Corchado, J. M., et al. (Eds.) (2014). *17th International Conference on Information Fusion (FUSION 2014): Salamanca, Spain 7-10 July 2014*. IEEE.

Geographic context configuration in fusion algorithms for maritime surveillance

Enrique Martí,
GIAA-Universidad
Carlos III de
Madrid
Colmenarejo, Spain
emarti@inf.uc3m.es

Borja González
GIAA-Universidad
Carlos III de Madrid
Colmenarejo, Spain
borjagps@gmail.com

Alvaro Luis,
Jesús García
GIAA-Universidad
Carlos III de Madrid
Colmenarejo, Spain
{aluis,
jgherrer}@inf.uc3m.es

José M. Molina
GIAA-Universidad
Carlos III de
Madrid
Colmenarejo, Spain
molina@ia.uc3m.es

Irene López García-
ELECNOR DEIMOS
DEIMOS Space, S.L.U.
Spain
ilopez@grupodeimos.com

***Abstract*—Real fusion system applications can be required to operate on wide areas for long periods of time. Adaptation is a basic capability under these circumstances. This paper presents a maritime surveillance platform designed to be flexible and robust. It features online configuration capabilities allowing to: (a) change the applied algorithms, (b) modify the operating parameters of the running algorithms, (c) tune the characterization of the available sensors. These configurations can be applied to limited spatial regions and time spans. This allows to use powerful or more specific configurations for localized scenarios (risks, clutter, alarms), or account for exceptional situations that can affect sensors, such as weather anomalies.**

Keywords—adaptive fusion; context-aided fusion; maritime surveillance; quality metrics

I. INTRODUCTION

A. *Maritime surveillance needs*

The priorities for Vessel Traffic Services (VTS) are oriented to efficient vessel transits with the best possible order and safe navigation, improving upon current operational procedures on the waterway. The general goal of any support system is providing enhanced operator situational awareness to overcome the limitations of traditional methods (direct sight, littoral radar, voice communication, etc). Another expected functionality is the detection of abnormal situations accordingly to expected behaviours (smuggling, route deviations, intrusions in non-authorized areas, etc).

To this goal, there are many sources providing raw sensor data in maritime domain which must be integrated to provide real-time decision support to operators. Data from cooperative sources (AIS transponders in vessels) must be correlated with non-cooperative sensors, such as shore radar, high-frequency radars or video (optical/infrared/satellites). However, sensory data alone can be insufficient to understand complex scenarios like harbours, and additional contextual information would be necessary to build a correct and complete description of situation.

B. *Context and fusion for maritime surveillance. Previous works*

Context information contains the complementary knowledge to understand situations. Context, defined as those information pieces that accompanies the focal entities of interest [1], is a key knowledge to understand their current states, predicted or any other inference process within information fusion.

The use of context in fusion systems is a quite hot topic in information fusion research. The use of static context in algorithms can be done at design time with off-line configuration processes. Physical context can be seen as the most direct use of context, when this information is helpful to model the performance of entities and data sources in the environment. In maritime domain there are abundant examples of context exploitation [2-4], such as geographic knowledge of the coastline, currents, tides, bathymetry, weather, sea state and ice, etc, which enables better prediction of vessels behaviour. So, in con-tracker system [5,6], water depth in channels (calculated from tabulated tidal height plus bathymetric depth) affects the motion of vessels, together with marked channel information, maximum speeds, restricted areas, etc.

Another aspect where context can be very useful is in track management, a particular case of level-4 fusion adaptation. Context can be exploited to adapt and improve the sensor fusion process accordingly to the situation [7-9]. For instance, feedback strategies –i.e. commands flowing from contextual situation level to the data fusion node–can yield improvement in adverse conditions, such as high traffic or heavy clutter scenarios with small probability of target detection [10]. In this work, a system allowing the automatic tuning or selection of algorithms (multi-algorithm fusion) based on contextual configuration is shown. It has been used to allow fine design of fusion performance accordingly to context configuration and inclusion of on-line events directly interacting with the tracking and fusion algorithms. It features online configuration capabilities allowing to: (a) change the applied algorithms, (b) modify the operating parameters of the running algorithms, (c) tune the characterization of the available sensors. These configurations can be applied to limited spatial regions and time spans. This allows to use powerful or more specific configurations for localized scenarios (risks, alarms), or account for exceptional situations that can affect sensors, such as weather anomalies.

II. SYSTEM ARCHITECTURE

A. General structure

The basic idea is reflected in Fig. 1. The contextual configuration represents all the geographic areas with associated parameters and algorithms, to be activated accordingly to current state (with the necessary logic to guarantee seamless transitions during changes). Besides, the operator is able to modify on-line the active configuration considering for instance the quality metrics provided by system (explained later).

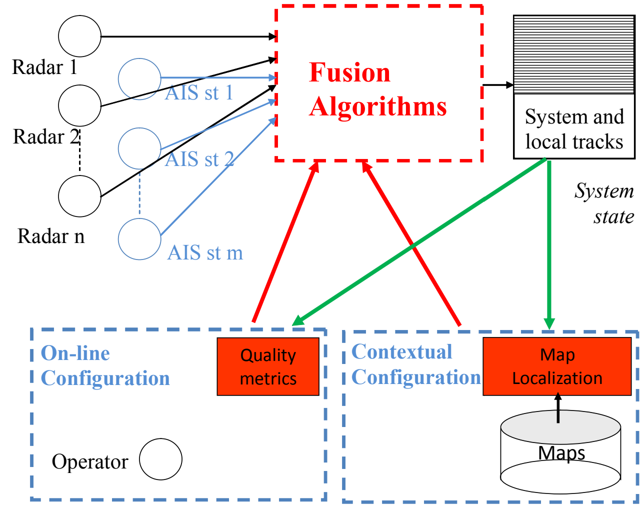

Fig. 1. General structure of configurable fusion

B. Fusion Algorithms

The system is structured as a distributed, event/message driven architecture: processing elements are abstracted as modules/boxes that can receive messages, compute over them, and emit new messages to other entities. In advance, we will call these entities “Message Processors”

An example instance of the architecture can be found in Fig. 2: sensors are abstracted as message processors that emit messages with the acquired data. Each sensor forwards data to a local tracker, whose output consists on messages describing the composition and evolution of a set of labelled tracks. All local trackers are connected with a global tracking module (another message processor) that fuses the different sources to produce the final result.

The figure describes some additional capabilities of this simple and powerful architecture. Some examples:

- Processing can be parallelized, distributed across several computers just by changing the messaging mechanism (network vs. in-memory). This is shown with the bottom box labelled as “Zonal Filter”, which describes a region of interest for exchanged messages. Only the AIS fixes inside that region (or outside it, depending on the configuration) will reach the tracker. This can be useful for ignoring crowded zones with no interest (a port), or for balancing computational load (instancing several trackers in separate machines and splitting the AIS fixes sent to them).
- The processing topology is completely dynamic: it is possible to add new features by just connecting additional message processors into the correct place. For example, logging policy can be easily modified by adding/removing the specialized message processors, or editing the connections between trackers and loggers. Also the process for extracting fusion quality metrics (see next sections) is implemented as message processors that receive messages from the trackers.

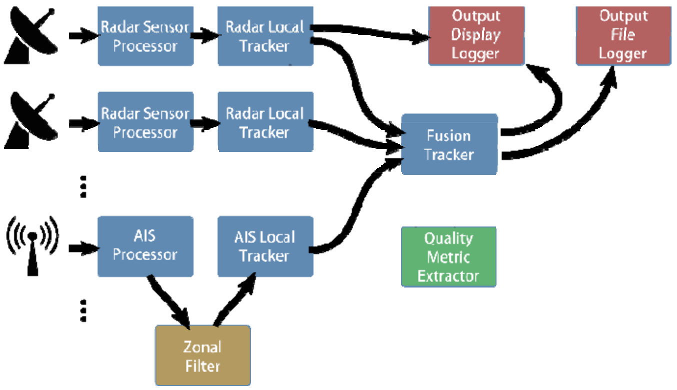

Fig. 2. Instance of the implemented fusion software, configured for a set of radar sensors and AIS stations

Individual trackers (both local –radar, AIS–, and global) are single message processors that execute an internal cyclic process with a predefined structure. This process has places where different algorithms are plugged. Thus, the configuration capabilities of the fusion system consist on modifying the plugged modules.

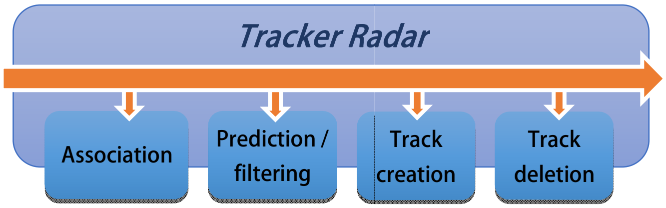

Fig. 3. Tracker structure, as a fixed processing scheme that relies on some configurable, plugged modules

Although the algorithms that have been implemented and tested so far are basic, the tracking process have some custom built-in processes and rules to make the final output stable and robust. The rules applied to make fusion stable include:

- Dominance of AIS MMSI codes for labelling the tracks. This applies even in the case of a transponder switched off and started again later: the track remembers previously associated MMSI.
- Sensor prioritization according to zonal quality indicators. Discrepancies between sensors or fusion events are resolved taking into account the reliability of the involved measures.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

- Limited memory of past actions to avoid cyclic switching behaviors.

Since the developed software is intended to serve as support system for human operators, the philosophy here is that sometimes a wrong but stable answer is preferable over a constantly changing situation: if two vessels cross their trajectories and the system switches their labels, it will not switch them back until some time has passed, and the evidences supporting the decision are strong enough.

III. GENERAL APPROACH: ZONAL CONFIGURATION OF ALGORITHMS AND PARAMETERS

The term "zonal configuration" defines a set of data structures that relates a set of algorithms and their parameters with spatial regions where they must be applied. Each tracker has its own configuration, which can be changed independently from the other trackers.

On its simplest form, the zonal configuration of a tracker is limited to a single default configuration applied to its whole operating area. TABLE I. describes an example configuration for a radar tracker:

TABLE I. SAMPLE BASE CONFIGURATION FOR RADAR LOCAL TRACKER

```
Config 'default'
|-gating:
  |-sigma_factor: [char] '3'
  |-min_box: [char] '50'
  |-type: [char] 'radarSimpleGating'
|-filter:
  |-max_accel: [char] '0.15'
  |-type: [char] 'kalman'
|-confirmation:
  |-N: [char] '7'
  |-M: [char] '8'
  |-type: [char] 'confirmationNOutOfM'
|-deletion:
  |-N: [char] '4'
  |-M: [char] '6'
  |-type: [char] 'deletionNOutOfM'
|-association:
  |-type: [char] 'hungarianAlgorithm'
```

This configuration specifies the technique/algorithm to be applied on each one of the configurable tracking steps, along with the *desired* parameter values.

Starting from this point, the implemented system allows to:

- Modify the value of any parameter. E.g.: change the N/M values of the deletion algorithm to improve behavior on high-clutter conditions.
- Change any of the selected technique/algorithm. E.g.: Switch from Kalman filtering algorithm to an Interactive Multiple Model (IMM) filter, with the appropriate logic to do the hand-over between algorithms (state/covariance initialization and other parameters depending on each algorithm).
- Add a new spatial region that will have a different configuration (and also remove existing regions, or change their algorithms/parameters).

Such events can take place at any time, and are applied as soon as possible while guaranteeing the integrity of the data and processes.

A. Representing configurations and zones

The zonal configuration of each individual tracker is based on the information defined in a mix of XML files (for algorithms/parameters), and KML files (for the zones). KML (Keyhole Markup Language) is an XML-based standard language for describing geographical or geolocalized data.

Every element receives an identifier that can be used for referencing it. These files are translated to a tree-like structure where the root represents the default configuration, and the leaves are regions of the space with a specific configuration.

The zonal configuration can be changed at any time through an API call that sends a message to the affected modules. Modifications in the zonal configuration are reflected as an update of the corresponding tree. The sent message indicates the affected trackers (referenced by its assigned sensor), which configuration/parameterization must be applied, and over which zone.

As a working example: assume TABLE I. represents the default configuration of a radar tracker associated to Radar 1 (see Fig. 7), as loaded from the XML configuration file. The same file defines a partial configuration called "smallGating", which affects the maximum distance between two tracks that can considered associable, keeping the same algorithm "radarSimpleGating":

TABLE II. ADDITIONAL PARTIAL CONFIGURATION FOR A RADAR LOCAL TRACKER

```
Config 'smallGating'
|-gating:
  |-sigma_factor: [char] '2'
  |-min_box: [char] '20'
  |-type: [char] 'radarSimpleGating'
```

A human operator monitoring Radar 1 may notice than the performance in the port, a crowded zone, is degraded. A possible solution is to apply the "smallGating" configuration to reduce association ambiguity. Thus, at time t=120 seconds, a configuration message with the following information is injected into the system:

```
Configuration event
|-sensor_id: 1
|-zone_id:   'Port'
|-config_id: 'smallGating'
```

As a result, the zonal configuration for this radar local tracker from t=120s in advance has the following morphology:

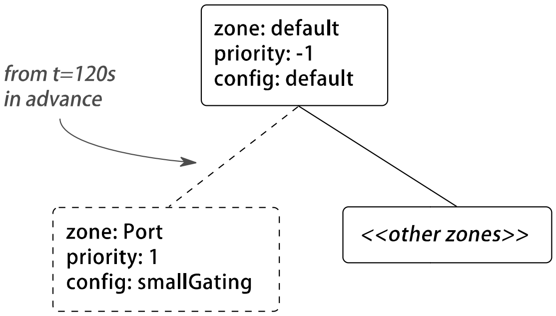

Fig. 4. Configuration transition

When the tracker extracts the configuration for Radar 1 inside the Port, the default configuration is overwritten with the zone specific elements, resulting in the structure shown in TABLE III.

TABLE III. COMBINED TRACKER CONFIGURATION FOR TRACKS/PLOTS INSIDE 'PORT' REGION

```
Tracker configuration in zone 'Port'
(combination: 'default'+'smallGating')
|-gating:
  |-sigma_factor: [char] '2'
  |-min_box: [char] '20'
  |-type: [char] 'radarSimpleGating'
|-filter:
  |-max_accel: [char] '0.15'
  |-type: [char] 'kalman'
|-confirmation:
  |-N: [char] '6'
  |-M: [char] '8'
  |-type: [char] 'confirmationNOutOfM'
|-deletion:
  |-N: [char] '4'
  |-M: [char] '8'
  |-type: [char] 'deletionNOutOfM'
|-association:
  |-type: [char] 'hungarianAlgorithm'
```

One of the motivations for using standard representation languages such as XML and KML is that the implemented system can be complemented with external edition and visualization tools. The decision of defining zones and configurations offline in files is part of client requirements. However, this can be replaced by a completely online strategy, where both elements are defined/modified through a GUI.

B. *Integrating zonal configuration into fusion logic*

Tracking and data fusion is a cyclic process, where a set of tasks are repeatedly executed over time. Some tasks are fed with the output of other processes. This can be a problem with a scheme like the proposed above, where the same step can be performed by several algorithms operating on different areas. The problem is especially acute when the data association can be parameterized in different zones, affecting to entities which can be moving through the transitions.

Let us use as example the configuration described above, where measure-track association is decided by applying the Hungarian algorithm [11]. This tool works over a cost matrix that describes how likely is for each measure to be associated with every existing track, and returns a 1-to-1 equivalence between measures and tracks. However, in spite that association step is performed by a single algorithm, its input (the cost matrix) is calculated using two different parameterizations for the "radarSimpleGating" algorithm: one inside the Port zone, and the other for the rest of the scenario. We could also be in the case of using different gating algorithms (such as Euclidean and Mahalanobis distances) or even association algorithms, as JPDA in heavily cluttered zones.

The proposed solution is illustrated in Fig. 5. Since gating has two different parameterizations, the involved tracks are split according to the regions they are in, and each set is processed with its corresponding instance of the algorithm. The partial outputs are combined back into a single data element that is suitable for the association step.

This approach has many advantages: the process is clear and well defined for any type of zonal configuration (different algorithms / different parameters for the same algorithm), the configuration of every step is transparent for the others, and it does not prevent correct functioning on algorithms that require all the information for producing the optimal solution (as Hungarian algorithm). Besides, spatial partitioning by tracks (vs. partitioning by measures) has a fundamental advantage: combining the partial outputs to produce a consistent and coherent global output is much simple.

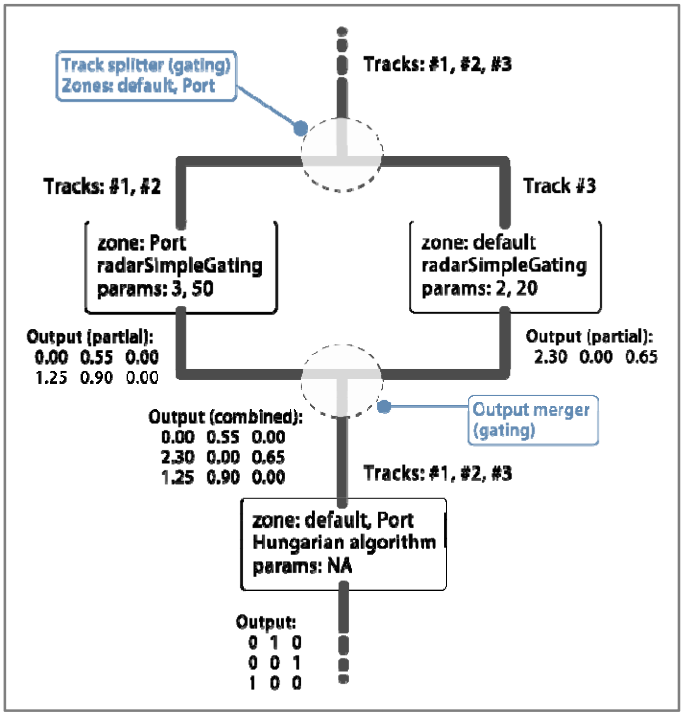

Fig. 5. Integration of zonal configuration in fusion logic. An example over the scenario displayed in Fig. 7

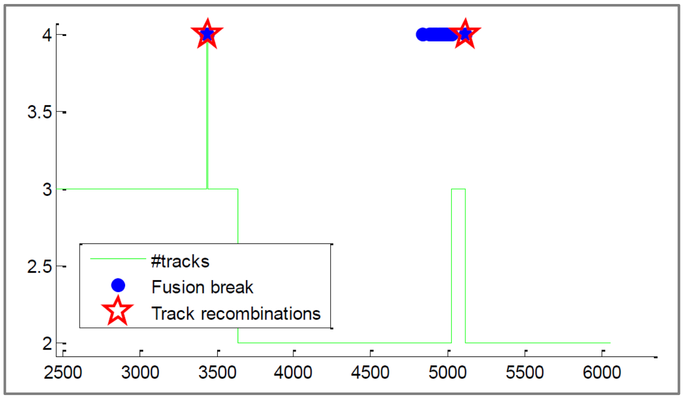

Fig. 6. Example of fusion quality metrics for a synthetic scenario.

C. *Quality metrics for self-evaluation and adaptation*

Finally, as part of the operator toolset, the sensor fusion system extracts some statistics that can be used to assess its correct functioning and detect anomalous situations that can be alleviated by a change of configuration. These metrics are based on analysing typical fusion events, including:

- Average filter residual: the residual error of Kalman-like filters. It is a measure of prediction models fitness with respect to observations.
- Average period between track updates: it helps to detect bad conditions or failures in the data sources, when compared to nominal update rates.
- Unassociated measures ratio: it is strongly related with the false positives rate of sensors. It also detects bad sensor behaviors, or can thus be used for adjustment of this parameter, if needed.
- Predicted tracks ratio: tracks that have not been updated with a measure/detection in the last cycle. It can be indicative of unusual amounts of false tracks and also sensor malfunctioning.
- Out-of-sequence data arrival: detect problems in data links
- Fusion break and track recombination: operators may want to inspect the scenario where these events took place. A high number of such events can be indicative of a malfunction in the fusion logic.
- Number of tracks: this value is expected to be stable, with small variations over time. Sudden changes can be related with a problem in sensor data acquisition or with a fusion logic malfunction.

Fig 6 shows an example in an illustrative scenario. Around t=3500s, the fusion logic breaks a system track, and the expelled component (a radar track) is immediately assimilated by another system track. This is a typical situation: one radar tracker performs a wrong association and exchanges the labels for two crossing vessels, but the fusion logic at global tracker notices the inconsistency (when the residual is significant) and rearranges the exchanged components at t=5000s. Later, around t=5000s, an abnormally long sequence of break/recombination events indicates a conflictive situation where fusion decisions are leading to an unstable output. In current implementation, a human operator can be noticed about the conflict so that he can tune the fusion parameters (for instance, increase maximum distance between components within a system track). Future lines of work involve developing an automatic response to such situations.

IV. PERFORMANCE ANALYSIS

A. *Demonstration tool and analysis scenarios*

A number of different tests have been defined to assess the system, mainly border cases like vessels crossing at certain distance and angle, vessels traveling at some distance from sensors to evaluate sensors precision impact on tracking, vessels doing complex manoeuvres, and so on. Performance evaluation is a critical aspect to validate solutions in this domain [12,13]. For designing this kind of scenarios, we developed a tool that helps defining complex maritime surveillance scenarios meeting all those requirements. This tool uses a geographical map representation that allows placing and drawing different entities like sensors, trajectories, mask zones, and both distance and angles rulers. In the following are briefly described those different configurable entities.

- Trajectories: defined by a set of waypoints, each one containing geodesic coordinates, speed, and turn rate. Moreover, it is possible to configure starting track delays in any waypoint, so it is easier to fine adjust trajectories, stops, and crossing distances accordingly to the situation to be simulated.
- Sensors: in this version they are basically Radar and AIS stations. A radar sensor is defined by some parameters like its position, period, maximum range, and precision both in azimuth and range. An AIS station instead only contains its maximum range from its location (the other parameters are in the simulated ship equipment).
- Distances and Angles: for monitoring distances and angles between tracks trajectories, tracks and sensors, and other custom locations. It is especially useful when monitoring crossing tracks, and it is required some specific crossing distance.
- Spatial Masks: for defining custom algorithms or algorithms parameters for certain zones. Masks are defined as polygons over the map, so it is easy to mask a seaport, a seaway, etc.

A sample screenshot of this tool presenting a simulated maritime scenario with different elements is shown in Fig. 7. A video sample with more details is also available at[1]. This tool not only allows displaying and configuring those different elements, it also supports trajectory and sensor detections simulation. In this way it is possible monitoring distances between tracks and sensors in real time as the simulation is executed. Once the simulation works as expected, the final step is exporting the result to files

---

[1] https://www.youtube.com/watch?v=ytqfizjD-vU

containing sensors, tracks, and zones, so they can be reproduced and processed by algorithms and tools like those described in this paper.

Fig. 7. Screenshot of scenario edition tool. It displays a simulated maritime surveillance scenario that configures two tracks with their trajectories (in white and red), and a radar (yellow) and AIS station (blue) with a given coverage area. Also a sample mask is defined in white for seaport boundaries.

B. *Analysis scenarios*

In Fig. 8 we can see a representation of the scenario used to make all the test and algorithms analysis.

The scenario contains three target and four sensors, it is important to realize that some of the sensors are detecting the targets in the limit of their maximum range, with their poorest resolution and accuracy. Some targets are equipped with AIS transponder, whose data is fused with radar sensors following the logic described.

Targets begin in separated places, and their trajectories converge later to the same area, where the targets maneuver to keep certain separations and avoid conflicts. TABLE IV. describes the targets appearing in this scenario.

TABLE IV. DESCRIPTION OF VESSELS IN THE ANALYZED SCENARIO

Target	Dimensions W/L/H	Distance	Speed	AIS Transponder
Target1	50/110/14 m	14 km	20 m/s	Yes
Target2	40/60/5 m	26 km	30–15 m/s	No
Target3	8/20/5 m	17.5 km	25 m/s	No

There are four sensors that can detect some stretch of target trajectories. They are described in TABLE V.

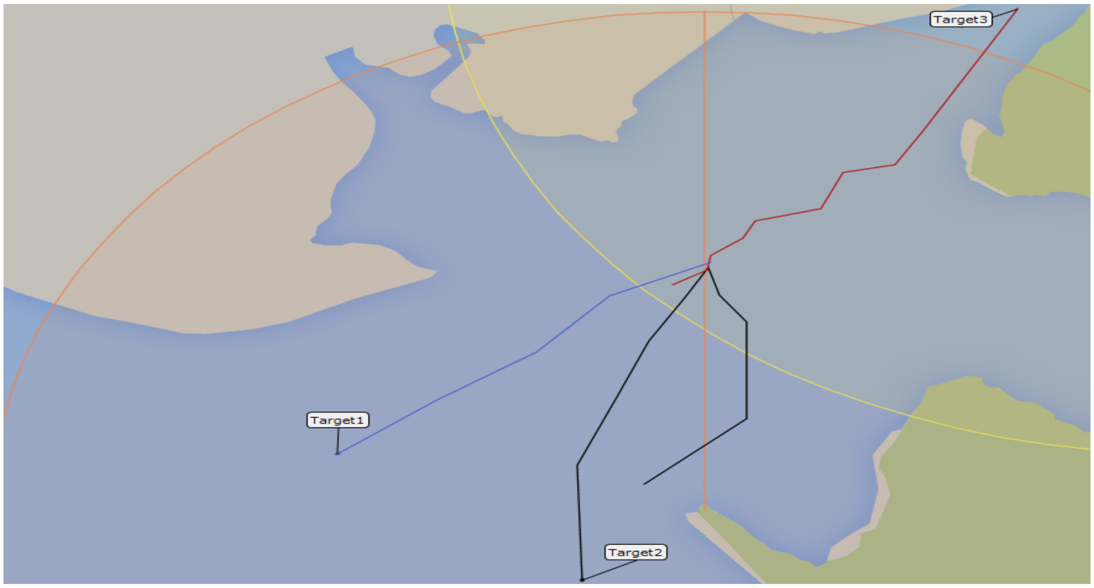

Fig. 8. Simulated scenario for analysis

TABLE V. DESCRIPTION OF SENSORS IN THE ANALYZED SCENARIO

Sensor	Type	Reach
5	Radar	55 km
6	Radar	23 km
7	Radar	22 km
4	AIS	255 km

1) ***Objectives*** *(test maneuvering targets in different áreas with algorithms)*

The principal objective of this section is illustrating some system possibilities to improve the result in different situations. So, we are going to show the scenario´s result with a default configuration and then, we will do different tests with others algorithms or different values in the parameters, to compare the results.

2) Results using default settings

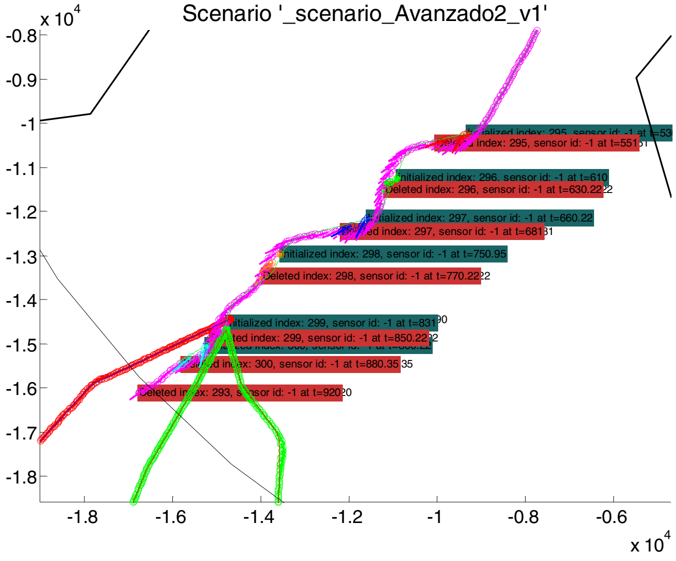

Fig. 9. Association results with default parameters

As we can see in figure 9, the system generates the fused results with some errors. Most of the errors come from the upper track, with some discontinuities due to bad association in one or more sensors. For instance, in figure 10 we can see the output data coming from a radar tracker, sensor 5, which loses target trajectory in the manoeuvres.

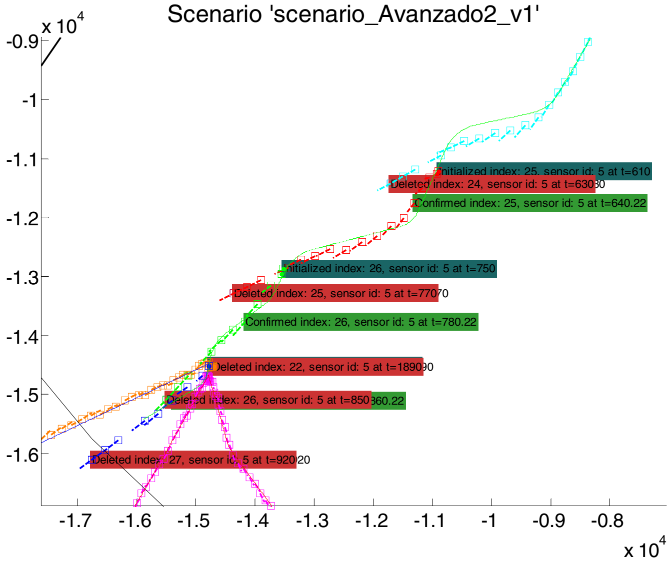

Fig. 10. Association results

In figure 11, the number of created and deleted tracks is displayed, as part of quality metrics.

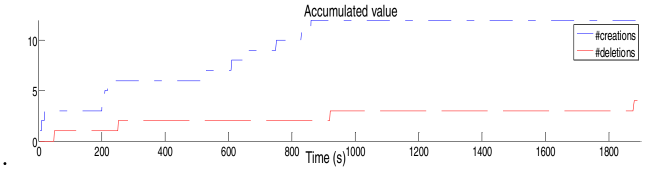

Fig. 11. Tracking results with enlarged association

To resolve this situation, there are several options. One of these options is changing the gating parameter with a bigger size for association. Another option is using a more advanced tracker like IMM´s algorithm. That algorithm will do a better adjustment based on target´s manoeuvring models.

3) Results. Modify gating criterion in maneuver zone

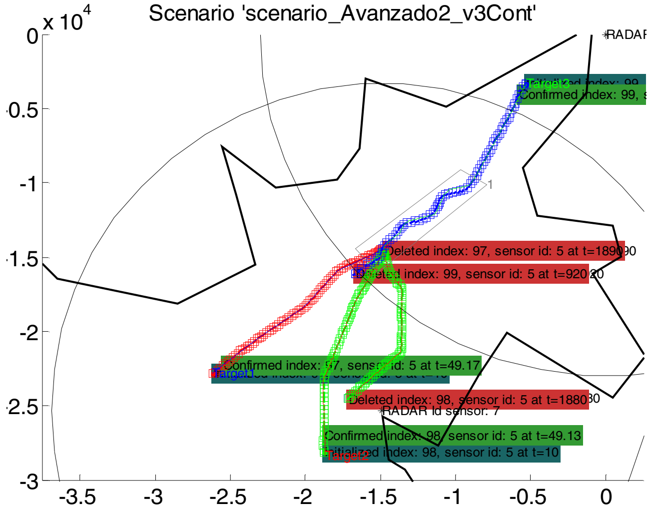

Fig. 12. Tracking results with enlarged association in zone

Figure 12 represents the result of the scenario after applying the new settings in gating´s parameter, accordingly to the defined area. The configuration is applied only when the track gets into the defined area and we can notice the correct association that sensor 5 does when gate is increased.

The possibility to change the parameters depend the area where targets are, this possibility allows specifying the best configuration for the different situations, without compromising the rest of the scenario.

4) Results. Use IMM in maneuver zone

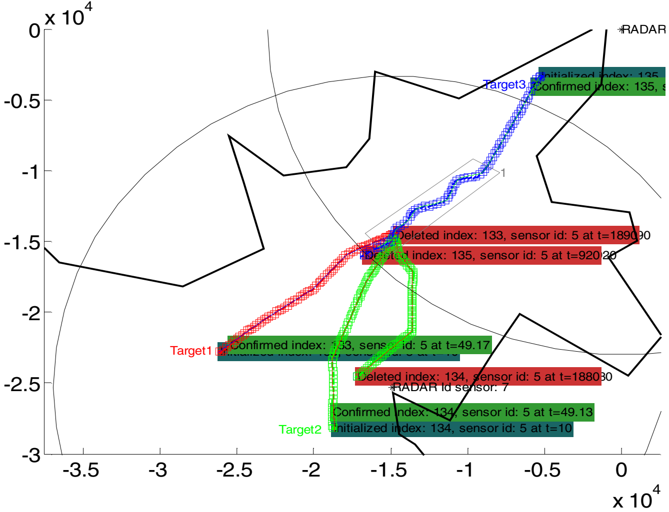

Fig. 13. Tracking results with alternative tracker in zone

Next, Fig. 14 illustrates the mode probabilities for IMM algorithm. Until time 520 approximately, target follows a straight trajectory with no manoeuvres and probabilities are constant, while the algorithm reacts to manoeuvres appearing from this time.

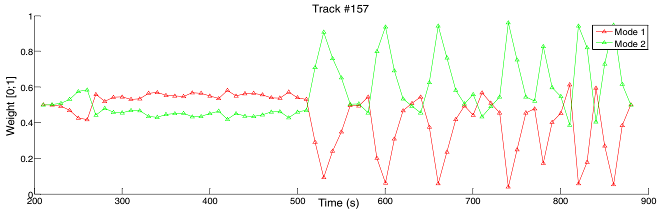

Fig. 14. IMM mode probabilities

The green line specifies the probability of the mode 2 along the time and the red one corresponds with mode 1.

5) IMM weight with different probabilities.

The parameters associated to the IMM filter were changed in the defined area (transition probabilities), with results shown in next figure.

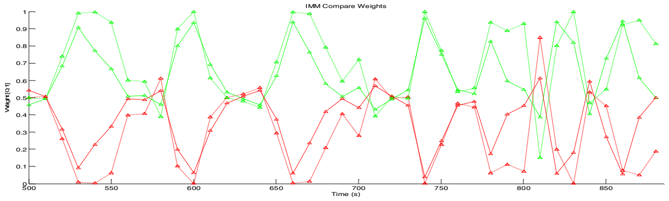

Fig. 15. IMM mode probabilities with different configurations

Finally, in the following figure, we can compare results with different algorithms. The one that has a lower error since second 520 approximately (time when the target enters the zone 1) is the execution with the IMM algorithm and the red one (track 99), is the execution with Kalman and bigger association gate configuration.

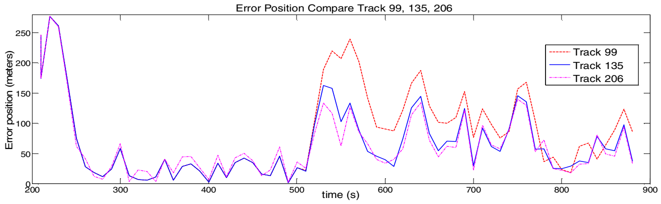

Fig. 16. Tracking error with alternative filters (Kalman, IMM1, IMM2)

CONCLUSIONS

The system presented in this work has been designed to provide highly configurable sensor fusion logic, where the algorithms and parameters can be modified accordingly to predefined areas and also on-line from operator commands, taking care of smooth transitions in the changes of configuration. Some simulated scenarios have been analysed to illustrate the capability to be adapted and solve typical tracking problems in representative situations.

ACKNOWLEDGMENT

This work was funded by contract between DEIMOS SPACE, S.L.U. and Universidad Carlos III, by Spanish Ministry of Economy and Competitiveness under grants TEC2012-37832-C02-01, TEC2011-28626-C02-02, and by Madrid Region Gov., grant CAM CONTEXTS (S2009/TIC-1485). DEIMOS Space, S.L.U. is the owner of all Intellectual Property Rights developed under said contract.

REFERENCES

[1] I. Visentini L. Snidaro Integration of Contextual Information for Tracking Refinement 2011 Proceedings of the 14th International Conference on Information Fusion (FUSION), 5-8 July 2011 Chicago, USA

[2] G. Battistello, Giulia ; M. Ulmke; F. Papi ; M. Podt; Y. Boers. Assessment of Vessel Route Information Use in Bayesian Non-linear Filtering p.447. 2012 15th International Conference on Information Fusion; 447-454.

[3] M. Vespe, M. Sciotti, G. Battistello Multi-Sensor Autonomous Tracking for Maritime Surveillance 2008 International Conference on Radar; 525-530

[4] Knowledge based radar detection, tracking and classification. John Wiley, 2008 Gini Rangaswamy

[5] G. Jemin J.L. Crassidis T. Singh, M. Fosbury. Anomaly Detection using Context-Aided Target Tracking. Journal of Advances in Information Fusion Vol. 6, No. 1 June 2011 pg 39-56

[6] Martí, E.D.; García, J., Crassidis, J.L. Improving Multiple-Model Context-Aided Tracking through an Autocorrelation Approach. In Proceedings of the 15th International Conference on Information Fusion, Fusion 2012, Singapore, July 2012

[7] H. Fargetton, J.G Siedler. Control of multi sensor system based on anomaly mapping and expert system. Sensor Data Fusion: Trends, Solutions, Applications (SDF 2011) Berlin 2011

[8] A. Benavoli, L. Chisci, A. Farina, S. Immediata, L. Timmoneri, G. Zappa Knowledge-Based System for Multi-Target Tracking in a Littoral Environment IEEE Transactions on Aerospace and Electronic Systems; 42, 3; 1100-1119 2006

[9] Tarun K. Bhattacharya and Peter Scarlett Artificial intelligence techniques for enhanced tracking performance in VTS radars. Radar Conference, 1996., Proceedings of the 1996 IEEE National 13-16 May 1996, 106 – 111

[10] Capraro, G.T. ; Wicks, M.C. Knowledge-based radar signal and data processing: a tutorial review IEEE Signal Processing Magazine; 23, 1; 18-29

[11] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistic. Quarter.*, vol. 2, no. 1–2, pp. 83–97, Mar. 1955

[12] M. Guerriero, P. Willett, S. Coraluppi and C. Carthel, “Radar/AIS Data Fusion and SAR Tasking for Maritime Surveillance" Proceedings of FUSION 2008, Cologne Germany, July 2008.

[13] J. Garcia, J.L. Guerrero, A. Luis, J.M. Molina, “Robust Sensor Fusion in Real Maritime Surveillance Scenarios" Proceedings of FUSION 2010, Edinburg, July 2010.