Passive Coherent Locator Tracking and Data Fusion

Krzysztof Kulpa, Mateusz Malanowski

Institute of Electronic Systems

Warsaw University of Technology

Nowowiejska 15/19, 00-665 Warszawa

k.kulpa@elka.pw.edu.pl, m.malanowski@ elka.pw.edu.pl

ABSTRACT

Passive radar is mainly aimed for detection and tracking of airborne targets using emitters of opportunity for target illumination. As they multistatic in nature, the main problem is that the measurements are based on bistatic range and range velocity estimation and main problem is in data fusion, while it is necessary to fuse the data from different pairs of transmitter-receiver to form and update tracks. In most cases the measurement provide only range Doppler information and no angular information, so in such a case target position can be estimated solving nonlinear set of equations and finding best possible association. In the notes the fundamentals of positioning and tracking are presented. Theoretical considerations are illustrated with selected simulation results.

1.0 INTRODUCTION

The typical scenario of passive radar measurements is presented in Fig. 1.

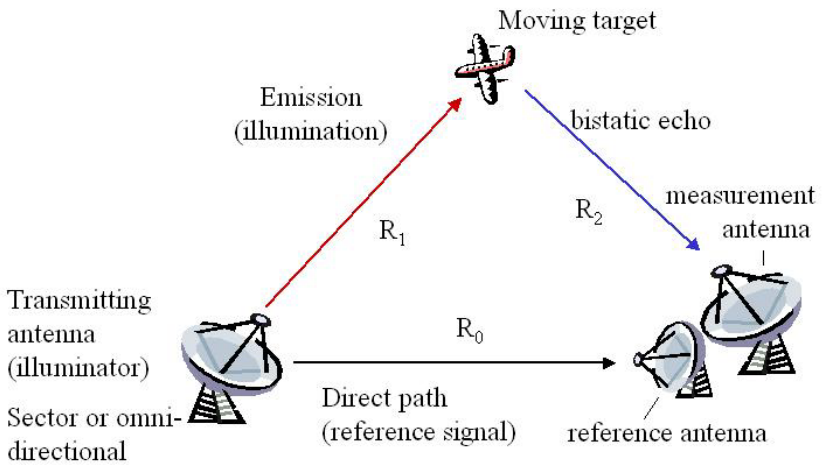

Fig. 1.Typical scenario of target detection using PCL radar.

The PCL radar measures the target position and position changes in the bistatic coordinates. Tt measure the time difference of arrival between two signals – direct end reflected, which is proportional to the bistatic range and the Doppler shift between those two signals which is proportional to the bistatic velocity (time derivative of bistatic range).

The two basic ideas are used for target localization and tracking. One is based on the bistatic range and angular of arrival concept. If the passive radar can measure the angle of arrival of the echo signal (using e.g. interferometric technique, when two or more reference antennas are in use) as depicted in Fig. 2.

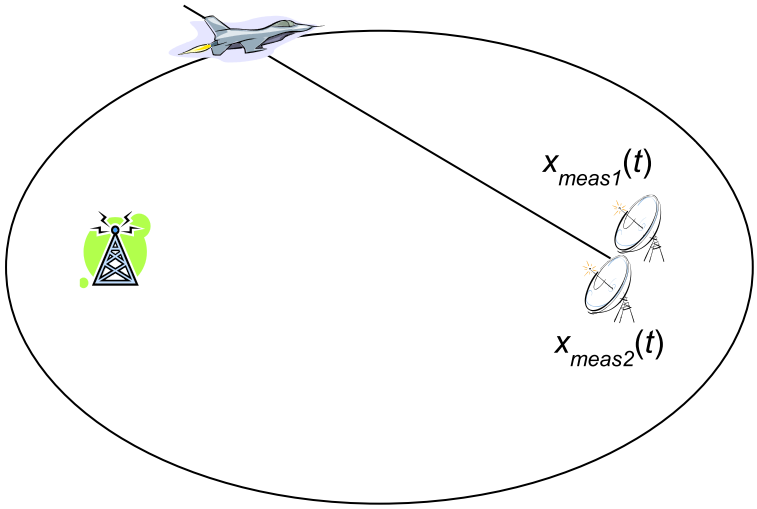

Fig. 2.Target localization based on angular measurements.

In most cases angular information is not accurate enough, so the precise tracking can be performed finding the crossing of at least 3 ellipsoids, as depicted in Fig.3.

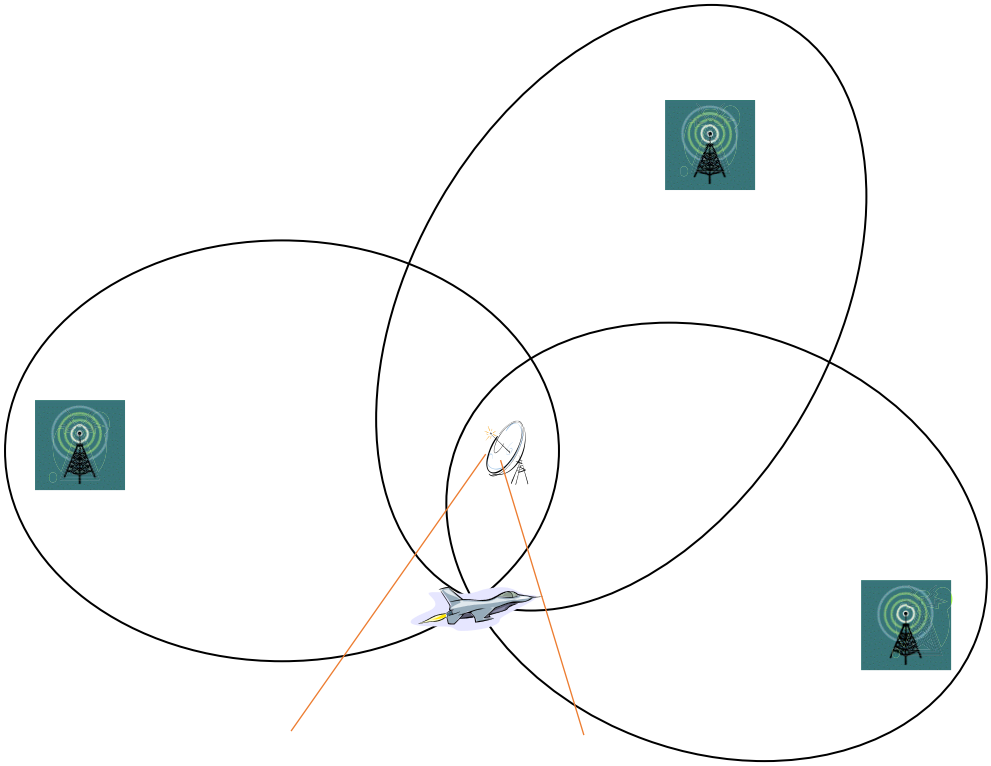

Fig. 3.Target localization based on range measurements only – finding the ellipsoid crossing.

In such a case the angular measurements can also be used for improving accuracy (in case of precise angular measurements) and for improving association problem, when only ellipsoid crossing in the direction pointed by angle estimator are considered ad true one.

Please note, that in presented case we have single point of 3 ellipsoid crossing and two punts where two of them crossing. Such points can be also take into account and it could create false targets.

There are several aims of the tracker – the primary one is to provide the initiation and tracking of all detected targets, but to perform this it is necessary to eliminate false bistatic plots, initiate new probe tracks, associate bistatic plots to the existing tracks, estimate the track quality, update tracks and terminate tracks when no sufficient track updates are available.

In this paper, a two-stage tracking algorithm is presented. First, the plots corresponding to different transmitters are used by separate bistatic trackers. In this way most of false measurements are separated from true target detections. In the second state the bistatic tracks are combine to form the Cartesian tracks. It can be performed using different approach, but one of the most promising is to use Extended Kalman Filter (EKF) feed by the bistatic plots taken from bistatic tracker. As the result the second stage is based on the EKF. The filter uses the raw plots associated by the bistatic tracker.

A similar approach of multi-stage algorithm was presented in [4] in the context of target tracking using a single frequency network of DAB/DVB-T transmitters.

2.0 BISTATIC TRACKING

The instantaneous bistatic range (proportional to the time difference of arrival) can be calculated from :

$$r(t)=\sqrt{(x-x_t)^2+(y-y_t)^2+(z-z_t)^2}+\sqrt{(x-x_r)^2+(y-y_r)^2+(z-z_r)^2}-r_b, \tag{1}$$

where (x, y, z) is the target position, (x_t, y_t, z_t) is the transmitter position, (x_r, y_r, z_r) is the receiver position, and r_b is the base line length (distance between transmitter and receiver). The instantaneous bistatic velocity (proportional to the Doppler shift) calculated as the first derivative of the bistatic range has the following form:

$$v(t)=\frac{(x-x_t)v_x+(y-y_t)v_y+(z-z_t)v_z}{\sqrt{(x-x_t)^2+(y-y_t)^2+(z-z_t)^2}}+\frac{(x-x_r)v_x+(y-y_r)v_y+(z-z_r)v_z}{\sqrt{(x-x_r)^2+(y-y_r)^2+(z-z_r)^2}}, \tag{2}$$

where (v_x, v_y, v_z) is the vector of target velocities. In the sequel, the equations (1) and (2) will be used for constructing the EKF. For the purpose of the tracking in the bistatic coordinates, the bistatic range can be approximated by a second-degree polynomial:

$$r(t)=R+Vt+\frac{At^2}{2}, \tag{3}$$

where R is the bistatic range, V is the bistatic velocity and A is the bistatic acceleration.

One can construct a tracker operating in bistatic coordinates based on (3) using an almost-constant-acceleration model [5, 6]:

$$\mathbf{x}_b(k)=\mathbf{F}_b\mathbf{x}_b(k-1)+\mathbf{v}_b(k), \tag{4}$$

where $\mathbf{x}_b(k)=[R(k), V(k), A(k)]'$,

$$\mathbf{F}_b = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

and $\mathbf{v}_b(k)$ is uncorrelated Gaussian process noise with covariance:

$$\mathbf{Q}_b = E\left[\mathbf{v}_b(k)\mathbf{v}_b'(k)\right] == \sigma_w^2 \begin{bmatrix} T^4/4 & T^3/2 & T^2/2 \\ T^3/2 & T^2 & T \\ T^2/2 & T & 1 \end{bmatrix}. \tag{6}$$

The measurement is modeled as:

$$\mathbf{z}_b(k) = \mathbf{H}_b \mathbf{x}_b(k) + \mathbf{w}_b(k), \tag{7}$$

where $\mathbf{z}_b(k) = \left[\hat{R}(k), \hat{V}(k)\right]$,

$$\mathbf{H}_b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{8}$$

and $\mathbf{w}_b(k)$ is uncorrelated Gaussian measurement noise with covariance:

$$\mathbf{R}_b = E\left[\mathbf{w}_b(k)\mathbf{w}_b'(k)\right] = \begin{bmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_V^2 \end{bmatrix}. \tag{9}$$

Based on (3) and (4), standard linear Kalman filtering equations can be used [5, 6].

State prediction

$$\hat{\mathbf{x}}_b(k \mid k-1) = \mathbf{F}_b \hat{\mathbf{x}}_b(k-1 \mid k-1), \tag{10}$$

Covariance of prediction

$$\mathbf{P}_b(k \mid k-1) = \mathbf{F}_b \mathbf{P}_b(k-1 \mid k-1)\mathbf{F}_c' + \mathbf{Q}_c, \tag{11}$$

Kalman Gain estimation

$$\mathbf{S}_b(k) = \mathbf{H}_b(k)\mathbf{P}_b(k \mid k-1)\mathbf{H}_b'(k) + \mathbf{R}, \tag{12}$$

$$\mathbf{K}_b(k) = \mathbf{P}_b(k \mid k-1)\mathbf{H}_b'(k)\mathbf{S}_b^{-1}(k), \tag{13}$$

State filtration

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{K}(k)\left(\mathbf{z}(k) - h\left(\hat{\mathbf{x}}(k \mid k-1)\right)\right), \tag{14}$$

$$\mathbf{P}_b(k \mid k) = \left(\mathbf{I} - \mathbf{K}_b(k)\mathbf{H}_b(k)\right)\mathbf{P}_b(k \mid k-1). \tag{15}$$

3.0 CARTESIAN TRACKING

The relationship between bistatic and Cartesian parameters is nonlinear, as can be seen from (1) and (2). There are several solutions for such problem. In radar field very often the measurements are converted from their native coordinates to the Cartesian coordinates and then tracking is performed using classical Kalman or Particle filters. But much simple solution can be achieved using extended Kalman Filter which uses linear approximations of the nonlinear equations.

The target state vector consisting of position and velocities can be expressed in Cartesian coordinates as : $\mathbf{x}_c(k) = \left[x(k), v_x(k), y(k), v_y(k), z(k), v_z(k)\right]$. The state evolution process can then be described as :

$$\mathbf{x}_c(k) = \mathbf{F}_c \mathbf{x}_c(k-1) + \mathbf{v}_c(k), \tag{16}$$

where $\mathbf{F}_c = diag(\mathbf{F}, \mathbf{F}, \mathbf{F})$, with

$$\mathbf{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \tag{17}$$

and $\mathbf{v}_c(k)$ is uncorrelated Gaussian process noise with covariance $\mathbf{Q}_c = diag\left(\sigma_{wx}^2\mathbf{Q}, \sigma_{wy}^2\mathbf{Q}, \sigma_{wz}^2\mathbf{Q}\right)$ where

$$\mathbf{Q} = \begin{bmatrix} T^2/2 & T \\ T & 1 \end{bmatrix}. \tag{18}$$

The measurement corresponding to the *m*-th transmitter can be modeled as :

$$\mathbf{z}_b^m(k) = h^m\left(\mathbf{x}_c(k)\right) + \mathbf{w}_b^m(k), \tag{19}$$

where $\mathbf{z}_b^m(k) = \left[\hat{R}^m(k), \hat{V}^m(k)\right]$ is the measurement vector, and $h^m(\cdot)$ is a nonlinear function transforming the Cartesian parameters to bistatic parameters according to equations (1) and (2). The covariance of the measurement noise is $\mathbf{R}_b$ as stated in equation (9). The Extended Kalman Filter works almost as the standard one, but the matrix H is calculated as linearization of the function $h^m(\cdot)$ at the predicted position $\hat{\mathbf{x}}(k \mid k-1)$ with respect to state vector elements:

$$\mathbf{H}_{[j,i]}^m(k) = \frac{\partial h_{[i]}^m\left(\hat{\mathbf{x}}(k \mid k-1)\right)}{\partial \mathbf{x}_{[j]}}. \tag{20}$$

The explicit form of H can be found in [10. The prediction in the considered case has the following form:

$$\hat{\mathbf{x}}_c(k \mid k-1) = \mathbf{F}_c \hat{\mathbf{x}}_c(k-1 \mid k-1), \tag{21}$$

$$\mathbf{P}_c(k \mid k-1) = \mathbf{F}_c \mathbf{P}_c(k-1 \mid k-1)\mathbf{F}_c' + \mathbf{Q}_c, \tag{22}$$

where $\mathbf{P}_c(k \mid k-1)$ is the *a priori* state covariance matrix and $\mathbf{P}_c(k-1 \mid k-1)$ is the *a posteriori* covariance matrix.

There are two methods for updating the estimate in a multisensory scenario: parallel and sequential [6]. In the

parallel method, the measurements $\mathbf{z}_b^m(k)$, and the matrices $\mathbf{H}^m(k)$ are stacked and the update is performed in one step. In the sequential method, the state vector and covariance are updated with measurements corresponding to each transmitter separately. When the system is linear, the both methods are equivalent, however, the first is more computationally expensive as it requires operations on larger vectors and matrices and can be less numerically stabile, while it is necessary to invert much bigger matrices. In our case the system is not linear, but the results are almost this some as long as the initialization of process is close to the true object position. , as in the considered case, one should start with the most accurate measurement to minimize linearization errors.

3.1 PARALLEL UPDATING

In parallel update concept all measurement data is used in one single step as depicted in Fig. 4.

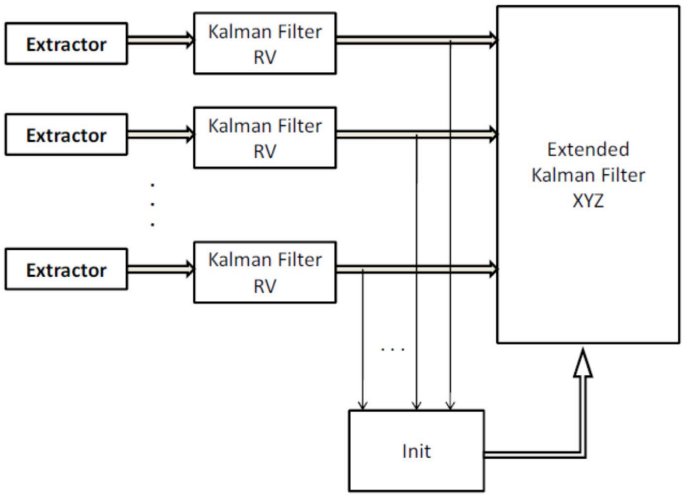

Fig. 4.Parralel update Kalman Filter.

The measurement vector as well as the state-to-measurement matrix is stacked to form the quantities corresponding to all three (or more) transmitters:

$$\mathbf{z}(k)=\begin{bmatrix}\mathbf{z}_b^1(k)\\ \mathbf{z}_b^2(k)\\ \mathbf{z}_b^3(k)\end{bmatrix}, \tag{23}$$

$$\mathbf{H}(k)=\begin{bmatrix}\mathbf{H}^1(k)\\ \mathbf{H}^2(k)\\ \mathbf{H}^3(k)\end{bmatrix}. \tag{24}$$

The covariance matrix of the measurement error for all transmitters is also created from matrices for separate transmitters:

$$\mathbf{R}=diag(\mathbf{R}_b,\mathbf{R}_b,\mathbf{R}_b). \tag{25}$$

The update equations are as follows: Gain update

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}_c(k \mid k-1)\mathbf{H}'(k) + \mathbf{R}, \tag{26}$$

$$\mathbf{K}(k) = \mathbf{P}_c(k \mid k-1)\mathbf{H}'(k)\mathbf{S}^{-1}(k), \tag{27}$$

State vector update

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{K}(k)(\mathbf{z}(k) - h(\hat{\mathbf{x}}(k \mid k-1))), \tag{28}$$

Covariance matrix update

$$\mathbf{P}_c(k \mid k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}_c(k \mid k-1). \tag{29}$$

Of course, the structure of the filter is varying when number of measurements assigned to the track is varying. As result it is necessary to design different Kalman filters for different number of measurements, and all filters have different sizes of matrices. It is also possible to design single filter for maximum number of TxRx pairs and in the case of lack of target detection corresponding to one of the tx-rx pair use the predicted value from the bistatic tracker and modify the measurement covariance matrix $\mathbf{R}_b$ corresponding to that pair (set to an infinite value what is numerically difficult or at least to very high value assuring stable result) to provide information of the tack of measurement.

3.2 SEQUENTIAL UPDATING

Instead of using parallel update it is possible to use sequential update. In such approach the whole filtering is divided into several stages. In the first stage the prediction is being performed as presented in the Fig. 5.

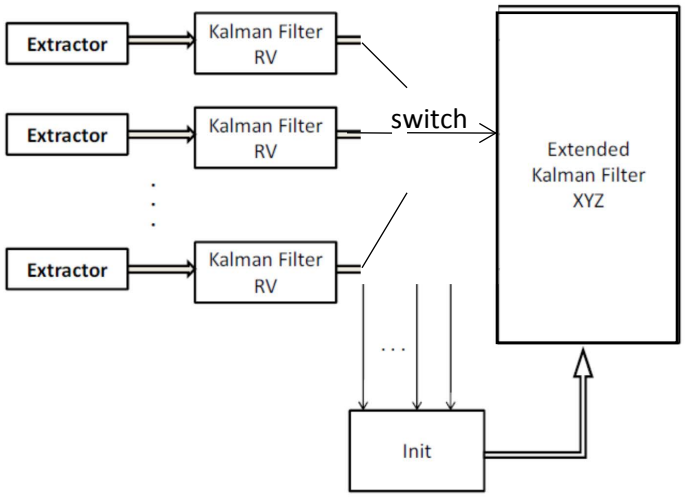

Fig. 5.Serial update Kalman Filter.

In the sequential method, the state estimate and covariance are updated by each measurement separately. First, temporary variables are created:

$$\hat{\mathbf{x}}^0(k \mid k-1) = \hat{\mathbf{x}}(k \mid k-1), \tag{30}$$

$$\mathbf{P}_c^0(k \mid k-1) = \mathbf{P}_c(k \mid k-1). \tag{31}$$

Next, the sequential updating is performed for each of the M measurements ($m=1..M$)

$$\mathbf{S}^m(k) = \mathbf{H}^m(k)\mathbf{P}_c^{m-1}(k \mid k-1)\mathbf{H}^{m\prime}(k) + \mathbf{R}_b, \tag{32}$$

$$\mathbf{K}^m(k) = \mathbf{P}_c^{m-1}(k \mid k-1)\mathbf{H}^{m\prime}(k)\left(\mathbf{S}^m\right)^{-1}(k), \tag{33}$$

$$\hat{\mathbf{x}}^m(k \mid k-1) = \hat{\mathbf{x}}^{m-1}(k \mid k-1) + \mathbf{K}^m(k)\left(\mathbf{z}_b^m(k) - h^m\left(\hat{\mathbf{x}}(k \mid k-1)\right)\right), \tag{34}$$

$$\mathbf{P}_c^m(k \mid k-1) = \left(\mathbf{I} - \mathbf{K}^m(k)\mathbf{H}^m(k)\right)\mathbf{P}_c^{m-1}(k \mid k-1). \tag{35}$$

The final stage is consists of updated of state estimate and covariance matrix:

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}^M(k \mid k-1), \tag{36}$$

$$\mathbf{P}_c(k \mid k) = \mathbf{P}_c^M(k \mid k-1). \tag{37}$$

In sequential update procedure the structure of Kalman Filter is fixed – reacted to single measurement and updates are perform only to measurements assigned to the track. In such case there is no necessity to artificially marked missing plots, and algorithm is very universal. It can use as many receiver-transmitter pairs and can be also adapted to different measurements (PCL radars with/without angle estimation, active monostatic radars, PET devices based on DOA or TDOA methods and many more).

3.2 SIMULATIONS

In the first example, three targets were simulated. The target trajectories were generated using the assumed model (10) with σ_{wx} = σ_{wy} = 1 m/s and σ_{wz} = 0.1 m/s. The receiver *Rx* was located at (0, 0, 0) km. The transmitters *Tx1*, *Tx2* and *Tx3* were located at (42, 40, 0) km, (-20, 0, 0) km and (10, -40, 0) km, respectively. The refresh interval T was equal to 1 s. The target trajectories, the transmitters and the receiver are shown in Fig. 6.

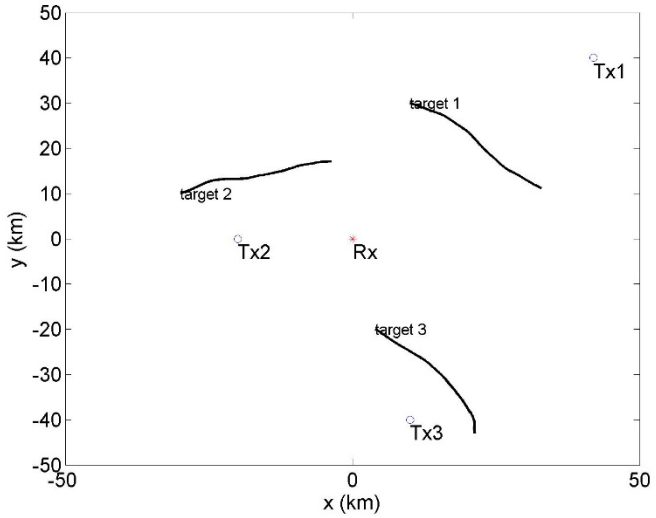

Fig. 6. Target trajectories in Cartesian coordinates

The Cartesian parameters were converted into bistatic ones using (1) and (2). The probability of detection was set to 0.7 for every transmitter and every target. The bistatic measurements were disturbed by adding Gaussian errors according to (7) with σ_R = 500 m and σ_V = 1 m/s – parameters typical for FM-based system. Figs. 7, 8 and 9 show true, measured and tracked bistatic parameters corresponding to one of the targets (target 1). The error of bistatic measurements can be clearly seen, especially for the bistatic range. This is the result of low bandwidth of the FM signal (approx. 100 kHz), which leads to low range resolution (app. 3 km resolution cell). As can be seen, the Kalman filters working in the bistatic coordinates increase the accuracy of bistatic range measurement significantly. However, as was indicated earlier, the bistatic trackers are used only for data association – the EKF uses the raw plots and not the state estimate of the bistatic trackers.

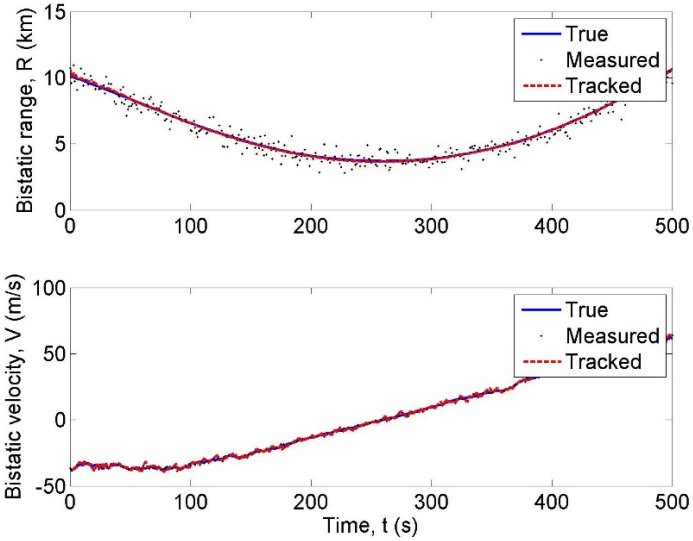

Fig. 7. True, measured and tracked bistatic parameters versus time for transmitter Tx1 and target 1.

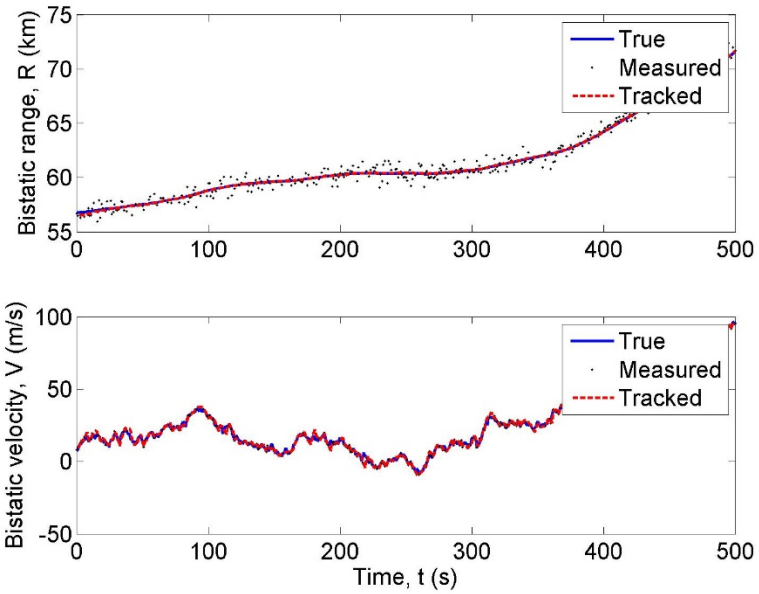

Fig. 8. True, measured and tracked bistatic parameters versus time for transmitter Tx2 and target 1.

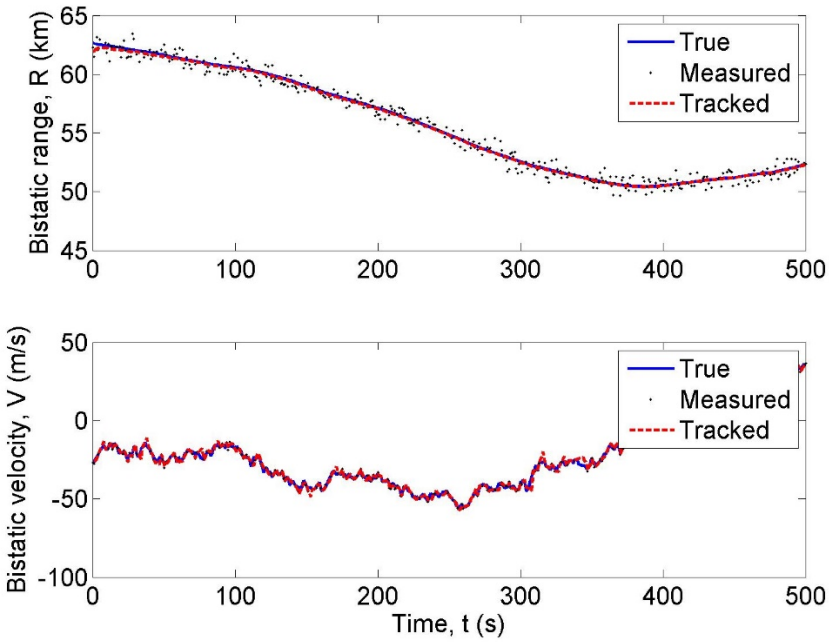

Fig. 9.True, measured and tracked bistatic parameters versus time for transmitter Tx3 and target 1.

Fig. 10 shows the true and tracked Cartesian coordinates of target 1 versus time using parallel updating. In can be seen that parameters in x and y coordinates are tracked with high accuracy. The estimate of the z coordinate is characterized by a large error, which decreases slowly in time. This is a well-known phenomenon – due to the geometrical relationships, the accuracy of the height estimation is worse than the accuracy of x and y coordinates.

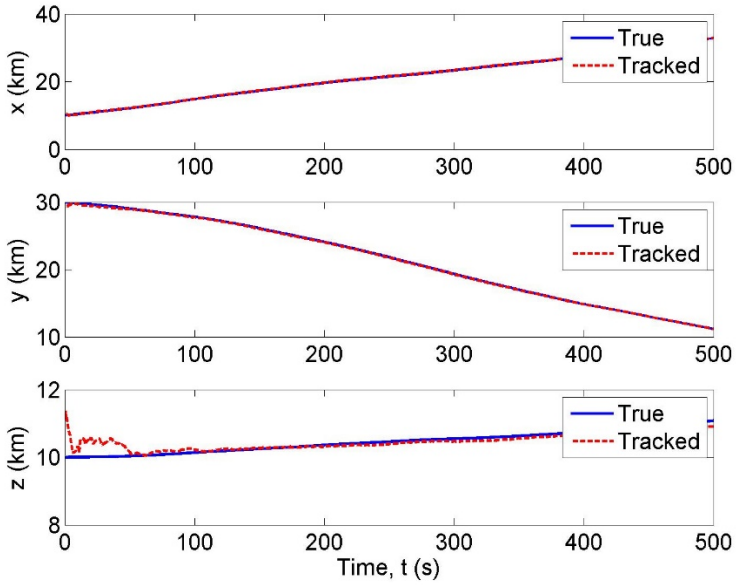

Fig. 10. True and tracked Cartesian coordinates versus time for target 1 (parallel updating).

In Fig. 11, a comparison of the theoretical (filter-calculated) and the actual position errors Δ_x, Δ_y and Δ_z for parallel updating is presented. The real error was calculated as a standard deviation of the difference between true and tracked position coordinate. Filter-calculated accuracy was the averaged value of the appropriate element of the $\mathbf{P}_c(k|k)$ matrix. As could be inferred from the previous figure, the accuracy of the height estimation is worse than the accuracy of x and y coordinates. In the case of x and y coordinates accuracy of hundreds of meters can be expected, which decreases to tens of meters during the course of tracking. The initial height accuracy is over 1 km and decreases to hundreds of meters. One can also observe that the actual and theoretical accuracies are very similar. This would not be true if the state estimates from the bistatic tracker were used in EKF – the filter would calculate the covariance too optimistically because of the correlation of the input measurements.

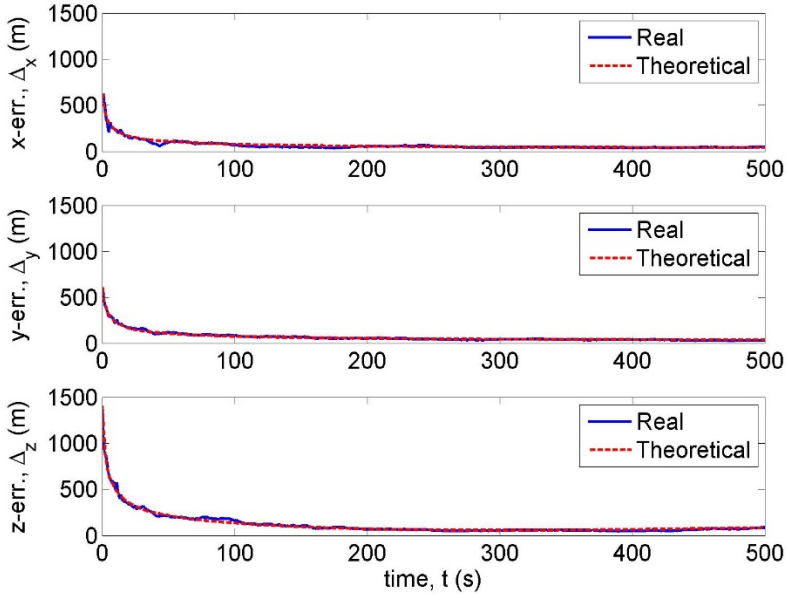

Fig. 11. Real and filter-calculated x, y, and z position accuracies versus time for target 1 (parallel

updating).

Fig. 12 shows the same position errors obtained using the sequential updating approach. As can be seen, the results are very similar to the structure with parallel updating. It is worth mentioning that using only bistatic measurements, without bearing information, it is possible to achieve 3D tracking with acceptable accuracy, not worse than in classical active radars.

The actual and theoretical errors corresponding to the three velocity components Δ_{Vx}, Δ_{Vy} and Δ_{Vz} are shown in Fig. 13. In this case, inferior accuracy in *z* dimension is also visible.

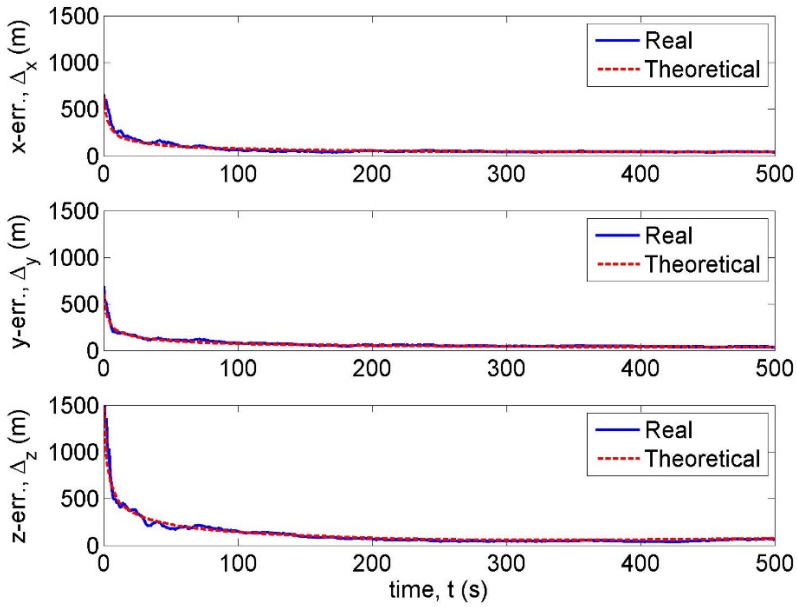

Fig. 12. Real and filter-calculated *x*, *y*, and *z* position accuracies versus time for target 1 (sequential updating).

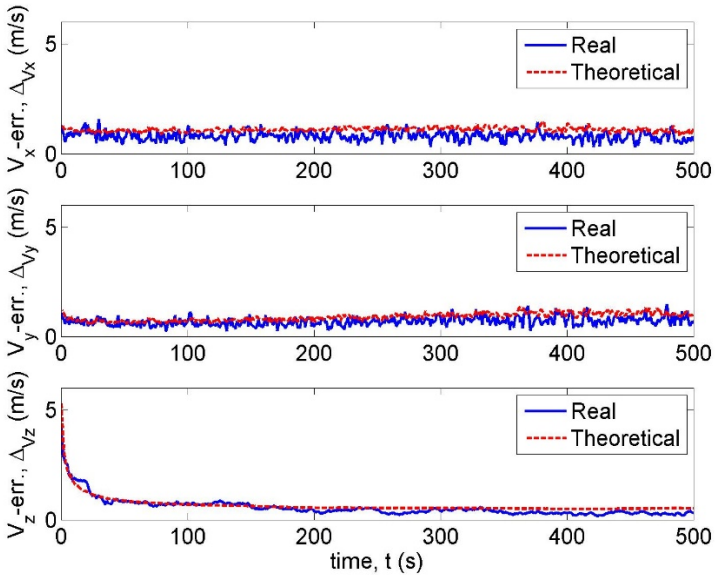

Fig. 13. Real and filter-calculated *x*, *y*, and *z* velocity accuracies versus time for target 1 (parallel

updating).

In order to compare the proposed two-stage algorithm with a one-stage tracker using the EKF only, the following simulation was carried out. Three targets were simulated and their bistatic parameters were calculated. Apart from the target-originated detections, false alarms were added to the bistatic measurements. The detections were generated uniformly on the bistatic range – bistatic velocity surface. It was assumed that the crossambiguity of the reference and surveillance signals is calculated for bistatic range (0, 300) km and bistatic velocity (-400, +400) m/s. The bistatic range resolution cell was 3 km and the bistatic velocity resolution cell was 2 m/s. As a result, 100 range resolution cells and 400 velocity resolution cells were obtained – 40000 cells in total. The number of false detections were generated according to the Poisson distribution based on the number of the resolution cells and the probability of false alarm (P_{fa}). In the one-stage tracker, the raw bistatic plots were used. In the two-stage algorithm, the bistatic trackers used 2/3 cascaded initialization logic. Only the bistatic plots corresponding to the initialized bistatic tracks were used in the EKF-based tracker. In both cases, one- and two-stage algorithm, a Cartesian track was created for each output of the ellipsoid intersection algorithm, which tests every possible combination of the bistatic measurements in search for simultaneous ellipsoid intersection. The simulation was repeated 100 times.

The table I shows the result of the comparison of the two approaches for different values of the P_{fa}. The number in the table indicate the average number of "ghost Cartesian tracks" – Cartesian tracks initialized by ghost targets. The results show that the number of ghost tracks in the case of two-stage tracker is reduced, especially in the case of high P_{fa}. It results from the fact that bistatic trackers in the two-stage algorithm eliminate the false detections almost completely. The remaining ghost targets result from the random intersections of the bistatic ellipsoids corresponding to the true targets.

The elimination of the false alarms also leads to reduced computational complexity of the intersection finding algorithm. The number of the possibilities which the algorithm has to test is $\prod_i N_i$, where N_i is the number of bistatic measurements corresponding to the *i*-th transmitter. As can be easily seen, the number of possibilities to test grows very fast with N_i. In the one-stage tracker, N_i is equal to the number of targets (assuming probability of detection equal to 1) plus the number of false detections. In the two-stage tracker, N_i very close to the number of targets – random false detections cause track initialization very rarely.

P_{fa}	One-stage tracker	Two-stage tracker
10^{-3}	212.90	0.30
10^{-4}	1.40	0.24
10^{-5}	0.43	0.23

Tab. I. A comparison of the average number of ghost tracks for one- and two-stage algorithm.

The disadvantage of the two-stage approach is the time delay associated with the process of bistatic track initialization. The Cartesian track is created only after the bistatic tracks corresponding to all transmitters are initialized. In the case of *M*/*N* initialization logic, at least *M* observations are required to confirm a track. If the probability of detection is low, the average number of observations needed for track confirmation may be much larger.

The computational complexity of additional bistatic trackers in the two-stage approach is compensated by the reduced number of ellipsoid intersections tests due to the false detections elimination.

References

[1] P. E. Howland, D. Maksimiuk, and G. Reitsma, “FM Radio Based Bistatic Radar,” *IEE Proc.-Radar, Sonar and Navigation*, vol. 152, 3 Jun. 2005, pp. 107–115.

[2] S. Herman, and P. Moulin „A Particle Filtering Approach to FM-Band Passive Radar Tracking and Automatic Target Recognition,” in *Proc. IEEE Aerospace Conference 2002*, vol. 4, pp. 4-1789– 4-1808.

[3] M. Tobias, and A. D. Lanterman, “Probability Hypothesis Density-Based Multitarget Tracking With Bistatic Range and Doppler Observations,” *IEE Proc. Radar, Sonar and Navigation*, vol. 152, Issue 3, 3 Jun. 2005, pp. 195–205.

[4] M. Daun, W. Koch, “Multistatic target tracking for non-cooperative illumination by DAB/DVB-T,” in *Proc. IEEE Radar Conference 2008*, 26-30 May 2008, pp. 1-6.

[5] Y. Bar-Shalom, and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, 1993.

[6] Y. Bar-Shalom, and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Yaakov Bar-Shalom, 1995.

[7] F. R. Castella, “Sliding window detection probabilities,” *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-12, no. 6, pp. 815–819, Nov. 1976.

[8] M. Malanowski, “An Algorithm for 3D Target Localization from Passive Radar Measurements”, in *Proc. Signal Processing Symposium 2009*, 28-30 May 2009, Jachranka, Poland.

[9] K. C. Ho, Y. Chan, “Solution and performance analysis of geolocation by TDOA”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 29, October 1993, pp. 1311–1322.

[10] R.K. Mehra, "A Comparison of Several Nonlinear Filters for Reentry Vehicle Tracking", *IEEE Trans. on Automatic Control*, vol. ac-16, no. 4, August 1971