Proceedings of the 7th WSEAS International Conference on Automation & Information, Cavtat, Croatia, June 13-15, 2006 (pp161-166)

Clustering-based Control of Discrimination Power in Target Tracking Applications

ABDER REZAK BENASKEUR, FRANÇOIS RHÉAUME, & PATRICK MAUPIN
Decision Support Systems Section
Defence Research and Development Canada - Valcartier
2459 Pie XI North, Québec (Qc) G3J 1X5
CANADA
www.valcartier.drdc-rddc.gc.ca

Abstract: - The paper addresses the problem of the target-tracking in military surveillance operations. More specifically, a closed-loop approach to adapt the sensing and tracking operations is proposed and compared to the conventional open-loop approach. The objective is to control and maintain, over a certain area, the level of discrimination power required by the mission objectives. The control strategy is based on two cascade loops. The outer loop uses clustering techniques to characterize the area in terms of discrimination power. This high level information is exploited by the inner loop to compute optimal track update and sensor scheduling strategy.

Key-Words: - Data Fusion, Target Tracking, Military Surveillance, Discrimination Power Control, Clustering.

1 Introduction

In military Command and Control (C^2) applications, target tracking uses data fusion technology to provide accurate and timely identification, classification, and kinematics information about the entities within the area of interest. To improve the process of data fusion, and make it cope with a larger class of problems, the processing and the resources it uses must be constantly managed and coordinated [1–3]. This defines the discipline of adaptive data fusion. The adaptation closes the loop over the fusion sensing and processing capabilities and develops options for collecting further information or tuning the processing/sensing parameters for the real-time improvement of the process effectiveness.

This paper presents part of the research activities, conducted at Defence R&D Canada, that aim at defining, developing, and demonstrating adaptation concepts. More specifically, the paper addresses the problem of adapting the target-tracking operations and attempts to demonstrate the benefits of such adaptation compared to the open-loop operation mode. The adaptation is achieved thanks to a two-level cascade loop, the objective of which is controlling the discrimination power over a certain area of interest.

In situations involving multiple targets, the latter may come too close to be clearly distinguishable from each other by the surveillance system. Targets are said to be distinguishable when the overlap between their localization probability distributions is below a specific level called the *disparity level*. The concepts of disparity is applied here to the problem of multiple target tracking, and a quantity called the *discrimination power* is used to adjust the scan rate of the surveillance system. The discrimination power measures the distance between the least disparate among the n pairs of target tracks at a given instant. Operations such as contact/track correlation, target identification and classification are very sensitive to the disparity between tracks. This may also be an issue in the target engagement operations. Given the criticality of the latter situation, a high discrimination power is often required to maximize the chance of threat neutralization and minimize the risk of collateral damages. In this paper, a solution is presented that: i) characterizes the whole area of interest based on the concept of discrimination power, where distinct regions are created and managed dynamically based on the targets spatial structure using clustering and classification techniques; and ii) controls the discrimination power over that area of interest, by an appropriate selection of the track update frequency and sensor scheduling within each distinct region.

The remaining part of the paper is organized as follows. Section 2 defines the (adaptive) data fusion problem. The specific problem of track-tracking is dis-

Proceedings of the 7th WSEAS International Conference on Automation & Information, Cavtat, Croatia, June 13-15, 2006 (pp161-166)

cussed in Section 3. Section 4 introduces the concept of discrimination power control and applies it in the context of target tracking. The results of this application are presented and discussed in Section 5. Section 6 gives some concluding remarks.

2 Adaptive Data Fusion Problem

Data fusion aims at supporting the decision maker in improving his situation awareness. To achieve higher performance, a modern data fusion system needs an active feedback or adaptation. The adaptation may concern the data fusion process itself or the related sensor management problem.

Adaptation in the specific context of target tracking aims at producing a system that can readily adapt to changing operating environment and needs. An adaptive tracking system must be able to detect variations in its performance index and respond to them by performing structural changes. The performance measurement and control process can be performed recursively at different levels of abstraction, where the loop of level n sets the objectives for the loop of level $n-1$. The latter selects the appropriate actions to achieve those objectives. Associated with each adaptation loop is a performance measure that provides the necessary feedback from the environment. Fig. 1 shows an example of the two-level cascade loop.

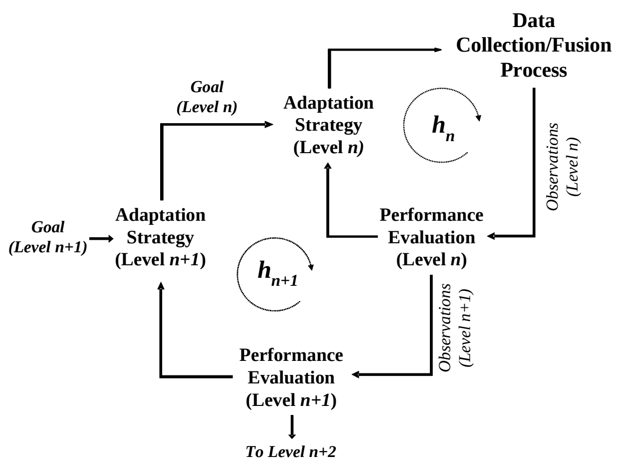

Fig. 1: Two-level Cascade Control Loop

3 Target Tracking Problem

In this section, the control structure of Fig. 1 is applied to the specific problem of target-tacking, as depicted in Fig. 2. The two loops operate at two different time scales, with the following objectives

1. **Outer-loop**: characterizes of the whole area of interest based on the concept of discrimination power, where distinct regions are created and managed dynamically based on the targets spatial distribution and by using clustering techniques. To ensure a certain level of cluster persistence, this loop operates at low frequency, as defined by the basic period $h_c > h_s$ in Fig. 2. Thus sensor scheduling can be made by considering several sampling intervals.

2. **Inner-loop**: adaptively controls of the discrimination power over the area of interest, by an appropriate selection of the track update frequency and sensor scheduling within each distinct sub-area. This adaptation loop operates at a faster time scale than the outer loop $(h_s < h_c)$. Note that the minimal value for h_s is a characteristic of the sensors.

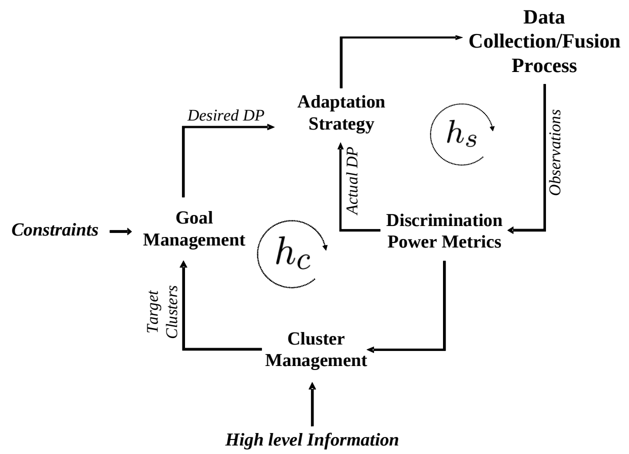

Fig. 2: Clustering/Discrimination Power-based Two-Level Control Loop

3.1 Scenario

A scenario where a single (phased array like) sensor [4] has to track several (N) targets is considered. The latter has the capability to switch the direction of its beam very quickly (assumed instantaneously for the problem being addressed) without inertia. The goal of the control is then to maintain the appropriate disparity level between the tracks based on the region they belong to. The control strategy should make the sensor spend more time over critical regions, *e.g.* where engagements involving hardkill weapons are planned.

3.2 Tracking Algorithm

The following give the equations of the dynamical model and the Extended Kalman Filter (EKF) [5] used by the underlying tracking algorithm. The discrete-time dynamical model of the targets is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{\Gamma}\boldsymbol{v}_k \tag{1}$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{h}(\boldsymbol{x}_{k+1}) + \boldsymbol{w}_{k+1} \tag{2}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{v}$ is the process noise with covariance matrix $\boldsymbol{Q}$, and $\boldsymbol{w}$ is the measurement noise, whose covariance matrix will be denoted $\boldsymbol{R}$. $\boldsymbol{\Gamma}$ is the discretized continuous time process noise transition matrix. The time-update equations for the EKF algorithm are given by

$$\hat{x}_{k+1|k} = \boldsymbol{F}_k \hat{x}_k \tag{3}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_k \boldsymbol{P}_k \boldsymbol{F}_k^T + \boldsymbol{\Gamma}\boldsymbol{Q}\boldsymbol{\Gamma}^T \tag{4}$$

where $\boldsymbol{F}_k$ is the Jacobian of $\boldsymbol{f}$. The above-given predicted values are used to calculate the updated version of the state and the corresponding covariance matrix.

$$\boldsymbol{P}^{-1}_{k+1|k+1} = \boldsymbol{P}^{-1}_{k+1|k} + \boldsymbol{P}_z^{-1} \tag{5}$$

$$\boldsymbol{P}^{-1}_{k+1|k+1}\hat{\boldsymbol{x}}_{k+1|k+1} = \boldsymbol{P}^{-1}_{k+1|k}\hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{P}_z^{-1}\boldsymbol{z}_{k+1} \tag{6}$$

with

$$\boldsymbol{P}_z^{-1} = \boldsymbol{H}^T_{k+1}\boldsymbol{R}^{-1}H_{k+1} \tag{7}$$

and where $\boldsymbol{H}_k$ is the Jacobian of $\boldsymbol{h}_k$, and k represents the discretized time.

4 Discrimination Power Control

The discrimination power is a metric of the disparity between multiple target tracks distributed in the space. The disparity between two tracks is defined as a statistical distance between the tracks' latest state estimates and is based on the probability distribution of each estimate. Track disparity expresses how much tracks are dissimilar. Fig. 3 illustrates pairs of tracks' estimates with different levels of disparity, represented by a statistical distance d. The ellipsoids represent contours of constant probability for the two-dimension and normally distributed position estimates of the tracks. As can be seen in Fig. 3, the farther the position estimates $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ are from each other in the space, the higher is the track disparity ($d_1 > d_2$). Also, the smaller the estimation error covariance matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are, the higher is the track disparity ($d_2 > d_3$).

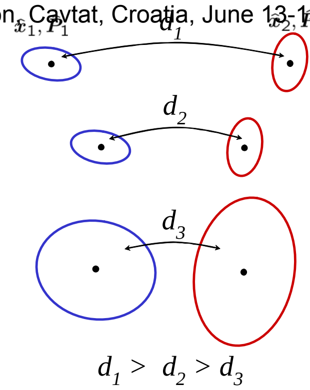

Fig. 3: Statistical distance d as a track disparity metric

The evaluation of the discrimination power requires calculating the statistical distance for each pair of tracks. A statistical distance d between two estimates $\hat{\boldsymbol{x}}_i$ of track i and $\hat{\boldsymbol{x}}_j$ of track j is provided by the Mahalanobis distance [6], which considers both the estimates and their corresponding covariance matrices.

$$d(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j) = \left[\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\right]^T \left[\boldsymbol{P}_i + \boldsymbol{P}_j\right]^{-1} \left[\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\right] \tag{8}$$

where $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ are the covariance matrices for the state estimates $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$ respectively. The distances for each possible pair of tracks are represented in the $N \times N$ proximity matrix $\mathcal{D} = [d(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j)]$.

4.1 Track Update Control

A solution to control the track disparity (and the related discrimination power) consists of adjusting adaptively the time intervals between updates for the different tracks. The time interval between measure updates will also be referred to as the update period, which is the inverse of the update frequency. It has a significant influence over the variation in time of the track accuracy (*i.e.*, covariance matrix). Shorter periods between updates should yield higher track accuracy and therefore higher disparity between tracks. Therefore, the goal of an adaptive tracking system is to choose appropriate update periods for each track in order to keep a certain level of discrimination power within the area of interest.

A straightforward discrimination power control solution is to have each track's own update period h be proportional to the distance with its nearest neighbor, according to the Mahalanobis distance. A track i would have an update period h_i determined such that

$$h_i = \zeta\left[\min_k d(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_k)\right], \quad k \in 1, ..., N \tag{9}$$

where $d(\hat{x}_i, \hat{x}_j)$ is as defined by Eq. 8 and ξ is a function that expresses the relation between h and d.

4.2 Hierarchical clustering of tracks

When regions of the area of interest are to be considered for controlling sensor scans, separating tracks into clusters can provide a way to determine the desired update strategy. Clusters of targets represent different regions of the space with different discrimination power levels. Agglomerative hierarchical clustering [6] is suitable for controlling the discrimination power using the region-based approach. With agglomerative hierarchical clustering, a binary tree is constructed based on a linkage measure, where each leaf represents the predicted state estimate of a track. The linkage measure is determined and related to the type of hierarchical clustering algorithm.

A hierarchical cluster tree is created based on the $N \times N$ proximity matrix $\mathcal{D} = [d(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j)]$. The process of hierarchical clustering is as follows

1. Assign each track to a cluster to have N clusters.
2. Merge the closest pair of clusters where cluster C_i and C_j are merged together to result in $N-1$ clusters.
3. Compute distances between the new cluster and each of the old clusters[1].
4. Repeat Steps 2 and 3 until all tracks are clustered into a single cluster of size N.

Fig. 4 shows an example of a cluster tree resulting from the single-linkage algorithm. The tree is binary with $N-1$ nodes excluding the leaves representing tracks. The highest nodes in the hierarchy have the largest distance values. A threshold L_k applied on the distances allows to group tracks into regions. The value of the threshold determines how the tree is separated. All nodes that are below the threshold will have their corresponding tracks regrouped into the same cluster. Each cluster of tracks occupies a particular region of the space. The value of the threshold should depend on the problem at hand, that is the number of targets and their properties, the number of sensors and their properties and most importantly the applications that makes use of target tracking (e.g. surveillance, target engagement).

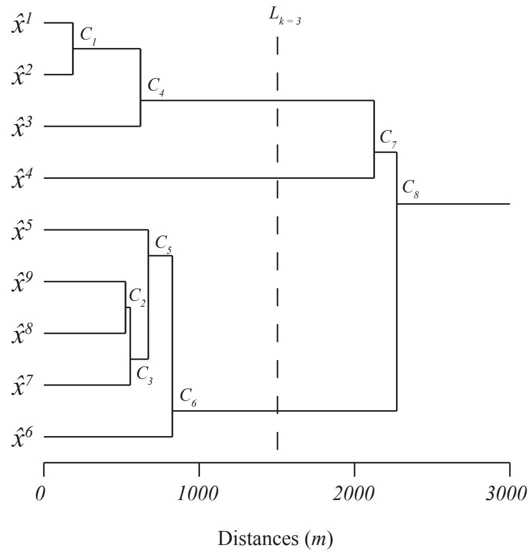

Fig. 4: Agglomerative clustering of tracks

4.3 Sensor Scheduling Heuristics

Let R_n be the region occupied by cluster C_n, for which the discrimination power d_n and the track similarity s_n are defined as follows

$$d_n^{-1} = s_n = \min_{i,j} d(\hat{\boldsymbol{x}}_k^i, \hat{\boldsymbol{x}}_k^j), \quad \hat{\boldsymbol{x}}^i, \hat{\boldsymbol{x}}^j \in R_n \tag{10}$$

The two metrics d_n and s_n will be used to define the optimal track update frequency and the scheduling strategy. The update cycle length of the clusters is defined as

$$h_c = h_s \xi \psi = h_s \xi \sum_{n=1}^{K} \eta_n \tag{11}$$

$$= h_s \xi \sum_{n=1}^{K} \lfloor \frac{d_n^{-1}}{\min_{i \in \{1,\dots,K\}} d_i^{-1}} \rfloor \tag{12}$$

$$= h_s \xi \sum_{n=1}^{K} \lfloor \frac{s_n}{\min_{i \in \{1,\dots,K\}} s_i} \rfloor \tag{13}$$

where K is the number of clusters and ξ is an integer defined arbitrarily. The integers η_n define the number of updates that will be performed for each region within a single update cycle ψ. The latter is defined

[1] The distance computation depends on the type of clustering used. The following gives, for two clusters C_i and C_j, the resulting distances $d_{min}(C_i, C_j)$ and $d_{max}(C_i, C_j)$, when single-linkage and complete-linkage are used respectively

$$d_{min}(C_i, C_j) = \min_{\hat{x} \in C_i, \hat{x}' \in C_j} d(\hat{x}, \hat{x}')$$

$$d_{max}(C_i, C_j) = \max_{\hat{x} \in C_i, \hat{x}' \in C_j} d(\hat{x}, \hat{x}')$$

such that the tracks within the cluster that has the highest discrimination power (*i.e.*, smallest s_n) will be updated only once, while the tracks within the other clusters will be more than once.

It is assumed that the sensor has to spend a minimum time t_s over a region to report a contact. This defines a maximum report frequency f_s. The update frequency for each cluster, as function of the sensor frequency f_s, is defined as follows,

$$f_n = \xi^{-1}\psi^{-1}\eta_n f_s$$

For regions that require more than one update, different update scheduling strategies may be possible. Since it is assumed that the sensor can be directed instantaneously, without extra cost, a scheduling strategy that maximizes the quality (*i.e.*, minimizes the uncertainty ellipsoid) of the obtained tracks is adapted. This strategy is described by Algorithm 1, where the objective is to separate (in time) the updates of the same region as much as possible in order to maintain a best quality of track with the same number of updates.

$S \leftarrow$ schedule vector of size ψ (update cycle length)
$j \leftarrow k \mid \eta_k = \max_{l\in\{1,\dots,K\}} \eta_l$
$i \leftarrow 1$
while $j > 0$ **do**
  $S(i) \leftarrow j$
  $\eta_k \leftarrow \eta_k - 1$
  if $j > 1$ **then**
    $S(\psi + i - 1) \leftarrow j$
    $\eta_k \leftarrow \eta_k - 1$
  end if
  $i \leftarrow i + 1$
  $j \leftarrow k \mid \eta_k = \max_{l\in\{1,\dots,K\}} \eta_l$
end while

Algorithm 1: Sensor Scheduling Strategy

5 Results and Discussion

An application example with seven (7) targets has been coded and the results are presented in this section. Within the considered configuration, certain targets will come too close, at given time instants, to be clearly distinguishable from each other, *e.g.* **Target 1** and **Target 7** at $5s$, and **Target 1** and **Target 6** at $14s$, as shown on Table 1 where the closest pairs of targets are given in terms of different simulation times.

Table 1: Closest pairs of targets in scenario.

Time (s)	Closest pair	Distance (m)
0	1 - 7	1300
5	1 - 7	10
10	1 - 6	1400
14	1 - 6	5
20	1 - 5	1100

Note that any well-defined distance on $\mathbb{R}^n$ may be used as a proximity metric. The presented results and the underlying development are based on the Mahalanobis distance.

Figures 5 and 6 present and compare the resulting track quality based on two different track update strategies, for a tracking duration of $20sec$. Dashed ellipses give initial position and uncertainty for each target. Plain ellipses are represented only to show final position and uncertainty for each target, and also where discrimination problems are expected, *e.g.*, intersections (**Target 1**, **Target 6**) and (**Target 1**, **Target 7**).

In Fig. 5 a static periodic update strategy was used, where the different tracks are being allotted the same attention without consideration of any additional information. On the contrary, Fig. 6 presents adaptive update strategy whose objective is the control of the discrimination over all the area of interest. More attention is allotted to critical regions, *i.e.*, where discrimination power tends towards zero. Clusters are created dynamically based-on spatial proximity, and the update strategy is defined based on the track similarity in each cluster.

Fig. 6 shows clearly the superiority of the adaptive approach over the static update strategy. The discrimination power is adaptively improved where required, *e.g.*, over intersections (**Target 1**, **Target 6**) and (**Target 1**, **Target 7**), by increasing the update rate over the clusters created by the targets proximity. Targets that do not need high discrimination power will belong to distinct clusters that will be updated less frequently, *e.g.*, **Target 3** and **Target 4**. For the least frequently updated tracks by the adaptive strategy (*i.e.*, **Target 3** and **Target 4**), the obtained results show $1.32km^2$ vs. $1.0km^2$ with static for **Target 3** and $1.11km^2$ vs. $.86km^2$ with static for **Target 4**.

6 Conclusion

The target tracking application is used to illustrate control and adaptation concepts in data fusion applica-

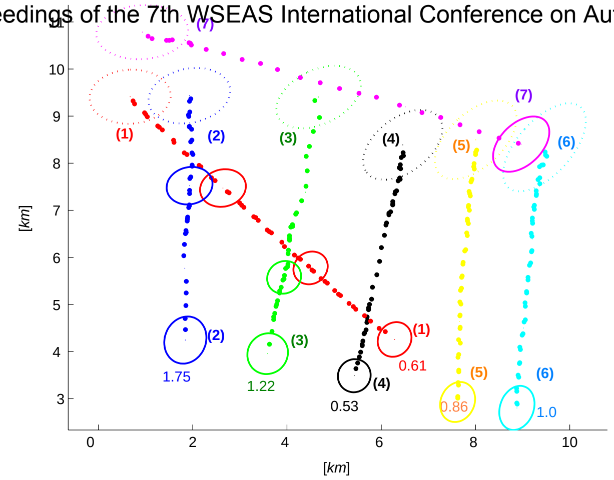

Fig. 5: Track quality yielded by the static periodic update strategy. Probability distribution contours are still overlapped few seconds after targets have crossed.

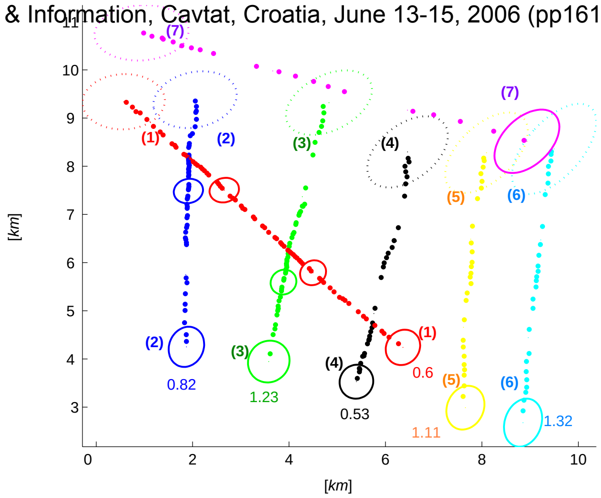

Fig. 6: Track quality yielded by the discrimination power control-based strategy. Probability distribution contours do not overlap few seconds after targets have crossed.

tions. A two-loop adaptation structure is defined to tackle the specific adaptive tracking problem. The outer loop dynamically defines and manages the clusters based on the concept of discrimination power, while the inner loop exploits the clusters for the adaptation purposes. The presented results showed the superiority of the adaptation-based strategy over a static policy. Extension of the proposed approach to the algorithmic adaptation, where the fusion processing is controlled instead of the sensor revisit rate, is straightforward. The performance of the different tracking algorithms replaces the performance of the sensors, and algorithms are dynamically allocated and scheduled, instead of sensors, to increase the discrimination power within the clusters. A more general approach, that combines both adaptation problems, is being addressed. Therein, the adaptation module selects the most appropriate strategy for both sensing and tracking. Further improvement is being achieved through the use of the Kinetic Data Structures (KDS) [7]. The latter allows the management of the clusters and the computation of the associated of discrimination power. KDS provide an efficient means for the computation of the different features that can be used in adaptive fusion problems.

References

[1] El-Fallah, A. I., Ravichandran, R. B., Mehra, R. K., Hoffman, J. R., Zajic, T., Stelzig, C., Mahler, R. P., and Alford, M. G. (2001). Scientific performance evaluation for distributed sensor management and adaptive data fusion. In *Proc. SPIE Vol. 4380, p. 328-338, Signal Processing, Sensor Fusion, and Target Recognition X, Ivan Kadar; Ed.*, pp. 328–338.

[2] Benaskeur, Abder Rezak, Rhéaume, François, and Bossé, Éloi (2006). Adaptation Hierarchy for Data Fusion and Sensor Management Applications. In *Proc. COGnitive systems with Interactive Sensors, Paris (France)*.

[3] Steinberg, A., Bowman, C., and White, F (1998). Revision to the JDL data fusion model. In *Joint NATO/IRIS Conference*, Quebec City.

[4] Alan J. Fenn, William P. Delaney, Donald H. Temme and Courtney, William E. (2000). The Development of Phased-Array Radar Technology. *Lincoln Laboratory Journal*, **12**(2).

[5] Anderson, B. D. O. and Moore, J. B. (1979). Optimal Filtering, NJ: Prentice-Hall.

[6] Jain, A. K. and Dubes, R. C. (1988). Algorithms for Clustering Data, NJ: Prentice-Hall.

[7] J. Gao, An Nguyen, L. J. Guibas (2005). Distributed Proximity Maintenance in Ad Hoc Mobile Networks. In *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'05)*.