

Implementation of tactical ballistic missile tracking

M. Okoń-Fąfara* & B. Wajszczyk
Faculty of Electronics, Military University of Technology,
gen. Urbanowicza Str. 2, 00-908 Warsaw

ABSTRACT

In the article the idea and the main implementation problem of a tactical ballistic missile (TBM) tracking chain are presented. The processing chain implements the Track-Before-Detect (TBD) method to confirm a TBM detection and as the next step an extended Kalman filter as a tracker is used. The structure of the system and the descriptions of the purpose of particular modules are presented in the paper. A complex data structure for storing of the measurements, the organizational data and the algorithms parameters has been designed in order to improve the computational effectiveness and use of system resources. The proposed data storing solution allows for the reduction of the amount of stored measurement data without a loss of tracking results quality. In the paper selected results of TBM tracking for different data organization solutions and different methods of the data amount reduction are presented.

Keywords: tactical ballistic missile, tracking, Track-Before-Detect, computational optimization

1. INTRODUCTION

In the classical radar signal processing chain, the detector decides about the occurrence of an object before the tracking step. In the case of the object's low level of radar cross section or the high linear speed, the lowering of the detection threshold is necessary. As a consequence side effects such as higher false alarm rate and increased chance for false plot to track association occur. Therefore the tactical ballistic missile (TBM) tracking may be impeded.

The described side effects can be minimized by using one of the alternative methods called track-before-detect (TBD). In this approach the detection and tracking stages are combined into one process [1] and association of several consecutive scans allows to take a detection decision [2]. There are many TBD algorithms that can be divided into two main categories. The parametric algorithms that use the *a priori* information about the object's dynamics and the non-parametric ones that allow to create the associations of the candidate plots into tracks in a dynamic way dynamically [3]. The first class of algorithms includes the Bayesian methods and multiple hypothesis tracking [4] and the most popular method in the second group is the dynamic programming (the Viterbi algorithm) [5].

The publications about the TBD algorithms are mainly focused on the results quality and only few describe the solutions that can speed up the computation. In this article a comprehensive implementation of a TBM tracking process using one of the TBD methods is presented, considering the data storage containers and the management of the flow of plots and appropriate control commands. During the development of the TBD processor the authors have taken into account that as the consequence of lowering the threshold of the Neyman-Pearson detector a large quantity of input data may appear. At the same time the storage and computation resources burden and the time of computations should be balanced. The description of the proposed implementation solutions and the results of its efficiency testing are included in the following chapters.

2. THE DESIGN OF TBM TRACKING CHAIN

The TBM tracking chain enables the parallel computation in several modules. The design has been based on the following assumptions:

- the computations are made in a selected part of the observed area,
- in the TBD algorithm many models of rockets trajectory are considered,

*marta.okon@wat.edu.pl; phone +48 261 839 243; fax +48 261 837 461; wel.wat.edu.pl

- the result of TBD can be used for further work in the TBD algorithm or the initiation of classical tracking,
- tracking using TBD has higher priority than classical tracking
- the already used data is removed from containers.

The proposed implementation of the TBM tracking chain contains three programming modules (Figure 1.):

- data storage and management (DSM),
- TBD algorithm module (TBDM)
- the extended Kalman filter (EKF) tracker (EKFT)

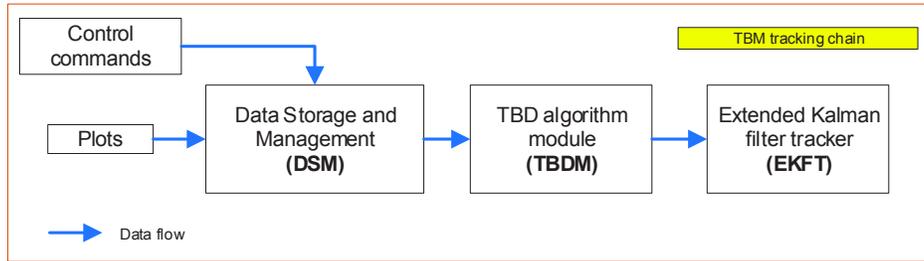


Figure 1. The tactical ballistic missile tracking chain.

The detailed tactical ballistic missile tracking chain is presented in the figure below.

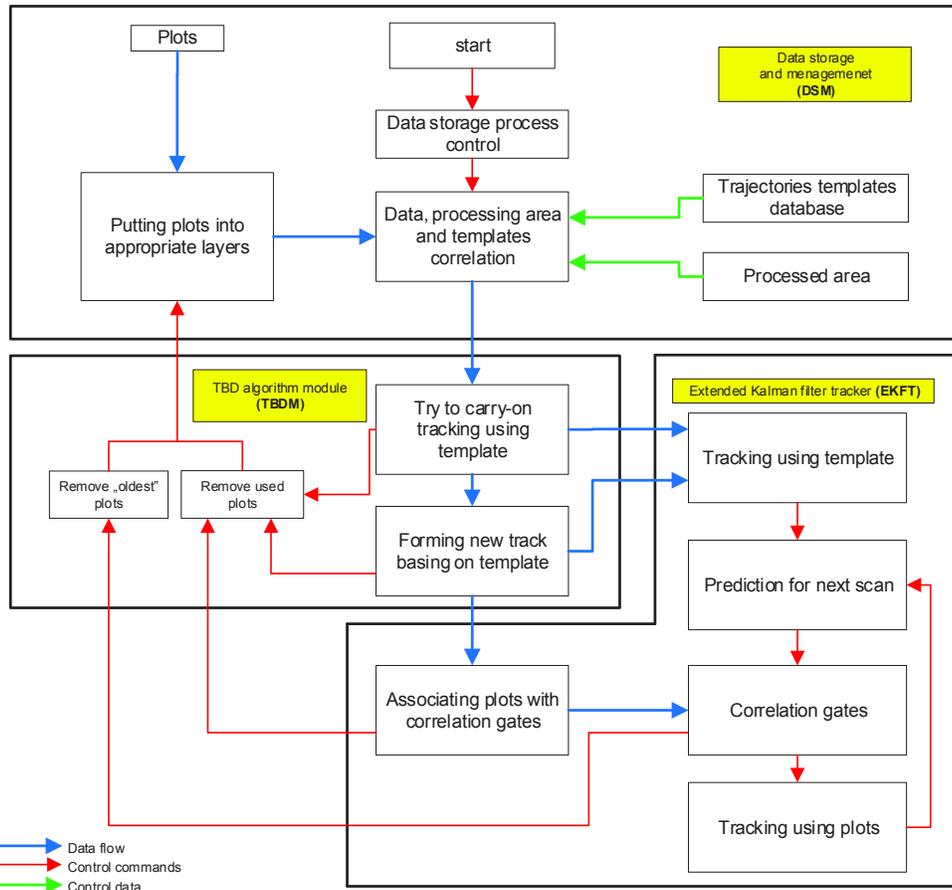


Figure 2. The detailed design of tactical ballistic missile tracking chain.

2.1 Data storage and management

The main goal of the data storage and management module is the appropriate acquisition of measurement data (plots) and the management of the tracking chain work. Plots are stored in previously prepared containers. Their configuration is dependent on the radar working mode and the algorithm control commands values. The proposed solution collects plots from N scans. It is assumed that the data from one scan (e.g. one revolution of antenna) corresponds to one layer of the container where the lowest layer number contains the oldest data. For the visualization of the whole process this layer is marked at the bottom, as showed in the Figure 3, where $N=4$.

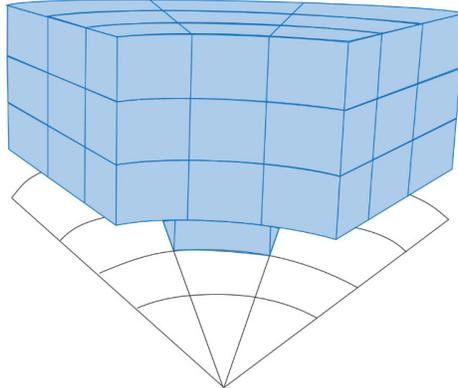


Figure 3. The visualization of the data storage container with distinguished layers for $N=4$.

Beside the distribution of the data to layers accordingly to its time of arrival, there is a spatial separation of the data in range and azimuth. The angular distribution is the consequence of the data acquisition process from detector and the work of the TBD algorithm, EKF tracker and deletion of used or expired data. This also enables the independent access to the data by particular processes that can work in parallel. In order take a full advantage of the capabilities of multicore processors, the Intel Threading Building Blocks (TBB) technology was used in proposed solution. The division of the range domain into sectors gives the possibility of further parallelization of computations and in the same time it limits the amount of data processed in one process.

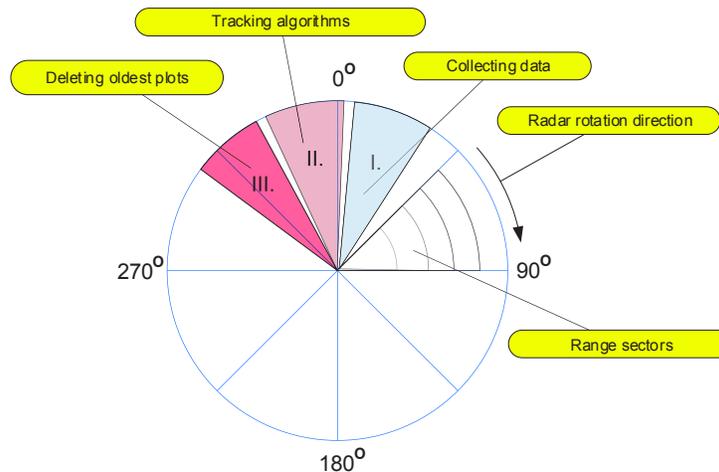


Figure 4. The spatial separation of the TBM tracking chain processes.

A side effect of the assumed spatial separation of data is the possibility of initiating multiple tracks with the same plots. Therefore an algorithm of finding and eliminating such tracks had to be implemented.

The data storage and management module is also responsible for storing the TBM trajectories templates and the areas of data processing. The templates database is range dependent, which is the effect of the following facts:

- from the radar point of view the trajectory model changes with the distance from the radar to the launch point due to the curvature of the Earth,
- at different distances another part of trajectory will be covered by the radar antenna beam,
- the computations should be done only for this part of the object trajectory that can be visible to the radar.

This phenomenon are explained in Figure 5.

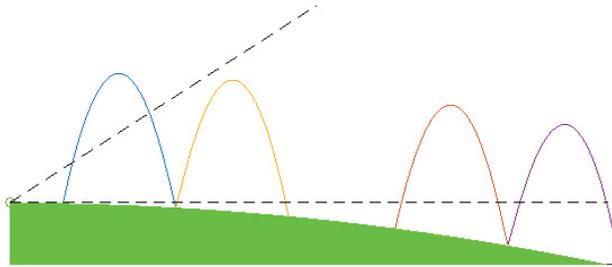


Figure 5. An object trajectory visibility for different launch points and course angles.

The processing areas database includes earlier prepared range and azimuth sectors. Before the start of the TBD algorithm the virtual connection between all sectors and chosen templates is made.

2.2 TBD algorithm module

From the point of view of the whole tracking chain the type of TBD algorithm is insignificant. Depending on the chosen solution data structures of plots or tracks may be extended with additional information. In this article the M/N method called also the binary integrator was used due to its low computational burden and its simple principle of operation. The plots collected from N consecutive scans are associated with earlier prepared templates. If the plots fit into the correlation gate of the template, then a flag variable is set to “one” and otherwise it is set to 0. In a such created sequence of flags there should be at least M “ones” to give a positive decision about object detection.

The templates database contains the correlation gates sizes that has been calculated in a separate application. The type of rocket and its trajectory (lofted, max-range, depressed) along with the earlier described range-dependent phenomenon has been taken into account, as a consequence, the templates are also range- and course-dependent. During the design of the templates the antenna time of revolution has been also considered and the lack of correlation between the launch time and antenna rotations has been assumed. Due to this fact a certain amount of dislocation between the first observable detection of the object and its theoretically assumed position had to be admissible.

Similar assumptions were made in [3], where the correlation gates dimensions are the referred to the radar resolution cell size and the templates include the maximal value of the two. The position of the middle point of the gate is tied to the last detected plot of the candidate track. However this solution has serious consequences. In the case of a stationary or a slowly moving target a false TBM detection could be formed. Moreover, a false candidate plot can dramatically relocate the correlation volume and as a result the object may not be detected. To avoid these side effects, in the presented solution the correlation gate position is dependent on the first analyzed plot and the gates dimensions are described by the minimum and maximum admissible relocation in every coordinate. An example of such prepared template is represented by the black squares in figure 6, where the first plot is marked by the yellow point, the red ones are consecutively detected plots of the object’s trajectory drawn with the blue line.

Considering every coordinate separately allows to narrow the correlation gates dimensions, because in every direction the displacement will vary depending on the object course angle and the range. This idea is presented in figure 7, where by the yellow areas the allowable object’s displacement in reference to the radar in chosen direction is schematically presented. If the object is moving towards the radar, the possible displacement in range is larger than in azimuth. Similarly, if the object course angle is perpendicular to the radar, the displacement in azimuth is dominant. Additional advantages of such solution are the increased separation between templates and the decreased probability of false alarm.

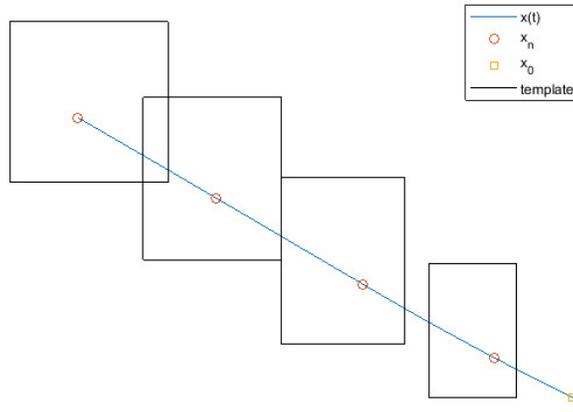


Figure 6. An example of template for $N = 5$.

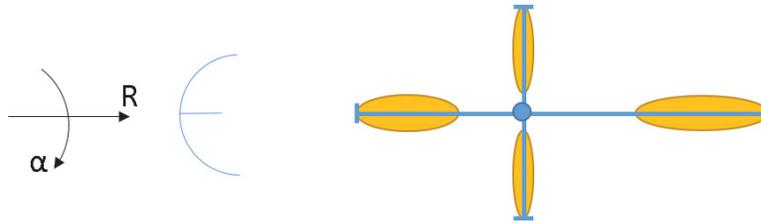


Figure 7. The maximum displacement of object in reference to its course angle.

The decision about a TBM object detection should be formed as soon as possible. Thus, the TBD algorithm starts working from layer 0, 1... up to $N-M$, so the first track will be initiated after M scans. Plots used in this processed are marked by appropriate flag and will not be taken into account in further processes. In this way the number of formed tracks is minimized.

New plots stored in the container's zero layer are firstly used in the updating process of a already formed candidate tracks by their association with the template. Plots uncorrelated neither with older nor newly formed tracks are send to the tracking procedure in the extended Kalman filter.

2.3 Extended Kalmna filter tracking

Basing on the candidate tracks formed by the TBD algorithm the tracking by the extended Kalman filter is initiated. The EKF works with the Newton's model and the 9-elements long state vector including the position, velocity and acceleration in Cartesian coordinate system. The prediction can be described as [6]:

$$\begin{aligned} X_{kp} &= F \cdot X_{k-1} \\ P_{kp} &= F \cdot P_{k-1} \cdot F^T + Q \end{aligned} \quad (1)$$

where:

X_{k-1} – state vector at scan $k-1$,

F – state transition matrix,

X_{kp} – predicted state vector at scan k ,

Q – covariance matrix of discrete process disturbances,

P_{k-1} – covariance matrix of prediction errors at scan $k-1$,

P_{kp} – predicted covariance matrix of filtration errors at scan k .

The filtration can be described as [6]:

$$\begin{aligned}
 u_{kp} &= \begin{bmatrix} \sqrt{X_{kp,1}^2 + X_{kp,4}^2 + X_{kp,7}^2} \\ \operatorname{arctg} \left(\frac{X_{kp,1}}{X_{kp,4}} \right) \\ \operatorname{arctg} \left(\frac{X_{kp,7}}{\sqrt{X_{kp,1}^2 + X_{kp,4}^2}} \right) \end{bmatrix} \\
 H &= \begin{bmatrix} \frac{X_{kp,1}}{u_{kp,1}} & 0 & 0 & \frac{X_{kp,4}}{u_{kp,1}} & 0 & 0 & \frac{X_{kp,7}}{u_{kp,1}} & 0 & 0 \\ \frac{X_{kp,4}}{X_{kp,1}^2 + X_{kp,4}^2} & 0 & 0 & -\frac{X_{kp,1}}{X_{kp,1}^2 + X_{kp,4}^2} & 0 & 0 & 0 & 0 & 0 \\ -\frac{X_{kp,1}X_{kp,7}}{u_{kp,1}^2 \sqrt{X_{kp,1}^2 + X_{kp,4}^2}} & 0 & 0 & -\frac{X_{kp,1}X_{kp,7}}{u_{kp,1}^2 \sqrt{X_{kp,1}^2 + X_{kp,4}^2}} & 0 & 0 & \frac{\sqrt{X_{kp,1}^2 + X_{kp,4}^2}}{u_{kp,1}^2} & 0 & 0 \end{bmatrix} \\
 S &= H \cdot P_{kp} \cdot H^T + R \\
 K_k &= P_{kp} \cdot H^T \cdot S^{-1} \\
 X_k &= X_{kp} + K_k (u_k - u_{kp}) \\
 P_k &= (I - K_k \cdot H) P_{kp}
 \end{aligned} \tag{2}$$

where:

u_{kp} – predicted measurement error,

H – observation model,

S – matrix of innovation,

R – covariance matrix of measurement errors,

K_k – Kalman gain matrix,

u_k – measurement error,

I – identity matrix

and index xx, n means the n -th element of vector.

Dimensions of correlation gates are calculated based on the matrix of innovation with certain coefficients and the middle point of the gate is located at predicted plot position. Most of the TBM objects do not make turning maneuvers, therefore a single EKF is used. A problem may appear near the apogee of the trajectory, where the direction of the movement in elevation is drastically changing. In effect, the use of the second filter should be considered in further work. This feature could also allow to narrow the correlation gates.

The EFKT module can work in two ways. The first option is designed to initiate the tracking basing on the results of the TBD algorithm. The plots from the candidate track are filtered by the EKF to settle the values of particular matrixes and

to shorten the time of its unstable work. Next, the new gate is calculated for the further scan. The plots unused in previous module will be correlated to track, which is the basic function of the EKFT. If any candidate track should be refreshed with a new plot, then the track corresponding to it will also be updated. At the end of this processing chain the plots in the oldest layer are removed from the container leaving the memory space for new data.

Every track contains a parameter describing the number of consecutive prediction passed without updating by a new plot. In the described tracking chain such situation cannot take place more than three times.

3. TESTS OF EFFICIENCY

The test of efficiency of the designed TBM tracking chain was evaluated by two factors: the correctness of tracking evaluated using graphical user interface (GUI) and the time of computations for its different configurations. The reference results were obtained for already described tracking configuration and without any measurement errors. The influence of switching off the options responsible for deleting used plots in the TBD and EKFT modules and the application of candidate track updating using templates. Different values of the detection threshold of the TBD algorithm was examined, the values 3/5, 4/5 and 5/5 were considered. The test scenario contained simultaneously 1, 20, 40, 60, 80 and 100 TBM objects. The launch points and the flight courses were generated randomly. To generate testing data a simulator described in [7] was used.

The way of displaying in GUI different stages of tracking is presented in figure 8. Additionally, points drawn by different colors represent plots stored in different layers of container.

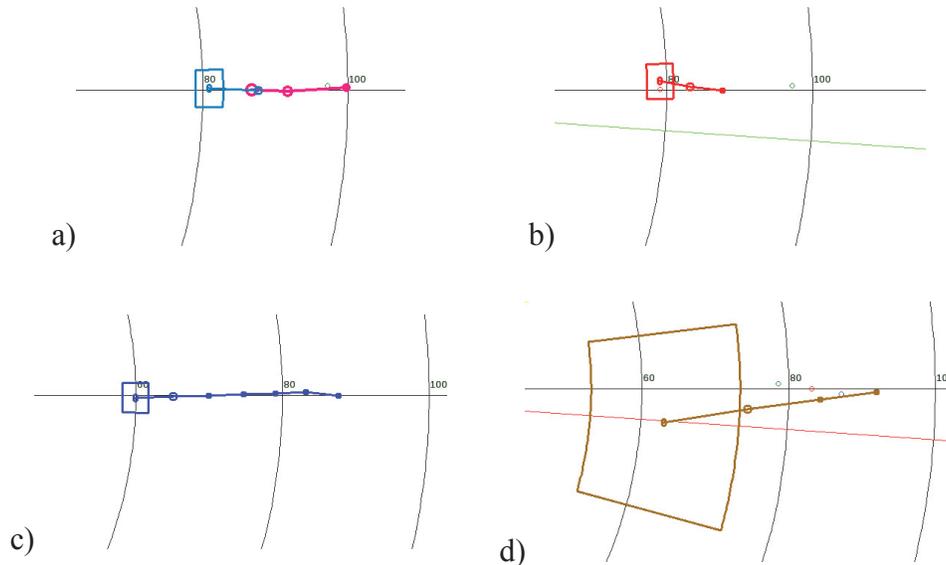


Figure 8. The way of displaying different stages of tracking; a) candidate track formed in the TBD and initiated track in EKFT; b) the track updated using template in TBD module; c) tracking in EKFT; d) prediction without updating by plot in EKFT.

According to the assumptions of TBD tracking chain design, the deletion of data at different stages of software was examined. The results for selected situations from scenario containing 60 TBM objects at once were observed. The example are presented in figure 9.

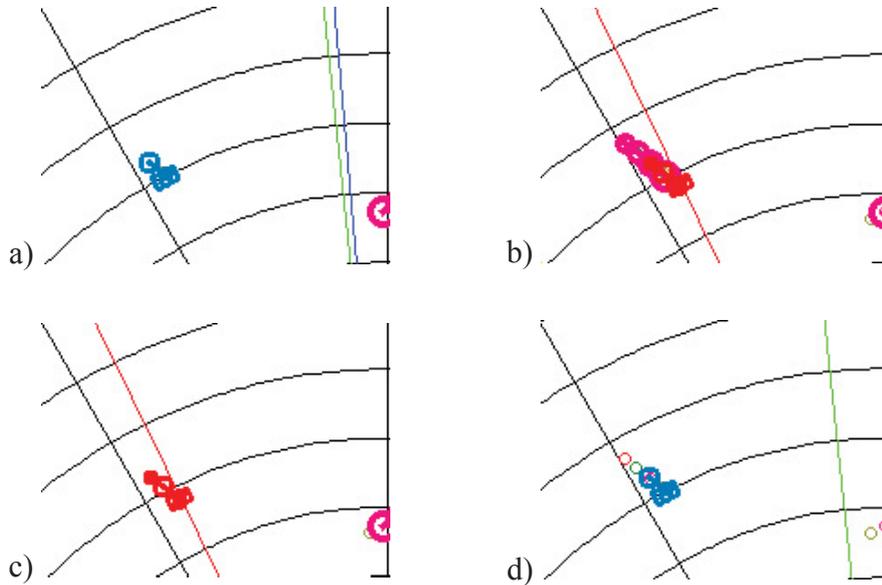


Figure 9. Selected results of TBM object tracking for different configuration of tracking chain; a) the reference picture; b) switched off the deletion of plots used in TBD; c) switched off the deletion of plots used in EKFT; d) switched off deletion of any used plots.

In the reference picture a) the position of last detected plot and the correlation gate predicted for next scan can be easily observed. In the case of picture b), the deletion of the plots used in TBD was switched off and in effect two tracks containing common fragments are formed. They are marked by different colors (pink and red) and they differ from each other in the time of initiation. During the next test the deletion of plots used by the Kalman filter was switched off and the result is in the picture c). Comparing this situation with the reference picture it can be seen that in this case the candidate track updating is made longer. In the last case none of the data are deleted and the result of tracking is the same as in the reference one, but even used plots are still displayed which is the evidence of an unnecessary memory burden.

As the second part of tests, the time of operation of particular parts of tracking chain was measured. Firstly, the dependence on the TBD threshold and the number of object in the scenario were examined. The time was measured by the TBB library. To avoid unnecessary operations, the application's GUI was switched off. The result of 200 s long test is presented in figure 10.

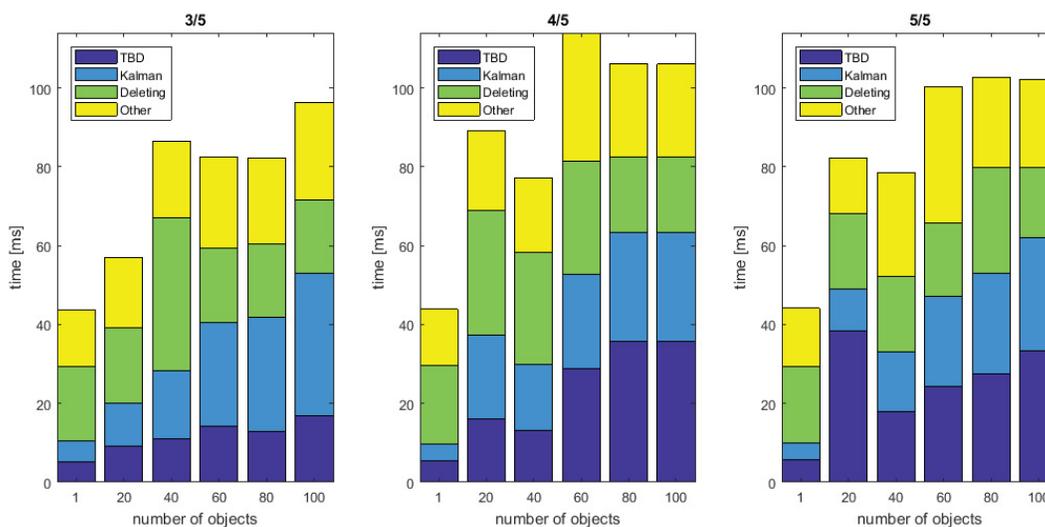


Figure 10. Time of operation of TBM tracking chain for varying TBD threshold and number of objects

The obtained results shows that for increasing the number of objects, the time of operation is also increasing, which should be expected. The nonlinearity of the increase may be caused by the fact that scenarios with higher number of objects are not the extension of the previous ones but are newly generated random situation. In effect the tests are made on completely new data.

The comparison of results with taking into account the TBD threshold shows that its increase does not strictly corresponds to the extended time of computations. But it can be observed that the lower value of M is associated with longer time of work of the TBD algorithm. This may be caused by the increased number of successful candidate tracks updates.

Some of the results are not consistent with the majority, however this may be caused by another system process working in the background simultaneously with the application. To confirm this, additional tests should be performed.

As the next step, the time of computations with disabled deletion of the tracks has been measured. Due to relatively high values of the earlier obtained results, the threshold of $4/5$ has been chosen. The time of operation of particular parts of TBM tracking chain in such case is presented in figure 11.

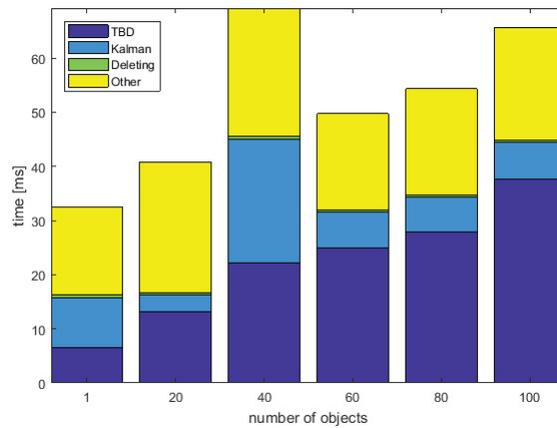


Figure 11. Time of operation of TBM tracking chain for different number of objects when no deletion of used plots is proceeded

As it should be expected, the time of deletion step is significantly shorter because it removes plots only from zero layer of the data container. In such configuration the computational burden related to searching elements in containers is avoided. In consequence the total time of operation decreased, however there is an increased probability of using the same plot to update many tracks which was considered earlier.

The influence of a candidate track updating at the TBD algorithm stage on time of tracking chain operation was also examined. The results are presented in figure 12.

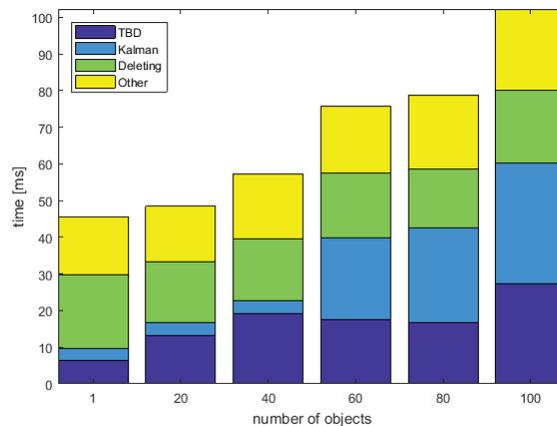


Figure 12. Time of operation of TBM tracking chain for different number of objects when candidate track update is not proceeded.

As it can be seen, the lack of candidate track update causes the decrease of time operation of TBD algorithm and of the whole tracking chain as a consequence. But at the design stage it was assumed that the TBD algorithm is more reliable than the classical tracking using EKF. Therefore, shorter time of operation is not a priority in this case.

4. SUMMARY

In the article a tracking chain of tactical ballistic missiles based on the TBD algorithms and extended Kalman filter was presented. The results of tracking of the simulated TBD objects were shown. The time of operation of particular steps of the chain was measured. The tests proved that the disabling some functions will increase computation speed but it is associated with a decreased reliability of tracking. In spite of this, the time of tracking chain work is admissible to practical solution.

The presented tracking chain realizes its basic function and it allows to extend it with additional options. The easiest way to increase its effectiveness is to provide the regulation of algorithms parameters at every step of the chain, particularly, the correlation gates coefficients and TBD thresholds that are not used any more. Also the multi-hypothesis approach should be considered to increase the quality of results.

REFERENCES

- [1] Gish H. and Mucci: R., "Target state estimation in a multi-target environments, " IEEE Trans. A. Elec. Sys. 23(1), 60-72 (1987)
- [2] Blackamn S. and Popoli R., [Design and Analysis of Modern Tracking Systems], Artech House, Norwood, (1999)
- [3] Sankowski M., Uruski P. and Blok E., "Application of Track-Before-Detect Algorithm to Detection of Ballistic Missiles in Air-Defense Radar," 15th Int. Conf. on Mic., Radar and Wireless Communications, 97-100 (2004).
- [4] Amditis A., Thomaidis G., Maroudis P., Lytrivis P. and Karaseitanidis I., "Multiple Hypothesis Tracking Implementation" in [Laser Scanner Technology] edited by Rodriguez A.M., InTech, (2012).
- [5] Barniv Y., "Dynamic programming algorithm for detecting dim moving targets" in [Multitarget-Multisensor Tracking: Advanced Application] edited by Bar-Shalom Y., Artech House, Norwood, USA (1990).
- [6] Kaniewski P., Smagowski P. and Konatowski S., "Ballistic Target Tracking with Use of Cinetheodolites," International Journal of Aerospace Engineering 2019(3240898),1-13 (2019).
- [7] Wajszczyk B., Motyl K. and Sienicki K., "The concept of the radar environment simulator software," Proc. SPIE 10715, 107150G (2018).