Distributed sensory devices with features extraction capability communicating with a recognition neural server using spikes adaptive protocol on local area network

Luca Marchese

Syn@ptics

www.synaptics.org

Abstract

In this feasibility study, neural code is evaluated as reliable language to build a protocol enabling distributed sensory devices to transmit simultaneously data to centralized data collectors. An application proposal is used to have a more practical view of the system.

The main result of this study is the realization of a system that can de-multiplex superimposed spikes trains coming from different synchronized sensors. This system can recognize from which sensors data are coming and identify the data related to the single sensor. Given the vector based nature of this communication language , a proposal for an adaptive protocol is suggested.

Distributed sensory devices with features extraction capability communicating with a recognition neural server using spikes adaptive protocol on local area network

Identifying the problem

The target of this feasibility study is a distributed system for monitoring environment with CCD cameras that can communicate vision data to a recognition server. The server is based on RBF neural VLSI (1) with hierarchical recognition capability in different tasks. The server should know identity and geographical position of any sensor and analyze incoming data in such a hierarchical recognition process in order to identify danger situations.

Two main problems related to the global system can be evaluated:

- how much intelligence in the distributed sensory devices ?
- how much network band is required and how this requirement can be minimized ?

Advantages of NST

Neural Server Technology born from the integration of the neural VLSI fast recognition capability and the actual available technology for client-server systems, networking and distributed computing. The basic concept is that distributed clients can collect data using local sensory devices and request recognition services to a central neural server. Normally clients should have inside policy to manage recognition results and possibly command local actuators.

The advantages are:

- low cost clients without expensive recognition capability

- low cost centralized maintenance of recognition knowledge

In this study the scenario is a bit different because we want to have sensory devices without intelligence that simply must collect data and send them to a server that has the functionality of recognition and policy management of it. The server, in this case, must recognize data knowing from which sensors are incoming and then decide actions to perform.

Requirements of the system

-The sensory devices should be sufficiently small and not expensive .

-The sensory devices should be not task-related (general purpose).

-The network connection should have low cost implementation and maintenance.

-The sensory devices must transmit data simultaneously without priorities.

-Transmission speed must be unaffected by removing or adding (up to a maximum number) sensory devices

Single wire or wireless network for reliable communication

I have considered the use of radio transceivers with low bandwidth but large field action (~300m) reducing data size in the sensory devices using one or two neural chips to perform features extraction and spikes encoding.

Spikes trains are associated to features and contain a higher order lower size information. In order to evaluate the bandwidth requirement it is needed to understand the spikes trains encoding process and the related entropy that are analyzed in the next paragraph.

No special radio protocols for multiple transmitting devices should be used to carry spikes trains information, because spikes could be transmitted simultaneously: the only requirement should be the synchronization of transmission between sensors that could be made using a synchronization signal generated by the server. This is at the moment a feasibility study and then some tests have been made only connecting sensors with a single copper wire, but problems that can born in a radio local area network have not been yet evaluated. A single wire implementation could be used for automotive applications.

Radial Basis Function neural chip for local features extraction and spikes trains encoding

We need to have sensory devices with features extraction capability that can reduce the data size increasing the order (fig.1). In this study we have CCD cameras as sensors and we must generate spikes trains to transmit information on image features. The possible methods for spikes train encoding evaluated have been reconstruction starting from data with linear filters or associations pattern to spikes train. In the particular task that we are analyzing the second solution looks to fit very well. The image from camera in m*n pixels format is scanned in (m/k)*(n/k) elementary squares of size k, using predefined prototypes learned by a neural RBF chip. The recognition or categorization of the squares is the first step of feature extractions and data reduction but the association of the features (categories for neural network chip) with spikes trains completes the work.

Having :

ρ = number of transmitting sensors

ϕ = number of features of the image

The required entropy of one spike train is :

ε = log{b2} ($\rho * \phi$) [bit] **[1]**

So if **ϕ = 5** and **ρ = 100** then **ε = log{b2} 500 = 9 [bit]**

But we need to transmit simultaneously data from distributed sensory devices and we don't have a data-fusion point: any sensory device is directly connected to the network and the trivial association spike = bit in the global information cannot be useful.

Considering the characteristics of the neural processor, we need to verify how we can encode sensor spikes trains composed of up to 64 spikes (the maximum dimension of the vector recognizable inside the neural processor) and 255 positions of timing resolution between spikes (byte wide vectors components size), that can be multiplexed (really overlapped on the same channel) and then de-multiplexed.

Having:

ν = number of relative positions of any spike

μ = number of spikes in the train

The entropy of one spike train is:

ε = log{b2} (νμ) [bit]** **[2]**

The entropy of one spikes train using only 3 spikes in this system is:

`ε = log{b2}(255**3)  ~= log{b2} () > 23 [bit]`

The superposition of spikes trains of different sources and the de-multiplexing process can behave correctly considering the following constraints:

- 1) the spikes trains from sensors must be synchronized
- 2) by 1) the duration of the spikes trains must be fixed
- 3) the subspace (in the ensemble of possible spikes trains) containing the fail conditions (fig.2) must be removed.

The space where the fail condition doesn't exist can be formalized as follow:

α = sensor

β = feature

K= absolute spike timing

δ = position (1>64)

ς = vector

Π& = every (in AND condition)

Π| = every (in OR condition)

!= = different

= = equal

χ = fuzzy different (LSUP distance => NOT falling in the same NAIF)

⌉ = fuzzy equal (LSUP distance => falling in the same NAIF)

m>n = from m to n

NAIF = Neuron Active Influence Field

Hypothesis: $\mathrm{K}\,[\Pi|\,\delta]\,[\Pi|\,\alpha' \mathrel{!=} \alpha]\,[\Pi|\,\beta]\ \rceil\ \mathrm{K}\,[\delta]\,[\alpha]\,[\beta]$ **(fail conditions) [3]**

Thesis: $\varsigma\,[\alpha]\,[\beta]\,(0>\delta)\ \ \chi\ \ \varsigma\,[\alpha]\,[\Pi\&\ \beta' \mathrel{!=} \beta]\,(0>\delta)$ **[4]**

(space where fail conditions don't exist)

We need to find an algorithm generating timing vectors satisfying the condition [3] and L1 distance > NAIF.

Ensembles of timing vectors generated by this algorithm should be assigned to sensory devices for the representation of image features.

The first step is the generation of timing vectors considering only L1 distance (fig.7).

The second step is the creation of vectors composed of absolute timing values as follow:

$$C[m] = \Sigma(k = 0 > m-1)\ C[k] \qquad [5]$$

Where C[m] is the m component value of the vector C

The third step is finding equal absolute values of time inside vectors owned by different sensors and mark them. This task require an high number of comparisons between vector components in different sensors. The components are marked when is verified the condition:

$$|C[s][m] - C[s'][n]| < NAIF \qquad [6]$$

The number of comparisons that must be performed is calculated as follow:

$$N = ((K * J)^{**}2) * (S * (S - 1) / 2) \qquad [7]$$

K = number of component in one vector

J = number of vectors in one sensors

S = number of sensors

The fourth task is the verification of LSUP distance between marked vectors and all the others owned by the same sensor, considering the vector terminated at the marked component.

For any LSUP distance D must be:

D = max (V[i] -V'[i]) (i = 0 => i = ç) > NAIF

Where ç = marked component index

This process could be very long in the third step that cant be executed in the neural processor.

It is possible to perform a less optimized algorithm completely executable in the neural chip. This algorithm (fig.7) generates random vectors (the first is ever accepted) verifying the L1 distance with any previously accepted vector. The accepted vectors are compared in LSUP mode with all the previously accepted vectors related to the same sensor: this task is performed in incremental mode on the length of the vector stopping when the LSUP distance is greater than the NAIF or when the dimension of the vector divided by a tuning factor is reached. More near the first element of the vector the LSUP distance is greater than the NAIF, more relaible is the use of the vector because the probability of a fail condition is strongly reduced.

Spikes train de-multiplexing and decoding using fuzzy states machines implemented on radial basis function neural chip

The spikes trains can be transmitted by sensory devices. Overlapped spikes trains generated by different local sensors must be de-multiplexed in order to build spikes trains related to any singular sensor. This process is based on an incremental fuzzy state machines that perform the recognition of any transaction deciding if it is a valid transaction for the sensor at which the machine is related: this means that one state machine must be implemented for any sensor (fig.3). The state is represented by the sequence of accepted spikes timing while the event is the last incoming spike. Relative timings of incoming spikes are collected and then the resulting vector(growing in size up to M components = M spikes in a train) is compared with the possible paths associated with the sensors, using a LSUP(fig.4) distance. The LSUP distance behavior makes that the acceptance / not-acceptance of the state transaction is depending only by the last event (or last spike timing), so it can be evaluated in a fuzzy manner without the influence of the previous accepted transactions. An example of de-multiplexing is shown in fig.5. Any state machine related to a specific sensor completes all the transactions and returns to the start point when the M^{th} event is arrived or when the maximum time for the completion of a spikes-train is reached. At this time the system has built N vectors describing spikes-trains for the N sensors and the de-multiplexing work is completed. The successive step is decoding this spikes-train for any sensor extracting the associated information. This step can be made using again the RBF network that has previously learned the exact vectors describing the trains (fig.6). The recognition is now performed in L1(fig.4) mode so that the fuzziness of any spike timing is now evaluated in the total vector and a perfect associated timing vector describing the spikes-train can be recalled. Not only the imprecision in spike-timing can be managed, but also the possible loss of one spike reception can be corrected by this behavior: while the states machine blocks at the first loss of spike (because the successive cant be acceptable), the recognition based on L1 pseudo-Euclidian distance calculation can recall the completed original vector. If the spike loss position is near the vector end, then the complete vector can be identified while if it is more near the vector start an uncertainty situation (2 or more complete vectors can be recalled by the broken vector) can be reached: in this case the choice of the complete vector should be supported by the Euclidian distance and (if available on chip [1])) by probabilistic information associated with prototypes. The pattern completion process can behave correctly because the vector identification is performed on a vector size equal to the number of received spikes and not fixed to M. So, in the process described, the spikes trains related to single sensors are recognized and classified by the RBF neural chip(s) and the associated image features can be recalled.

Adaptive protocols for sensory devices

The interesting aspect of this communication based on neural code is the capability to transmit information from different sources, really in parallel mode, on the same channel without protocols dedicated to test it's busy condition before any transmission. Though an other interesting aspect is that this interpretation of neural code is based on encoding information on vectors and then adaptive protocols can be evaluated. An adaptive protocol is, in this context, a system of rules that enables sensory devices to learn a language (based on spikes trains) that is not in conflict with other sensory devices during parallel transmission. In this environment, timing vectors of sensors should be initialized randomly. At the first start, the server detects many fail conditions and then instead of transmit the synchronization signal (a reserved spikes train), it sends an other spikes based message to request sensors protocol adjustment. Sensors then start to count up to a random number: the first ready start to transmit his pattern with spikes train, while the other sensors listen. Any listening sensor then adjusts his timing vectors in order to have not conflict with the transmitting sensor. The sensors that don't need to adjust their timing vectors will not try to transmit after the first time. The process is repeated until no more sensors are transmitting. During this period, the server listens the spikes trains coming from single sensors in order to build the new database of timing vectors. While during normal behavior the server can recognize the sensors by the timing vectors as well as the single information , in this phase of “protocol adaptation” a sensor identification signal must be used as header of the trains sequences: in this manner the server can associate spikes trains to sensors. The identification signal by sensors is not linked to the adaptive protocol because used only in a “single device transmitting mode”, but should be based again on spikes here more trivially associated to bits of a device address settable on the device itself.

At this point the server wait a fixed time of silent and then send again a synchronization signal. Sensors start to transmit again in a parallel manner. The server will decide after if the new protocol is correct (and it will send again a synchronization signal) or not (and then it will send again a signal for protocol adjustment) (fig.8/9). The spikes trains used for synchronization (SYNCH) and for protocol adaptation (ADAP) are “reserved” and must be included in the comparing algorithms.

At the moment adaptive protocols based on spikes is only a proposal and none testing or more detailed feasibility studies have been made yet.

References

Fred Rieke, David Warland, Rob de Ruyter van Steveninck, William Bialek (1999) SPIKES - Exploring the neural code - MIT Press.

Compnent Development Laboratory IBM France, Silicon Recognition (1998) ZISC036 User Manual.(http://www.fr.ibm.com/france/cdlab/zisc.htm)

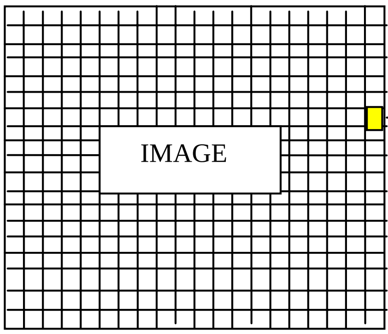


Fig.1

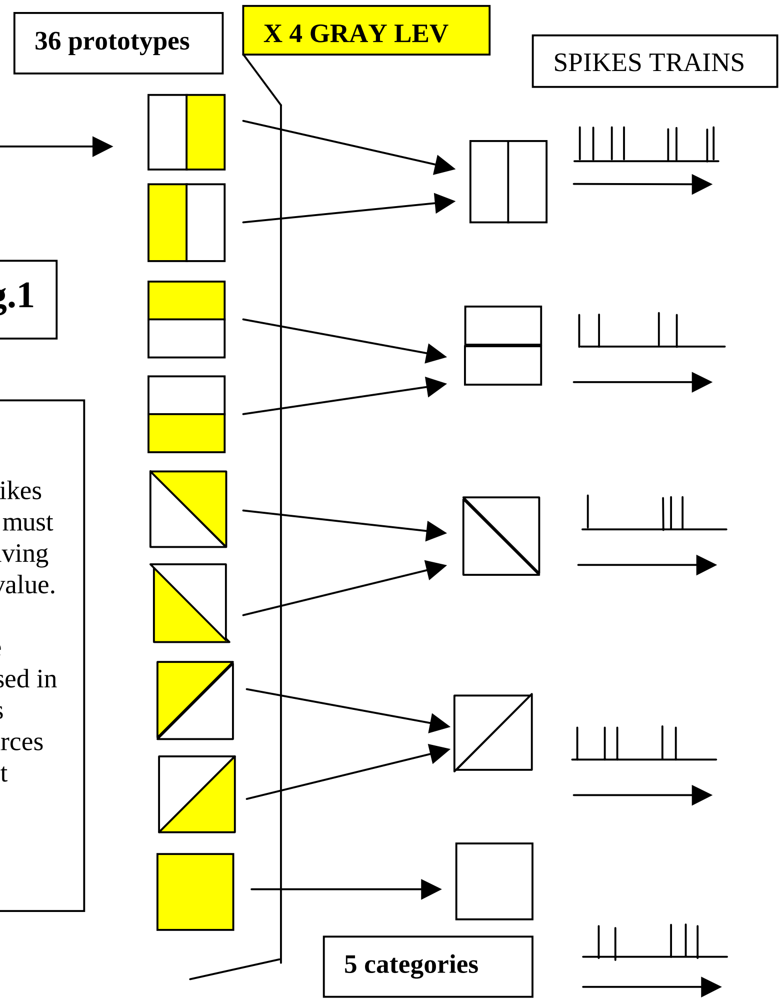
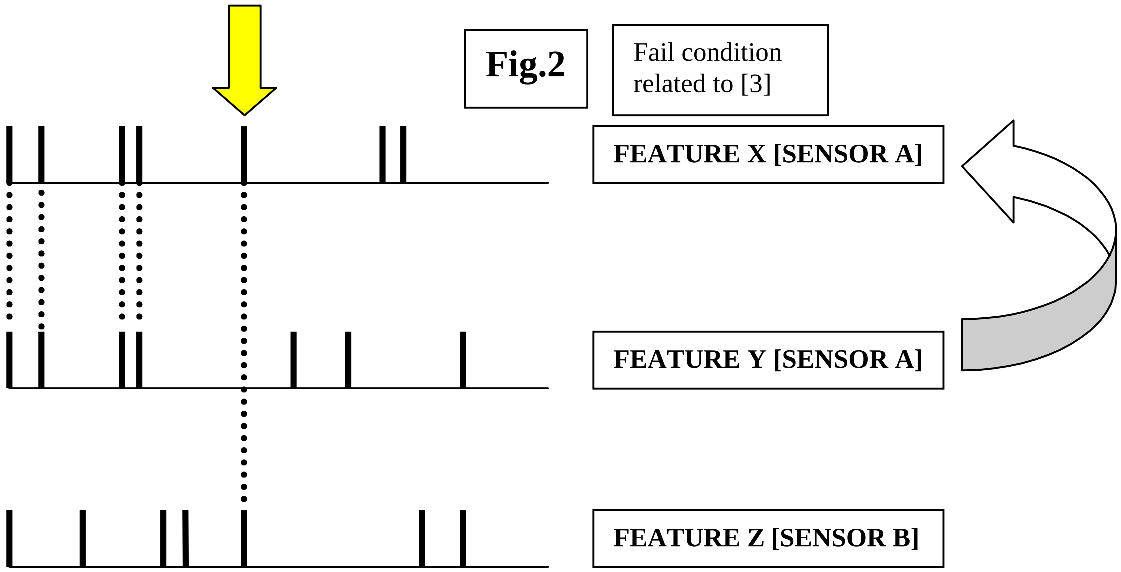


Image contours extraction using
Pre-defined patterns as prototypes.
Any category (5) is associated with a single spikes train: all the spikes train must be different and must be characterized by timing intervals vectors having Euclidean distances grater than a pre-defined value. Any source must have its own spikes train to transmit a feature and the position used by one spike in one feature of one source cannot be used in the spikes trains owned by an other source(this constraint could b optimized). Having 100 sources we need to have 500 different spikes trains that means to have an entropy:
E = log(b2) 500 = 9 bit

Fig.2


Fail condition
related to [3]
FEATURE X [SENSOR A]
FEATURE Y [SENSOR A]
FEATURE Z [SENSOR B]

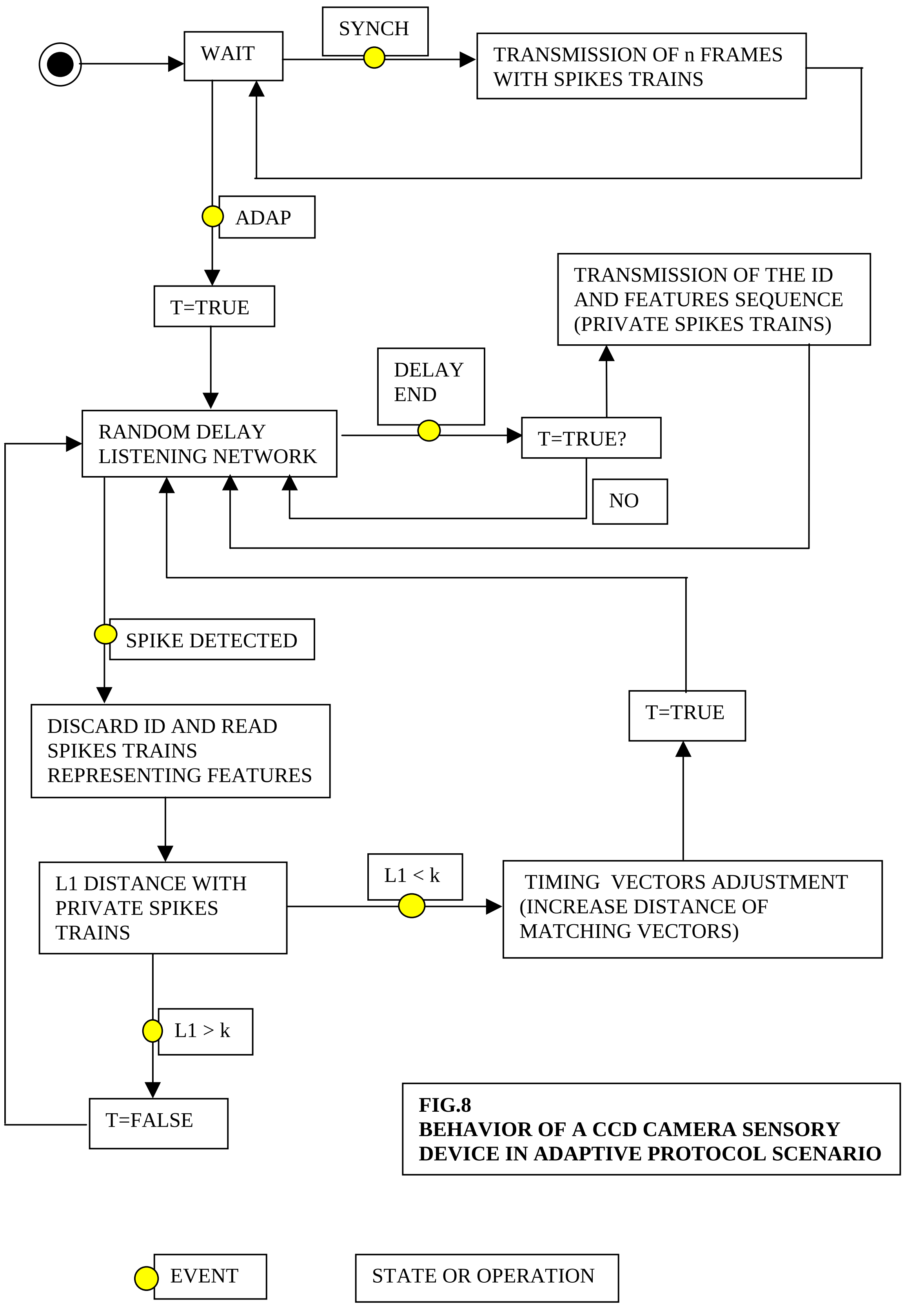
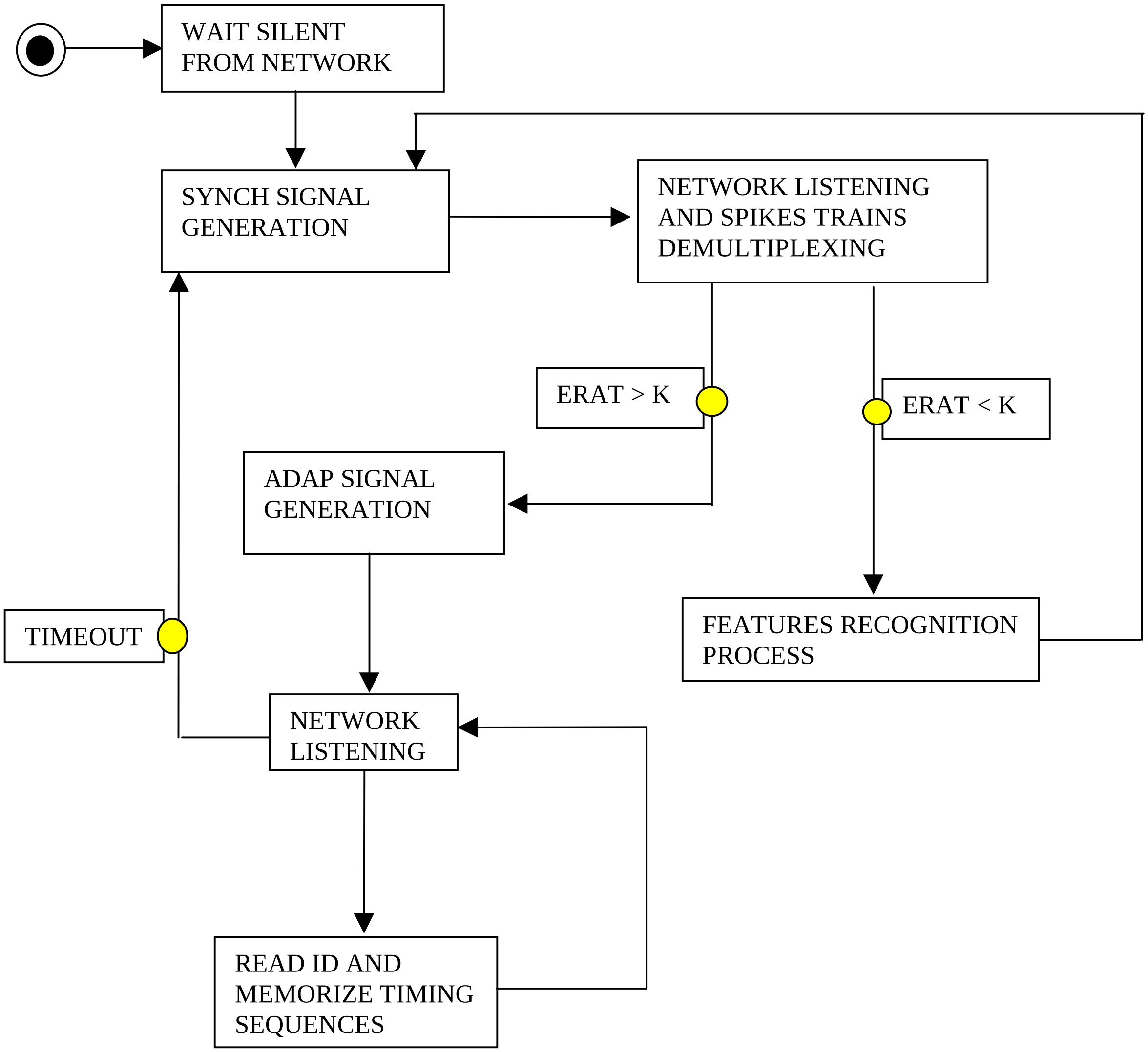


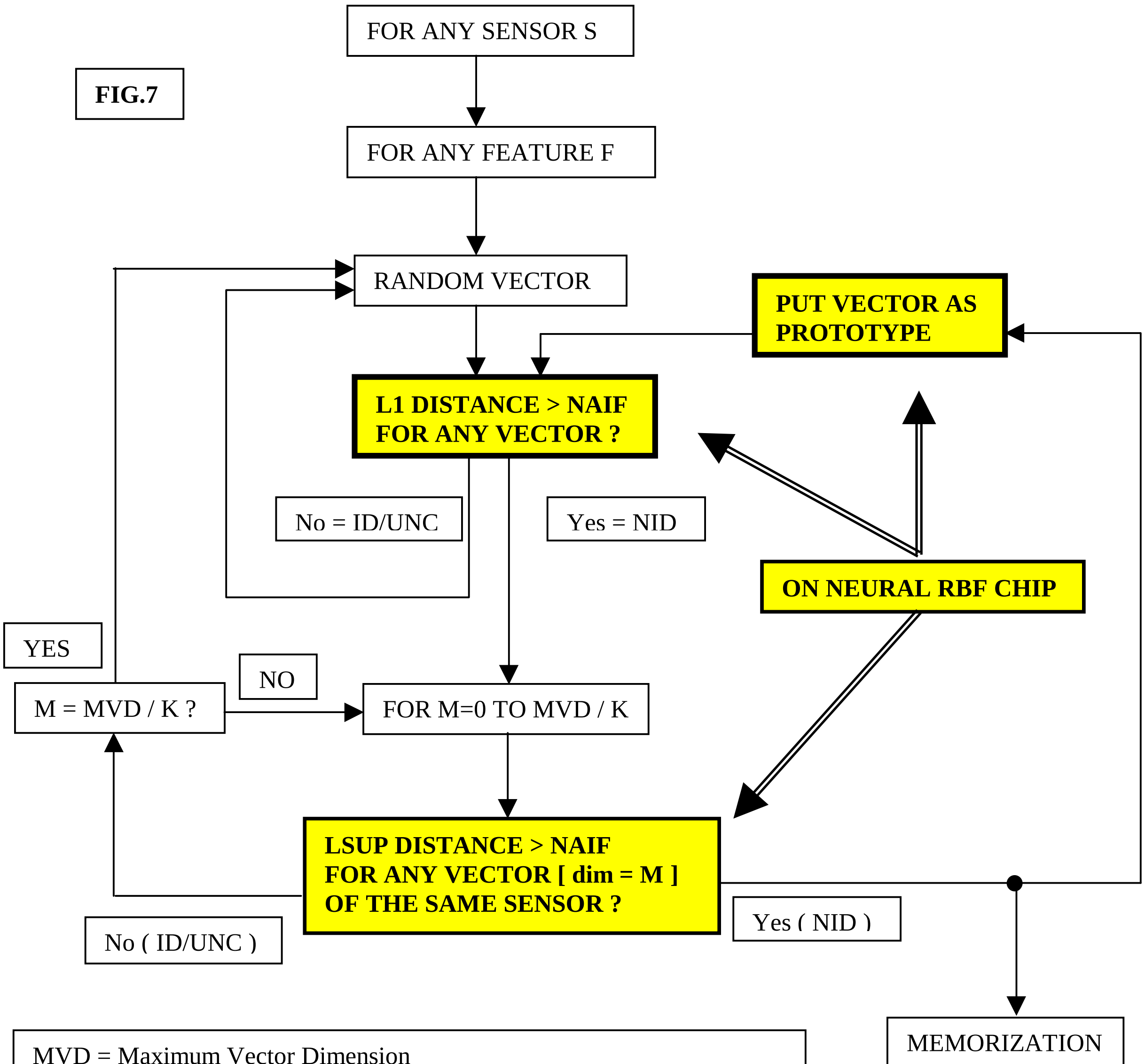
FIG.8
BEHAVIOR OF A CCD CAMERA SENSORY DEVICE IN ADAPTIVE PROTOCOL SCENARIO



ERAT = ERROR RATE

FIG.9
SERVER SIDE BEHAVIOR IN ADAPTIVE PROTOCOL SCENARIO

FIG.7



MVD = Maximum Vector Dimension
1 < K < MVD is the tuning factor for the process
If K = MVD => probability of fail condition = 0 but the space of possible vectors is strongly reduced

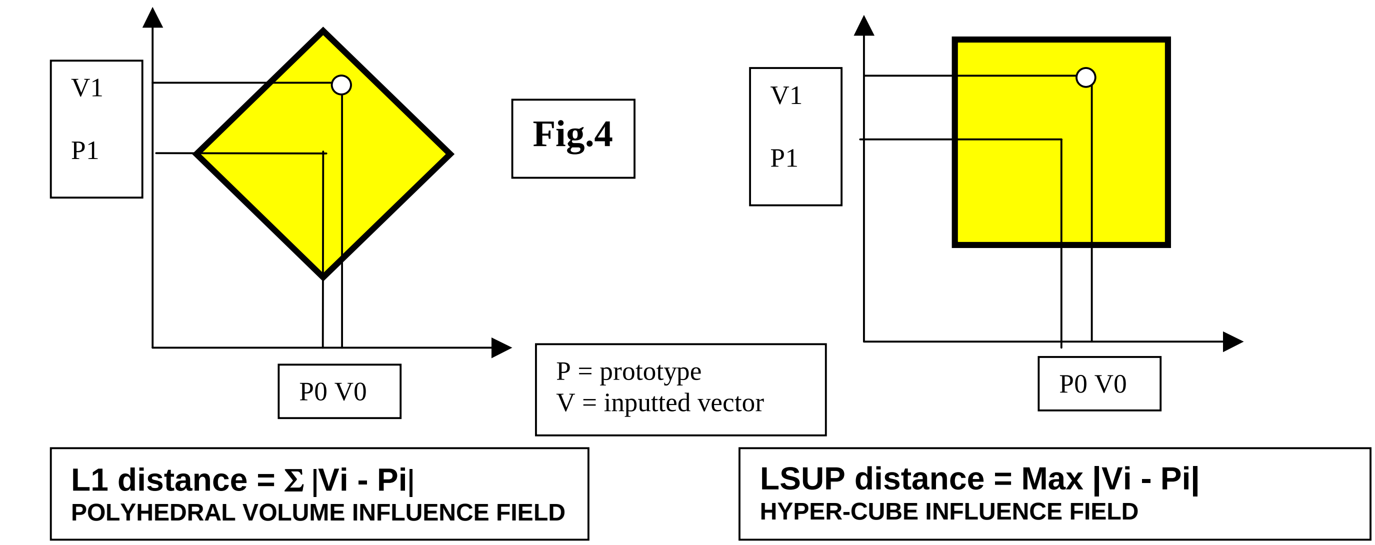

V1
P1
P0 V0
P = prototype
V = inputted vector
V1
P1
P0 V0
L1 distance = Σ |Vi - Pi|
POLYHEDRAL VOLUME INFLUENCE FIELD
LSUP distance = Max |Vi - Pi|
HYPER-CUBE INFLUENCE FIELD

Fig.4

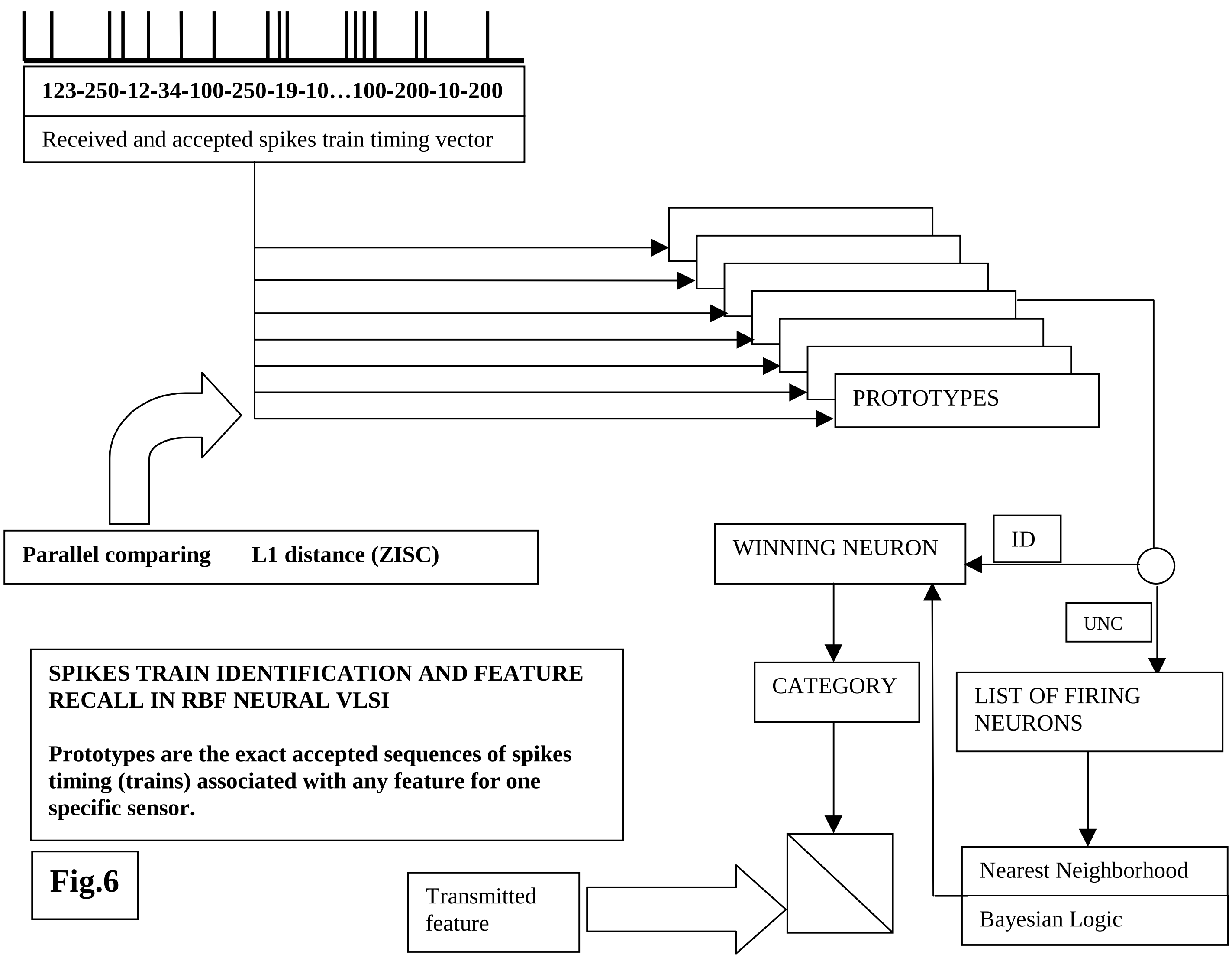

123-250-12-34-100-250-19-10…100-200-10-200
Received and accepted spikes train timing vector
PROTOTYPES
Parallel comparing L1 distance (ZISC)
WINNING NEURON
ID
UNC
CATEGORY
LIST OF FIRING NEURONS
SPIKES TRAIN IDENTIFICATION AND FEATURE RECALL IN RBF NEURAL VLSI
Prototypes are the exact accepted sequences of spikes timing (trains) associated with any feature for one specific sensor.
Transmitted feature
Nearest Neighborhood
Bayesian Logic

Fig.6

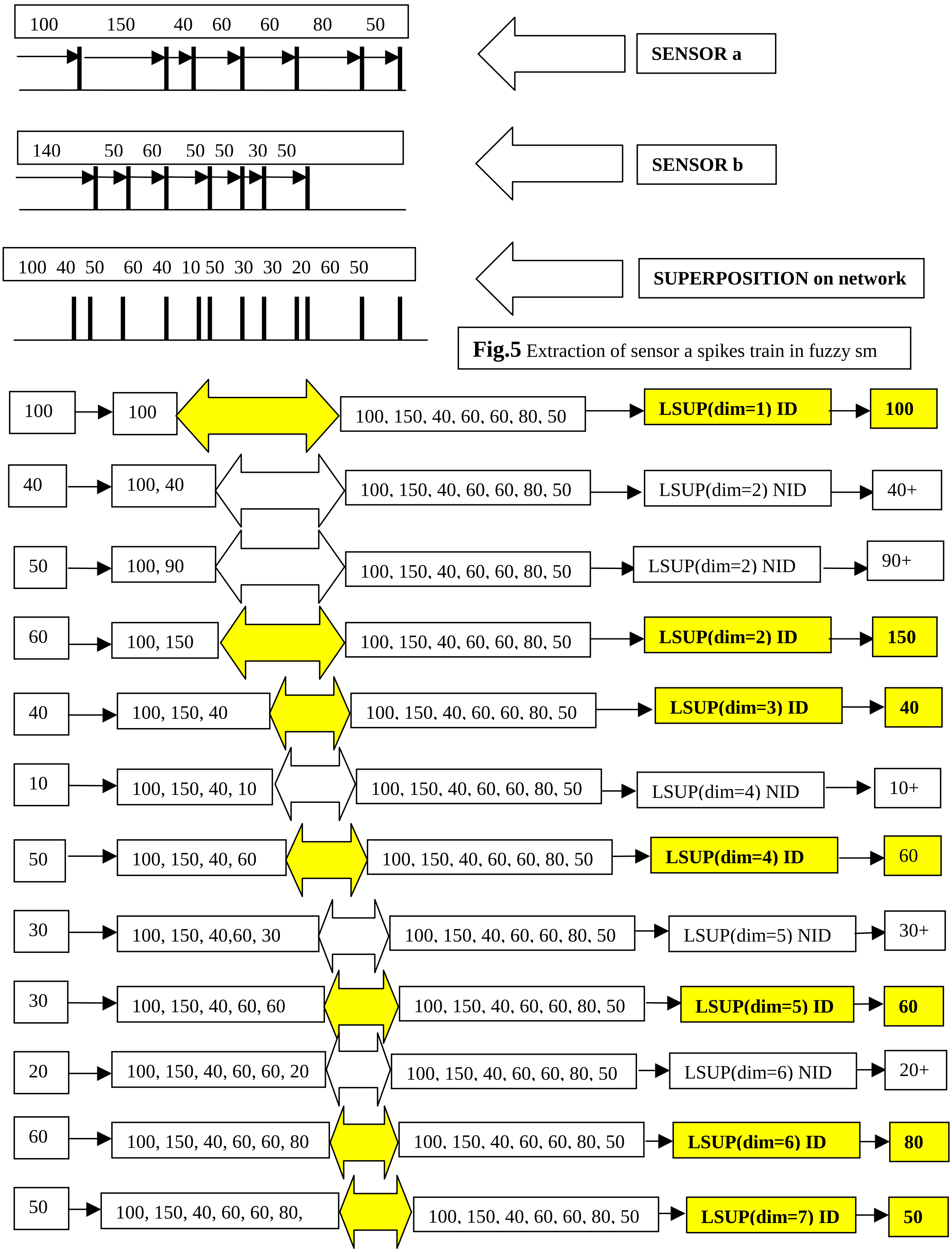
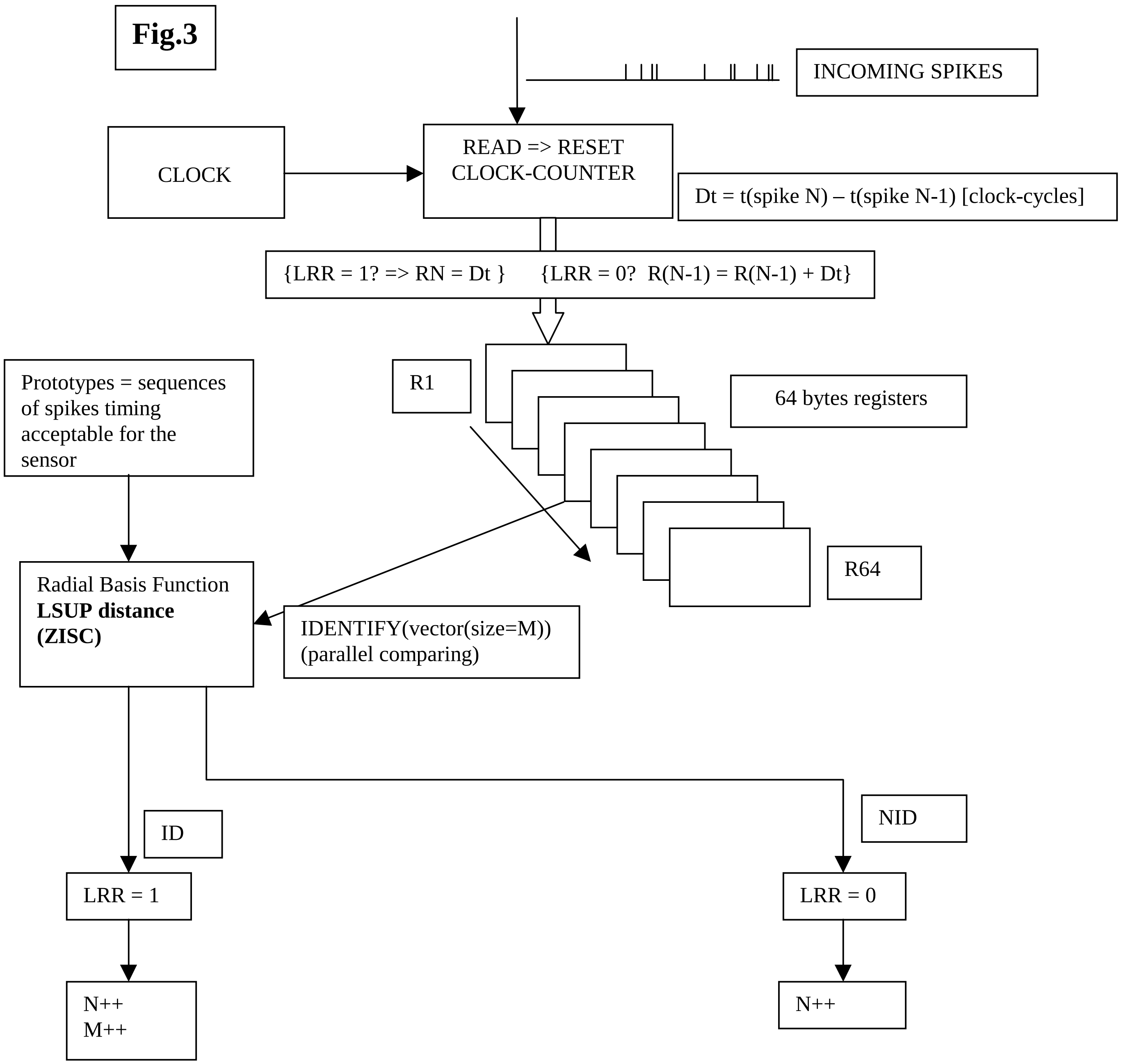


Fig.5 Extraction of sensor a spikes train in fuzzy sm



Behavior of the states machine related to one single sensor contained in the server. One of this Sm must be implemented in the receiver server for any transmitting sensor. The first spike must be a SYNCH signal that is synchronized for all the transmissions of the sensors and behaves as reference point.
The counter counts the clock cycles between spikes transmitting the value and self-resetting at any occurrence of spike. Depending on the value of the register LRR (Last Recognition Register) the timing value is putted in the next byte register(in the array of 64 byte registers) or is added to the last written. If the last previous spike timing has not been accepted the LRR is set to 0, so any successive spike timing will be added up to the occurrence of a spike whose timing addition makes an acceptable timing for the Sm. In this manner the spikes that arrive with timing not acceptable are discarded but are used only to build the time interval from the last accepted one. The recognition of the timing vector is performed by a Radial Basis Function neural network chip that must be set to evaluate the maximum distance for any component (max {v[n] – v'[n])} (In ZISC is L1)). The maximum distance value remove the influence of the possible imprecision related to the previous spikes timing, so the fuzziness of the spike timing admitted is evaluated only on the last incoming spike that is processed.
The recognition of the timing vector is performed on the dimension of the accepted spikes + 1 (M): the +1 is related to the spike under recognition. If the timing vector is recognized (ID) the LRR is set to 1 and both N (total number of spikes) and M(number of accepted spikes) are incremented. If the timing vector is not recognized (NID) the LRR is set to 0 and only the N is incremented .