A MATLAB TOOL FOR DEVELOPMENT AND TESTING OF TRACK INITIATION AND MULTIPLE TARGET TRACKING ALGORITHMS

Kiril ALEXIEV

1. Introduction

Digital computer simulation is a valuable tool, used for the design, analysis, and testing of complex systems whose behavior cannot be easily evaluated by means of analysis. Simulation includes three main steps, as follows: input data generation, modeling of examined system and performance evaluation with proper visualization. By their very nature, radar data processing algorithms are well suited to computer simulation. Simulation programs are often coded in general purpose high-level languages, such as C++, Pascal, Ada, or Java. This is mainly due to the popularity of these languages among programmers and computer users, as well as to their availability and portability. But the most complex algorithms can be easily coded by means of Matlab. The Matlab language can be learnt quickly and after that the engineers can fully exploit its power with high productivity. Matlab compiler translates *.m files to C code for real time implementation purposes. In spite of the fact, that the resulting code consists almost completely of calls to Matlab *.dll functions and the performance is similar to that of standard *.m files, the translation can be regarded as a good initial step of migration to C code.

The purpose of this paper is to describe simulation tools for analysis and design of a radar data processing system, to outline the techniques used to generate the input data, to simulate the algorithms and to analyze the results for evaluation of system’s performance.

This tool can be useful to practicing radar engineers for the purposes of both analysis and design.

2. Architecture of the simulation programs

The simulation of radar data processing can be defined as a set of algorithms which allows:

- Complex scenario generation;
- Recognition of a pattern of successive detections as pertaining to the same target (track initiation);
- Estimation of the kinematics parameters (position, velocity and acceleration) of a target, thus establishing the so called "target track";
- Extrapolation of the track parameters;
- Distinguishing of different targets and thus establishing a different track for each target;
- Adaptive scheduling of the time dwells of a phased-array radar in order to follow a maneuvering target with constant accuracy and to interleave in an optimum manner the tracking phases with search looks and other radar functions;
- Efficient managing of the detections or the tracks, provided by the different radar sets of a netted system looking at the same portion of the controlled space, in order to provide a better picture of the latter.

A generalized scheme of the proposed Matlab tool is presented on figure 1.

3. Input data simulation

There are two different approaches to the input data simulation. The first of them uses data, recorded from real radar. This approach simulates the real operating conditions of the testing system and there are no errors caused by data modeling. But there is a severe drawback – it is very difficult, dangerous, expensive and some times impossible to explore estimated algorithms in a complex scenario. Such a scenario is of low probability, although it can exist in real life critical conditions. Another, more mild drawback is that the true target path and the true target maneuver parameters are unknown and the researcher has no exact reference data for accurate evaluation of the algorithm under exploration.[2] Nevertheless, the Matlab tool has an entry for real life data, using a common data format for data exchange.

The use of simulated data has considerable flexibility in the selection of a complex target and clutter scenario and an a priori known reference input is provided. The simulation program generates hundreds of targets moving rectilinearly or maneuvering with given transversal and longitudinal accelerations. The radar parameters (scan rate T and detection probability P_D) can vary in wide intervals.

The simulation program has the ability to synchronize position of generated targets in the space and thus to create complex and critical scenarios. However, only an approximate representation of the operating conditions can be obtained. Another simulation program is used for noise and clutter generation. The noise can be generated in the whole surveillance volume or only in the current targets gates. The last feature is very useful for testing algorithms with hypotheses generation. Another useful feature of the simulation program is the possibility of generating given number of trajectories with fully random parameters or parameters, randomly chosen in given intervals. In this way, the input data are generated for Monte Carlo analysis of explored algorithms.

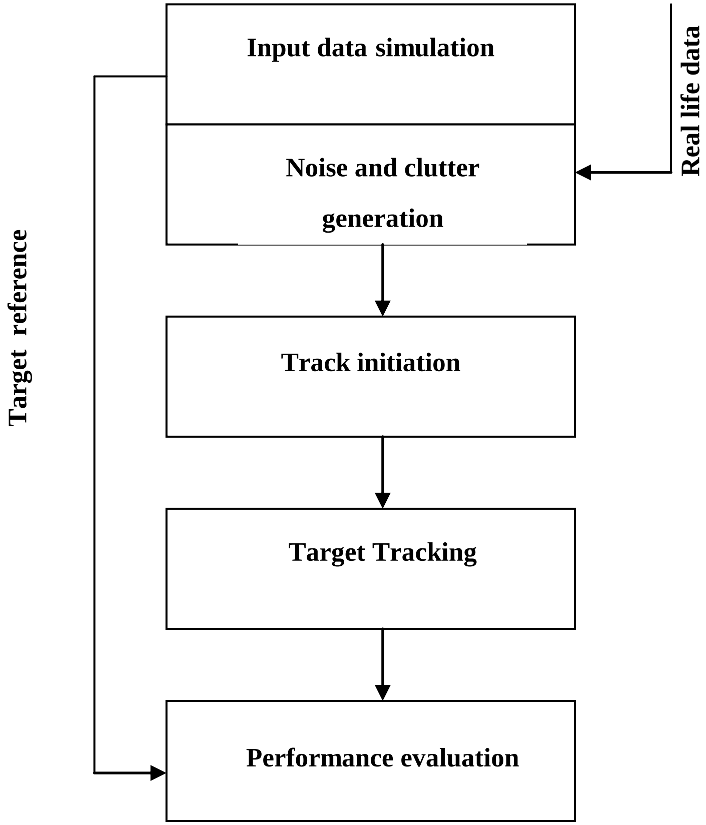
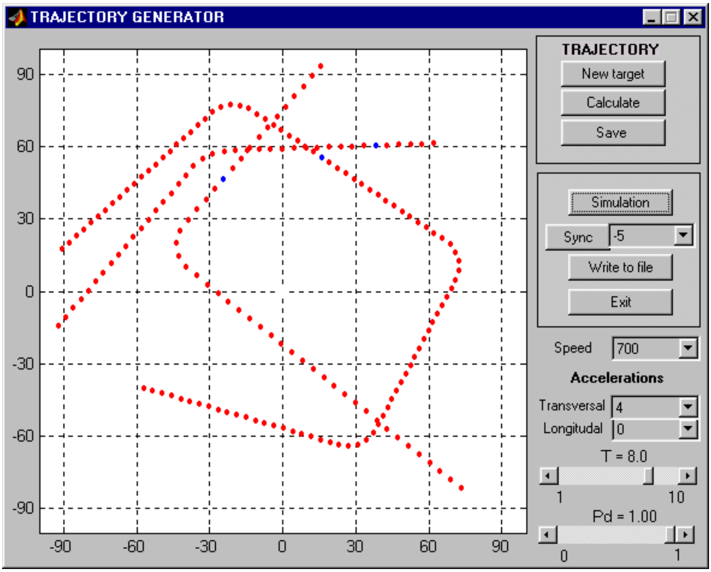


Figure 1: Flowchart of the Matlab radar data processing tool



Sync -5

Speed 700

Transversal 4

Longitudal 0

T = 8.0

Pd = 1.00

Figure 2: The graphic user interface of target simulation program

The trajectory generator is generally better suited for algorithm estimation and tuning. Recorded real sensor data can be used as a more realistic test in an advanced stage of the design or as a last approval of system characteristics.

The trajectory is assumed to be planar; it may consist of straight and circular sections. Initially, several points on the map define its sections. Every point consists of target position, target speed, target transversal acceleration and time.[9] The time is calculated using longitudinal acceleration and parameters of two sequential points. In every point (except the first and the last one) the direction of the target is changed. The maneuver is considered with constant longitudinal speed and constant transversal acceleration. The target motion is modeled by computing the position, velocity and acceleration at the equal time instants T . The time interval between two consecutive detections of a target may differ slightly from radar scan period. This simplification does not affect the accuracy of the simulation since the Kalman filtering does not require a constant sampling interval, being based on the effective detection instant. This assumption compares to a modulation of the antenna scan period, which is commonly encountered in practice, e.g. due to the wind.

The radar sensor is modeled by a program, which takes into account measurement error distribution. Analyses of radar measurements range and azimuth errors showed that the best approximation of the error distributions is a double Gaussian distribution.[3] This model is used in the radar modeling.

4. Track initiation programs

Track initiation programs associate sensor measurements with potential track trajectories using different correlation techniques. The task is to find several measurements ordered in space and time. The most common approach, considered as classical, uses N sequential measurements (usually 4-5 measurements) and implements weighted least squares to find initial approximation of target state. The classical approach can be very computationally intensive, because the number of hypotheses grows exponentially with the number of measurements under consideration. This hypothesis growth can be overcome by careful hypothesis pruning. A gating technique is introduced in order to reduce the combinatorial problem, but this algorithm does not solve the problem completely.

In dense target and clutter environments, however, the number of hypotheses remains too big enough and the classical approach fails to initialize the trajectories, thus leading to poor results. In this case, different type of track initiation procedure has to be used. The vast surveillance volume is fragmented in a set of cells and the combinatorial problem is decomposed on many such problems of smaller size, solved in small fragments. Two types of such track initiation procedures are implemented. The first of them uses uniform surveillance volume fragmentation. The measurement selection method typically uses a mosaic grid to group the measurements in subsets. The track initiator uses these subsets for potential track determination. The problems of optimization in this case are determination of cell size and how to process measurements on (or near to) the cell borders. The second algorithm uses template matching technique like Hough transform,[5] Fourier transform, etc. The cells in the surveillance volume in this case correspond to the initialized target trajectories. Both methods require additional computer resources to resolve the combinatorial problem in the case of dense target and clutter environment. The main parameters to be estimated are the probability of detection of a trajectory, the probability of false track detection as a function of the number of considered measurements N, radar probabilistic characteristics like P_D and P_{fa}, the size of gate, cell or template and so on.

5. Target tracking programs

The modern surveillance systems using radar as sensor require rapid and highly accurate data to be subsequently processed. Location, velocity, maneuver and possible identification of each target of interest can be provided by radar data processing with accuracy and reliability greater than that available from single-look radar report. Furthermore, radar data processing can enhance the signal-processing function by removing false detections caused, for example, by residual clutter.

This suit of programs reproduces the radar data processing algorithms, which allow the formation of estimated target states on the basis of incoming measurements provided by radar sensors.

Advanced multitarget/ multisensor/ multiplatform tracking algorithms have to possess the following characteristics:

- Tracking hundreds of targets;
- Work successfully with potentially long revisit rates (several seconds);
- Continuous tracks of weaker targets at lower SNRs (low value of P_D);
- Continuous tracks of closely-spaced targets;
- Creates common fused tactical picture of surveillance volume of several sensors.

Several estimation procedures are implemented in the proposed Matlab tool. They use $\alpha-\beta$, $\alpha-\beta-\gamma$, Kalman and extended Kalman filters in different realizations. All these procedures are intended to improve estimation accuracy. Note, that only in Europe more than 30 different trackers are currently in use.[4] Some of them use the same filters but work with different coordinate systems (polar, orthogonal) or the state vectors have different length. Most of these filters are available for use in the Matlab subroutine library.

Classically, plots are associated to the potential tracks using the “nearest-neighbor” algorithm. Wrong nearest-neighbor assignments, however, cause tracking filter divergence. Such is the case when there are false alarms in the target gate or in the case of closely spaced targets.

The first of these cases can be resolved using the probabilistic data association (PDA) algorithm.[1] This is a basic algorithm for plot-to-track association, which uses all measurements in the target gate. PDA allows more than one measurement to be associated to a track, each with a different probability and corresponding weight, according to its distance to the target prediction. The PDA filter is very simple and robust against false alarms.

The Joint Probabilistic Data Association (JPDA) algorithm is another advanced technique, implemented as a tracking algorithm in the Matlab tool. This algorithm resolves the case of closely spaced targets with common measurements. In this case measurement to track association for one track cannot be performed independently of other tracks in the cluster (cluster is a set of closely spaced tracks). *Joint* means that all possible measurement track combinations have to be evaluated. Furthermore, the track state vector update must, in principle, be done also jointly. Through appropriate approximations in the JPDA algorithm, however, the latter may not be necessary. Still, the complexity of JPDA grows exponentially with the number of tracks and measurements involved in the resolution situation. The advantage of JPDA is that, even in resolution situations, the track quality can be maintained at a high level. Several modification of this algorithm have been realized and estimated.[7,8]

The Interacting Multiple Models (IMM) filter is a robust filter, used for tracking of maneuvering targets. It assumes that a target is in one of a number of modes of movement, each of which may be modeled by its own equations of motion. This approach uses several filters. Every filter corresponds to a mode of movement of the target. All filters process each measurement. The particular filter innovation and the probability of holding target in (or moving target to) this mode define the weight of particular filter estimate on the common estimate. In the next interaction step, the information from all particular filters is combined and fed back into the filters. The choice of filters and suitability of their parameters remains a difficult problem to solve. It is obvious that robustness of IMM filter is achieved at the expense of estimate accuracy. For example, if a filter matches exactly with target motion mode, its estimate is deteriorated by influence of the other filters, which give poorer estimates. Another disadvantage of the IMM filters is the increased computational complexity. The IMM filter may also be used in conjunction with PDA filter and JPDA filter.[1,6] The researchers have on hand several versions of the described algorithms in the Matlab tool library.

6. Statistical estimation and visualization programs

The input data simulation program works with real life data (received by radars) or simulates the movement of targets and calculates the values of measurements. The second case is used for Monte Carlo estimation of the algorithms. Sufficient number of statistically independent trials is performed in order to achieve a significant sample of output values from which reliable statistics can be estimated. The estimates are compared with reference values of the tracks and models. The accuracy and detail of every model may vary from a coarse functional description of the system to a very accurate one, according to the purpose of the simulation and required accuracy of the results.

The visualization of results can be achieved by means of power Matlab graphic output. The next two examples demonstrate the capabilities of the presented tool.

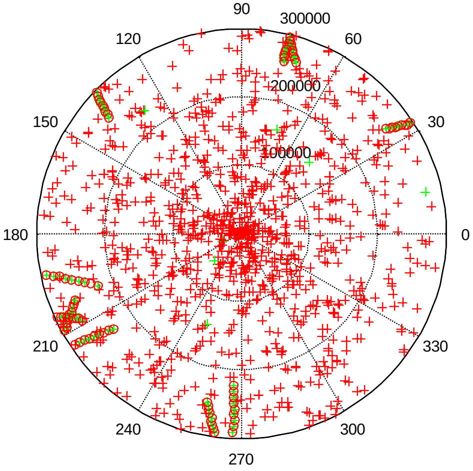


Figure 3: The test scenario includes ten approaching targets with randomly generated trajectory parameters and noise

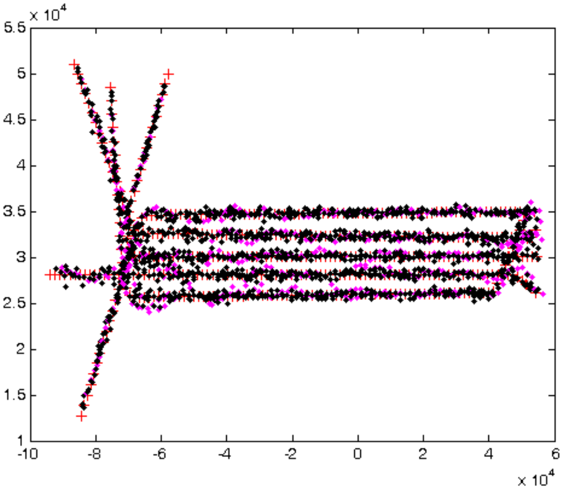


Figure 4: IMM JPDAF algorithm for the case of five-target scenario with $\beta V = 2$

Conclusion

A Matlab simulation tool is presented for multiple target tracking algorithm exploration and estimation. The built-in library of scenarios, models and algorithms provides an opportunity for easy implementation and testing of new versions of the track initiation and target tracking algorithms, comparative analysis and prompt estimation of their characteristics. This simulation tool protects us from surrogate target tracking system implementation.

Acknowledgment

The work on this paper was supported by the Center of Excellence BIS21 under Grant ICA1-2000-70016.

Notes

[1] Yaakov Bar-Shalom, *Multitarget multisensor tracking: Advanced applications* (Norwood, MA, Artech House, 1990).

[2] Alfonso Farina and Flavio A. Studer, *Radar Data Processing* (Letchworth, Hertfordshire, England: Research Studies Press, 1986).

[3] Samuel S. Blackman, *Multiple Target Tracking with Radar Applications* (Norwood, MA: Artech House, 1986).

[4] ARTAS, European Organisation for the Safety of Air Navigation (EUROCONTROL, 2001).

[5] Kiril M. Alexiev, “Implementation of Hough Transform as Track Detector,” in Proc. of the International Conf. on *Multisource Multisensor Information Fusion* (Paris, France, 2000), pp. ThC4-11-ThC4-16.

[6] Ljudmil Bojilov, Kiril Alexiev and Pavlina Konstantinova, “An Accelerated IMM JPDA Algorithm for Tracking Multiple Manoeuvring Targets in Clutter,” in the current issue of *Information & Security*.

[7] Ljudmil Bojilov, Kiril Alexiev and Pavlina Konstantinova, “A particular programme realization of JPDAF algorithm,” *Comptes rendus de l'Academie bulgare des Sciences* 55, 9 (2002): 37-44.

[8] Pavlina Konstantinova and Kiril Alexiev, “A Comparison of Two Hypothesis Generation Algorithms in JPDAF Multiple Target Tracking,” in *Proc. of the International Conference on Computer System and Technologies*, vol. II (Sofia, 21-22 June 2001), pp. 18-1 – 18-5.

[9] Kiril Alexiev, Emanuil Djerassi and Ljudmil Bojilov, “Flight object modeling in radar surveillance volume,” in Proc. Sixth international conference *Systems for automation of engineering and research SAER'92* (Varna, Bulgaria: SAER, 1992), pp. 316-320, 01-03.

KIRIL METODIEV ALEXIEV see page 153.