A machine-learning approach to multiple-detection data association for ASDE radar

Jesús García
Antonio Berlanga
José M. Molina
Universidad Carlos III de Madrid
Avda. Univ. Carlos III, 22
28270 Colmenarejo (Madrid)
Spain
jgherrer@inf.uc3m.es
aberlan@ia.uc3m.es
molina@ia.uc3m.es

Gonzalo de Miguel
Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación
Ciudad Universitaria
28040 Madrid
Spain
gonzalo@grpss.ssr.upm.es

Abstract* - *Some types of sensors provide multiple detections per target, such as ASDE radar, infra-red or video cameras, bringing new challenges to the tracking systems based on their data. In these cases, data association needs to be extended to address blobs re-connection and target segmentation issues, losing the assumptions handled by classical approaches. In this work, the design is partially considered as a data analysis process performed over representative samples to infer appropriate rules for data association. The proposal is to apply a machine-learning paradigm based on available data samples and performance results assessed through simulation. It extends a previously proposed approach based on a efficient search in the hypotheses space, applying now data mining to develop a suitable heuristic function. The advantages of this methodology are analyzed in two representative complex scenarios of airport surface.*

Keywords: Data Association, Machine Learning, Data Mining, Performance Analysis

1 Introduction

The A-SMGCS Surveillance function for airport areas fuses data from different sensors to automatically detect and track all relevant targets located in the airport movement area (runways, taxiways and apron areas). It provides the controllers with a periodically refreshed picture with all interesting details of traffic state in the airport surface.

In this work we address tracking aspects when the sensor data are complex sets of blobs provided by non-cooperative sensors such as ASDE (Airport Surface Detection Equipment [1]) radar, or video cameras. Basically, the final system is conceived as a distributed structure, with a local processor operating on the data provided by each sensor. In the case of ASDE radar, due to the sensor resolution and relative sizes of targets, the reflected signal from targets on airport surface extends over a set of range and azimuth cells, forming an image of every object in the scenario. First, moving targets are detected against their local background to generate detected cells, connecting them later to form image regions referred to as blobs. Then, the tracker must distinguish all targets in the scenario and track their motion, applying association and filtering processes to the blobs extracted from the processed images.

The association problem in this context is a correspondence of multiple blobs to multiple tracks, where the usual constraints of one-to-one assignment exploited by conventional association algorithms such as Nearest Neighbour tracking [2] are not held here since multiple blobs may be originated by the same target. Bayesian extensions such as Multiple Hypotheses Tracking (MHT) [2] or multidimensional S-D association [3] also assume conditions of single plot updating each track and no more than one track updated by the same plot, so they are not either directly applicable to this problem. It is needed to remove the one-to-one constraint, opening other alternatives to cope with the possibility of several blobs updating each track.

In a previous work [4], a data association system was proposed, considering the estimated information about the targets and their attributes to jointly decide the best grouping of blobs and association to tracks in real conditions (target re-connection, close maneuvering targets, false alarms, etc.). There, enumerations of blob grouping hypotheses and assignment to tracks were carried out, making use of an extended distance exploiting the targets attributes extracted from images to take the best decisions. Since that type of solution demands excessive computation load in moderately complex situations (the dependence on the number of blobs and tracks is exponential), it was complemented with a pruning method to guarantee a constant worst-case use of

* This work has been funded by the Spanish CICYT contracts: TIC2002-04491-C01/02 and CAM (07T/0034/2003 1)

computation. The heuristic criterion proposed there to prune the hypotheses was an estimation of "conflict degree" of every blob, based on distances to the closest tracks. Based on that heuristic, blobs were removed from the active set hypotheses until the number of hypotheses was acceptable accordingly to the computation resources.

Here, it is explored a new heuristic to prune the hypotheses set based on a machine-learning algorithm to improve the performance. Data mining techniques derive from Statistical and Machine-learning fields, and are typically used to "discover" hidden patterns and relations or to build useful models in a process named Knowledge Discovery in Databases (KDD [5]). The key aspect in that process is the preparation of data, including the selection of input data, problem formulation and model representation (rules, decision/regression trees, etc.). After this preparation, the algorithms search in the space of model parameters for those most suitable to the specific data set used in training. In this work, this type of techniques are applied to select an appropriate heuristic for complex-data association, allowing a good tradeoff between performance and resources demanded.

The following section details the data association algorithm, with the role of a heuristic to prune the hypotheses enumerated and evaluated. Section three details the proposed methodology to build a new heuristic to select the pruning with better performance. Final section illustrates, with some results obtained by simulation, the advantages of this learning process from examples with respect to the previous implementation.

2 Multi-detection Data Association System with Hypotheses Selection

The blobs-to-tracks assignment is performed by enumeration of different hypotheses of grouping blobs and correspondence to tracks, or labeling them as false alarms. Finally, each one of the hypotheses in the remaining subset is evaluated according to an extended distance [6], considering to which track is assigned each group of blobs. The term considering residuals in centroids is enriched with terms for attributes to take into account the available structural characteristics of targets extracted from data. Therefore, the tracks here contain not only the usual vector states with estimated location and cinematic parameters, but also a spatial representation of target extension and shape, with attributes extracted from the images such as orientation, length, width and area.

However, the number of possible hypotheses becomes unaffordable very quickly, and a practical system operating in real time needs to reduce the enumerated hypotheses, being sure that this reduction will not remove the most likely hypotheses. The selection is based on pruning the least probable hypotheses with a heuristic rule assessing the "conflict degree" of every blob in order to select which one has the lowest level of conflict with several tracks. Then, it is directly assigned and removed from the set. The process is repeated until the number of hypotheses is left below the maximum number allowed, guaranteeing so a bounded computation dedicated to evaluate hypotheses in worst case. The following figure illustrates the algorithm for an example of four conflicting blobs to be assigned to two tracks or disregarded as false alarms. Initially there are 3^4=81 hypotheses, and the first two iterations remove blobs b1 and b4 to their closest track, Track2, so the number of remaining hypotheses is 9, and the pruning process stops.

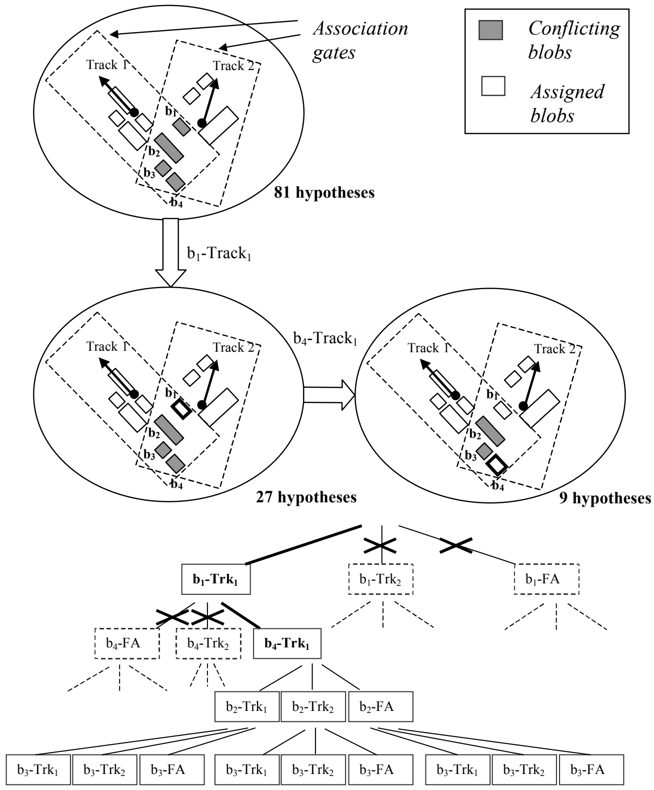

Fig. 2. Hypotheses pruning by direct blob assignment

In [4], the process was done with a heuristic based on the statistical distances of each blob centroid to all compatible tracks (including the null track representing false alarms). Then, the heuristic was computed as the ration between the two closest tracks to the blob. A blob had a lower conflict degree when this ratio was higher (it is much closer to the nearest track than to the rest), and also it got lower as the blob was closer to the closest track.

3 Design of pruning heuristic with Machine Learning

The final objective of all tracking process is to achieve enough reliability about the tracks provided, keeping a unique track per real target and minimizing the losses and switches of tracks even in complex conditions. With the previous approach, there was a mechanism to build a more robust system as the allowed number of hypotheses was enlarged (and so the computation demanded). However, it was not clear that the heuristic rule to select the blobs to prune were the optimum, and if the process could be probably improved, achieving better results with an equivalent allowed computational burden.

The design of a new heuristic has been carried out based on the paradigm of data-driven modeling, learning from examples an unknown relation. In this case, the selected algorithm was M5' [5]: a model tree induction algorithm for predicting numeric values (each leaf node has a linear regression model). From a training set with input attributes and numeric output values, it induces a functional model tree by selecting the most representative attributes and the nodes in the tree that better matches the training data, but avoiding over-fitting.

In this case, the selected unknown function is the cost of the errors when blobs are directly assigned to tracks, so that the ones with lower cost could be selected first to reduce the number of hypotheses to evaluate, minimizing the "risk" of wrong associations. The parameters selected for this function (attributes) should represent the different situations to discriminate them and distinguish the cases where decisions are more likely to produce larger errors. The cost function has been defined as the relative increase of tracking errors in position and velocity when a blob is assigned to an incorrect track or incorrectly labeled as a false alarm, relative increase with respect to the errors under situation of right assignment of all blobs. So, the cost was evaluated by means of simulation, analyzing for each blob the cost of direct assignment to the closest track and the cost of labeling as false alarm. In the cases of right assignment, when the blob was from that closest track or a false alarm, this cost was set to zero.

Regarding the input parameters (attributes for the algorithm) to describe situations, were the following:

- **Distance1**: normalized distance to the closest track. It is the physical distance to the closest track divided by the squared root of the target estimated area in the track.
- **Distance2**: the same as the above quantity, but with respect to the secondly closest track. If the blob was gated by a single track, this value was set to 10000.
- **Area1**: normalized area of blob, dividing area of blob by the target estimated area.
- **Area2**: the same as the above quantity, but with respect to the secondly closest track. If the blob was gated by a single track, this value was set to 10000.

So, the training process was carried out by computing the costs with the blobs generated and processed in a single run of the two scenarios described in the following section. For every blob available, the attributes indicated above were computed, and then, two cost functions computed. First, it was assigned to the closest track to evaluate the cost produced with respect to ideal assignment. The same was done disregarding this blob as false alarm and comparing the cost of no updating its track. In the cases that the blob was really generated from its closest track the first cost was zero, and in the cases that it was a real false alarm, the second cost was set zero. The resulting models with the simulated data (2200 samples of blobs), with rounded coefficients, were the following:

Cost of assignment to the closest track:
```
distance1 <= 0.97: LM1
distance1  >  0.97: LM2
LM1=0
LM2= 100 + 8*distance1 + 40*areaNorm1
        -50*distance2 + 50*areaNorm2
```

Cost of classification as false alarm:
```
distance1 <= 0.4: LM3
distance1  >  0.4: LM4
LM3= 162 - 400*distance1
LM4= 0.4
```

So, the learned model of cost for direct assignment of blob to a track closer than 0.97 is zero. Otherwise, it increases with distance to the closest track, it is reduced with distance to the second track (since it is less probable than come from a different target) and increases with normalized areas with respect to both. Regarding the cost of direct labeling as false alarm, it is fixed to 0.4 when normalized distance to closest track is higher than 40%. Otherwise, it continuously increases up to 162 with zero distance. After defining these costs, the new heuristic for removing blobs from the hypotheses set consists in computing both costs for every blob and then select the one with the lowest cost value, deciding its direct assignment or elimination.

4 Results

Results presented in this section were obtained by means of Monte Carlo simulation. The ASDE detections were generated and processed by the tracking system to analyze the effect of the assignment scheme with the two heuristics in the global performance.

Two sample scenarios were simulated, both of them with targets performing maneuvers at close distances so the tracking algorithm should group and assign the detected blobs to the right tracks. The targets trajectories are depicted in figures 2 and 3, presenting the whole history of track updates generated along a single-run simulation for each scenario.

Scenario 1 (figure 2) contains two aircraft moving on crossing taxiways. The first target makes two turns moving towards the upper taxiway, while the second one performs a stop-and-go maneuver, waiting until the first one enters the taxiway. Then, it moves behind the first one, separated 100 meters. Besides the targets detections, this scenario contains false alarms appearing in three areas in borders of taxiways. They were included to simulate the effect unmatched detection masks. The unmasked detections in these cluttered areas interacted with the tracks, although track initiation was not allowed out of taxiways. The association logic should distinguish them from target detections, to keep quality of tracks.

Scenario 2 (figure 3) presents an example of operation in the airport's apron area: an aircraft brakes in a loading position while two vehicles approach to it, stop separated 25 meters from the aircraft, and then turn almost 180º to go back to their original positions. This scenario illustrates the system's capability to track targets while they are closely maneuvering at distances shorter than targets sizes.

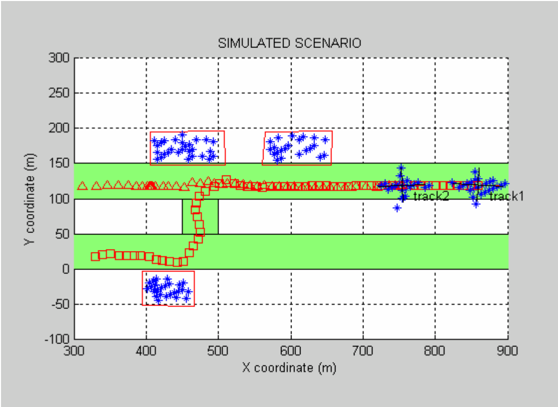

Fig. 2. Scenario 1

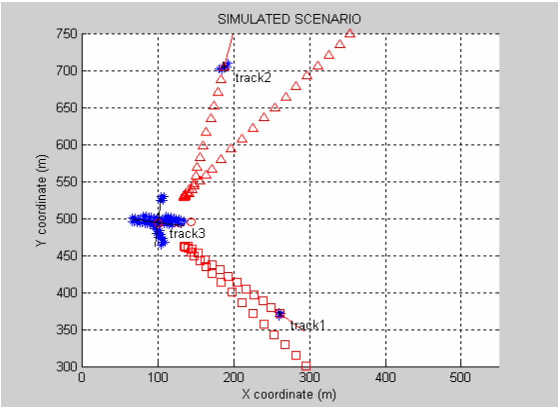

Fig. 3. Scenario 2

System performance was evaluated regarding two aspects: (i) tracking accuracy, reflecting the impact of blob-association errors over final accuracy and (ii) tracking continuity, assessing the robustness of tracker to maintain stable tracks. The figures evaluated were:

Track accuracy: these figures were assessed only for conditions of no track commutation, selecting the track with the highest number of associated target cells in the case of track-split situations.

- position error: horizontal deviation between real and estimated centroid.
- velocity error: deviation between estimated and real target groundspeed.

Track continuity:

- number of tracks: the averaged number of tracks represents the presence of extra tracks or track losses, when it differs from the number of real targets in each scenario.
- rate of track switches: it assess the probability of track switch, defined as situations when more than 50% of cells assigned to the track come from a different source than that corresponding to the initialization situation. Track switches are the situations in which a track is initiated representing a target and then represents other object (other target or false alarms).

The results were generated with 300 iterations of Monte Carlo simulation in the two scenarios described above. The association logic were configured varying the maximum number of allowed hypotheses to 1, 4, 8 and 64, comparing the performances with the previous and proposed heuristic to prune the hypotheses. The criteria for representing algorithm's configuration in all figures is the following : (-·-), (···), (---), (—) for 1, 4, 8 and 64 hypotheses allowed; red color for the old heuristic, blue color for the new data-driven heuristic.

Results in figures 4-8 correspond to scenario 1 with probability of false alarm in the cluttered areas set to 50%. We can see the accuracy results for target 1 (double turn) in figures 4-5, and the continuity performance in figures 6-8. The number of tracks (figure 6) is continuously increased along time for all configurations, indicating the generation of false tracks or effects of track split. The new heuristic to select hypotheses shows significant improvement in tracks stability (switch rates), and in track accuracy, especially when very few hypothesis are allowed. This improvement is due to the learning process aimed at improving performance for the same resources of computation allowed (number of hypotheses). Only for the cases of 64 hypothesis allowed it gets worse performance than the old heuristic. It is noticeable that the new heuristic for pruning leads to very similar performances with different number of hypotheses, if compared with the strong dependence with the previous approach.

Regarding scenario 2, results are presented in figures 9-14, for the aircraft under load operation, target 1. The configuration with a single hypothesis allowed in this scenario systematically grouped more close cells and produced the losses of tracks representing the vehicles, targets much smaller than the aircraft. That is why the number of tracks falls in the time interval around [30,70], improving when the number of hypotheses increases. In this case, the machine-learning heuristic brings improvement in all cases with the exception of a single hypothesis allowed, in which case the track switch for the aircraft and the rate of tracks deleted increase. Very significant improvement appear in the stability and accuracy for all tracks for the rest of cases.

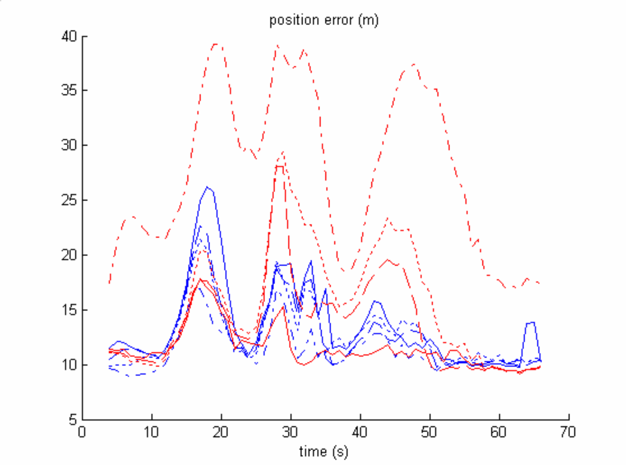

Fig. 4. Scenario 1. Position error for target 1

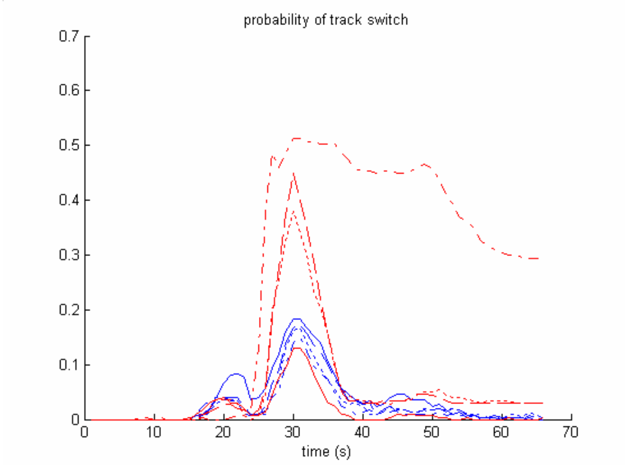

Fig. 7. Scenario 1. Track switch probability for target 1

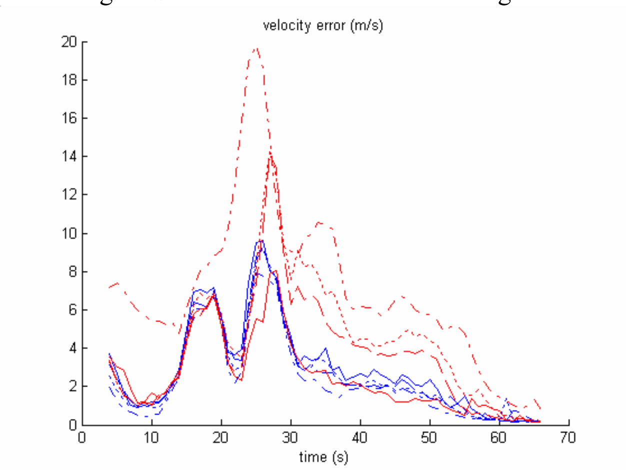

Fig. 5. Scenario 1.Velocity error for target 1

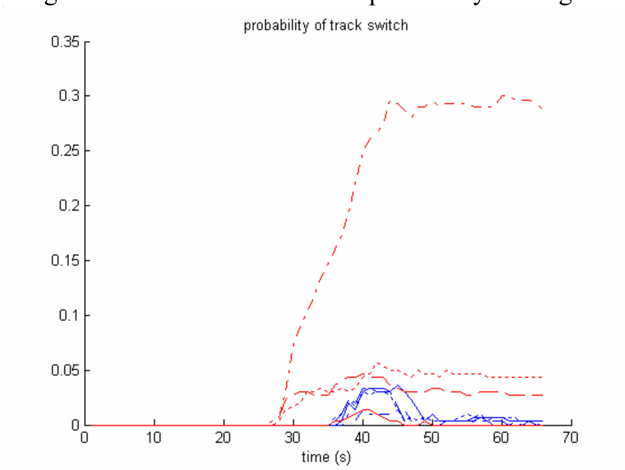

Fig. 8. Scenario 1. Track switch probability for target 2

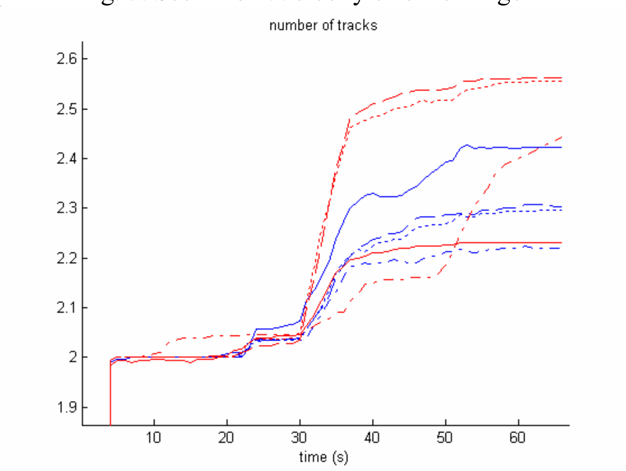

Fig. 6. Scenario 1. Average number of tracks

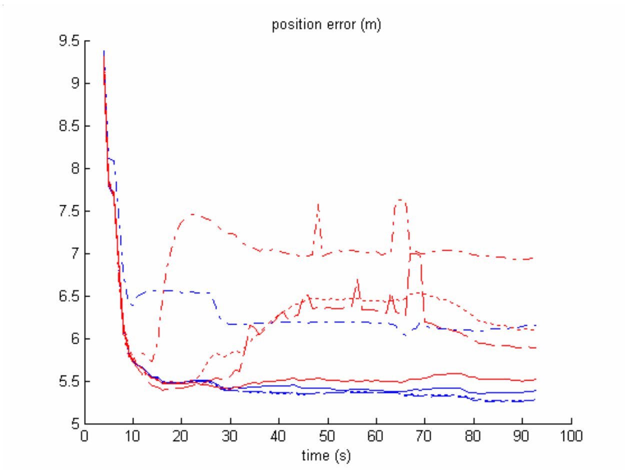

Fig. 9. Scenario 2. Position error for target 1

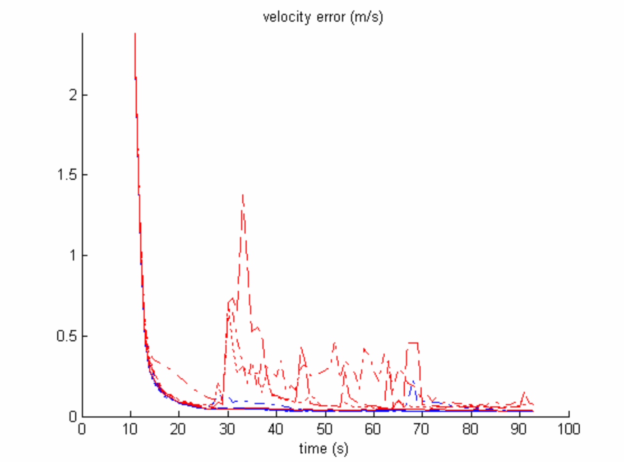

Fig. 10. Scenario 2. Velocity error for target 1

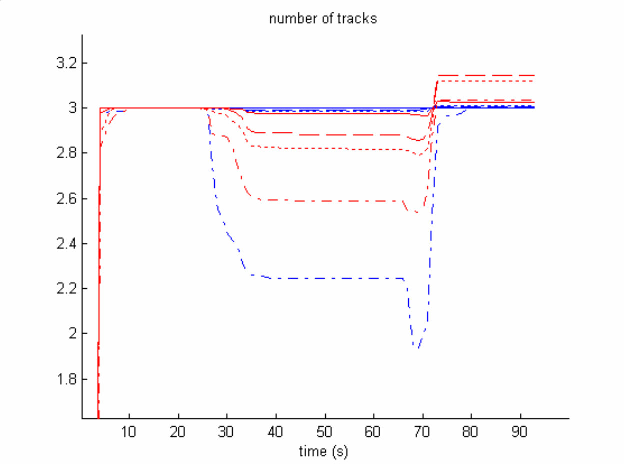

Fig. 11. Scenario 2. Average number of tracks

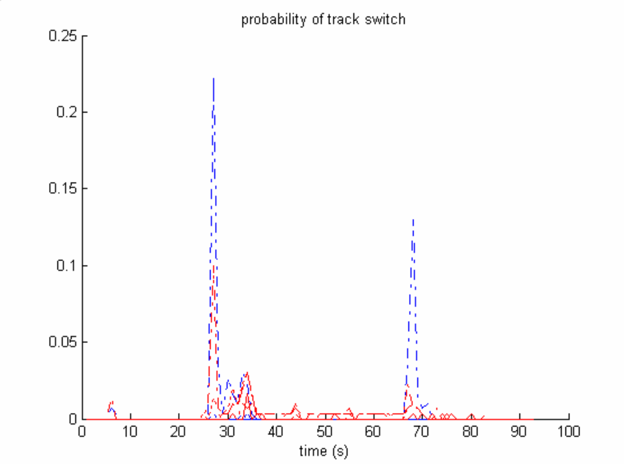

Fig. 12. Scenario 2. Track switch probability for target 1

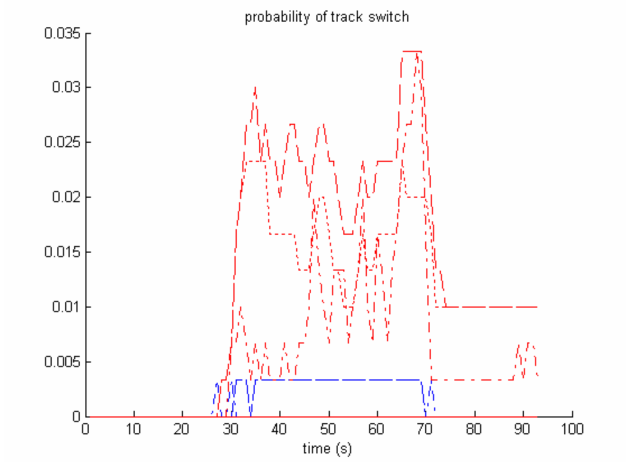

Fig. 13. Scenario 2. Track switch probability for target 2

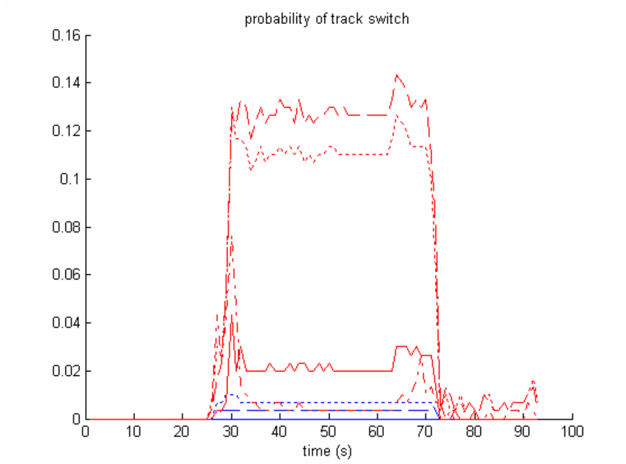

Fig. 14. Scenario 2. Track switch probability for target 3

2 Conclusions

Machine-learning based on performance improvement has shown to be a promising tool to improve the design of a complex tracking system, where the tuning of parameters and functions is not clear. This is due to the lack of available analytical models to relate the design variables with the expected performances. In this case, the tradeoff adjusted by the maximum number of hypotheses allowed (a constant bound of computation load in worst case) has been significantly improved by learning the best criteria to reduce problem complexity, minimizing the errors costs.

However, this is an exploratory approach yet and future works will analyze more scenarios and situations to derive functions able to cope with more general situations. Besides, machine learning will be applied to other aspects, such as the optimization of parameters in the extended distance used to evaluate hypotheses.

References

[1] C. E. Schwab, D.P. Rost. Airport Surface Detection Equipment. *Proceedings of the IEEE*, No. 2, February 1985.

[2] S. Blackman, R. Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House. 1999.

[3] S. Deb, M. Yeddanapudi, K. R. Pattipati, Y. Bar-Shalom. A Generalized S-Dimensional Assignment for Multisensor-Multitarget State Estimation. *IEEE Transactions on Aerospace and Electronic Systems*. Vol. 33, Nº2. Apr. 1997.

[4] J, García, G. de Miguel, A. Berlanga, J. M. Molina, A. Soto ASDE-RADAR Data Association for Robust and Efficient Tracking on Airport Surface. *International Radar Symposium 2003*. Dresden, Germany. Sept. 2003..

[5] S. Witten, Ian H. *Data mining : practical machine learning tools and techniques with Java implementations*. San Francisco Morgan Kaufmann, 2000.

[6] J. García, J. A. Besada, F. J. Jiménez, J. R. Casar. Surface Movement Radar Data Processing Methods for Airport Surveillance. *IEEE Transactions on Aerospace and Electronic Systems*. April 2001.