

# Supervised dFasArt: A Neuro-fuzzy Dynamic Architecture for Maneuver Detection in Road Vehicle Collision Avoidance Support Systems

Rafael Toledo<sup>1</sup>, Miguel Pinzolas<sup>2</sup>, and Jose Manuel Cano-Izquierdo<sup>2</sup>

<sup>1</sup> Dept. of Electronics, Computer Technology and Projects

<sup>2</sup> Dept. of Systems Engineering and Automation

Technical University of Cartagena

{Rafael.Toledo,Miguel.Pinzolas,JoseM.Cano}@upct.es

**Abstract.** A supervised version of dFasArt, a neuronal architecture based method that employs dynamic activation functions determined by fuzzy sets is used for solving support of the problem of inter-vehicles collisions in roads. The dynamic character of dFasArt minimizes problems caused by noise in the sensors and provides stability on the predicted maneuvers. To test the proposed algorithm, several experiments with real data have been carried out, with good results.

**Keywords:** dFasArt, Collision Avoidance, Maneuver Detection.

## 1 Introduction

The issue of collision avoidance in roads has been addressed from many different points of view in the current literature. An interesting approach is based on the creation and interpretation of a scene of vehicles in a potentially conflictive situation [1] – [7]. Vehicles exchange pose and geographical information, along with its current maneuver states through ad hoc WLAN networks, in order to determine their roles in the scene [8].

However, the problem of defining appropriate dynamics to represent all possible maneuver states of road vehicles is not simple. To recognize maneuver states, several authors employ multiple model filters [9], [10]. The use of multiple models allows more accurate positioning and noise estimation, and the possibility of recognizing different dynamic states of the vehicle [11]. In order to solve the problem of often unrealistic switches between models, several authors propose the use of interactive multiple model (IMM) filtering, in which maneuver states are formulated as Markovian processes. In works such as [10], the use of an IMM based method combining constant velocity (CV) and constant acceleration (CA) models has been found adequate to this problem. Some other authors like [9] and [12] employ similar assumptions to represent different vehicle dynamics. Nevertheless, despite of the improvements achieved, the tuning of the CA filter noise parameters has been found problematic. For example, in case of highways, typical accelerations do not last long enough to accomplish a transition from CV state to CA, while situations in urban scenarios are completely different [13].

In our case, market considerations encourage the use of low cost sensors, at the expense of high noise values in the measurements. Therefore, a method capable to interpret very different vehicle dynamics anytime, avoiding noise inconveniences is advisable.

The dFasArt neural architecture can naturally address these issues. Due to its dynamic quality, dFasArt allows taking into account the time span of input data, without the need of keeping buffers of past input or output values. The dynamic characteristic of the activation functions provides a natural way of filtering noise in inputs, while the dynamic evolution of the reset signals allows stability in the predictions.

In this work, a supervised version of dFasArt architecture has been developed and tested for maneuver detection. In the following Section, a brief description of the neural architecture is given. In Section 3, experimental setup is described and obtained results commented. Finally, Section 4 summarizes our conclusions about the work.

## 2 Supervised dFasArt

FasArt model links the ART architecture with Fuzzy Logic Systems, establishing a relationship between the unit activation function and the membership function of a fuzzy set. On the one hand, this allows interpreting each of the FasArt unit as a fuzzy class defined by the membership-activation function associated to the representing unit. On the other hand, the rules that relate the different classes are determined by the connection weights between the units.

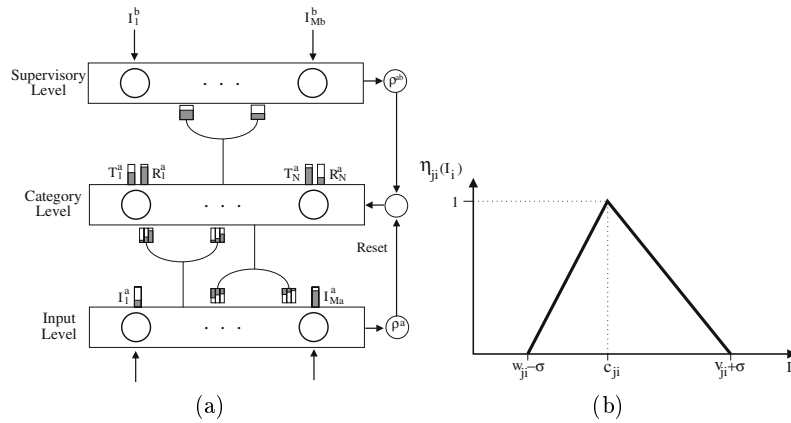


Fig. 1. (a) Supervised dFasArt model architecture. (b) Membership-activation function.

Derived from FasArt, dFasArt uses dynamic activation functions, determined by the weights of the unit. These weights can be regarded as the defining parameters of a fuzzy set membership function [14]. In dFasArt, learning is unsupervised

and incremental. In this work, a supervised version on the dFasArt algorithm has been developed. This modification follows the ARTMAP philosophy, maintaining the maximum generalization-minimum prediction error principle.

The Supervised dFasArt (SdFasArt) architecture is represented in Fig. 1(a). SdFasArt uses a dynamic activation function determined by the weights of the unit as the membership function of a fuzzy set. The signal activation is calculated as the AND of the activations of each one of the dimensions when a multidimensional signal is considered. This AND is implemented using the product as a T-norm. Hence, the activity  $T_j$  of unit  $j$  for a M-dimensional input  $\mathbf{I} = (I_1 \dots I_M)$  is given by:

$$\frac{dT_j}{dt} = -A_T T_j + B_T \prod_{i=1}^M \eta_{ji}(I_i(t))$$

Where  $\eta_{ji}$  is the membership function associated to the *ith-dimension* of unit  $j$ , determined by the weights  $w_{ji}$ ,  $c_{ji}$  and  $v_{ji}$ , as it is shown in Figure 1(b). The  $\sigma$  parameter determines the fuzziness of the class associated to the unit by:

$$\sigma = \sigma^* |2c_j| + \epsilon$$

The election of the winning unit  $J$  is carried out following the winner-takes-all rule:

$$T_J = \max_j \{T_j\}$$

The learning process starts when the winning unit meets a criterion. This criterion is associated to the size of the support of the fuzzy class that would contain the input if this was categorized in the unit. This value is calculated dynamically for each unit according to:

$$\frac{dR_j}{dt} = -A_R R_j + B_R \sum_{i=1}^M \left( \frac{\max(v_{ji}, I_i) - \min(w_{ji}, I_i)}{|2c_{ji}| + \epsilon} \right)$$

The  $R_j$  value represents a measurement of the change needed on the class associated to the  $j$  unit to incorporate the input. To see if the  $J$  winning unit can generalize the input, it is compared with the design parameter  $\rho$ , so that:

– If:

$$R_J \geq \rho$$

the matching between the input and the weight vector of the unit is good, and the learning task starts.

– If:

$$R_J < \rho$$

there is not enough similarity, so the Reset mechanism is fired. This inhibits the activation of unit  $J$ , returning to the election of a new winning unit.

If the Reset mechanism is not fired, then the learning phase is activated and the unit modifies its weights. The *Fast-Commit Slow-Learning* concept is commonly used.

When the winning unit represents a class that had performed some other learning cycle (*committed unit*), the weights are *Slow-Learning* updated according to the equations:

$$\begin{aligned}\frac{d\mathbf{W}}{dt} &= -A_W \mathbf{W} + B_W \min(\mathbf{I}(t), \mathbf{W}) \\ \frac{d\mathbf{C}}{dt} &= A_C (\mathbf{I} - \mathbf{C}) \\ \frac{d\mathbf{V}}{dt} &= -A_V \mathbf{V} + B_V \max(\mathbf{I}(t), \mathbf{V})\end{aligned}$$

For the case of the uncommitted units, the class is initialized with the first categorized value, hence *Fast-Commit*:

$$\mathbf{W}_J^{NEW} = \mathbf{C}_J^{NEW} = \mathbf{V}_J^{NEW} = \mathbf{I}$$

Supervision is carried out in the supervisory level, by means of vector  $\mathbf{I}^b = (I_1^b \dots I_{Mb}^b)$ . In this level, for each time instant,  $\mathbf{I}^a$  activates the corresponding unit. The  $\mathbf{W}_k^{ab}$  matrix of adaptive weights associates, in a many-to-one mapping, units of the category level to units on the supervisory one. When a unit  $J$  is activated for the first time in the category level, weights are adapted by means of a fast-learning process:

$$\mathbf{W}_J^{ab} = \mathbf{I}^b$$

If unit  $J$  is a committed unit, a matching between the membership value to the predicted category and the “crisp” desired value is carried out:

– If:

$$|\mathbf{W}_J^{ab} \wedge \mathbf{I}^b| \geq \rho^{ab} |\mathbf{I}^b|$$

So that prediction corroborates supervision.

– If:

$$|\mathbf{W}_J^{ab} \wedge \mathbf{I}^b| < \rho^{ab} |\mathbf{I}^b|$$

then matching between prediction and supervision is not strong enough. In this case, the Reset signal is fired, and a new prediction is made.

When no supervision is present, SdFasArt will predict as output the value associated to the weight vector of the winning unit, that is,  $\mathbf{W}_J^{ab}$ .

### 3 Experimental Tests

#### 3.1 Experimental Setup

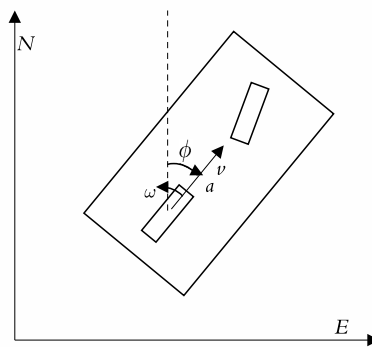
For navigation purposes GPS, an odometry captor, INS sensors and GIS maps are installed aboard the vehicle prototype. The use of inertial sensors, accelerometers and gyroscopes, supplies continuous positioning even in cases without GPS coverage, and high frequency measurements [15]. To integrate data coming from different sensors extended Kalman filters (EKF) run a set of different

kinematical models as detailed in [15]. Thanks to the MEM (Micro-Electro-Mechanical) technology, low cost inertial sensors can be considered, at the expense of higher measurement noises and low level of performance [16]. For vehicle to vehicle (V2V) communications WLAN ad hoc networks are used [8]. The test vehicle deployment consists of wireless LAN IEEE 802.11b and antenna by Cisco, EGNOS capable GPS and DGPS (differential GPS) sensors by Novatel and Trimble, and MEM based IMUs (Inertial Measurement Unit) by Crossbow and Xsens.

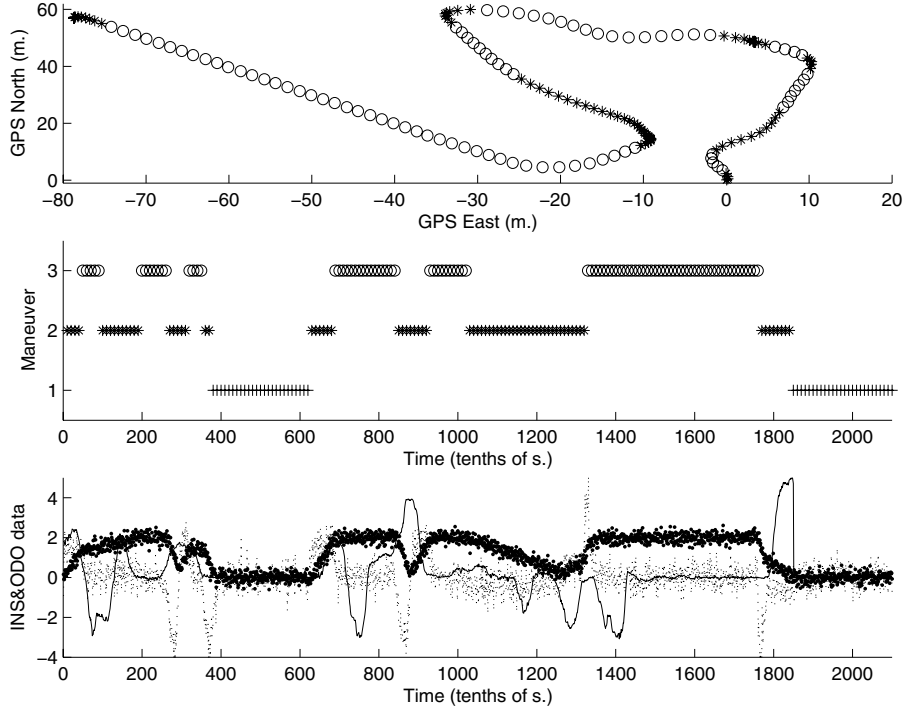
In the measurement collection phase, acceleration and heading turn rate coming from the INS device are gathered  $(a, \omega)$ , while velocity is obtained from the odometry captor installed in the back wheels axis of the vehicle  $(v)$ . Next, sorting and synchronization processes are performed. In this phase, a spurious detection method based on simplified Nyquist inequation is applied [15]. Finally, these data are used as inputs of the supervised dFasArt algorithm. In this work, although GPS North and East measurements are collected by our system and used for positioning purposes, only odometry and inertial measurements are used by the maneuver detector algorithm. Thus, system performance is not determined by GPS signal outages, very often in built-up environments.

### 3.2 Vehicle Models

Maneuver states are typically defined by kinematic vehicle models. In order to test the performance of the proposed SdFasArt algorithm, three different vehicle models are used as reference. The outcomes of the algorithm will be then compared with assumed kinematical maneuver truth. The three kinematical models proposed are based on a simplified bicycle model, in which the orientation of velocity and acceleration are assumed to be equal (Fig. 2). Previous results achieved by the authors proved that convenience of this assumption. Three different maneuver states are distinguished: AD (acceleration and deceleration), CR (cruise) and STA (stationary) states.



**Fig. 2.** Simplified bicycle model of the kinematic behavior of the road vehicle. In this model it is assumed that the orientation of velocity and acceleration are defined by heading angle  $\phi$  and rate  $\omega$ , and referred to East, North  $(E, N)$  coordinates.



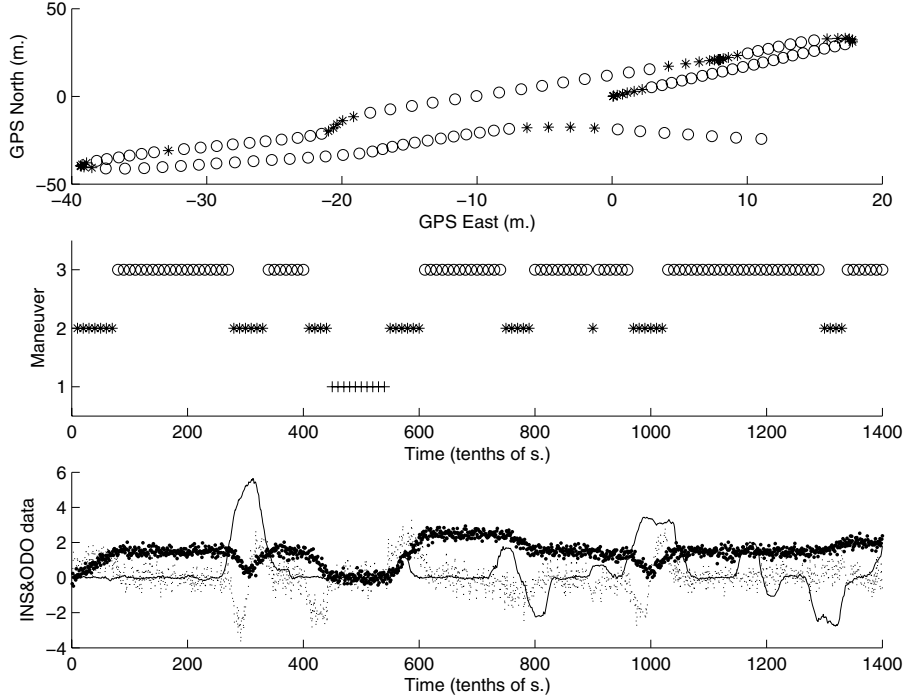
**Fig. 3.** Circuit used for training. From top to bottom: GPS trajectory; desired maneuver classification: AD (\*), CR (o), STA (+); output from the odometry,  $v$  (thick line), and INS,  $a$  (dotted line),  $\omega$  (solid line). Real values of  $a$  and  $\omega$  have been scaled to fit in figure limits.

**Acceleration/Deceleration (AD).** State vector of the acceleration/deceleration model is  $\mathbf{x}_{AD} = (x, y, \phi, v, \omega, a)$ , representing east, north, velocity angle, velocity, yaw rate of turn, and the acceleration, in the center of mass of the vehicle. The dynamics of this model are described by

$$\dot{\mathbf{x}}_{AD} = \begin{bmatrix} (v + at) \cos(\phi) \\ (v + at) \sin(\phi) \\ \omega \\ a \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \eta_{\omega_{AD}} \\ \eta_{a_{AD}} \end{bmatrix} \quad (1)$$

where  $\eta_{\omega_{AD}}$  and  $\eta_{a_{AD}}$  are white noise terms representing the errors due to model assumptions of constant acceleration and constant yaw rate.

**Cruise (CR).** State vector and differential equation of the cruise model are the same as in the AD model (1). As previously commented many authors proposed constant velocity models for cruise maneuver state. However, this has been found



**Fig. 4.** Circuit used for validation. From top to bottom: GPS trajectory; desired maneuver classification: AD (\*), CR (o), STA (+); output from the INS:  $V$  (thick line),  $a$  (dotted line),  $\omega$  (solid line). Real values of  $a$  and  $\omega$  have been scaled to fit in the figure limits.

problematic in several cases. In our approach, both AD and CR maneuver states are defined by similar CA models, defining higher noise values for AD state ( $\eta_{\omega_{AD}}, \eta_{a_{AD}}$ ) in order to fulfill higher dynamics.

**Stationary (S).** In this case, state vector is simplified being  $v = \omega = a = 0$ , and the differential equation

$$\dot{\mathbf{x}}_S = [\eta_{x_S} \ \eta_{y_S} \ \eta_{\phi_S} \ 0 \ 0 \ 0]'$$

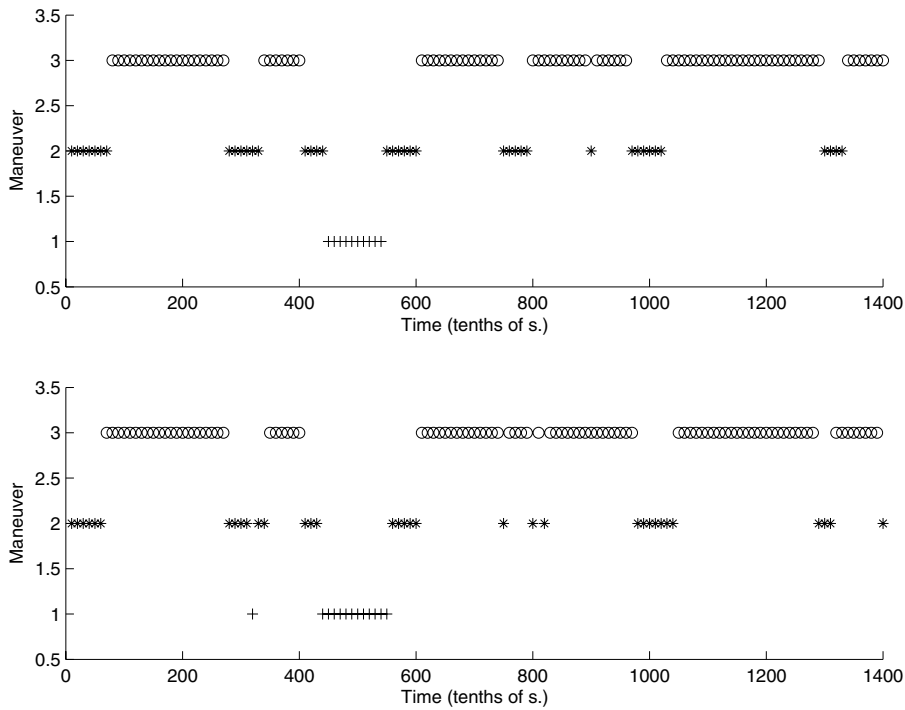
where  $\eta_{x_S}$ ,  $\eta_{y_S}$  and  $\eta_{\phi_S}$  are white noise terms representing the errors due to the model assumptions. Noise parameters of the models are tuned starting from the sensor specifications.

### 3.3 Experimental Results

After several experiments to fine-tune the parameters of SdFasArt, values shown in Table 1 have been chosen.

**Table 1.** Parameters of Supervised dFasArt

Parameter	Value	Description
Aw	0.1	Time constant of weight's dynamic
Av	0.1	Time constant of weight's dynamic
Ac	0.1	Time constant of weight's dynamic
$\delta$	0.2	Minimum fuzziness of the fuzzy categories
$\alpha$	$1e^{-7}$	Activation value for new classes
RESET	0.2	Reset level
Ar	1.21	Time constant of RESET's dynamic
At	0.98	Time constant of activation's dynamic



**Fig. 5.** Results on the validation circuit. Top: desired maneuver classification: AD (\*), CR (o), STA (+); Bottom: predicted maneuver classification: AD (\*), CR (o), STA (+).

Two different circuits have been employed, one for training (shown in Fig. 3) and the other for validation (see Fig. 4). Results obtained for the validation data are shown in Fig. 5. It can be seen that only a few maneuvers have been misclassified. As referred to the assumed truth, 86.36% of the samples are properly classified. However, manual labeling can only be considered as another proper estimate of the maneuver truth, since there are not unique criteria to classify complex dynamics. Thus, some of the errors can be attributed to errors in the



manual labeling for the estimated maneuver truth of the training data. Furthermore, we can appreciate that transitions from STA to CR or from CR to STA maneuver states are never predicted by the algorithm, showing the algorithm prediction consistency.

## 4 Conclusions

In this work, a Supervised version of dFasArt has been proposed and tested for maneuver detection in road vehicles. The combination of the dynamic character of dFasArt with a supervisory module results in a robust classifier, capable to provide stable outputs in spite of noisy time-varying input data.

The neural architecture has been tested using real data gathered from inertial and odometry sensors mounted on a vehicle, showing good performance and consistent results.

**Acknowledgments.** We would like to thank the people of the NEUROCOR Group for their support in this work, and the Spanish Ministerio de Fomento and the European Space Agency (ESA) for sponsoring the research activities under the grants FOM/3929/2005 and GIROADS 332599 respectively.

## References

1. Kokar M, Matheus C, Letkowski J: Association in Level 2 fusion., SPIE (2004).
2. Ceruti M, Ontology for Level-One Sensor Fusion and Knowledge Discovery. SPIE (2004).
3. Matheus C, Mieczyslaw M. Kokar M, Baclawski K,: A Core Ontology for Situation Awareness. Proceedings of Sixth International Conference on Information Fusion, Cairns, Australia, July (2003), pp. 545–552.
4. Mieczyslaw M. Kokar M , Matheus C, Baclawski K, Letkowski J, Hinman M, Salerno J: Use Cases for Ontologies in Information Fusion. Proceedings of Sixth International Conference on Information Fusion, Cairns, Australia, July (2003), pp. 545–552.
5. Ceruti M, Kaina J: Enhancing Dependability of the Battlefield Single Integrated Picture through Metrics for Modeling and Simulation of Time-Critical Scenarios. Proc. of the IEEE 9th Intl. Workshop on Real-time Dependable Systems, (WORDS 2003F), Oct. (2003).
6. Ceruti M, Kamel M: Preprocessing and Integration of Data from Multiple Sources for Knowledge Discovery. International Journal on Artificial Intelligence Tools, vol. 8, no. 3, June (1999), pp. 159-177.
7. Waltz E, Llinas J: Multisensor Data Fusion. Artech House, Boston (1990).
8. Toledo R, Sotomayor C, Gomez-Skarmeta A: Quadrant: An Architecture Design for Intelligent Vehicle Services in Road Scenarios. Monograph on Advances in Transport Systems Telematics (2006) : 451–460.
9. Huang D, Leung H: EM-IMM based land-vehicle navigation with GPS/INS, Proceedings of the IEE ITSC Conference. Washington DC USA. Oct. (2004). pp. 624–629.

10. Hoffmann C, Dang T: Cheap Joint Probabilistic Data Association Filters in an Interacting Multiple Model Design, Proceedings of the 2006 IEEE-MFI 2006. September 3-6, (2006), Heidelberg, Germany. pp. 197–202.
11. Toledo R, Zamora M, Skarmeta A: A Novel Design of a High Integrity Low Cost Navigation Unit for Road Vehicle Applications, Proceedings of the IEEE-IV 2006, Tokyo, Japan, June (2006) pp.577–582.
12. Barrios C, Himberg H, Motai Y, Sadek A: Multiple Model Framework of Adaptive Extended Kalman Filtering for Predicting Vehicle Location, Proceedings of the IEEE-ITSC 2006, Toronto, Canada, September 17-20, (2006). pp. 1053–1059.
13. Kaempchen N, Weiss K, Shaefer M, Dietmayer K: IMM Object Tracking for High Dynamic Driving Maneuvers. Proceedings of the IEEE-IVS'2004 June (2004) pp. 825 – 830.
14. Cano-Izquierdo J, Almonacid M, Ibarrola J, Pinzolas M: Use of Dynamic Neuro Fuzzy Model dFasArt for Identification of Stationary States in Closed-loop Controlled Systems, Proceedings of EUROFUSE 2007 Jaen, Spain (2007) (in press).
15. Toledo R,: A High Integrity Navigation System for Road Vehicles in Unfriendly Environments Phd. Dissertation. Universidad de Murcia Publishing. (2005).
16. Barshan B, Durrant-Whyte H F,: Inertial Navigation Systems for Mobile Robots. IEEE International Transactions on Robotics and Automation. June (1995), Vol. II NO. 3: 328–342.