This document is published in:

García Pedrajas, N. et al. (eds.) (2010) *Trends in Applied Intelligent Systems*: *23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Cordoba, Spain, June 1-4, 2010, Proceedings, Part III*. (Lecture Notes in Computer Science, 6098) Springer, pp. 498- 507.
Doi: http://dx.doi.org/10.1007/978-3-642-13033-5_51

Air Traffic Control: A Local Approach to the Trajectory Segmentation Issue

José Luis Guerrero, Jesús García, and José Manuel Molina

Group of Applied Artificial Intelligence (GIAA), Computer Science Department
Carlos III University of Madrid, Colmenarejo - Spain
{joseluis.guerrero,jesus.garcia,josemanuel.molina}@uc3m.es

Abstract. This paper presents a new approach for trajectory segmentation in the area of Air Traffic Control, as a basic tool for offline validation with recorded opportunity traffic data. Our approach uses local information to classify each measurement individually, constructing the final segments over these classified samples as the final solution of the process. This local classification is based on a domain transformation using motion models to identify the deviations at a local scale, as an alternative to other global approaches based on combinatorial analysis over the trajectory segmentation domain.

Keywords: Air Traffic Control, segmentation, movement model, model fitting.

1 Introduction

Air Traffic Control (ATC) is a critical area related with safety, requiring strict validation in real conditions [3]. The basic considered data are sensor plots having the following components: stereographic projections of their x and y components, covariance matrix and detection time. The coordinates may be affected by errors, containing biases and noise. These sensor plots are then divided into segments sharing the same mode of flight or MOF (this division is known as the segmentation process, not to be mistaken for the one in contexts like [8]). The difficulty of that process is to differentiate accurately the different segments, especially at their edges (where it is difficult to determine whether position variation is caused by the measuring errors or by a different MOF). To improve that accuracy, as we are handling recorded data, we may use both past and future measures for our estimations.

Even though we have presented it for the ATC domain, this problem is presented in a wide range of domains such as tracking and segmentation of an object's trajectory in video data [2] (relating it to dimensionality issue), or the pattern recognition domain [9], (presenting segmentation as an optimization problem which trades off model fitting error versus the cost of introducing new segments, and introducing a solution based on dynamic programming).

In our current domain, some of the ideas we will be proposing in our segmentation algorithm are already found in available works, but in different contexts and applications. Machine learning techniques are applied in [5], but with very different attributes for our trajectory's measurements (in the reference they are based on IMM filtering [10]). Also the idea of needing several basic MM's (or movement models, a

simplification of the MOF's) is commonly covered ([5], [12], [6]), but their use differs to the one included in our proposal (for example, as individual models on an IMM filter or in the reconstruction process). It is interesting, as well, to consider that this segmentation problem is usually presented as a first step in the larger issue of trajectory reconstruction [12], [6].

In this study we will discuss an approach to the segmentation of trajectories where the three possible MM's are uniform, turn and accelerated movements [7]. With the presented input attributes, we will look for an algorithm that will sequentially use a different model to classify the measures belonging to each individual MM. This paper will be centered in the uniform MM.

In most cases available in the current literature on this topic, this segmentation problem and its solution are exposed as a global optimization issue [9]. Even so, all through this paper a local approach will be used. This implies that each of the trajectory's measurements will be individually classified according to the local information around it, and segments built with the classified isolated measurements will be the last step of our solution.

The formalization for our problem will be explained in the second section of this paper. The third section will present our general approach to the solution, while the fourth will analyze some initial issues of that proposal. The fifth section will present the validation experiments for the solution presented, along with some general results using that solution. Finally we will present the conclusions obtained from the solution's design and the overall results.

2 Problem Definition

2.1 General Problem Definition

As we presented in the introduction section, each analyzed trajectory (T^i) is composed by a collection of sensor reports (or measurements), which are defined by the following vector:

$$\vec{x}_j^i = \left(x_j^i, y_j^i, t_j^i, R_j^i\right), j \in \{1, \dots, N^i\} \quad (1)$$

where the j sub-index indicates the measurement number, the i super-index indicates the trajectory number, x_j^i, y_j^i are the stereographic projections, t_j^i is the detection time, R_j^i is the covariance matrix and N^i is the last measurement of the analyzed trajectory. From this problem definition our objective is to divide our trajectory into a series of segments (B_k^i), according to our estimated MOF. This is performed as an off-line process (meaning that we may use past and future information from our trajectory). The segmentation problem is formalized in (2)

$$T^i = \cup B_k^i \qquad B_k^i = \{x_j^i\} \qquad j \in [k_{min}, k_{max}] \quad (2)$$

where k is the segment number and k_{min}, k_{max} the given measurement boundaries for that segment. In the general definition of this problem these segments are obtained by the comparison with a test model of some windows of measurements coming from

our trajectory, in order to obtain a fitness value, deciding finally the segmentation operation as a function of that fitness value [5], [9].

On the one hand, the segments obtained can be seen as the problem's basic division unit (using a global approach [9]), being division the basic operation of the algorithm, or we may consider classifying each of the measurements from the trajectory alone (along with their local information) and obtaining the segments as a synthesized final solution, built upon the classification of the measurements (basically, by joining those adjacent measurements sharing the same MM into a common segment, being this the approach chosen in this paper).

2.2 Local Approach Problem Definition

We have presented our problem as an offline processing, meaning that we may use information both from our past and our future. Introducing this fact into our local representation, we will restrict that information to a certain local segment around the measurement which we would like to classify. These intervals are centered on that measurement, but the boundaries for them can be expressed either in time (4) or in number of measurements (3).

$$S_j^i = \{\vec{x}_k^i\}, k\epsilon\{j-p,..,j+p\} \quad p\epsilon[j-1, N-j] \tag{3}$$

$$S_j^i = \{\vec{x}_k^i\}, \quad t_k^i\epsilon\{t_j^i - m, t_j^i + m\} \quad m\epsilon[t_j^i - t_0^i, t_N^i - t_j^i] \tag{4}$$

Once we have chosen a window around our current measurement, we will have to apply a function to that segment in order to obtain its classification. This general classification function $F(\vec{x_j^i})$, using measurement boundaries, may be represented with the following formulation:

$$F(\vec{x_j^i}) = F(\vec{x_j^i}|T^i) \Rightarrow F(\vec{x_j^i}|S_j^i) = F_p(\vec{x}_{j-p}^i,.., \vec{x}_j^i,.., \vec{x}_{j+p}^i) \tag{5}$$

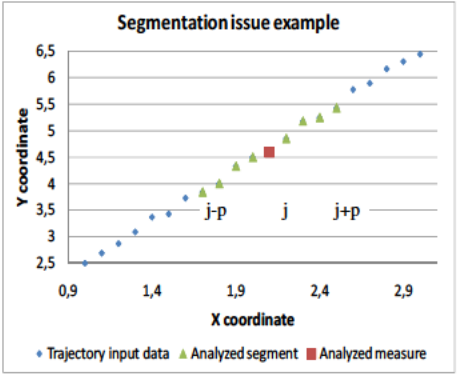

Fig. 1. Problem example

From this formulation of the problem we can already see some of the choices available: how to choose the segments (according to (3) or (4)), which classification function to apply in (5) and how to do the final segment synthesis.

3 Solution Proposal

As presented in the introduction section, we will consider three basic MM's and classify our measurements individually according to them [7]. If a measurement is classified as unknown, it will be included in the input data for the next model's analysis. This general algorithm introduces a design criterion based on the concepts of true positives rate (TPR, determining how many measurements belonging to our model are correctly classified) and false positives rate (FPR, determining how many unknown measurements we incorrectly classify), respectively equivalent to the type I and type II errors explained in [1]. The design criterion will be to keep a FPR as low as possible, understanding that those measurements, already assigned to a wrong model, will not be analyzed by the following ones. The proposed order for this analysis is the same in which we have introduced our MM's, and the choice is based on how accurately we can represent each of them.

In the local approach problem definition section, the segmentation problem was divided into two different sub-problems: the definition of the $F_p(\overrightarrow{x_j^i})$ function (to perform measurement classification) and a final segment synthesis over that classification.

We will divide our classification function $F(\overrightarrow{x_j^i})$ in a domain transformation $Dt(\overrightarrow{x_j^i})$ (domain specific, which may be seen as a data preprocessing [4]) and a final classification $Cl(Dt(\overrightarrow{x_j^i}))$ (based on general classification techniques). The domain transformation, $Dt(\overrightarrow{x_j^i})$ will convert our input data into a transformed domain (based on model fitting value) where a classification threshold will be chosen to determine whether our measurement belongs to the analyzed model or not. The output of that first phase will be several possible pre-classifications (according to parameters such as segment resolution, which will be explained in the first phase section) for each measurement of the trajectory ($\overrightarrow{x_j^i}$). For this first phase we will need to perform an analysis over the different parameters involved and a design of their value for the final algorithm proposition. This paper will be centered on these parameters' analysis.

The introduced final classification, $Cl(Dt(\overrightarrow{x_j^i}))$, will use the output data from the first phase to obtain a final classification for each measurement. After that classification has been performed, the isolated measurements will be joined into different segments, according to that classification (segment synthesis).

The formalization of these phases and the subsequent changes performed to the data is presented in the following vectors, representing the output data for our three processes:

Input data: $T^i = \{\overrightarrow{x_j^i}\},\ j\epsilon\{1..N^i\} \quad \vec{x}_j^i = (x_j^i, y_j^i, t_j^i, R_j^i)$.

Domain transformation: $Dt(\overrightarrow{x_j^i}) \Longrightarrow F(\overrightarrow{x_j^i}|T^i) \Rightarrow F(\overrightarrow{x_j^i}|S_j^i) = \{Pc_k^j\},\ k \epsilon \{1..M\}$

Pc_k^j = pre-classification *k* for measurement *j*.

Classification process: $Cl(Dt(\overrightarrow{x_j^i})) = Cl(\{Pc_k^j\}) = C_j$

C_j = automatic classification result for measurement *j*.

Final output: $T^i = \cup B_k^i \quad B_k^i = \{x_j^i\}\ j \epsilon [k_{min}, k_{max}]$

B_k^i= Final segments obtained by the union process.

4 First Phase: Domain Transformation

The first phase of our algorithm covers the process where we must synthesize an attribute from our input data to represent each of the trajectory's measurements in a transformed domain and choose the appropriate thresholds in that domain to effectively differentiate those which belong to our model from those which do not do so. This process has the following representative parameters: transformation function, segment management, extension and resolution, and threshold choosing technique.

The transformation function decision is the most crucial one involving this first phase of our algorithm. In [7] the discussion of whether introducing noise information in the domain transformation function allows us to improve our results was presented. The results proved that, as expected, that noise information improves the overall results. The transformed value presented was a normalized BLUE residue (6).

$$res = \frac{1}{(kmax-kmin+1)} \sum_{k=kmin}^{k=kmax} \left(x(k) - x_{int}(k) \;\; y(k) - y_{int}(k)\right) R_k^{-1} \begin{pmatrix} x(k)-x_{int}(k) \\ y(k)-y_{int}(k) \end{pmatrix} \quad (6)$$

where $x(k)$, $y(k)$ are the sensor measurements values, R_k is the covariance matrix (associated to the sensor) and $x_{int}(k)$,$y_{int}(k)$ are interpolated values using BLUE equations.

The rest of the parameters were briefly covered in [7] as well, even though we will review them here. The segment management determines whether we analyze the measurements alone or classify their surrounding segment according to the center measurement value. The segment extension is defines how we choose the units and boundaries for our segments (basically number of measurements or time interval constraints). Segment resolution refers to the choice of the length of those segments, and how it affects our results. The threshold choosing technique involves how we determine if a measurement belongs to our model or not. This parameter, not covered in previous works, will be covered in the next section.

4.1 Threshold Choosing Technique

The threshold choice involves determining the boundary above which transformed measurements will be considered as unknown. According to our design criterion, we would like to obtain a TPR as high as possible keeping our FPR ideally at a zero value. Graphically over figure 2, that implies getting the red line as low as possible, leaving only the central section over it (where the maneuver takes place, making its residue value high enough to get over our threshold).

The presented residue value in (6) follows a Chi-squared probability distribution function (pdf) normalized by its degrees of freedom, n. "n" is given by twice the number of 2D measurements contained in the interval minus the dimension of P (P=4 in uniform segment). For a valid segment residual, "res" behaves with distribution $\frac{1}{(kmax-kmin+1)} \chi^2_{2(kmax-kmin+1)-P}$, which has the following mean and variance:

$$\mu = 2 - \frac{P}{(kmax-kmin+1)} \qquad \sigma^2 = \frac{4}{(kmax-kmin+1)} - \frac{2P}{(kmax-kmin+1)^2} \quad (7)$$

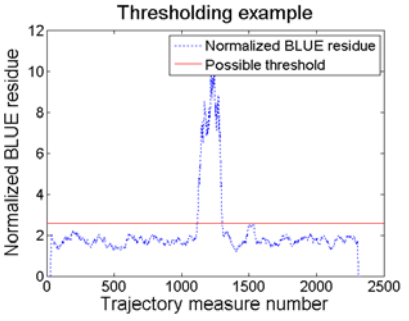

Fig. 2. Threshold choosing example

The residue distribution allows us to establish our criterion based on the TPR value, but not the FPR (we have a distribution over the uniform measurements, not the unknown ones), which is the one constrained by the design criterion. We may use the Chevychev's inequality [11] to determine a threshold which should leave the 99% of the measurements belonging to our model above it (TPR>=0.99), with $\mu + 3\sigma$ value. From the values exposed in (7) we get the following threshold value:

$$\text{thres}=2 - \frac{4}{N} + 3\sqrt{\frac{4}{N} - \frac{8}{N^2}} \quad N = (kmax - kmin + 1) \tag{8}$$

This threshold depends on the resolution of the segment (N), which also influences the residue value in (6). It is interesting to notice that the highest threshold value is reached with the lowest resolution. This is a logical result, since to be able to keep our TPR (having fixed it with the inequality at 99%) with short segments, we need to have a high threshold, in order to counteract the noise effects (while longer segments are more resistant to that noise and thus the threshold value may be lower).

We would like to determine how precisely our χ^2 distribution represents our normalized residue in non-uniform trajectories with estimated covariance matrix. In the following figures we compare the optimal result of the threshold choice (dotted lines), manually chosen, to the results obtained with equation (8). Figure 3 shows the used trajectories for this comparison, whereas figure 4 shows the actual comparison for the proposed trajectories between the optimal TPR and the one obtained with (8) for increasing threshold values.

In the two trajectories in figure 4 we may appreciate two distortion effects introduced by our approximation. The turn trajectory shows an underestimation of our TPR due to the inexactitude in the covariance matrix R_k . This inexactitude assumes a higher noise than the one which is present in the trajectory, and thus will make us choose a higher threshold than necessary in order to obtain the desired TPR margin.

In the racetrack trajectory we perceive the same underestimation at the lower values of the threshold, but then our approximation crosses the optimal results and reaches a value over it. This is caused by the second distortion effect, the maneuver's edge measurements. The measurements close to a maneuver beginning or end tend to have a higher residue value than the theoretical one for a uniform trajectory (due to their proximity to the non-uniform segments), making us increase the threshold value

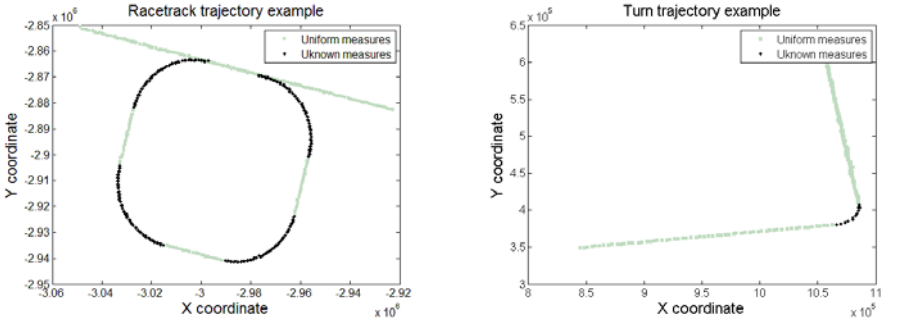

Fig. 3. Considered trajectories

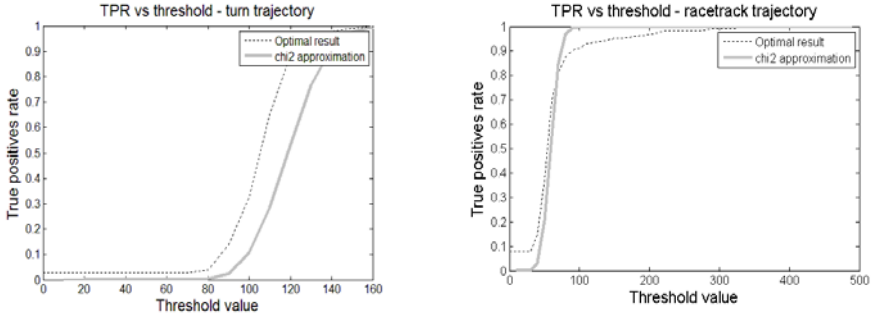

Fig. 4. χ^2 approximation comparison

to classify them correctly (which causes the optimal result to show a lower TPR in the figure). These two effects show that we may need a heuristic tuning in our χ^2 distribution in order to adapt it to these distortion effects.

5 Experimental Algorithm Design and Validation

The first step for the validation of the proposal is the generation of a set of test trajectories as representative as possible. We will include specific trajectories for each particular MM and also racetrack ones, which represent typical situations during landing procedures. To carry out the validation process we will add an additional component C_j^i to our measurement's data, which contains the real classification value of the measurement (one of our analyzed MM's), to obtain the results of the TPR and FPR indicators. This validation must be based on two different processes: the experiments performed in order to determine the design of the algorithm and those used to validate that design's results, covered in the next two sections.

5.1 Algorithm Design Parameters

This section will present the validation experiments needed to determine the design of the algorithm in the first phase section. Each of these alternatives must be compared and a decision over its value taken. One of the main difficulties arising in that process is that most of those parameters are related, so that global elections need to be made.

According to previous sections, we will use a BLUE reside, with the number of measurements as our segment extension and a χ^2 approximation based threshold choice. Resolution values allow us to obtain different effects according to their values, so a multi-resolution approach is chosen. Figure 5 shows the results for a sample turn trajectory of this pre-classifications, appreciating that high resolutions allow us to have better results (91 measurements) up to a certain limit, above which our maneuvers' boundaries start to increase, obtaining worse TPR results (121 measurements)

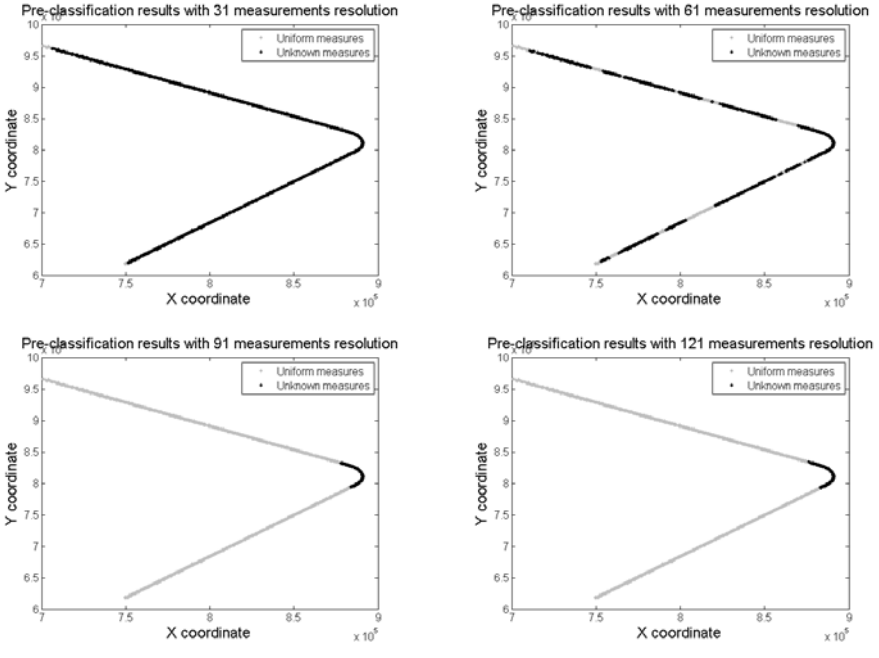

Fig. 5. Pre-classification results with different resolutions

5.2 Initial Validation Overview

Validation over our first phase pre-classification is a rather difficult process, as the complete solution algorithm must be validated as a whole. Even so, each of the different steps needs individual performance assessment in order to be able to design it correctly, so "ad hoc" validation methods are required. This is particularly true for this first phase, where the algorithm is required to find the input data for a second phase in order to be able, after that second phase, to classify each measurement correctly.

To make this achievement possible, we will choose a representative set of trajectories (two examples of each of the non-uniform possibilities presented) and obtain their TPR and FPR results over a set of different resolution values: 11, 31, 51, 71, 91 and 111. Due to space requirements we will only show the best result for each trajectory and its associated resolution. The idea behind this validation is to test the representativeness of our results for different resolutions, being able to measure whether, by joining these results, we will be able to obtain an accurate final classification. Table 1 shows these results.

Table 1. Classification results over simulated data

Trajectory	Resolution chosen	Results	
		TPR	FPR
Racetrack 1	31	0,9031	0
Racetrack 1	51	0,7591	0
Accel. 1	111	0,9935	0,0206
Accel. 2	111	0,9928	0
Turn 1	11	0,9876	0
Turn 2	91	0,9874	0

These results show that different resolutions allow us to accurately define the uniform segments of our trajectories (obtaining TPR values above 90 % in 5 out of 6 trajectories from the data set) while keeping the ideal zero value of FPR in almost every case (even in the trajectory where it exceeds the zero value, it is very low). This defines the boundaries for non-uniform MM segments and allows the application to those segments of the different possible non-uniform models (less accurate than the one presented in this work) to perform the final reconstruction

6 Conclusions

In this paper we have presented the general segmentation issue and its formulation in our particular ATC domain, where it is of capital importance for the automation process which the domain is going through. The basic idea is to segmentate an aircraft's trajectory, by means of a simplification into three basic different MM's. This can be performed from a global or a local approach, each of them with different advantages and difficulties. We have performed a local approach, covering in this paper its main issues: the domain transformation (along with the parameters determining how to choose the local information to classify each measurement), the classification techniques required and the final classification refinement needed (some of these only introduced, due to space restrictions). We have shown as well validation experiments over some critical parameters and a final overview over the quality of the general classification results, where we have obtained encouraging TPR and FPR values. Future work includes the complete description of the algorithm, introducing non-uniform models, along with its application to real data, in order to test its complete performance in real environments.

Acknowledgements

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

1. Allchin, D.: “ErrorTypes”. Perspectives on Science 9(1), 38–58 (2001)
2. Bashir, F.I., Khokhar, A.A., Schonfeld, D.: Segmented trajectory based indexing and retrieval of video data. In: Proceedings for the International Conference on Image Processing, ICIP (2003)
3. Kennedy, D., Gardner, A.B.: Tools for analysing the performance of ATC surveillance radars. IEE Colloquium (1998)
4. Famili, A., Shen, W.-M., Weber, R., Simoudis, E.: Data Preprocessing and Intelligent Data Analysis. Intelligent Data Analysis 1(1), 3–23 (1997)
5. Garcia, J., Perez, O., Molina, J.M., de Miguel, G.: Trajectory classification based on machine-learning techniques over tracking data. In: 9th International Conference on Information Fusion, Florence, Italy (2006)
6. Garcia, J., Molina, J.M., de Miguel, G., Besada, A.: Model-Based Trajectory Reconstruction using IMM Smoothing and Motion Pattern Identification. In: 10th International Conference on Information Fusion, Quebec, Canada (July 2007)
7. Guerrero, J.L., Garcia, J.: Domain Transformation for Uniform Motion Identification in Air Traffic Trajectories. In: International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI). Springer, Heidelberg (2008)
8. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting Time Series: A Survey and Novel Approach. In: Data Mining in Time Series Data Bases. Series in Machine Perception Artificial Intelligence, vol. 57, pp. 1–22
9. Mann, R., Jepson, A.D., El-Maraghi, T.: Trajectory segmentation using dynamic programming. In: Proceedings for the 16th International Conference on Pattern Recognition (2002)
10. Mazor, E., Averbuch, A., Bar-Shalom, Y., Dayan, J.: Interacting multiple model methods in target tracking: a survey. IEEE transactions in Aerospace and Electronic Systems (1998)
11. Meyer, P.: Introductory Probability and Statistical Applications, 2nd edn. Addison Wesley, Reading (1970)
12. Pérez, O., García, J., Molina, J.M.: Neuro-fuzzy Learning Applied to Improve the Trajectory Reconstruction Problem. In: International Conference on Computational Intelligence for Modelling, Control and Automation CIMCA 2006, Sydney, Australia (2006)