Fuzzy Logic and Target Estimation

Emmanuel Druon

Département Signaux et Systèmes
Institut Supérieur
d'Electronique du Nord
41 Bld Vauban
59046 Lille Cedex
FRANCE

Didier Willaeys

Laboratoire d'Automatique et de
Mécanique Industrielle et Humaine
URA CNRS 1118
Université de Valenciennes
59326 Valenciennes Cedex
FRANCE

Philippe Vanheeghe

Laboratoire d'Automatique et
d'Informatique Industrielle de Lille
URA CNRS D1440
Ecole Centrale de Lille
Cité Scientifique BP48
59651 Villeneuve d'Ascq Cedex
FRANCE

Abstract: Since its definition in 1965 by Professor L.A. Zadeh [16] fuzzy logic has been used in a large number of domains [15]. Fuzzy algorithms can now be found in various fields such as estimation, decision-making and automatic control.The problem considered here is to estimate with a fuzzy algorithm, the real target of a manoeuvring object moving in a 2D space. Numerical values of position and velocity are either sent by sensors (and therefore considered as noise-corrupted) or estimated by a Kalman filter. The proposed method uses fuzzy logic algorithms to analyze data obtained from different sources.

Emmanuel Druon was born in 1968. He received a Ph.D degree in Automatic Control from "Université de Valenciennes et du Hainaut - Cambrésis", France, in 1995. He is assistant professor at "Institut Supérieur d'Electronique du Nord", Lille, France. His research interests include soft computing and computer systems.

Didier E.G. Willaeys was born in 1946. He is professor at "Université de Valenciennes et du Hainaut - Cambrésis", France. He is teaching automatic control and information technology. He is a research manager at "Laboratoire d'Automatique et de Mécanique Industrielle et Humaine", Valenciennes, France. This laboratory is associated with CNRS.His research interests include diagnosis methods using Boolean logic, fuzzy set theory and Artificial Intelligence. He has some patents in the field of diagnosis systems for the certification of P.L.C. program.

Philippe M. Vanheeghe was born in 1956. He received a Ph.D degree in Computer Science from "Université des Sciences et Technologies de Lille", France, in 1984. He is head of the Signals and Systems Department at "Institut Supérieur d'Electronique du Nord", Lille, France. His research interests include modelling, analysis and control of complex systems.

I. Introduction

The problem is to estimate, as soon as possible, the real target of maneuvering objects directed towards different possible targets using a fuzzy logic algorithm.

In this paper, manoeuvring objects are considered as controlled by a Pure Proportional Navigation law (PPN law) [3],[8],[13],[14] and [17]. Each manoeuvring object is assumed to be directed towards a predefined non-manoeuvring target. The velocities of both targets and manoeuvring objects are assumed to be constant. Target velocities are supposed to be very low if compared to manoeuvring object velocities.

Before going any further in this study, the main parameters of PPN laws need to be briefly defined.

Figure 1 indicates the reference axes and the main notations used throughout this Section.

I.A. PPN law Definition

Using these notations, a PNN law can be defined [3] as a law setting the evolution of angle γ_O according to a proportional navigation constant A and to angle η:

$$\dot{\delta}_O = A \cdot \dot{\eta} \qquad (1)$$

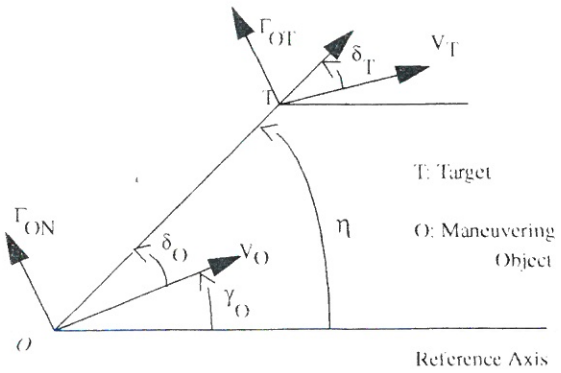

Figure 1. Reference Axes and Notations

Usually, the corresponding velocity vector evolution is obtained through the control of the

normal acceleration of the object. Using Figure 1 and Equation (1), Γ_{ON} can be written as:

$$\Gamma_{ON} = A \bullet V_O \bullet \dot{\eta} \quad (2)$$

In this paper, the term "proportional navigation coefficient μ" is often used. It is defined as:

$$\mu = A - 1 \quad (3)$$

It is also assumed [17] that μ is within the interval I_μ defined by:

$$I_\mu = [2,6] \quad (4)$$

In order to simplify the equations,the different targets are supposed to be moving along the reference axis. Moreover, since their respective velocity is chosen very low compared to the velocities of the manoeuvring objects, the targets are assumed to have no acceleration. Therefore,

$$\delta_T = \eta \quad (5)$$

$$\Gamma_{TN} = 0 \text{ m} \bullet \text{s}^{-2} \quad (6)$$

Targets and manoeuvring objects are defined as points and no outside perturbation is considered.

The different manoeuvring objects are assumed to be independent. This paper will therefore limit the study to one manoeuvring object moving towards a predefined target among different possible ones.

I.B. Definition of the Problem Parameters

In order to solve the target estimation problem previously described, a set of target estimation parameters has been derived [2], [4] and [6]. This set can be expressed as follows.

I.C. Definition of the Estimation Parameters [4]

Consider a system (Σ) which includes a manoeuvring object O and a set of n target points T_i (Figure 2). In the subsystem (Σ_{T_i}) of target points T_i, an observer is defined.

Noise-corrupted values of $\overline{OT_i}$, η_i (the angle between the reference axis and (OT_i)), V_O and V_{T_i} are sent to this observer every time interval Δt.

The observer also knows that object O is moving towards a predefined target of (Σ_{T_i}) using a PPN law with a proportional navigation coefficient μ defined within I_μ (see Equation (4)).

In order to solve the target estimation problem, the observer assumes at each time t that all the targets T_i are aimed at by object O. This hypothesis makes it possible to compute, for each target point T_i, an estimation vector v_i, the analysis of which characterizes the real target of O.

A precise definition of each parameter of this estimation vector can be found in [4]. In this paper, v_i will be defined as follows:

$$v_i(t) = \left[\mu_{estimated_i}(t), \delta o_i(t), \Delta c_i(t)\right]^T \quad (7)$$

where $\mu_{estimated_i}(t)$ is the proportional navigation coefficient estimated for target point T_i at time t; $\delta o_i(t)$ is the estimated angle between vector $\overline{V}_O$ and line (OT_i) at time t; $\Delta c_i(t)$ is a variable derived from the study of PPN laws in terms of optimal control [2].

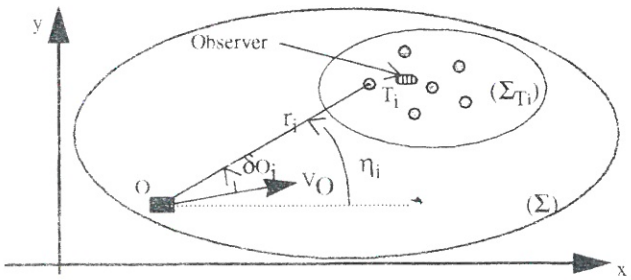

Figure 2. Notations Used To Define the Estimation Parameters of the Problem

$\mu_{estimated_i}(t)$ and $\Delta c_i(t)$ are computed with the Kalman filter derived in [6] which uses the noise-corrupted data of position and velocity of each component of system (Σ).

The variable $\delta o_i(t)$ can be computed either in a straightforward geometrical way or using the previous Kalman filter.

I.D. Theoretical Evolutions of the Parameters

The evolution of each element of estimation vector vi can be used to identify the real target of object O. This study has been conducted by V. Boulet and E. Duflos in [2] and [6]. Only the main results are going to be presented in this paper.

I.D.1 Theoretical Evolutions of $\mu_{estimated_i}(t)$

$\mu_{estimated_i}(t)$ is the estimated coefficient which is computed to adapt the kinematic reality (in terms of positions and velocities) to the hypothesis that object O is moving towards target T_i.

Using Equations (1) and (3) and using the fact that

$$\eta_i = \delta_{O_i} + \gamma_{O_i} \tag{8}$$

$\mu_{estimated_i}(t)$ can be defined as:

$$\delta_{O_i} = -\mu_{estimated_i} \bullet \eta_i \tag{9}$$

The study presented in [6] shows that according to the position of T_i with regard to the trajectory followed by O, the evolution of $\mu_{estimated_i}(t)$ can be derived.

Three different evolutions are identified:

- targets of Type I: $\mu_{estimated_i}(t)$ is always decreasing for $t \in [t_0, t_f]$ where t_0 represents the initial time and t_f the final time (usually the impact between O and the real target)
- targets of Type II: at first $\mu_{estimated_i}(t)$ is increasing. Then, there is a discontinuity when $\delta_{O_i} = 0$ and finally, $\mu_{estimated_i}(t)$ is decreasing.
- the real target: it is supposed that the object is controlled by a PPN law with a constant proportional navigation coefficient. Hence $\mu_{estimated_i}(t)$ is constant.

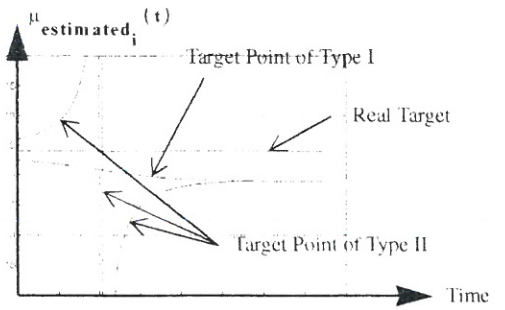

Figure 3. Theoretical Evolution of $\mu_{estimated_i}(t)$

Figure 3 shows these different behaviors.

I.D.2 Theoretical Evolutions of $\delta o_i(t)$

If T_1 is assumed to be the real target and considering that all the elements of (Σ) have a constant velocity and that $V_{T_i} \ll V_O$, it can be derived [6] that:

- if there is an initial error $(\delta_{O_1}(t_0) \neq 0)$, then $\delta_{O_1}(t)$ is strictly monotonic and converges towards 0.
- if there is no initial error, $\delta_{O_1}(t)$ is always close to 0

For the other target points, it can be shown [6] that:

- if T_i is of Type I: the monotonicity of $\delta_{O_i}(t)$ changes once and only once. $\delta_{O_i}(t)$ has the same variations as $\eta_i(t)$ at the end of the trajectory
- if T_i is of Type II: $\delta_{O_i}(t)$ is monotonic

Figure 4 shows possible theoretical evolutions of $\delta_{O_i}(t)$ if $\delta_{O_i}(t)$ is assumed positive.

I.D.3 Theoretical Evolution of $\Delta c_i(t)$

The evolution of $\Delta c_i(t)$ has been studied in [2]. This variable is defined as:

$$\Delta c_i(t) = u_i(t) - \hat{u}_i(t) \tag{10}$$

where $\hat{u}_i(t)$ is the optimal control to apply to object O so that O reaches its target with a criterion minimizing the terminal miss distance. That is:

$$\hat{u}_i(t) = -3 \bullet \dot{r}_i(t) \bullet \dot{\eta}_i(t) \tag{11}$$

The function $u_i(t)$ represents the control of a PPN law estimated by assuming that O is moving towards T_i. This gives, using Equation (2):

$$u_i(t) = A_i \bullet V_O \bullet \dot{\eta}_i \tag{12}$$

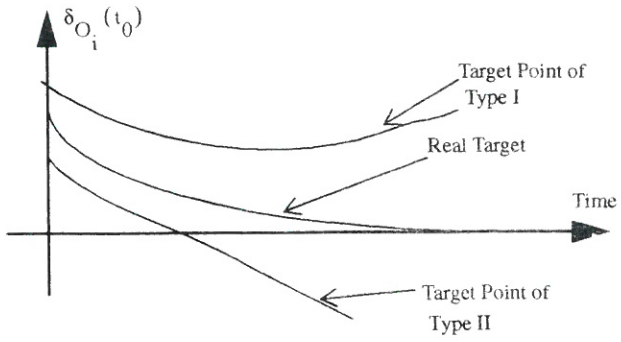

Figure 4. Hypothetical Theoretical Evolution of $\delta_{O_i}(t)$ if $\delta_{O_i}(t_0)$ Is Supposed Positive

The study presented in [2] shows that the evolution of $\Delta c_i(t)$ can be deduced from the evolution of $\delta_{O_i}(t)$ by exchanging the evolutions of Type I and Type II targets.

Figure 5 shows a hypothetical evolution for different $\Delta c_i(t)$ (assuming that $\Delta c_i(t_0)$ is positive).

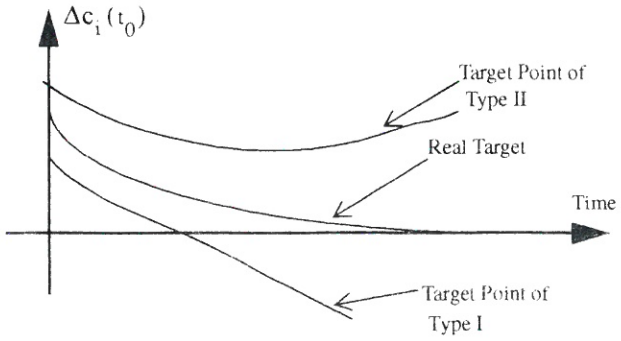

Figure 5. Hypothetical Theoretical Evolution of $\Delta c_i(t)$ if $\Delta c_i(t_0)$ Is Supposed Positive

I.E. The Estimated Evolutions of the Parameters

By analyzing the evolutions of each element of $v_i(t)$ and by comparing them with the previously presented theoretical evolutions, the real target can be estimated.

In reality, it is important to notice that all these parameters are computed from noise-corrupted data. Hence, a Kalman filter is used to estimate each component of $v_i(t)$ [6]. However, this filter is not perfect and it has drawbacks which are now going to be summarized.

I.E.1 Initialization Time

The way the Kalman filter has been implemented [6] implies an important initialization time in order to get correct estimations as soon as possible. In fact 101 sampling periods are necessary to obtain the first vector v_i. The first estimation is made at the middle point of these 101 samples: at sample number 50.

This implies that an estimation algorithm working only on these data would be delayed by at least 101 periods. Since the sampling period used is of 50 ms [4], the delay is therefore of 5.05 s.

I.E.2 Estimated Evolutions of $\mu_{estimated_i}(t)$

Because of the continuous characteristics of the implemented filter, the discontinuity presented in paragraph I.D.1 for Type II targets cannot be estimated. Consequently, a monotonic evolution is obtained: $\mu_{estimated_i}(t)$ for these targets is always increasing.

For the other kinds of targets, the evolutions are identical to those previously presented (if the initialization time is not taken into account).

Figure 6 presents the new evolutions obtained for $\mu_{estimated_i}(t)$.

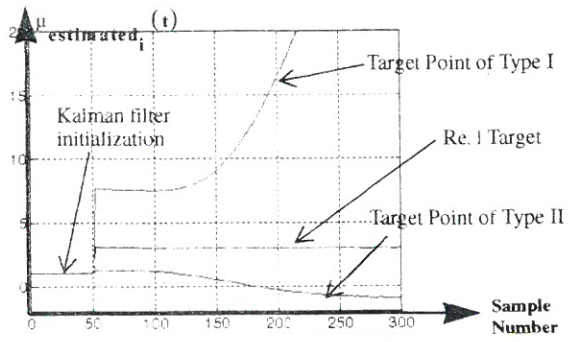

Figure 6. Evolutions of $\mu_{estimated_i}(t)$ Obtained with the Kalman Filter

Remark concerning target estimation using $\mu_{estimated_i}(t)$:

The differences between theoretical and estimated evolutions are not really significant as far as target estimation is concerned. It is still possible to identify the real target since its $\mu_{estimated_i}(t)$ is still the only constant estimated value.

I.E.3 Estimated Evolutions of $\delta o_i(t)$

The differences in the evolutions of $\delta_{O_i}(t)$ obtained with the Kalman filter are more significant. Due to the continuous characteristics of this filter, the estimated evolution of $\delta_{O_i}(t)$ for targets of Type II are similar to that of the real target.

Therefore, it becomes impossible to use the data estimated by the Kalman filter to «separate» targets of Type II from the real target.

Figure 7 shows the estimated evolutions of $\delta_{O_i}(t)$.

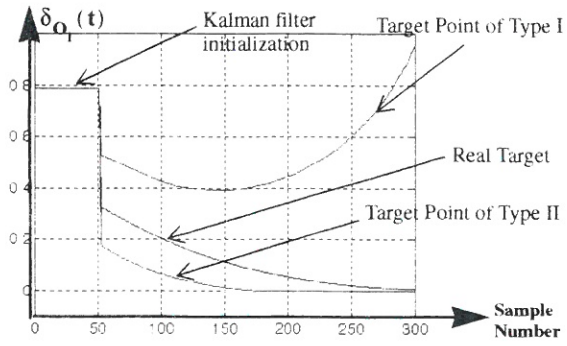

Figure 7. Evolutions of $\delta_{O_i}(t)$ Obtained with the Kalman Filter

I.E.4 Estimated Evolutions of $\Delta c_i(t)$

The problem for this parameter is identical to the one presented for $\delta_{O_i}(t)$. The evolutions of $\Delta c_i(t)$ for target points of Type II are similar to the evolutions of $\Delta c_i(t)$ for the real target. Therefore, this parameter cannot be used to correctly identify all kinds of targets.

Figure 8 shows the estimated evolutions of $\Delta c_i(t)$.

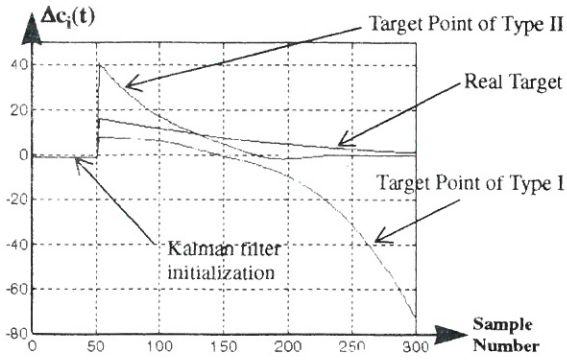

Figure 8. Evolutions of $\Delta c_i(t)$ Obtained with the Kalman Filter

I.F. Dealing with the Kalman Filter Imperfections

Based on the previous results, a target estimation model has been derived. This model takes into account the facts that:

- the initialization time of the Kalman filter is important (101 sample periods)
- the evolutions of the target estimation parameters computed by the Kalman filter are different from the theoretical ones.

Since this latter problem makes it impossible to separate Type II targets from the real target using $\delta_{O_i}(t)$ and $\Delta c_i(t)$, a direct geometrical estimation of $\delta_{O_i}(t)$ has been used. This estimation is based on the geometrical description of the problem (see Figure 1), and uses the noise-corrupted data.

The drawback of such a direct computation of $\delta_{O_i}(t)$ is that the obtained values of $\delta_{O_i}(t)$ are also noise-corrupted. Therefore, algorithms able to deal with such data have been developed.

The main advantage of using such a computation is that values are readily obtained and that the evolutions of $\delta_{O_i}(t)$ are globally (if noise-corruption is not considered) correct.

However, due to the necessity of knowing A_i (which is equivalent to estimating $\mu_{estimated_i}(t)$) to compute $\Delta c_i(t)$ (see Equations (10) and (12)), a direct computation of this optimal control based parameter is unfortunately impossible. Hence, its ability to estimate the real target of object O

becomes quite limited. This limitation will be more precisely expressed in the next Sections.

The target estimation algorithm is therefore based on two distinct phases. The first one only uses the direct computation of $\delta_{O_i}(t)$. The second one deals with all the components of $v_i(t)$: $\delta_{O_i}(t)$, $\mu_{estimated_i}(t)$ and $\Delta c_i(t)$ (the last two being estimated with the Kalman filter).

II. Fuzzy Logic Algorithms

The work needed to be performed by the target estimation algorithm is in fact reduced to multiple function analysis problems. It has to analyze each parameter of $v_i(t)$ to find out which T_i is the real target.

In order to take into account these different parameters and to be able to deal with the noise-corrupted $\delta_{O_i}(t)$, a fuzzy logic algorithm has been developed.

Since this algorithm has two different working phases, the study of two distinct types of algorithms has become necessary.

The first type deals with the noise-corrupted $\delta_{O_i}(t)$, the second one uses all the components of $v_i(t)$.

II.A. First Phase: Algorithms Working on $\delta_{O_i}(t)$

Different models of algorithms have been studied to identify the target of O as soon as possible. These algorithms are based on those presented by the authors in [5].

The estimation algorithms are based on a fuzzy logic controller. The idea is to create an estimation level, called $Danger\_Level_i(t)$, for each target T_i. This $Danger\_Level_i(t)$ is increased if T_i is identified as being the real target and it is decreased otherwise.

To perform this task, the recognition algorithm generates a variation value, called $\Delta level_i(t)$ for each T_i according to the computed $\delta_{O_i}(t)$.

The different parts of the target estimation algorithm are based on a fuzzy logic controller [9], [10] and [11]. Therefore, the first module consists in a fuzzification process. Then, the different inference rules on which the estimation process is based, are found. Finally, a defuzzification algorithm is run to obtain a discrete value for each $\Delta level_i(t)$.

Each $Danger\_Level_i(t)$ can then be updated by using the following equation:

$$Danger\_Level_i(t) = Danger\_Level_i(t-1) = \Delta level_i(t) \quad (13)$$

Schematically, these recognition algorithms can be represented as shown in Figure 9.

The general description of the target estimation algorithms having been given, they are now going to be presented. They are separated into two different groups. The first group concerns two algorithms directly using the noise-corrupted values of $\delta_{O_i}(t)$. The second one defines a target estimation algorithm based on data smoothed by a least-squares approximation algorithm.

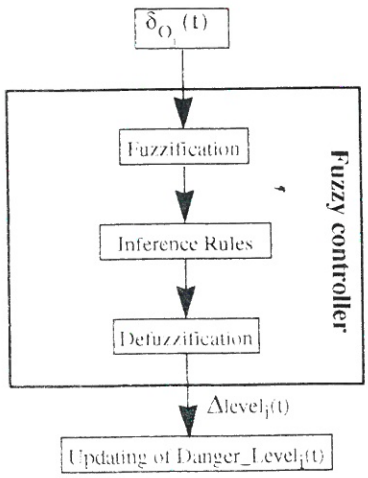

Figure 9. General Description of the Target Estimation Algorithms Based on the Evolutions of $\delta_{O_i}(t)$

II.A.1 Algorithms Using the Noise-corrupted Values of $\delta_{O_i}(t)$

II.A.1.a. Algorithm I: Algorithm Based only on $\delta_{O_i}(t)$

II.A.1.a.1. General Description

This first algorithm is based on a very simple fuzzy logic controller since it only uses one input: the noise-corrupted values of $\delta_{O_i}(t)$. The general description of this algorithm is exactly identical to the one shown in Figure 9. In the following, it is assumed that $\delta_{O_i}(t_0)$ is positive. The case $\delta_{O_i}(t_0)$ negative can easily be deduced from the presented results.

The analysis in Figure 4 gives some clues to how to derive different inference rules. In a noiseless environment, the fact that $\delta_{O_i}(t)$ crosses over the time-axis identifies T_i as being of Type II. However,since the input data used are noise-corrupted, it is necessary to build different fuzzy sets to take that noise into account. Therefore, two different fuzzy sets are defined in the negative area. The one closer to zero infers a low negative $\Delta level_i$ and the other set a large negative one. Such a design makes it possible to decrease the influence of a temporary negative value for the real target because of the noise.

The problem is more complex in the positive area. Considering the time evolution of each $\delta_{O_i}(t)$, the most natural way of dealing with the positive values is to consider $\dot{\delta}_{O_i}(t)$. However, since the data are noise-corrupted, this solution is not very realistic.

The other way of dealing with the positive values of $\delta_{O_i}(t)$ is to consider that the only critical area is the one close to zero. A large positive value of $\delta_{O_i}(t)$ is then considered as insignificant and infers no variation of $Danger\_Level_i$. In order to deal with these two cases, two fuzzy sets are also considered in the positive area.

The advantage of such a method is that the recognition level of the considered target is only increased if $\delta_{O_i}(t)$ is close to zero. This solves the problem of Type I targets.

However, a major drawback appears for targets of Type II if the hypothesis of a positive $\delta_{O_i}(t_0)$ is considered. For geometrical reasons (established from Figure 1), the starting value of $\delta_{O_i}(t)$ for Type II targets is always lower than that of the real target. Therefore, $\delta_{O_i}(t)$ for Type II targets always enters in the «around zero» area before the $\delta_{O_i}(t)$ of any other type. The immediate effect is an increase, for a given amount of time, of the danger level of Type II targets whereas a constant value is inferred for the real one. This result can be considered as an estimation error since the target T_i identified as being the real one is in fact of Type II.

For this algorithm, the fuzzification process is obtained by considering the truth value of each input for different fuzzy sets. The defuzzification process is based on a center of gravity method.

II.A.1.a.2. The Implemented Algorithm

In order to implement the target estimation algorithm the description of which has been derived, different simulations have been performed. The following fuzzy sets have been experimentally defined by analyzing the results.

As previously expressed, the universe of $\delta_{O_i}(t)$ is covered by four different fuzzy sets, named:

- Large Negative (LN) which concerns negative values of $\delta_{O_i}(t)$ far from zero
- Negative (N) which is for negative values of $\delta_{O_i}(t)$ close to zero
- Zero (Z) for the area around zero
- Large Positive (LP) which concerns positive values of $\delta_{O_i}(t)$ far from zero

These fuzzy sets are given in Figure 10. The boundaries are set by simulation results analysis.

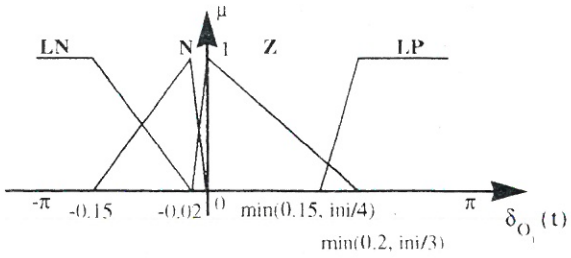

Figure 10. Fuzzy sets associated with $\delta_{O_i}(t)$ for the first recognition algorithm based on the noise-corrupted values of $\dot{\delta}_{O_i}(t)$ .The «ini» value is equal to $\delta_{O_i}(t_0)$

To simplify the defuzzification process, crisp values (non-fuzzy) are used for the output sets. They are labelled Very Large NegatiVe (VLNV), Large NegatiVe (LNV), Very Large PositiVe (VLPV) and ZeRo (ZR). They correspond respectively to the following $\Delta level_i$ -3, -2, 3 and 0.

The inference rules are rather simple. They are based on the general description of the algorithm presented at paragraph II.A.1.a.1. The operator used to combine them is defined as the maximum. They are given in the following decision table where each entry corresponds to a rule. For example, the first entry gives the rule:

$$\text{If } \delta_{O_i}(t) \text{ is N then } \Delta level_i(t) \text{ is LNV} \quad (14)$$

Table I Decision table for the first recognition algorithm based on the noise-corrupted values of $\delta_{O_i}(t)$

$\delta_{O_i}(t)$	LN	N	Z	LP
$\Delta level_i(t)$	VLNV	LNV	VLPV	ZR

Based on this model, it is now possible to derive the general evolution of $\delta_{O_i}(t)$ for each kind of target assuming non noise- corrupted data. It is given in Figure 11.

II.A.1.a.3. Simulation Results

This first target estimation algorithm has been implemented to determine its performance. The speed performance is defined by the target estimation distance. It corresponds to the distance OT from which the danger level of a given target is higher than any other one till the end of the simulation. The precision performance is given as an error ratio as shown in Figure 12.

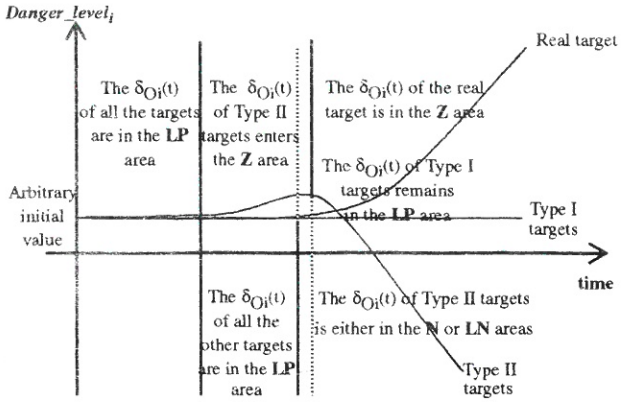

Figure 11. Estimated Evolutions of Danger_Level $_i$(t) for Noiseless $\delta_{O_i}(t)$ for Algorithm I

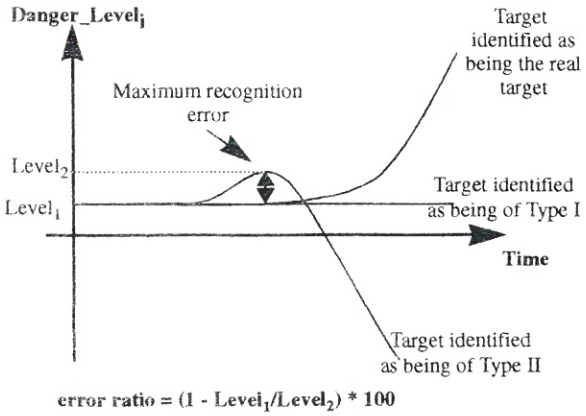

Figure 12. Graphical description of the error ratio used to derive the precision performance of the recognition algorithms. The Danger_Level$_i$ evolutions are hypothetical.

The simulation part is divided into two sections. The trajectory simulation, solving the kinematic equations presented in paragraph I.A and the estimation part implementing the proposed fuzzy estimation algorithm.

Runge-Kutta integration methods were used to solve the kinematic equations. The integration time (Δt) was chosen equal to the sampling period

used by the Kalman filter: 0.05 second.

This theoretical trajectory was then altered by corrupting the data with white noise. The noise standard deviations chosen were 0.0016 radians for η_i, 20 m for r_i and 10 m.s^{-1} for V_O.

The proposed scenario is shown in Figure 13. Manoeuvring object O is moving towards target T_1 using a PPN law with a coefficient μ equal to 3 and a $\delta_{O_i}(t_0)$ equal to 0.5 radians (28°60). The velocity of the targets was set to 15 m.s^{-1} and all the targets are supposed to be moving along the x-axis. The manoeuvring object velocity was set to 800 m.s^{-1}. All the targets are equidistant from T_1 and this distance is set to 3 kilometers. Manoeuvring object O is starting at a distance of 15 kilometers from target T_1. Simulations have been performed with different values of $\eta_1(t_0)$.

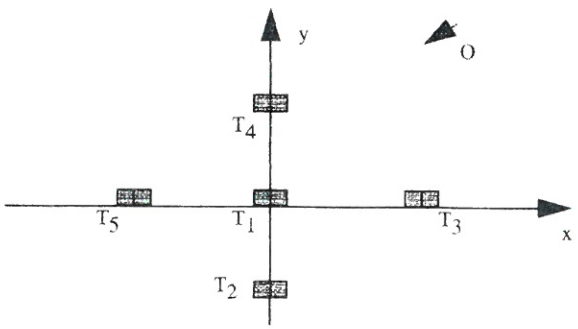

Figure 13. Simulation Scenario

Figure 14 shows the Danger_Level$_i$(t) evolutions obtained for $\eta_1(t_0) = 45^o$ with this first fuzzy algorithm.

As noticed if comparing Figures 11 and 14, the obtained evolutions correspond to the estimated ones.

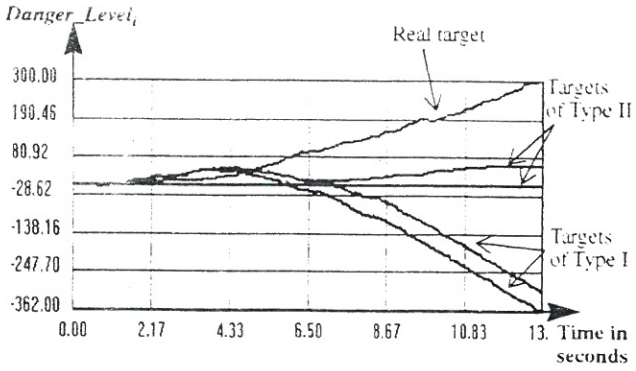

Figure 14. Simulated danger level evolutions obtained with algorithm I. The initial conditions are as described in the simulation scenario with $\eta_1(t_0) = 45^o$.

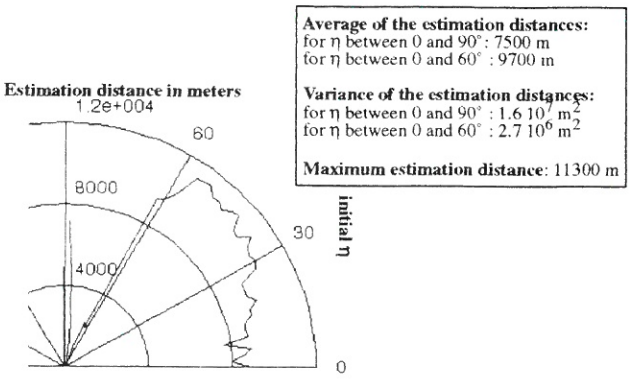

Figure 15. Estimation distance of the real target as a function of $\eta_1(t_0)$ for algorithm I. The initial conditions are those described in the simulation scenario.

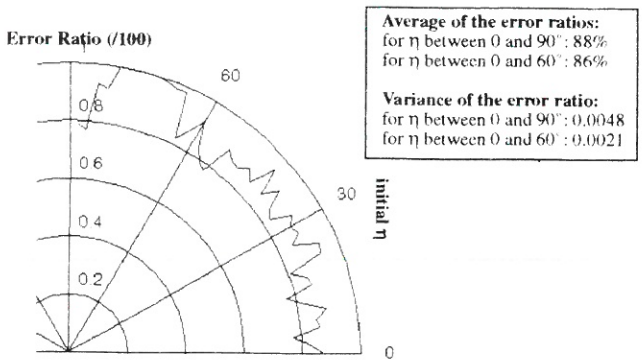

Figure 16. Target estimation error ratio as a function of $\eta_1(t_0)$ for algorithm I. The initial conditions are those described in the simulation scenario.

Figures 15 and 16 show respectively the target estimation distance and the error ratio for $\eta_1(t_0)$ varying between 0 and 90°.

Table II sums up the results obtained with this first algorithm.

Table II Performance of the first target estimation algorithm based on the noise-corrupted values of $\delta_{O_i}(t)$

Result type	Average	Variance
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 90°.	7500 m	$1.6\ 10^7\ m^2$
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 60°	9700 m	$2.7\ 10^6\ m^2$
Error ratio for $\eta_1(t_0)$ varying between 0 and 90°.	88%	0.0048
Error ratio for $\eta_1(t_0)$ varying between 0 and 60°.	86%	0.0021
Best estimation distance	11300 m	

II.A.1.a.4. Results Analysis

The results presented in Table II show that the estimation of the real target of a manoeuvring object controlled by a PPN is possible even with a simple algorithm. However, the observed performance is not impressive. The error ratio is very important. It indicates that there is an estimation error at the beginning of the simulation.As previously explained, this is mainly due to the poor analysis performed for positive values of $\delta_{O_i}(t)$. The estimation distance is not really good either. On the average, it takes one third of the simulation time to correctly identify the real target.

The obtained results also show that for $\eta_1(t_0)$ between 60 and 90°, the estimation algorithm has not always identified correctly the real target. In order to understand the problem in this now called "critical zone", a more precise study of this area has been performed.

II.A.1.a.5. Analysis of the Critical Zone Defined for $\eta_1(t_0)$ Between 60 and 90°

A geometrical analysis of this critical zone shows the estimation error around $\eta_1(t_0)=80°$ due to the fact that for the simulation conditions used, more than one target is in the line of sight of the manoeuvring object close to impact. The analysis of the results in this area shows that these troublesome targets are those aligned with the real target for η_{impact} (the value of η derived from the real target close to impact) equal to 90°. The problem is graphically described in Figure 17.

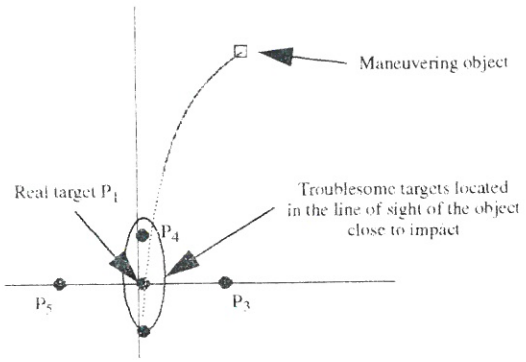

Figure 17. Graphical Explanation of the Critical Zone

This graphical explanation can be proved mathematically. In [4], the expression of η_{impact} is derived as follows:

$$\eta_{impact}=\eta(t_0)-\frac{1}{1-A}\cdot\delta_O(t_0) \qquad (15)$$

For $\eta_{impact}=90^{o}$, for the values of μ and δ_O defined by the simulation scenario (respectively 3 and 28°6) and using Equation (3), Equation (15) gives:

$$\eta(t_0)=80^{o}4 \qquad (16)$$

A more precise study of the estimation distance in the critical zone has been conducted. The results are shown in Figure 18.

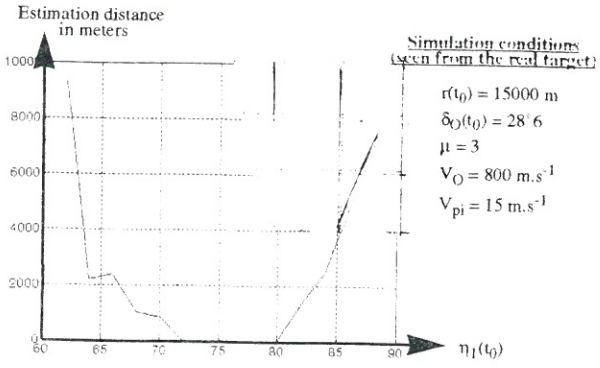

Figure 18. Estimation Distance of the Real Target in the Critical Zone

As can be observed in Figure 18, the closer $\eta_1(t_0)$ is to 80°4, the more difficult it is to estimate the real target. This is mainly due to the fact that targets P_1, P_2 and P_4 are aligned in the «close to impact» line of sight of the object for a longer time.

It is also important to notice that the estimation is more difficult for $\eta_1(t_0)$ values inferior or equal to 80°4. The reason is the curvature of the trajectory implied by a positive initial value of $\delta_{O_i}(t_0)$. If the same simulations are performed for a negative one, the estimation is harder for $\eta_1(t_0)$ values superior or equal to 80°4.

To conclude with this first algorithm, it seems important to say that its poor performance generates a large critical zone. A similar graph is plotted for a more elaborate algorithm in [1] and further on in this paper, proving that for better algorithms, this zone is thinner.

II.A.1.b. Algorithm II: An Algorithm Using Dynamic Fuzzy Sets

II.A.1.b.1. General Description

The main problem of the previous algorithm is that it uses the positive area as a static one. However, the analysis of the different $\delta_{O_i}(t)$ evolutions shows that this area has, in fact, two different roles.

On starting simulation, its role is not really significant since all the $\delta_{O_i}(t_0)$ are assumed as positive. As the simulation progresses, its role is more and more important since it allows the dissociation of Type I targets from the real one.

Considering this fact, a new algorithm based on the noise-corrupted values of $\delta_{O_i}(t)$ has been derived. The idea is to use time varying fuzzy sets in the positive area in order to take into account the two roles of this zone. The boundaries of these fuzzy sets are dynamically modified so that they have more and more importance as the simulation proceeds.

To precisely control the evolution of the danger level, a new fuzzy set is created for the input data. It is called Positive (P). This new set is necessary for two reasons. First, it allows the fuzzy logic controller to generate negative variations if the input is large positive. Second, it makes it possible to take the noise into account as done in the negative area. The fuzzy set Z infers a very large positive variation. P only generates a large positive one.

The fuzzification and defuzzification processes are obtained as previously. The fuzzy sets for the negative area are unchanged.

II.A.1.b.2. The Implemented Algorithm

An experimental study based on multiple trials with time decreasing functions has made it possible to define an acceptable evolution of the boundaries for the fuzzy sets in the positive area. Hence, this evolution is defined to get a full significant role for different sets after a time equal to $2 \times \left(t_{f\,direct_i}\right)/3$ where $t_{f\,direct_i}$ is defined as the time for object O to reach target T_i in a straight line, that is:

$$t_{f\,direct_i}=\frac{r_i(t_0)}{V_O} \qquad (17)$$

where $r_i(t_0)$ represents the initial distance OT_i

and V_O the module of the velocity vector of O.

The fuzzy sets associated with different input data are shown in Figure 19.

To simplify the defuzzification process, the output sets are crisp (non-fuzzy) sets. Four such sets have been defined. They are labelled Very Large NegatiVe (VLNV), Large NegatiVe (LNV), Large PositiVe (LPV) and Very Large PositiVe (VLPV). Their corresponding numerical values are: -3,-2, 2 and 3.

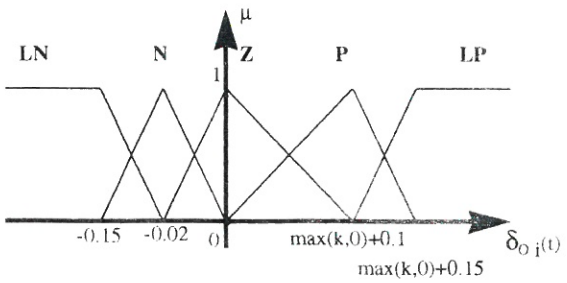

Figure 19. Fuzzy sets associated with $\delta_{O_i}(t)$ for the second recognition algorithm using dynamic fuzzy sets. The term k is time dependent and defined by:

$$\delta_{O_i}(t_0) - \left(\delta_{O_i}(t_0) \cdot t \cdot 2 \cdot t_{f\ direct_i} / 3\right)$$

The decision table shown as Table III gives the inference rules which have been derived from the general description of the recognition algorithm. As before, each entry defines an inference rule. The combination between different rules is performed with the maximum operator.

Table III Decision table for the second recognition algorithm based on dynamic fuzzy sets

$f_i(t)$	LN	N	Z	P	LP
$\Delta level_i(t)$	VLNV	LNV	VLPV	LPV	VLNV

Using the previous Table, it is possible to derive the theoretical evolution (based on noiseless information) of the danger level associated with each target. At first, each danger level is increasing for all the targets since $\delta_{O_i}(t)$ for each T_i is either in zone Z or P. Then, as the object proceeds towards its target, the danger level of Type II targets starts decrease since their $\delta_{O_i}(t)$ reaches either zone N or LN. The danger level of Type I targets either continues to increase (if $\delta_{O_i}(t)$ is in the P zone) or starts decrease (if $\delta_{O_i}(t)$ is in the LP zone). The danger level of the real target is always increasing since its $\delta_{O_i}(t)$ is always in zone Z or P.

These different theoretical evolutions are summed up in Figure 20.

II.A.1.b.3. Simulation Results

Target estimation algorithm II has been implemented and simulations have been performed under the same conditions as those presented for algorithm I.

Figure 21 shows the danger level evolutions obtained through simulation for an $\eta_1(t_0)=45^o$. The results obtained can be compared with those presented in Figure 20. The only difference between the two sets of results is that during a short time interval, the danger level of Type II targets is higher than that of real target. That is due to the fact that in simulations, noise-corrupted data are used whereas the theoretical evolutions are established from noiseless information. Therefore, during this time interval, the noise- corrupted $\delta_{O_i}(t)$ of some Type II targets are in the Z area whereas the $\delta_{O_i}(t)$ of the real target is in the P area.

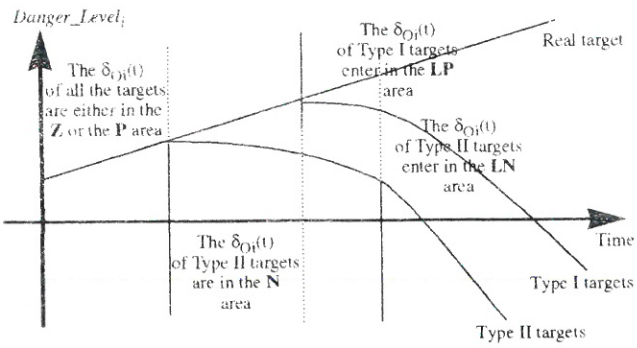

Figure 20. Estimated Evolutions of Danger_Level $_i(t)$ for Noiseless $\delta_{O_i}(t)$ for Algorithm II

Figures 22 and 23 give respectively the target estimation distance and the error ratio as a function of $\eta_1(t_0)$.

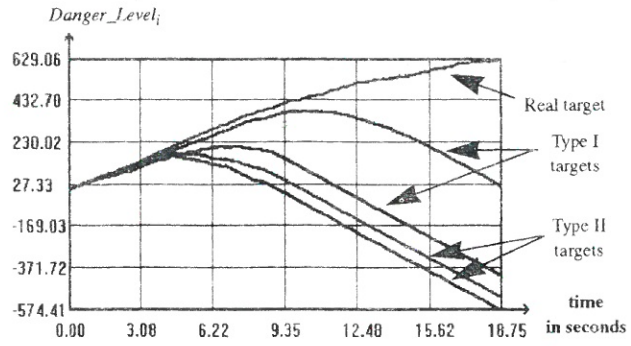

Figure 21. Simulated danger level evolutions obtained with algorithm II. The initial conditions are as described in the simulation scenario with $\eta_1(t_0)=45^o$.

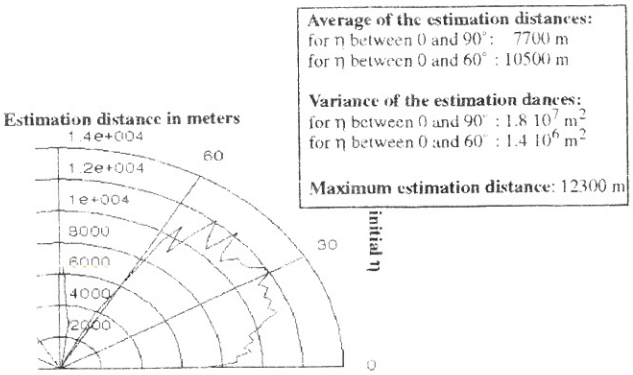

Figure 22. Estimation distance of the real target as a function of $\eta_1(t_0)$ for algorithm II. The initial conditions are those described in the simulation scenario.

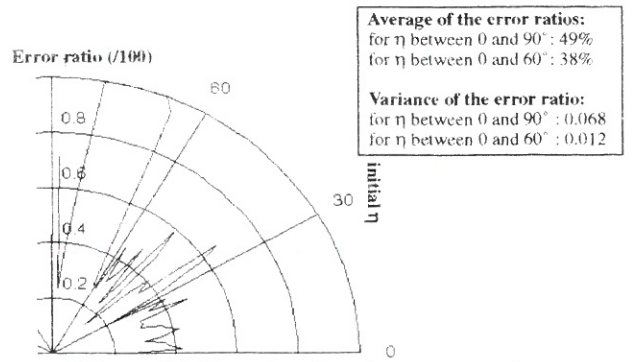

Figure 23. Target estimation error ratio as a function of $\eta_1(t_0)$ for algorithm II. The initial conditions are those described in the simulation scenario.

Table IV summarizes the averages and the variances of both estimation distances and error ratios obtained with this second estimation algorithm.

Table IV Performance of the second target estimation algorithm based on the values of $\delta_{O_i}(t)$

Result Type	Average	Variance
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 90°.	7700 m	$1.8\ 10^7\ m^2$
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 60°.	10500 m	$1.4\ 10^6\ m^2$
Error ratio for $\eta_1(t_0)$ varying between 0 and 90°.	49%	0.068
Error ratio for $\eta_1(t_0)$ varying between 0 and 60°.	38%	0.0124
Best estimation distance	12300 m	

II.A.1.b.4. Results Analysis

The previous results immediately show the advantage of separating the two roles of the positive area. The error ratio has been reduced by more than half in the non-critical zone and almost by half if $\eta_1(t_0)$ is chosen between 0 and 90°. The estimation distance has also been improved even if the difference is not as important as the one obtained for the error ratio.

The decrease of the error ratio is very important in case of the target estimation algorithm. It proves that the estimation error at the beginning of the simulation has been largely reduced. The resulting algorithm is therefore more reliable.

The two target estimation algorithms which have

been presented are based on noise-corrupted values of $\delta_{O_i}(t)$. In order to check if the performance can be improved if smoothed data are used, another algorithm is now studied.

II.A.2 Algorithm III: An Algorithm Based on Smoothed Data

II.A.2.a. General Description

The main advantage of using smoothed data is the possibility of taking into account the first derivative of $\delta_{O_i}(t)$. This implies that the use of dynamic fuzzy sets is no longer necessary: the separation of the two roles of the positive area is obtained by a combined analysis of $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$. The general description of this new target estimation algorithm is therefore modified. It is given in Figure 24.

In order to smooth the noise-corrupted data, a least-squares approximation algorithm is used [7]. In fact, a total of three different least-squares algorithms are used. Two of them are respectively used to smooth the values of $r_i(t)$ and $\eta_i(t)$ necessary to compute $\delta_{O_i}(t)$. The third one is then used to smooth these last values.

Based on an experimental study, the first two least-squares algorithms are chosen to fit a third degree polynomial function. The third one is designed to fit a fourth degree polynomial function.

Moreover, this experimental study shows that best results are obtained if two different methods are used to compute the smoothed values. For $r_i(t)$ and $\eta_i(t)$, the smooth values are always defined locally, using N_1 measures of $r_i(t)$ and $\eta_i(t)$respectively. On the other hand, the complete history of the values is kept to compute $\delta_{O_i}(t)$ starting with N_2 measures.

Therefore, for $r_i(t)$ and $\eta_i(t)$, the smooth value is always defined as the middle point of N_1 data. Only the first $N_1/2$ values and the last $N_1/2$ values are defined with the same polynomial function (respectively the first one and the last one).

On the other hand, the first $N_2/2$ smooth values of $\delta_{O_i}(t)$ are computed from the polynomial function estimated from the first N_2 noise-corrupted data. Then, each new entry estimates a new smooth value of $\delta_{O_i}(t)$.

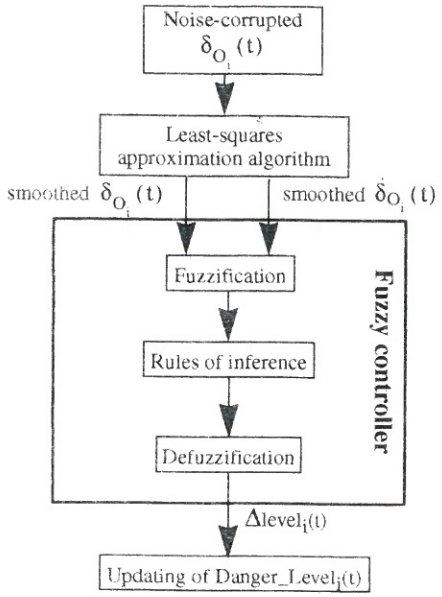

Figure 24. New general description of the target estimation algorithm using the smoothed data of $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$

The number of points to start with (N_1 and N_2) is however hard to define. Numbers too large delay the target estimation algorithm too much. Numbers too small do not sufficiently smooth the data. In order to get the optimal value, an auto-adaptive fuzzy algorithm [12] could be studied in order to automatically derive the best fuzzy sets for each parameter of the controller.

However, in this paper the fuzzy sets have been defined manually. The smooth data are obtained by taking N_1 and N_2 equal to 26. This number seems quite appropriate since the simulations performed show that it highly smoothes the noise-corrupted data without generating too important delays (a total of 2 seconds if the sampling period is chosen equal to 50 ms). Table V gives the variance and the average of the errors compared to the non noise- corrupted data.

Table V Average and variance of the error due to the least- squares algorithms for $N=N_1=N_2=26$ points

Variable	Average of the errors compared to non noise-corrupted data	Variance of the error
r_i	0.26 m	1.05 m^2
η_i	8.4 10^{-6} rad	4.3 10^{-9} rad^2
δ_{Oi}	6.1 10^{-3} rad	1.9 10^{-4} rad^2
$d\delta m_i/dt$	8.8 10^{-6} rad	1.27 10^{-7} rad^2

The fuzzification and defuzzification processes are realized as previously. The inference rules are created to identify the different function types using the smoothed values of $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$.

Since the target estimation algorithm is now working on smoothed data, the use of two fuzzy sets in the negative area is no longer necessary. Therefore, only the fuzzy set previously labelled Large Negative (LN) is used.

In the positive area, the Positive (P) fuzzy set introduced by the second algorithm is kept. Its purpose is only to ease the inference of a correct variation of the danger level in this area.

II.A.2.b. The Implemented Algorithm

Figure 25 and 26 show respectively the fuzzy sets derived by simulation for $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$. As previously,the numerical values for the boundaries are obtained experimentally.

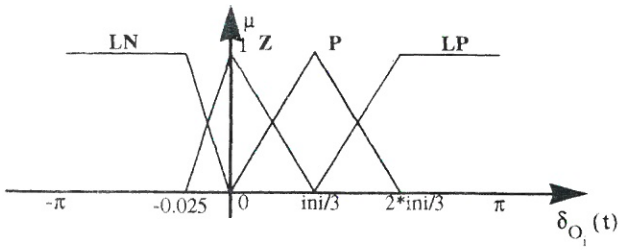

Figure 25. Fuzzy sets associated with $\delta_{O_i}(t)$ for the third estimation algorithm based on the smoothed values of $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$. The value «ini» is equal to $\delta_{O_i}(t=0)$

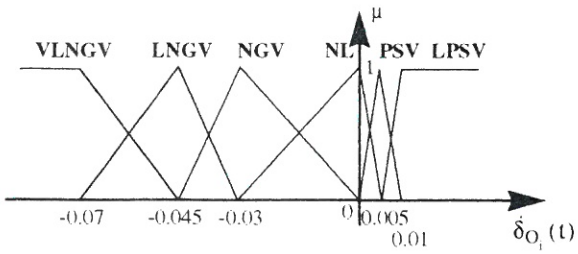

Figure 26. Fuzzy sets associated with $\dot{\delta}_{O_i}(t)$ for the third target estimation algorithm based on the smoothed values of $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$.

The fuzzy sets associated with $\dot{\delta}_{O_i}(t)$ are labelled Very Large NeGatiVe (VLNGC), Large NeGatiVe (LNGV), NeGatiVe (NGV), NuL1 (NL), PoSitiVe (PSV) and Large PoSitiVe (LPSV).

To simplify the defuzzification process, the output values are defined as crisp (non-fuzzy) numbers. Their labels are Very Large NegatiVe (VLNV), Large NegatiVe (LNV), NegatiVe (NV), ZeRo (ZR), PositiVe (PV), and Very Large PositiVe (VLPV). The value associated with each number is respectively: -3, -2, -1, 0, 1 and 3.

The decision table which defines the inference rules is given as Table VI. Each entry defines a rule. The two inputs are combined with the

minimum operator and the different rules with the maximum operator.

Table VI Decision table for the third recognition algorithm based on smoothed values of $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$

$\dot{\delta}_{O_i}(t)$ \ $\delta_{O_i}(t)$	LN	Z	P	LP
VLNGV	VLNV	NV	PV	PV
LNGV	VLNV	ZR	PV	PV
NGV	VLNV	PV	PV	ZR
NL	VLNV	VLPV	ZR	NV
PSV	VLNV	NV	NV	LNV
LPSV	VLNV	LNV	LNV	VLNV

The theoretical evolutions of the danger level of each kind of a target can be derived from the previous Table. They are quite similar to those presented for the second estimation algorithm. The main difference is a delay at the beginning of the simulation, a delay due to the least squares algorithms. Figure 27 presents these evolutions derived from non noise-corrupted data.

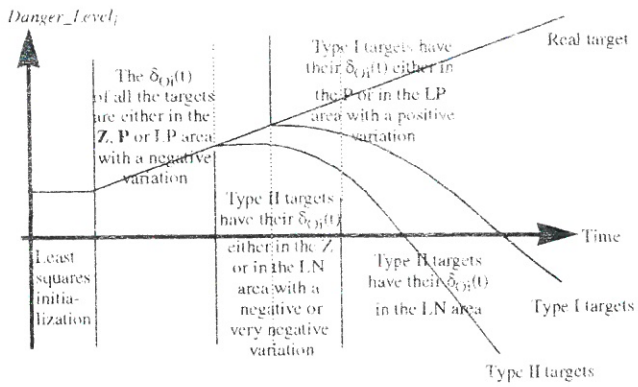

Figure 27. Estimated evolutions of Danger_Level$_i$ (t) for noiseless $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$: algorithm III

II.A.2.c. Simulation Results

Figure 28 shows the evolutions of the danger levels obtained by simulation. These results are consistent with those presented in Figure 27. Moreover, by comparing them to those obtained from the two previous algorithms, it can be noticed that the evolutions of the danger levels are more regular with this new algorithm. This advantage is due to the fact that the new model of the target estimation algorithm is based on smoothed data and allows the use of the first derivative of $\delta_{O_i}(t)$.

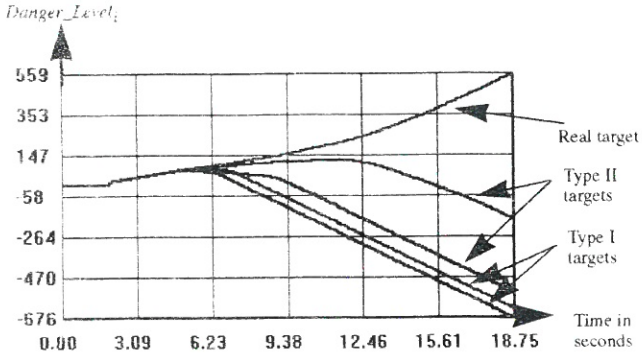

Figure 28. Simulated danger level evolutions obtained with algorithm III. The initial conditions are as described in the simulation scenario with $\eta_1(t_0) = 45^o$.

Figures 29 and 30 give respectively the target estimation distance and the error ratio as a function of $\eta_1(t_0)$.

Table VII summarizes the results obtained with this third target estimation algorithm. The simulation conditions are identical to those presented previously.

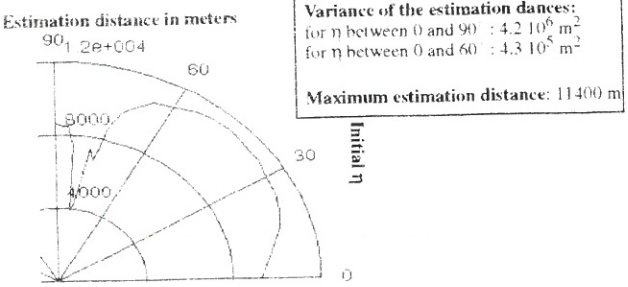

Figure 29. Estimation distance of the real target as a function of $\eta_1(t_0)$ for algorithm III. The initial conditions are those described in the simulation scenario.

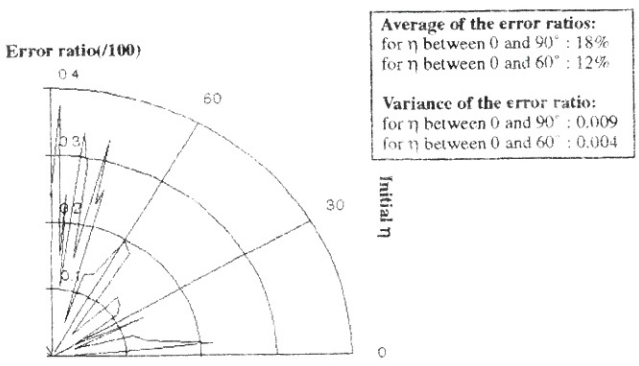

Figure 30. Target estimation error ratio as a function of $\eta_1(t_0)$ for algorithm III. The initial conditions are those described in the simulation scenario.

Table VII Performance of the third target estimation algorithm based on the smoothed values of $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$

Result Type	Average	Variance
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 90°	9200 m	4.2 106 m^2
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 60°	10700 m	4.3 10^5 m^2
Error ratio for $\eta_1(t_0)$ varying between 0 and 90°	18%	0.009
Error ratio for $\eta_1(t_0)$ varying between 0 and 60°	12%	0.004
Best estimation distance	11400 m	

II.A.2.d. Results Analysis

Previous results have shown the advantage of using smoothed data. A more precise control of Δlevel $_i$ is obtained. It implies better and more stable estimation distances and lower error ratios.

The only problem seems to be the best estimation distance which is the worst obtained so far. Simulation analysis shows that this is mainly due to the delays implied by the least-squares algorithms which are certainly not optimized.

However, the reliability of this new algorithm (enhanced by low error ratios) makes it a good choice for this first working phase of the general target estimation algorithm.

The following description concerns the second phase of the target estimation algorithm when all the components of estimation vector $v_i(t)$ are taken into account.

II.B. Second Phase: Algorithms Working on $v_i(t)$

It seems necessary to remind the reader of this second phase as working on all the parameters of estimation vector $v_i(t)$. It starts when the Kalman filter is initialized.

II.B.1. General Description

The fuzzy estimation algorithm is derived to compute a fast and accurate estimation of the real target of manoeuvring object O. It uses the different parameters of vector $v_i(t)$ as well as their first derivative.

II.B.2 The Implemented Algorithm

As far as $\delta_{O_i}(t)$ and $\dot{\delta}_{O_i}(t)$ are concerned, the smoothed values computed by the least- squares algorithms previously presented are used. Therefore, the fuzzy sets associated with these two variables are those presented in Figures 25 and 26.

Only one fuzzy set is defined for $\mu_{estimated_i}(t)$. Its role is to ensure that $\mu_{estimated_i}(t)$ is inside the interval of acceptable μ, defined by Equation (4). It takes into account the possible error induced by the Kalman filter. It is called ACCP which stands for ACCePtable values of μ. It is represented in Figure 31.

Since a constant evolution of $\mu_{estimated}(t)$ characterizes the real target of object O, fuzzy sets have been derived to analyze $\dot{\mu}_{estimated_i}(t)$. These sets are called MUNEG, MUNLL and MUPOS which identify respectively negative, null and positive variations of $\mu_{estimated}(t)$. Their

boundaries are defined by an experimental analysis of $\dot{\mu}_{estimated_i}(t)$. They are given in Figure 32.

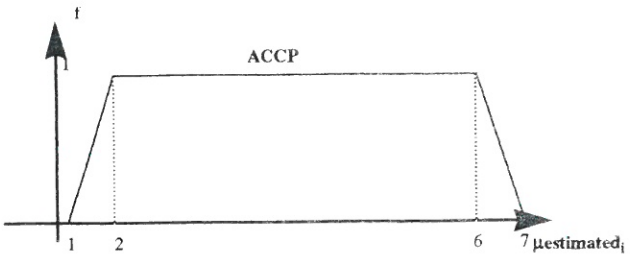

Figure 31. Fuzzy set associated with $\mu_{estimated_i}(t)$.
Letter f indicates the membership function of fuzzy set ACCP to avoid confusion.

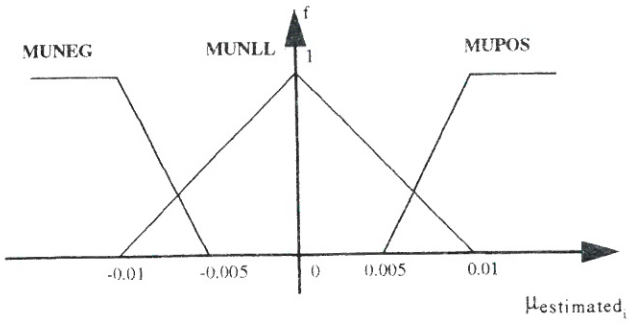

Figure 32. Fuzzy set associated with $\dot{\mu}_{estimated_i}(t)$. Letter f indicates the membership function of fuzzy set ACCP to avoid confusion.

Because of the imperfections of the Kalman filter used (see paragraph I.E), Type I targets are the only ones that can really be identified by analyzing $\Delta c_i(t)$. They can be recognized by a strictly negative $\Delta c_i(t)$. Moreover, since these values are computed with the Kalman filter, a precise noise analysis is not necessary. Hence, only one fuzzy set is defined for $\Delta c_i(t)$. This set is called DCNEG which stands for negative values of $\Delta c_i(t)$. It is represented in Figure 33.

In order to precisely control the evolution of each danger level, seven output (crisp) sets have been defined. Their labels are Very Large NegatiVe (VLNV), Large NegatiVe (LNV), NegatiVe (NV), ZeRo (ZR), PositiVe (PV), Large PositiVe (LPV) and Very Large PositiVe (VLPV). The value associated with each of them is respectively: -3, -2, -1, 0, 1, 2 and 3.

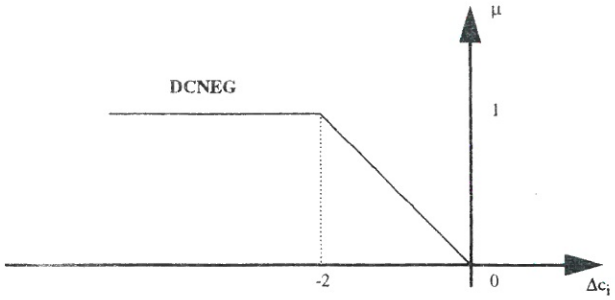

Figure 33. Fuzzy set associated with $\Delta c_i(t)$. This set identifies Type I targets.

The fuzzy algorithm is defined by the following rules. For each rule, the inputs are combined either with the minimum operator (and) or the maximum operator (or). The different rules are combined with the maximum operator.

R 1: If $\mu_{estimed\ i}(t)$ is not **ACCP** then *Δlevel$_i$(t)* is **VLNV**

R 2: If $\mu_{estimed\ i}(t)$ is **ACCP** and ($d\mu_{estimed\ i}(t)/dt$ is **MUPOS** or $d\mu_{estimed\ i}(t)/dt$ is **MUNEG**) then *Δlevel$_i$(t)* is **VLNV**

R 3: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and ($\Delta c_i(t)$ is **DCNEG** or $\delta o_i(t)$ is **LN**) then *Δlevel$_i$(t)* is **LNV**

R 4: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and $\Delta c_i(t)$ is not **DCNEG** and $\delta o_i(t)$ is **Z** and $d\delta o_i(t)/dt$ is **VLNGV** then *Δlevel$_i$(t)* is **ZR**

R 5: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and $\Delta c_i(t)$ is not **DCNEG** and $\delta o_i(t)$ is **Z** and $d\delta o_i(t)/dt$ is **LNGV** then *Δlevel$_i$(t)* is **PV**

R 6: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and $\Delta c_i(t)$ is not **DCNEG** and $\delta o_i(t)$ is **Z** and $d\delta o_i(t)/dt$ is **NGV** then *Δlevel$_i$(t)* is **LPV**

R 7: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and $\Delta c_i(t)$ is not **DCNEG** and $\delta o_i(t)$ is **Z** and $d\delta o_i(t)/dt$ is **NL** then *Δlevel$_i$(t)* is **VLPV**

R 8: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and $\Delta c_i(t)$ is not **DCNEG** and $\delta o_i(t)$ is **Z** and $d\delta o_i(t)/dt$ is **PSV** then *Δlevel$_i$(t)* is **ZR**

R 9: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and $\Delta c_i(t)$ is not **DCNEG** and $\delta o_i(t)$ is **Z** and $d\delta o_i(t)/dt$ is **LPSV** then *Δlevel$_i$(t)* is **NV**

R10: If $\mu_{estimed\ i}(t)$ is **ACCP** and $d\mu_{estimed\ i}(t)/dt$ is **MUNLL** and $\delta o_i(t)$ is not **Z** and $\delta o_i(t)$ is not **LN** and Δc_i is not **DCNEG** then *Δlevel$_i$(t)* is **ZR**

In the previous rule based algorithm, R1 allows the danger level of a given target to rapidly decrease if the estimated value of the proportional navigation coefficient is not in the correct interval. R2 checks the constant criterion of $\mu_{estimated_i}(t)$. R3 ensures that both $\delta_{O_i}(t)$ and $\Delta c_i(t)$ of a given target T_i characterize neither a Type I target (if $\Delta c_i(t)$ is DCNEG) nor a Type II target (if $\delta_{O_i}(t)$ is LN).

Rules R4 through R9 limit the maximum increase of any danger level according to the number of criteria corresponding to the real target. R7 represents the real target.

R10 is a kind of a watch dog. It implies a constant danger level if the evolutions of different parameters cannot identify a target type (Type I, II or real target).

The previous rules and the decision table presented for algorithm III makes it possible to derive the theoretical evolutions of this new target estimation algorithm. The different evolutions of the danger levels can be separated into three different zones.

The first one (labelled ZONE I) is defined from initial time (t = 0) to t = 2 seconds. It corresponds to a constant value of each danger level since the least -squares algorithms used in algorithm III are not initialized.

ZONE II is defined between time t= 2 s and t = 5.05. It corresponds to the time when only algorithm III is active, the Kalman filter being not yet initialized.

ZONE III is defined for t>5.05. The Kalman filter is completely initialized. The algorithm is the one using rules R1 through R10. It corresponds to a very decreasing evolution of the danger level associated with non target points.

These three zones are represented in Figure 34.

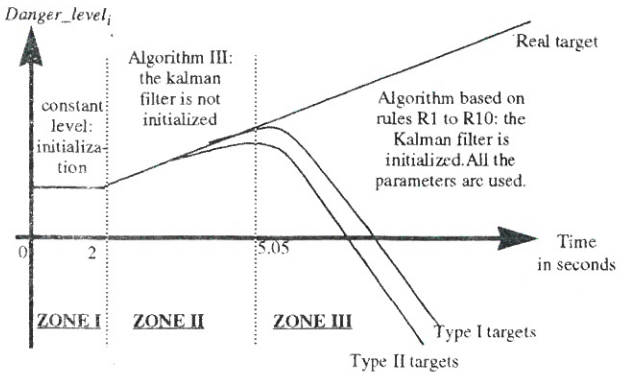

Figure 34. Estimated evolutions of Danger_Level $_i$(t) for noiseless input data: algorithm IV

II.B.3 Simulation Results

Figure 35 shows the evolution obtained by simulation. In this figure, it is possible to identify the three zones presented above. However, the boundary between Zone II and Zone III is a discontinuous one. This is due to the fact that when the Kalman filter is initialized, the danger levels are recomputed from the first estimation provided by the filter [4], that is from time t =2.5 s (50th sample). The algorithm using all the parameters being more accurate, the danger levels are lower for non target points implying a discontinuity in the Figure.

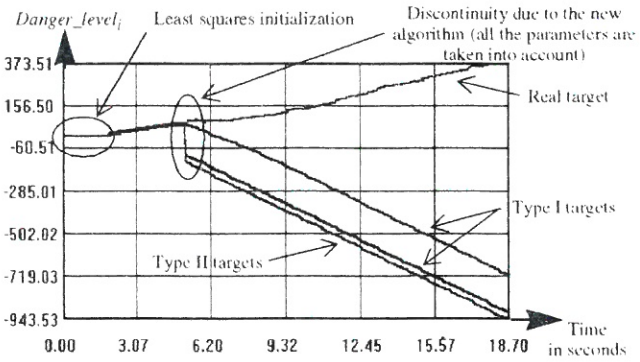

Figure 35. Simulated danger level evolutions obtained with algorithm IV. The initial conditions are as described in the simulation scenario with $\eta_1(t_0) = 45^o$.

Figure 36 and 37 give respectively the target estimation distance and the error ratio as a function of $\eta_1(t_0)$.

Table VIII summarizes the results obtained with this fourth target estimation algorithm. The simulation conditions are identical to those presented before.

II.B.4 Result Analysis

By comparing these new results with those obtained with algorithm III (see Table VII), it can be noticed that the general performance of algorithm IV is close to that of algorithm III. This can easily be explained by the fact that the estimation distances and the error ratios are established in most cases at the beginning of the simulations when algorithm IV is in fact equivalent to algorithm III.

The real advantages of the new algorithm (in terms of estimation distances and error ratios) are mainly visible when the estimation cannot be made at the beginning of simulation: in what has been called the critical zone.

In order to verify the previous statement, simulations have been performed for algorithms III and IV around the initial value of η derived in paragraph II.A.1.a.5, that is $\eta_{initial} = 80^o$ 4. These results are presented in Figure 38 and Table IX. Their analysis shows that the use of an algorithm based on all the estimation parameters has reduced the critical zone.

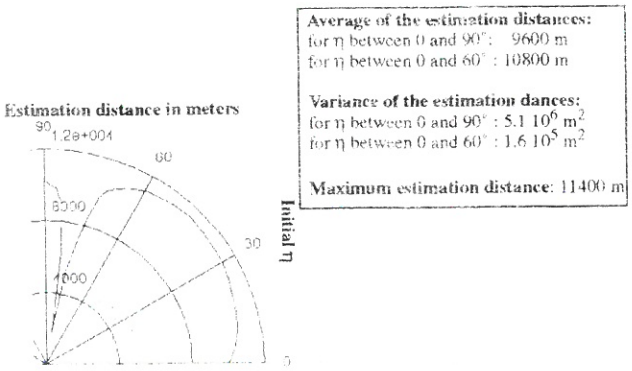

Figure 36. Estimation distance of the real target as a function of $\eta_1(t_0)$ for algorithm IV. The initial conditions are those described in the simulation scenario.

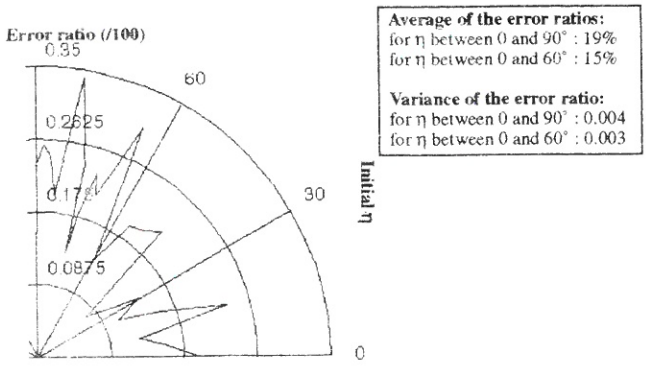

Figure 37. Target estimation error ratio as a function of $\eta_1(t_0)$ for algorithm IV. The initial conditions are those described in the simulation scenario.

Table VIII Performance of the fourth target estimation algorithm based on all the estimation parameters

Result Type	Average	Variance
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 90°	9600 m	$5.1\ 10^6\ m^2$
Target estimation distance for $\eta_1(t_0)$ varying between 0 and 60°	10800 m	$1.6\ 10^5\ m^2$
Error ratio for $\eta_1(t_0)$ varying between 0 and 90°	19%	0.004
Error ratio for $\eta_1(t_0)$ varying between 0 and 60°	15%	0.003
Best estimation distance	11400 m	

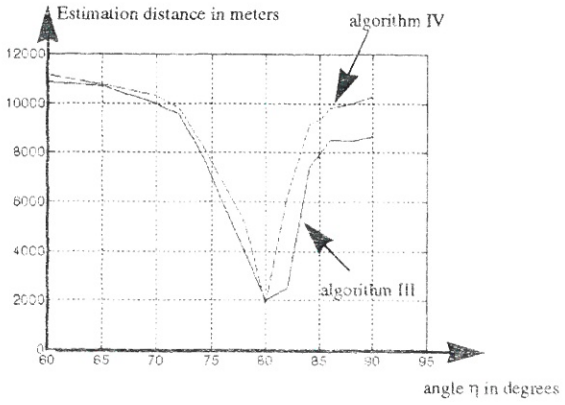

Figure 38. Estimation Distance in the Critical Zone for Algorithms III and IV

Table IX Performance of the third and fourth target estimation algorithms in the critical zone

Result Type	Algorithm III		Algorithm IV	
	Average	Variance	Average	Variance
Target estimation distance for $\eta_1(t_0)$ varying between 60 and 90°	7700 m	7.10^6 m^2	8400 m	$6.4\ 10^6$ m^2
Error ratio for $\eta_1(t_0)$ varying between 60 and 90°	22%	0.009	23%	0.003

III. Conclusion

This paper has presented the different steps followed to create a target estimation algorithm based on a fuzzy algorithm. The use of such a model has made it possible to analyze each estimation parameter separately. The fusion of these different analyses is then easily made through a rule based system.

The estimation is computed by using a danger level variable associated with each possible target. The fuzzy algorithm infers a variation of these danger levels, presenting a dynamic evolution of the estimation.

A precise control of the danger level variables has been obtained by combining all the information available at a given time. Thus, the final algorithm is divided into two phases according to the parameters already available.

The result of this study can be defined as a powerful and reliable estimation algorithm. Moreover, the use of fuzzy logic has proved to be an interesting and original way to deal with the problem of target estimation.

REFERENCES

1. BOULET, V., DRUON, E., WILLAEYS, D., and VANHEEGHE, P., **Target Estimation Using Fuzzy Logic**, Proceedings of 1993 IEEE Int. Conf. on Systems, Man, and Cybernetics, Le Touquet, October 1993, pp.674-679.
2. BOULET, V., **Contribution à la détermination de points cibles de mobiles en déplacement sous contraintes**, Ph.D Thesis, Université de Lille I, September 1995.
3. CARPENTIER , R., **Guidage des avions et missiles aérodynamiques**, Cours de l'Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Vol. I and Vol. II, 1989.
4. DRUON, E., **Application de la logique floue à la détermination de points de rencontre entre mobiles sous contraintes**, Ph.D Thesis, Université de Valenciennes et du Hainaut-Cambrésis, September 1995.
5. DRUON , E., BOULET, V., DUFLOS, E., WILLAEYS, D. and VANHEEGHE,P., **Comparison of Different Fuzzy Algorithms in a Function Behavior Recognition Problem**, Proceedings of 1995 IEEE Int. Conf. on Systems, Man and Cybernetics, Vancouver, October 1995, pp. 4107-4112.

6. DUFLOS, E., **Contribution à la modélisation des lois de guidage: caractérisation d'un point cible prédéfini ou non,** Ph.D Thesis, Université de Toulon et du Var, September 1995.

7. GERALD, C.F. and WHEATLEY, P.O., **Applied Numerical Analysis,** Fourth Edition, ADDISON-WESLEY PUBLISHING COMPANY, 1992.

8. GUELMAN, M., **A Qualitative Study of Proportional Navigation,** IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, Vol. 7, No. 4, July 1971, pp. 637-643.

9. KICKERT, W.J.M. and MAMDANI, E.H., **Analysis of A Fuzzy Logic Controller,** FUZZY SETS AND SYSTEMS , Vol. 1, 1978, pp. 29-44.

10. LEE, C.C., **Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I,** IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, Vol. 20, March/April 1990, pp. 404-418.

11. LEE, C.C., **Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II,** IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, Vol. 20, March/April 1990, pp.419-435.

12. PROCYK, T.J. and MAMDANI, E.H., **A Linguistic Self-Organizing Process Controller,** Automatica, PERGAMON PRESS LTD, Oxford, Vol. 15, 1979, pp. 15-30.

13. SHUKLA, U.S. and MAHAPATRA, P.R., **Generalized Linear Solution of Proportional Navigation,** IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, Vol. 24, No. 3, March 1988, pp.231-238.

14. SHUKLA, U.S., and MAHAPATRA, P.R., **The Proportional Navigation - Pure or True,** IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, Vol. 26, No. 2, March 1990, pp. 382-392.

15. TERANO, T., ASAI, K. and SUGENO, M., **Fuzzy Systems Theory and Its Applications,** ACADEMIC PRESS, 1987.

16. ZADEH, L.A., **Fuzzy Sets,** INFORMATION AND CONTROL, Vol. 8, June 1965, pp.338-353.

17. ZARCHAN, **Tactical and Strategic Missiles Guidance,** Vol. 157, Second Edition, AIAA, 1994.