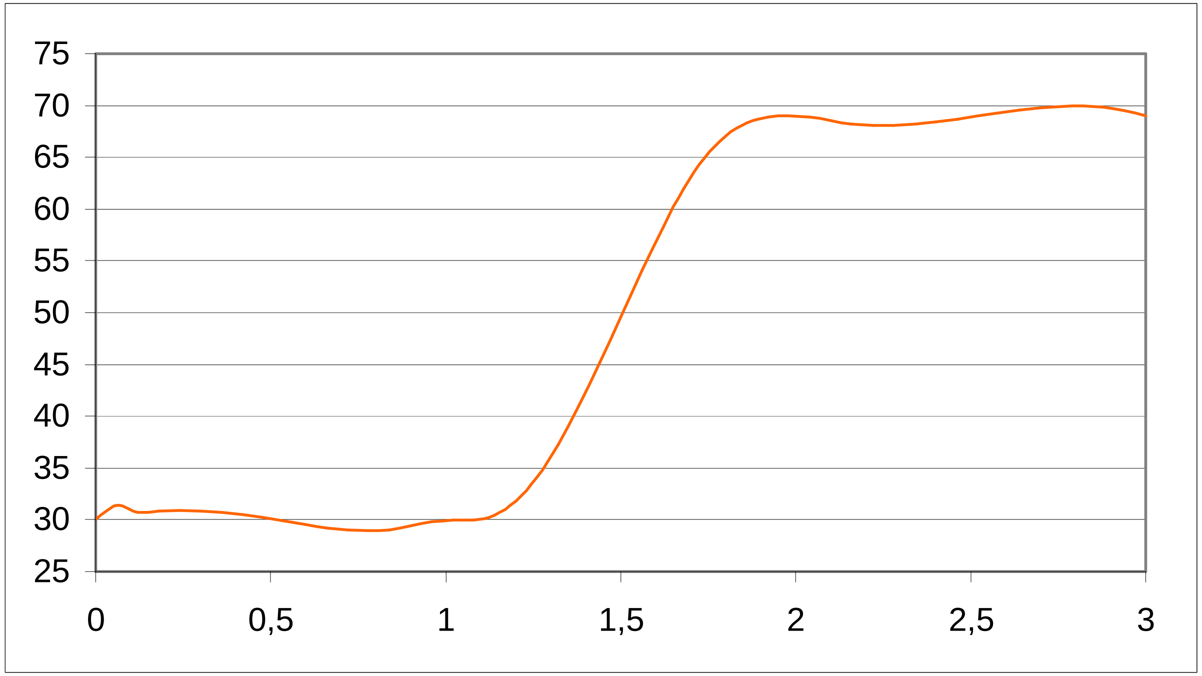
TORQUE TIME COURSE IN THE FEMUR JOINT



CLASSIFICATION OF THE SAME TORQUES BUT WITH THE INCREASED VIGILANCE TO

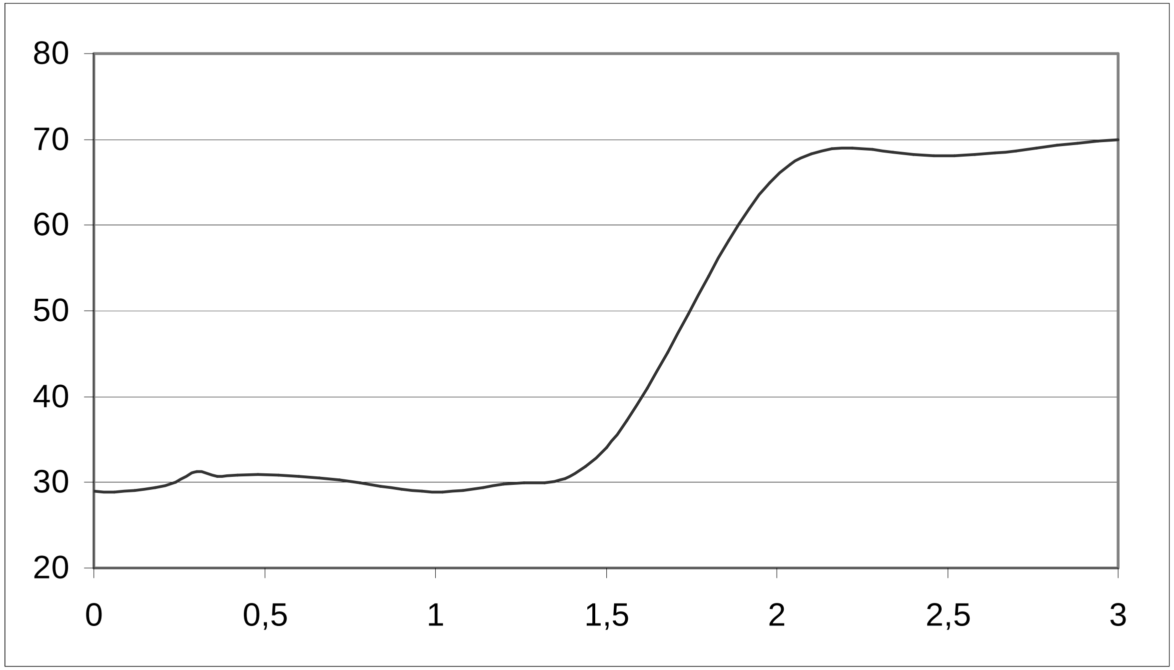
$$\rho = 0.9$$

Class 1   ρ = 0.9



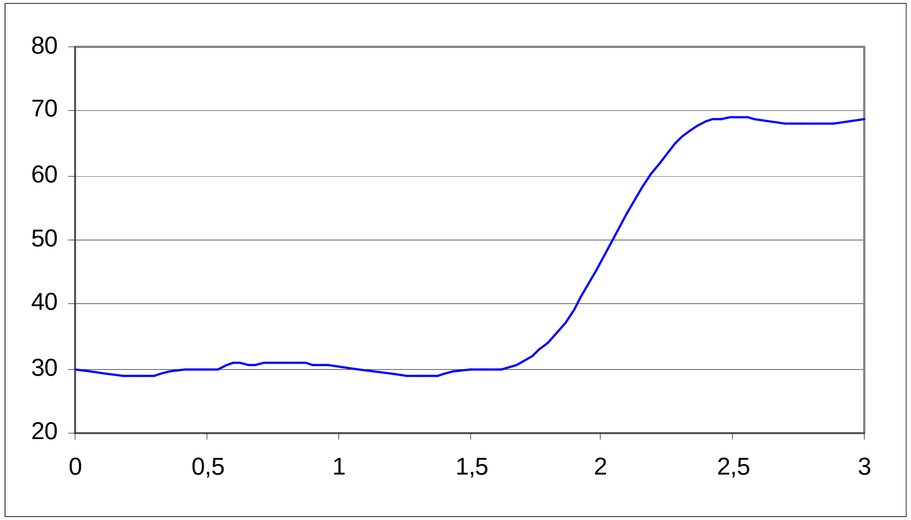
Normal torque influenced by a slight drag in bearings

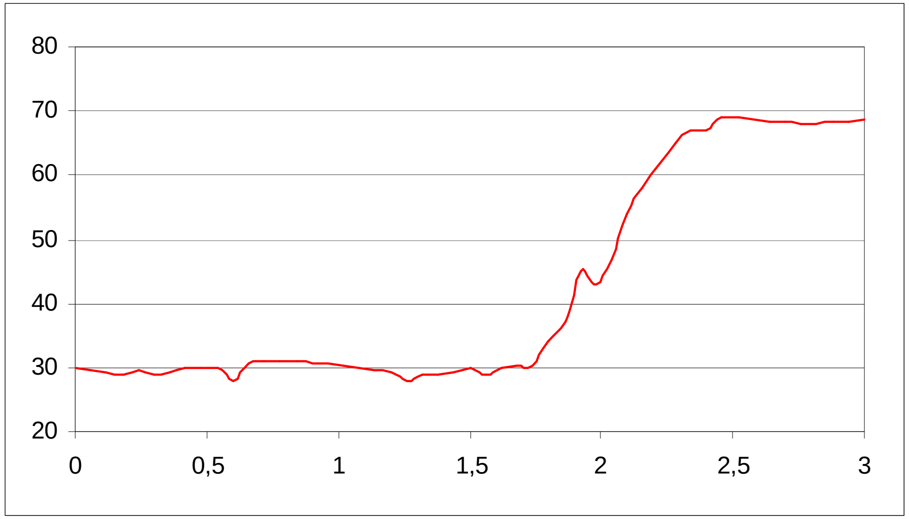
Class 2 $\rho = 0.9$



Normal torque delayed by 0.24 s

Class 3   ρ= 0.9

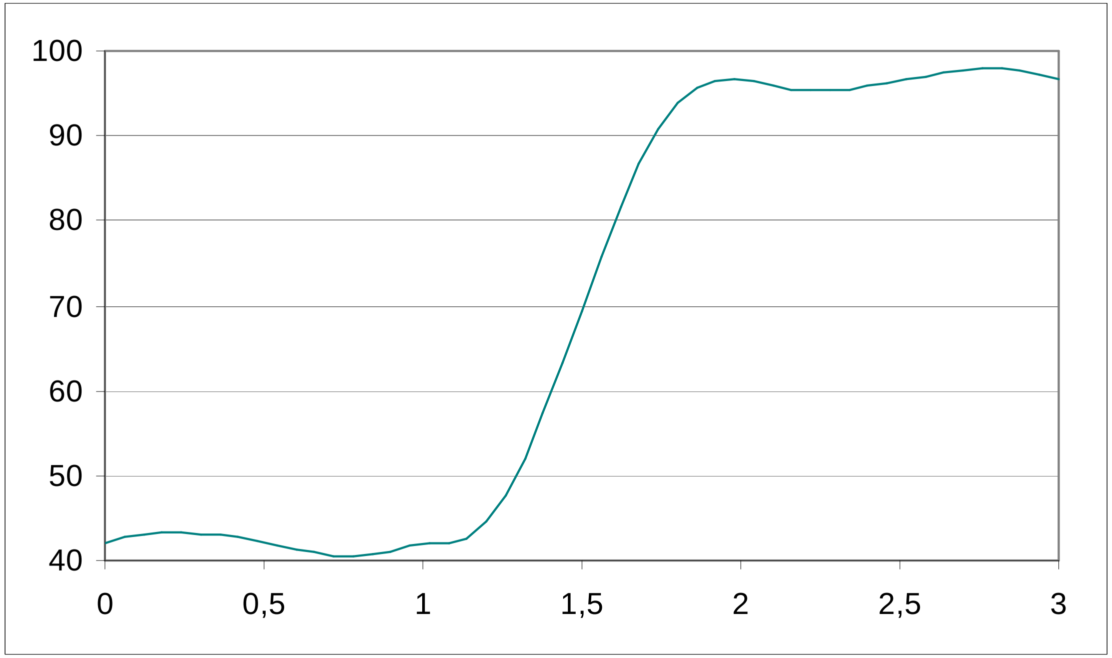




Torque delayed by 0.54 s

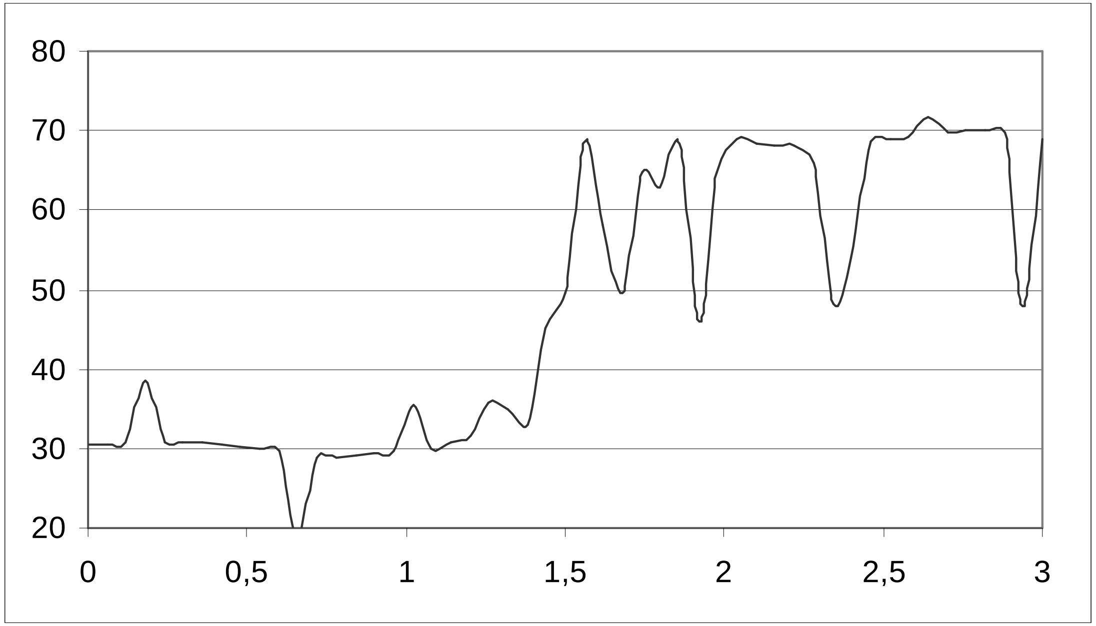
Normal torque influenced by dragging bearings and delayed by 0.54s

Class 4 $\rho = 0.9$



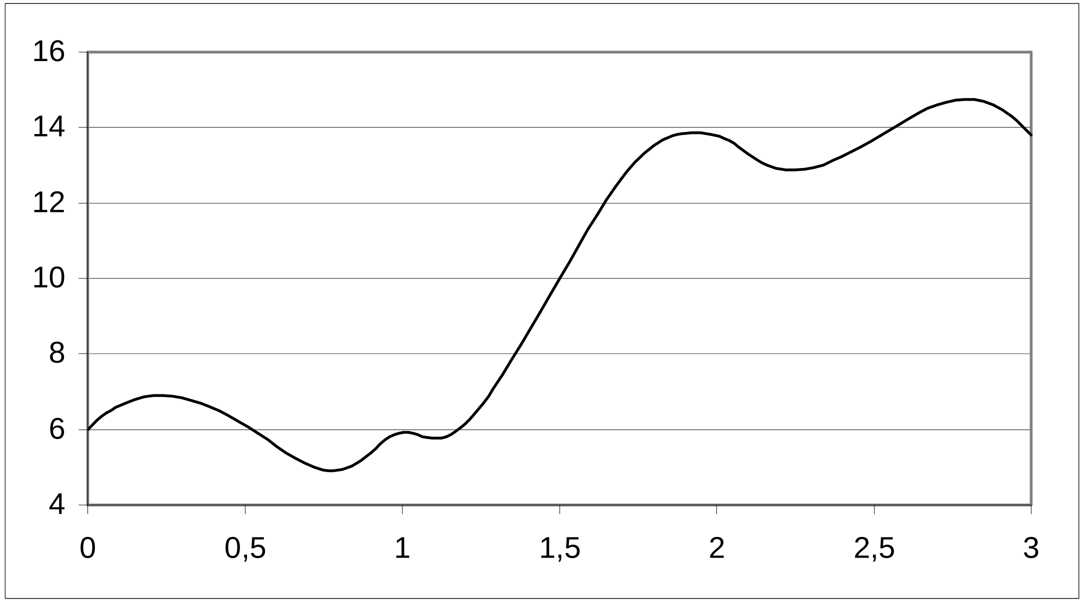
Increased torque caused by heavy loading

Class 5 ρ = 0.9



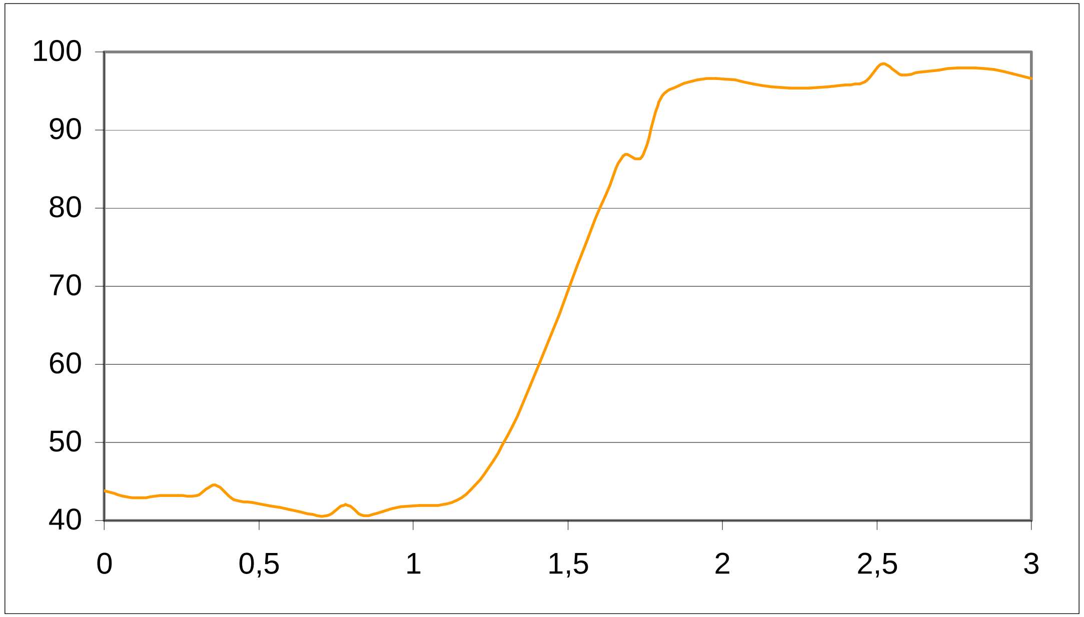
Normal torque with strong irregulations due to dragging and / or slipping clutch

Class 6 $\rho = 0.9$



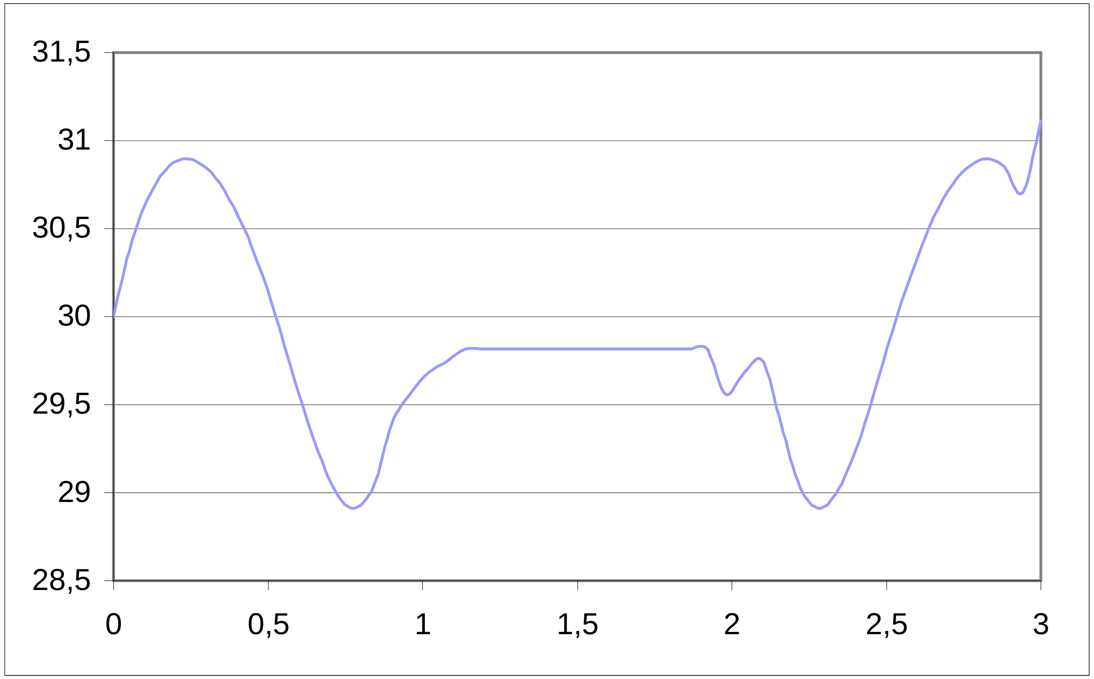
Torque course if a foot sinking into a soft ground (joint actuator is almost unloaded)

Class 7 $\rho = 0.9$

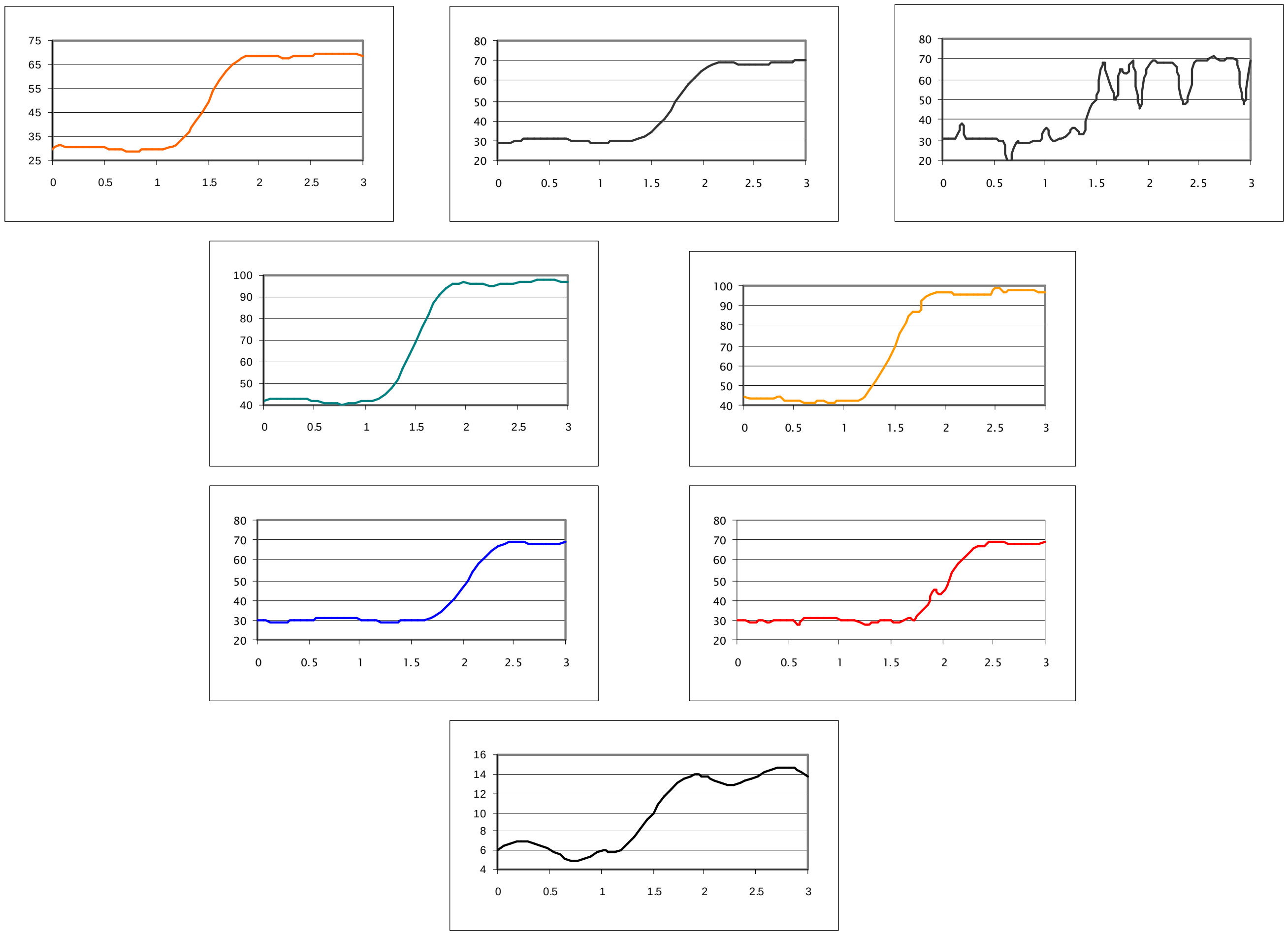


Increased and noisy torque due to the fiction in bearing; the clutch is slightly dragging.

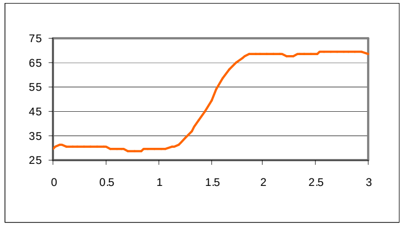
Class 8 $\rho = 0.9$

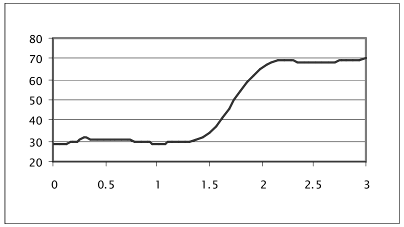


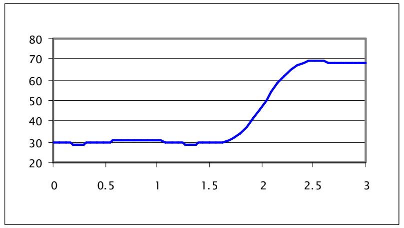
The foot was raised and laid back to the ground without rotation of the coxa joint what caused a plateau in the middle part

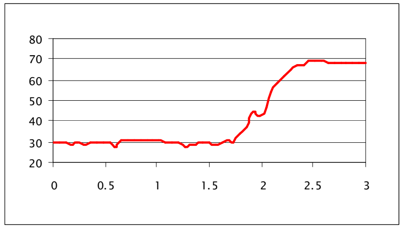


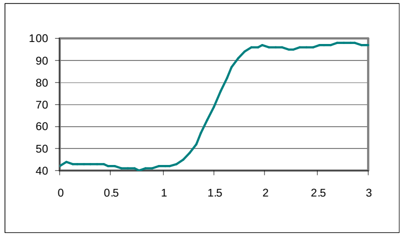
Results of classification for vigilance $\rho = 0.8$

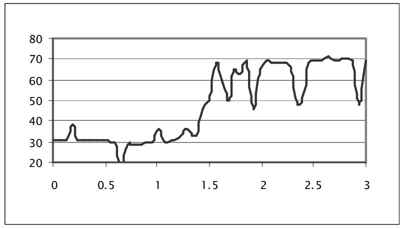


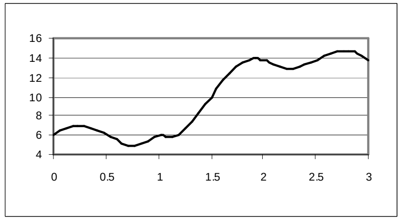


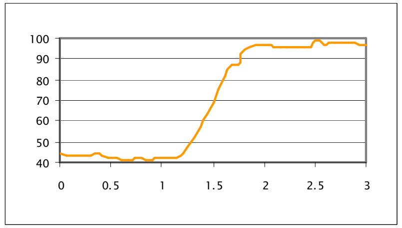


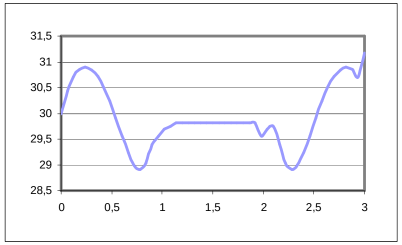












Classification of the same patterns for vigilance $\rho = 0.9$

THANK YOU
FOR YOUR ATTENTION