

RESEARCH

Open Access

Track-before-detect in distributed sensor applications

Felix Govaers^{1*}, Yang Rong², Lai Hoe Chee², Wolfgang Koch¹, Teow Loo Nin² and Ng Gee Wah²

Abstract

In this article, we propose a new extension to a Dynamic Programming Algorithm (DPA) approach for Track-before-Detect challenges. This extension enables the DPA to process time-delayed sensor data directly. Such delay might appear because of delays in communication networks. The extended DPA is identical to the recursive standard DPA in case of all sensor data appear in the timely correct order. Furthermore, an intense evaluation of the Accumulated State Density (ASD) filter is given on simulation data. Last but not least, we apply a combination of DPA and ASD on data of a real radar system and present the resulting tracks. Our experience concerning this combination is a seamless cooperation between the track initialization by DPA and a track maintenance by ASD filter.

Keywords: Track-before-detect, Out-of-sequence, Real data application, Dynamic programming approach, Accumulated state density, TBD, OOSM, DPA, ASD

1. Introduction

Since many years, security applications employing radar sensors for surveillance objectives are increasingly important. In situations where targets with a low signal-to-noise ratio (SNR) appear, it is convenient to apply tests on track existence utilizing raw sensor data instead of using thresholded measurements. This approach is generally called Track-before-Detect (TBD). It enables a radar system to search for low-observable targets (LOTs), i.e., objects with a low SNR. These targets can be invisible to conventional methodologies, as most of the information about them might be cut off by the applied threshold. The gain of a TBD algorithm is often paid by high computational costs. Even today, when computational power is cheap and highly available, most of the techniques for TBD still suffer from being hard to realize for a real time processing of sensor data. First and foremost, this is due to the huge amount of data to be considered in each scan.

Capacity and stability of communication channels such as 3G Networks, WLAN, HF, or WANs are subject to an ever increasing development. For many fusion applications, in particular for surveillance tracking, this enables a user to explore new approaches by exploiting

multiple sensor systems. When the link capacity is very low or temporarily unavailable, a common centralized tracking scheme is Track-to-Track Fusion (T2TF) [1]. However, T2TF neglects valuable information on LOTs, as track initialization is performed only on local sensor data. Therefore, we address the challenge of TBD and track maintenance (TM) in distributed sensor applications by processing all information available depending on the available bandwidth.

Applications evolving multiple distributed sensors often suffer from effects of the communication links. The major challenge therein constitute in particular time-delayed sensor data, so called Out-of-Sequence (OoS) measurements, which appear, e.g., by timely misaligned scan rates, varying communication delays, or asynchronous sensors caching their data in a local storage. To overcome this challenge, the Accumulated State Densities (ASDs) filter gives a neat and efficient scheme to process such OoS measurements [2-4]. Therefore, the ASDs give an optimal estimation filter for distributed sensor applications performing the TM part.

1.1. Structure

This article is structured as follows. In Sect. 2, an overview to related work is given. The main contribution of this article is a TBD algorithm which is able to process

* Correspondence: felix.govaers@fkf.fraunhofer.de

¹Fraunhofer-FKIE, Wachtberg, Germany

Full list of author information is available at the end of the article

OoS data sets. This algorithm is subject of Sect. 3 and has been tested intensively on real sensor data. The tracking results are presented in Sect. 4, which also includes a numerical evaluation of an ASD filter. The conclusion of this article is given in Sect. 5.

2. Related work

2.1. Out-of-sequence processing

Since the development of multi-sensor systems, the challenge of OoS processing is crucial for further development in tracking research. Bar-Shalom was the first, who picked up the problem and provided an exact solution for lags which are equal or smaller than one update period [5]. He extended his approach in [6] to a multi-step lag algorithm called *Al1* by applying the *equivalent measurement* [7,8] of recent sensor data. This enabled him to use the derived algorithm on OoS data with an arbitrary big lag, but as the equivalent measurement neglects some cross covariances, the result is not an optimal solution. Further generalizations to MHT and IMM scheme followed by various groups as [9-12].

In [13], the idea of augmenting past states and current states for a neat OoS processing occurred. This approach neglects information of current states about time-delayed measurements. In particular, when high maneuvering targets are observed, this results in a sub-optimal routine. An algorithm calculating the cross-covariances for each step in between the occurring lag is given in [14]. An obvious drawback of such an algorithm is the number of measurements to be stored and numerical costs. In [15], past states are considered to provide a more comprehensive treatment of issues in particle filtering. A solution for OoS processing using particle filters is presented in [16].

All filter techniques presented in this work are based on the ASD. In 2009, Koch presented a closed formula for an ASD posterior [2]. His work was continued and investigated more intensively in [4]. Extensions to *MHT* and *IMM* filtering are given in [3].

2.2. TBD methods

There exist various methodologies to realize TBD. One can separate four different classes of them: Dynamic Programming Algorithm (DPA), Particle Filters, Hough Space Transform, and Subspace Data Fusion. Due to computational reasons, a practical application of the Hough Transform on TBD is often limited to non-maneuvering targets [17,18]. While the numerical costs of particle filters are high in general, their accuracy (in theory) can achieve any degree desired. Therefore, many recent research activities concentrate on this approach for TBD [19]. However, these algorithms still face the problem that it takes a long time for the modes (i.e., the tracks) to appear. The subspace approach to TBD

algebraically calculates the posterior of the emitter's position given the sensor data with respect to properties of the antenna [20]. While the results on simulation data seem to exceed other techniques, it has not been tested on real data yet. Furthermore, the computational complexity is very high and therefore it might be difficult to implement for applications with real time requirements.

The DPA approach consists of a sequential Log-Likelihood-Ratio (LLR) test for existing targets in each sensor cell. Unlike conventional track extraction methodologies on thresholded measurements [21], it calculates the probability of a track existence without using an estimated spatial covariance matrix of the target state [22]. A score which is a function of this probability is calculated for each scan. Given the Markov property, this approach solves the global track search asymptotically in an efficient way. In the recent time, Orlando et al. showed that an application to an under-water sonar system is possible [23].

3. Track initiation using OoS-DPA

3.1. DPA algorithm

Assume a time series of sensor observations $Z^k = \{z_1, \dots, z_k\}$ is given, where $z_k = \{y_k^1, \dots, y_k^N\}$ is the set of measured amplitudes or SNRs y_k^i in the corresponding sensor bin θ_i , $i = 1, \dots, N$. For a complete track initialization, we are interested in both, the question of track existence and the associated time series of sensor bins $\hat{\theta}_k, \dots, \hat{\theta}_1$ for case of a positive result.

Following the description of Arnold et al. [22], we assume there is a function $s(\theta_k, \dots, \theta_1)$ which is maximized by the desired sequence of states. This scoring function respects the observed signal strength and the underlying target motion. Whereas for the general solution an exhaustive search over all possible combinations is necessary, the DPA splits the scoring function into temporary elements

$$s(\theta_k, \dots, \theta_1) = \sum_{i=2}^k s_i(\theta_i, \theta_{i-1}). \quad (1)$$

This is possible, if the target motion is modeled as a Markov random walk of first order. Then, the solution is given by

$$(\hat{\theta}_k, \dots, \hat{\theta}_1) = \arg \max_{\theta_k} \{ \max_{\theta_{k-1}} [s_k(\theta_k, \theta_{k-1}) + \max_{\theta_{k-2}} \{s_{k-1}(\theta_{k-1}, \theta_{k-2}) + \dots + \max_{\theta_1} \{s_2(\theta_2, \theta_1)\} \dots \}] \}. \quad (2)$$

An asymptotic solution to this maximization problem can be calculated stepwise by introducing auxiliary function chain $\{h_i\}_{i=1, \dots, k-1}$ which is defined by the following recursive expression:

$$h_1(\theta_2) = \max_{\theta_1} s_2(\theta_2, \theta_1) \quad (3)$$

$$h_i(\theta_{i+1}) = \max_{\theta_i} \{h_{i-1}(\theta_i) + s_{i+1}(\theta_{i+1}, \theta_i)\}. \quad (4)$$

For a given initialization $\hat{\theta}_1$, we obtain $\hat{\theta}_i$, $i \geq 2$, by

$$\hat{\theta}_i = \arg \max_{\theta_i} \{h_{i-1}(\theta_i)\}. \quad (5)$$

For the derivation of such a score function, we follow the idea of the conventional track extraction methodology [21] and use a sequential likelihood ratio test. Switching to the logarithmic version of it, we are able to prove the necessary splitting property of (1). To this end, we consider the following hypotheses.

- H_1 : $\theta_k, \dots, \theta_1$ is associated to a target.
- H_0 : There is no target.

Using the LLR test, we obtain for the cumulative scoring function s :

$$s(\theta_k, \dots, \theta_1) = \log \left(\frac{p(\theta_k, \dots, \theta_1 | Z^k)}{p(H_0 | Z^k)} \right). \quad (6)$$

Applying Bayes' Theorem on the argument, we obtain

$$\frac{p(\theta_k, \dots, \theta_1 | Z^k)}{p(H_0 | Z^k)} = \frac{p(z_k | \theta_k)}{p(z_k | H_0)} \cdot \frac{p(\theta_k, \dots, \theta_1 | Z^{k-1})}{p(H_0 | Z^{k-1})}. \quad (7)$$

Because of the Markov assumption, the following equation holds.

$$p(\theta_k, \dots, \theta_1 | Z^{k-1}) = p(\theta_k | \theta_{k-1}) p(\theta_{k-1}, \dots, \theta_1 | Z^{k-1}). \quad (8)$$

Combining the above equations yields for the cumulative scoring function

$$s(\theta_k, \dots, \theta_1) = \log \left(\frac{p(z_k | \theta_k)}{p(z_k | H_0)} \right) + \log(p(\theta_k | \theta_{k-1})) + s(\theta_{k-1}, \dots, \theta_1) \quad (9)$$

$$= s_k(\theta_k, \theta_{k-1}) + s(\theta_{k-1}, \dots, \theta_1) \quad (10)$$

$$= \sum_{i=2}^k s_i(\theta_i, \theta_{i-1}). \quad (11)$$

This satisfies the required assumption of (1). Therefore, the auxiliary functions $h_i(\theta_{i+1})$ are given by:

$$h_{k-1}(\theta_k) = \log \left(\frac{p(z_k | \theta_k)}{p(z_k | H_0)} \right) + \max_{\theta_{k-1}} \{ \log(p(\theta_k | \theta_{k-1})) + h_{k-2}(\theta_{k-1}) \}. \quad (12)$$

Various approaches have been discussed to estimate the signal dependent log-term of h_{k-1} (see [24] and literature cited therein). For sensors for which the assumption of a Gaussian distributed SNR with mean \bar{s} and additive noise holds, the expression simplifies to

$$\log \left(\frac{p(z_k | \theta_k)}{p(z_k | H_0)} \right) = \frac{(\gamma_k^{\theta_k} - \bar{s})^2 - (\gamma_k^{\theta_k})^2}{2}, \quad (13)$$

where $\gamma_k^{\theta_k}$ represents the measured SNR in sensor bin θ_k rescaled such that the noise covariance is unity.

3.2. Out-of-sequence DPA

As stated in Sect. 1, low computational costs of a TBD algorithm are crucial for real applications. Therefore, it would be highly inconvenient to reprocess stored data in situations where time-delayed measurements occur, i. e., OoS data. In this section, we propose an extension to the DPA algorithm described in Sect. 3.1 such that it can update its states directly on OoS data sets. In particular, we state how to establish the links between the states in order to obtain the estimated time series of bins $\hat{\theta}_n, \hat{\theta}_{n+1}, \dots, \hat{\theta}_k$.

3.2.1. Update of the score

Because of time limitations, it is generally not intended to retrospectively update the scores and links of the past states of time t_l for $t_l < t_k$. Therefore, the current score values for each sensor bin only reflects the exact posterior for a given state θ_k at time t_k . Let us now assume a time-delayed sensor data set z_m originating from time $t_m < t_k$ occurs. The goal is now to calculate the score conditioned on the new measurement data set $Z^{k,m} = Z^k \cup \{z_m\}$. As in the above scheme, we have

$$s(\theta_k, \dots, \theta_m, \dots, \theta_1) = \log \left(\frac{p(\theta_k, \dots, \theta_m, \dots, \theta_1 | Z^{k,m})}{p(H_0 | Z^{k,m})} \right). \quad (14)$$

Again, we might apply Bayes' Theorem on the argument of the logarithm and obtain

$$\frac{p(\theta_k, \dots, \theta_m, \dots, \theta_1 | Z^{k,m})}{p(H_0 | Z^{k,m})} = \frac{p(z_m | \theta_m)}{p(z_m | H_0)} \cdot \frac{p(\theta_k, \dots, \theta_m, \dots, \theta_1 | Z^k)}{p(H_0 | Z^k)} \quad (15)$$

$$= \frac{p(z_m | \theta_m)}{p(z_m | H_0)} \cdot \underbrace{p(\theta_m | \theta_k, \dots, \theta_1)}_{(*)} \cdot \frac{p(\theta_k, \dots, \theta_1 | Z^k)}{p(H_0 | Z^k)}. \quad (16)$$

The term $(*)$ needs a fully smoothed state time series $\theta_k, \dots, \theta_1$ for a precise calculation. However, during the track extraction phase we might assume the target to be not maneuvering very strong. Therefore, an appropriate approximation is given by

$$p(\theta_m | \theta_k, \dots, \theta_1) \approx p(\theta_m | \theta_k), \quad (17)$$

which is not covered by the Markov property, because we might have $k > m > 1$. In such a case, it would be necessary to incorporate the system dynamics from the past and the future to obtain an exact result on the conditional density of θ_m . For the sake of simplicity, we only incorporate the system dynamics from the recent

processing step k . Using this approximation, we end up with a straight forward score update by the auxiliary function

$$h_k(\theta_m) = \log \left(\frac{p(z_m|\theta_m)}{p(z_m|H_0)} \right) + \max_{\theta_k} \{ \log(p(\theta_m|\theta_k)) + h_{k-1}(\theta_k) \}. \quad (18)$$

Nevertheless, this score references to the states at time t_m , therefore a similar approximation might be necessary, if the following data set is originated at time t_{k+1} .

3.2.2. Obtaining the links

Assume, at time t_k the score $h_{k-1}(\hat{\theta}_k)$ for sensor bin $\hat{\theta}_k \in \{1, \dots, N\}$ exceeds the threshold μ_t for track confirmation. If the fixed length for track initialization is $k - n + 1$, we additionally need to gather $\hat{\theta}_{k-1}, \dots, \hat{\theta}_n$. As we can save the backward links which carry out the maximization in the auxiliary function $h_{i-1}(\cdot)$, this is a trivial task, if all sensor data appear in the timely correct order. An example is given in Figure 1 where a time-bin diagram is shown. The three paths in this figure overlap in some parts, while their score at the most recent instant of time belongs to distinct sensor bins.

Let us now consider the OoS case. If μ_t is exceeded by the score for some $\hat{\theta}_m$ referring to time t_m , we gather the sequence of states $\{\hat{\theta}_j\}_j$ by following the links starting at $\hat{\theta}_m$. Using the reversed order of the data appearance, this link is always unique. Therefore, we obtain a unique track sequence $\hat{\theta}_k, \dots, \hat{\theta}_n$ with $t_m \in [t_k, t_n]$ by ordering the elements of the path accordingly to their instant of time of origination. This procedure is visualized in Figure 2 for the timely ordered case (a) and OoS case (b).

4. Evaluation and application results

4.1. DPA evaluation

The evaluation of the OoS-DPA is separated into two parts. The first part considers the runtime duration when OoS sensor data appears in comparison to a reprocessing scheme, which starts at the last instant of time such that the remaining data can be used in the

timely correct order. The second part addresses the obtained track accuracy. To this end, the ordered DPA output is taken as a reference. For both parts, the data set provided by DSO National Laboratories from a 2D radar system is applied as input. While for the time measurements the whole set of 400×372 sensor bins are taken into account, the accuracy performance test concentrates on a small 10×10 bins subset. Processing 15 data scans in the correct order with a standard DPA algorithm yields exactly one target. By means of this result, we evaluate number, states, and processing speed estimated by the extended DPA in the OoS case.

Figure 3 presents the results of processing speed for both algorithms, the reprocessing DPA and the OoS-DPA. As the reprocessing takes a lot of time, it is obvious that the speed of such a scheme is much lower than a direct update. Furthermore, the time consumption increases linearly in the mean time delay. This behavior, of course, is as expected.

Next, we have a look at the DPA output. We examine the deviation between the ordered case and OoS case for a single target. To this end, we study a small subset of the radar data and compare the results in terms of bin deviations, non-detections and false tracks. As shown in Figure 4, the mean deviation of a track consisting of 15 states is up to 3 bins in the range axes. At a range bin size of 60 m, this corresponds to 180 m range off-set. The main reasons for this off-set is most probably the approximation in motion penalties mentioned in the section above. The mean deviation on the bearing axis is below 0.5° , thus all estimated bearing bins are almost the same. Note that the deviation in both, range and bearing, are highly dependent on the observed case. However, this shows that the OoS-DPA is able to establish a target track such that it can be maintained by a tracker. Furthermore, Figure 5 shows that the chosen target was detected by the OoS-DPA in almost every run. There were only three non-detections at a mean time delay of 5s out of 1000 runs (blue line). As mentioned above, a quite small subset of 10×10 bins was considered for this evaluation. However, the red line shows, that there was no run with a second (false) target detection in it.

4.2. Numerical evaluation of the ASD filter

This section analyses the performance of an ASD filter in comparison to other existing techniques. Appropriate candidates for such a comparison are a standard Kalman filter (KF), which has to reprocess some stored sensor data in case of an OoS measurement, and the algorithm called A/1 from Bar-Shalom et al. [6]. On the one hand, the KF needs a lot of storage for all measurements within a time window and extra time for reprocessing depending on the size of the lag of an OoS measurement. On the other

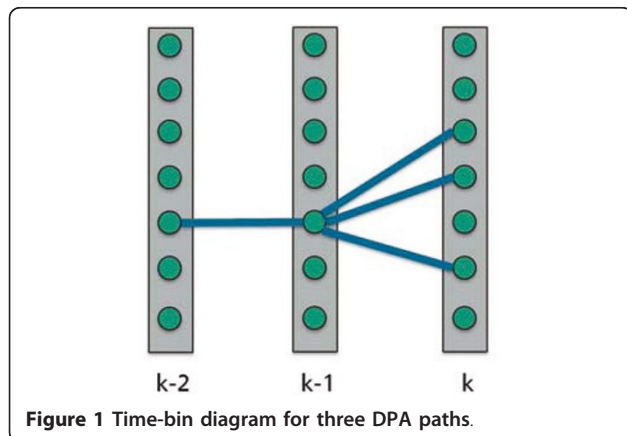


Figure 1 Time-bin diagram for three DPA paths.

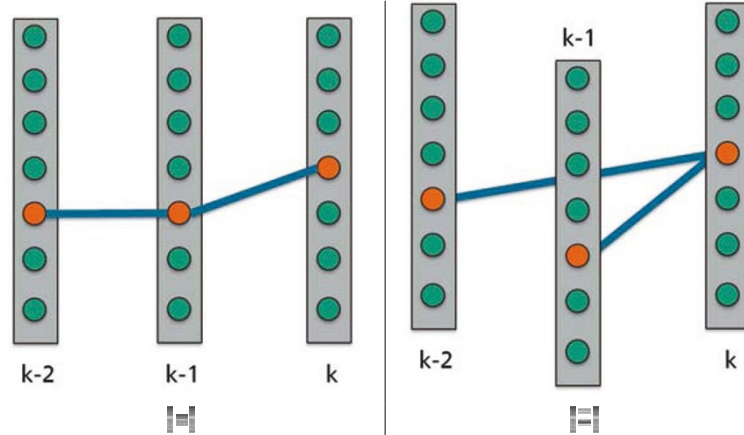


Figure 2 Links obtained by some θ_m where $m = k$ (a) and $m = k - 1$ (b).

hand, the A/I consists of the algorithm A1 from [5] applied to the *equivalent measurement* [7,8] of the set of sensor data since the time of the last update before the time of the OoS measurement. Therefore, it is an approximation of the optimal estimate, but it does not require a storage of all sensor data. The degree of approximation depends on the level of process noise. In the evaluation below, we compare the resulting performance for different evolution noise levels.

In the following simulation scenarios, a target moves in a non-deterministic manner through a two-dimensional space. A virtual sensor observes this target once a second by measuring its range and bearing. These measurements suffer from an additional zero-mean Gaussian distributed noise. In particular, the noise level we use a variance of

$\sigma_\varphi^2 = 1$ and $\sigma_\varphi^2 = 1$ millidegree² in range and bearing, respectively. All mentioned filters are initialized with the perfect start values of the targets position and velocity. Furthermore, they use a perfect matching evolution model. For the latter, we chose a *Continuous White Noise Acceleration Model* [25], i.e., the transition probability density function for a given state \mathbf{x}_k at time t_k to \mathbf{x}_{k+1} at t_{k+1} is given by the following linear Gaussian model:

$$p(\mathbf{x}_{k+1} | \mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{F}_{k+1|k} \mathbf{x}_k, \mathbf{Q}_{k+1|k}), \quad (19)$$

where

$$\mathbf{F}_{k+1|k} = \begin{pmatrix} 1 & T1 \\ 0 & 1 \end{pmatrix}, \quad (20)$$

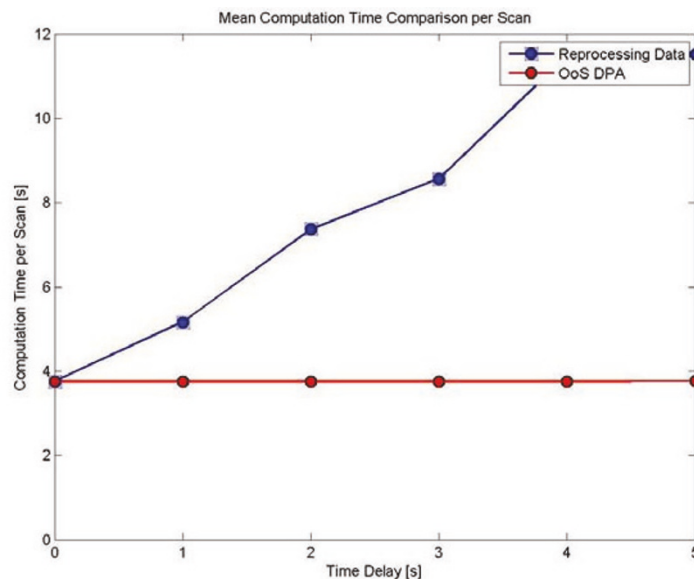


Figure 3 Mean processing time per scan over increasing mean time delay.

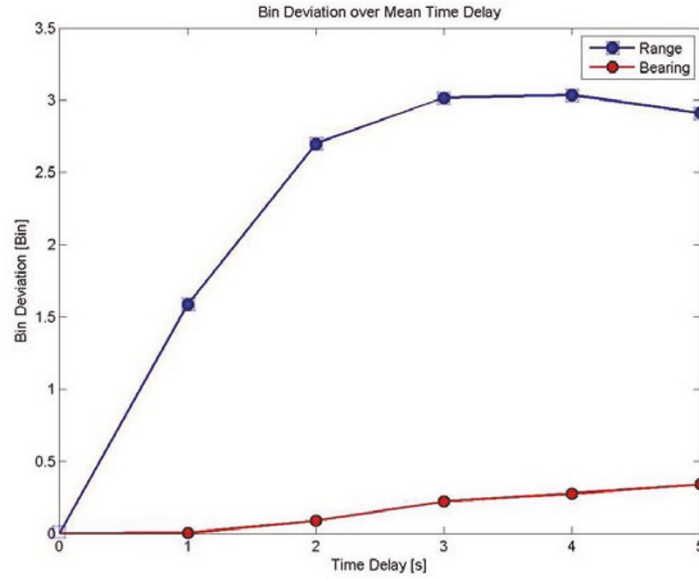


Figure 4 Mean track deviation for OoS-DPA.

$$\mathbf{Q}_{k+1|k} = q \cdot \begin{pmatrix} \frac{1}{3}T^3 \cdot \mathbf{1} & \frac{1}{2}T^2 \cdot \mathbf{1} \\ \frac{1}{2}T^2 \cdot \mathbf{1} & T \cdot \mathbf{1} \end{pmatrix}, \quad (21)$$

$$T = (t_{k+1} - t_k). \quad (22)$$

Here, the parameter q describes the speed variance for an update interval of $T = 1$ s. We use the abbreviation $\mathbf{1}$ for an identity matrix in the dimension 2. For each setup of this parameter, 1000 Monte Carlo simulations were run, where we tracked the target for 500 steps.

The communication link effect is simulated by an additional Poisson distributed delay μ_k with mean and variance $\bar{\Delta}_k$ for each measurement transfer from the sensor to the fusion center. This causes a regular appearance of OoS measurements. After every update, we quantify the root mean squared error (RMSE) of the filter's estimate according to the real target position. Furthermore, we meter the processing time of the filter for each run. This reveals the efficiency regarding to the numerical complexity of the algorithm.

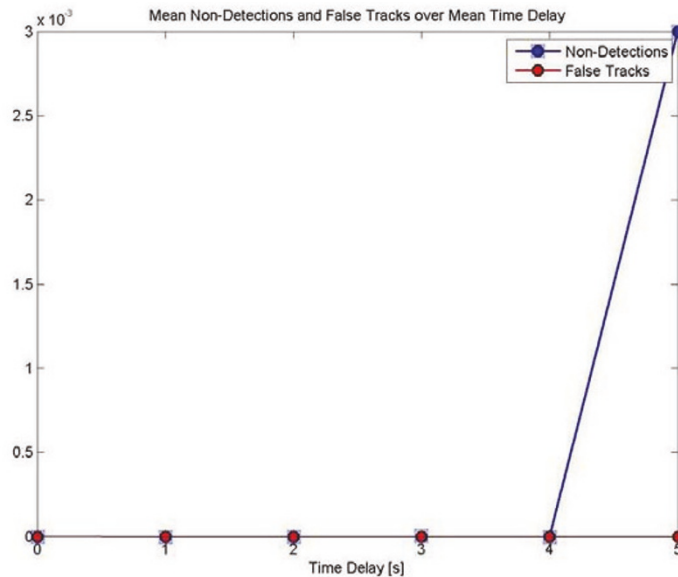


Figure 5 Mean number of false tracks and non-detections.

In Figures 6, 7, 8, 9, 10, and 11, the results of the RMSE over an increasing mean time delay $\bar{\Delta}_k = E[\Delta_k]$ are presented. It can easily be seen that the accuracy of the A/I algorithm decreases for stronger maneuvering targets and highly delayed measurements. For almost deterministic targets ($q = 0.01$), a difference in the performance between the three algorithms cannot be observed. In any case, the ASD filter has an equal RMSE to the reprocessing KF. However, the ASD filter is much more efficient regarding to the numerical complexity of the state smoothing and OoS processing. This can be seen in Figures 12 and 13, where a boxplot of the processing time is given for each algorithm. Here, one can also see that the effect of reprocessing measurements in this simple case (i.e., perfect data-to-target association, no false measurements, perfect detection) is very low. This effect, however, will increase in more realistic scenarios, where those conditions are not given. Furthermore, the advantage of a unified handling of filtering and retrodiction^a becomes obvious, as the ASD algorithm handles it in less than half of the time required for a separate retrodiction.

4.3. Implementation setup for real data set

Depending on the given circumstances, either communication channels might be of limited capacity, or high bandwidth links might be available. To cover these possible situations, we consider three different setups:

- (i) Single sensor performing TBD on a local sensor site which is connected to a Fusion Center (FC)

which maintains the tracks. OoS measurements appear due to varying delays on this link. This setup is visualized in Figure 14.

- (ii) Multiple sensors with separated local TBD modules connected to a FC performing an ASD (see Figure 15). In this scenario, as well as in the previous one, only new tracks and thresholded measurements are sent via the network.

- (iii) Sensors connected to a FC performing a centralized TBD methodology and maintaining the tracks. In this scenario, the raw sensor data are sent to the FC, depicted in Figure 16.

The data set used was provided by DSO National Laboratories (DSO) and consists of raw sensor data obtained by a two-dimensional radar system. It includes 389 scans, which corresponds to about 15 min at a given scan rate of $T = 2.2$ s. At certain instants of time, up to ten targets can be found. The test for existing targets is done by a DPA algorithm. If the test for target existence turns out with a positive result, a new track is initialized. It consists of the Least Squared Errors (LSE) approximation of the recent 15 states and is passed to the TM. This maintenance also essentially consists of a Sequential-Likelihood-Ratio test, as proposed in [21]. The result of it is called *LR-score* and depends on the choice of various parameters. Most of them will be explained below.

4.3.1. Filter parameters

Essentially, there are two thresholds $A < B$ to test for track continuation. A LR-score below A will lead to a

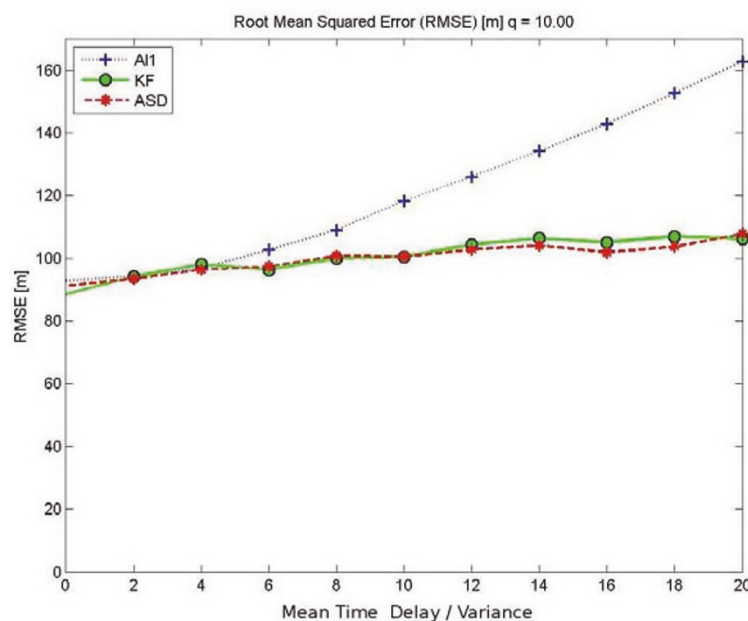


Figure 6 Simulative results for $q = 10.0$.

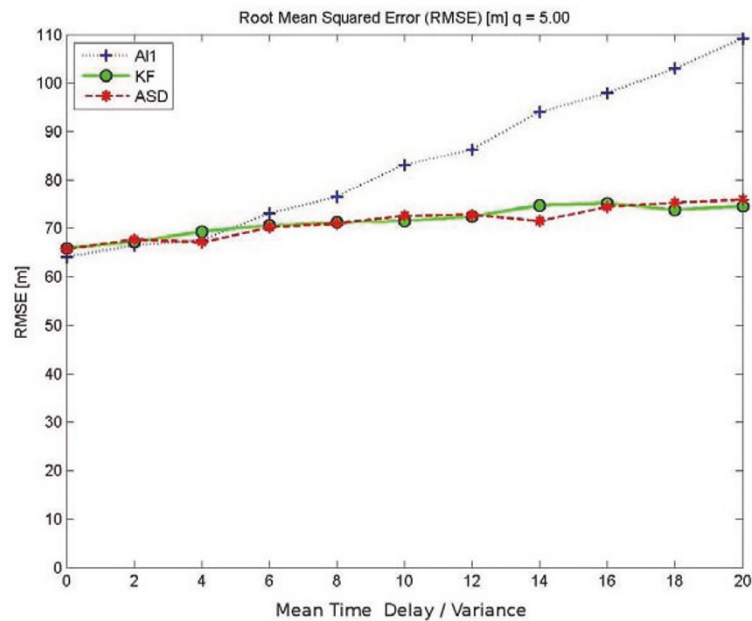


Figure 7 Simulative results for $q = 5.0$.

track deletion, while a score above B indicates a *track confirmation*. If the score is in between of both, the track is continued, but not be confirmed, i.e., its tracking results are not displayed. A new track arising from the DPA is initialized with an LR-score $LR_0 = 1/10 \cdot B$. Therefore, it is not assumed to be confirmed, unless it can be observed by the TM for at least one additional

step, depending on the chosen values for *track detection* P_D and the *mean false measurement density* ρ_F .

Another important parameter for TM is μ_{tm} , which gives a lower bound for the distance of two targets before their corresponding tracks are merged. A similar threshold is given on a lower level, as we use a Multiple-Hypotheses-Tracker (MHT) [24] extension for the

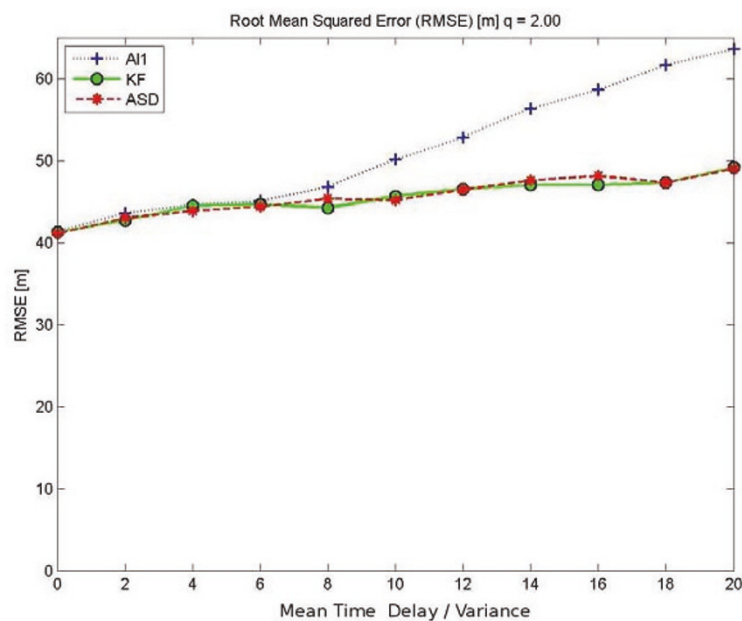


Figure 8 Simulative results for $q = 2.0$.

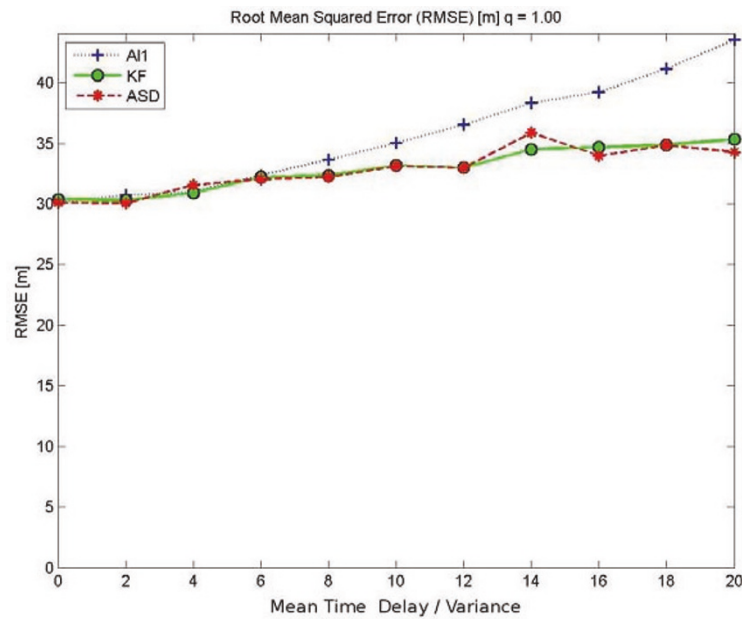


Figure 9 Simulative results for $q = 1.0$.

ASD filter [3]. In order to keep numerical expenses at a decent level, similar hypotheses are merged, if their weighted distance $d = (\mathbf{x}^1 - \mathbf{x}^2)^T (\mathbf{P}^1 + \mathbf{P}^2)^{-1} (\mathbf{x}^1 - \mathbf{x}^2)$ is lower than the threshold μ_{hm} [26]. Here, \mathbf{x}^i and \mathbf{P}^i represent the state and the covariance, respectively, for the i th track. Furthermore, hypotheses with a probability lower than a threshold μ_p are deleted immediately.

An overview of all mentioned parameters is given in Table 1.

4.3.2. Results

In order to give a comparison in the mean tracking error, an exact ground-truth trace for some targets would be necessary. However, such a trace is not available for the real data set. Therefore, we present the

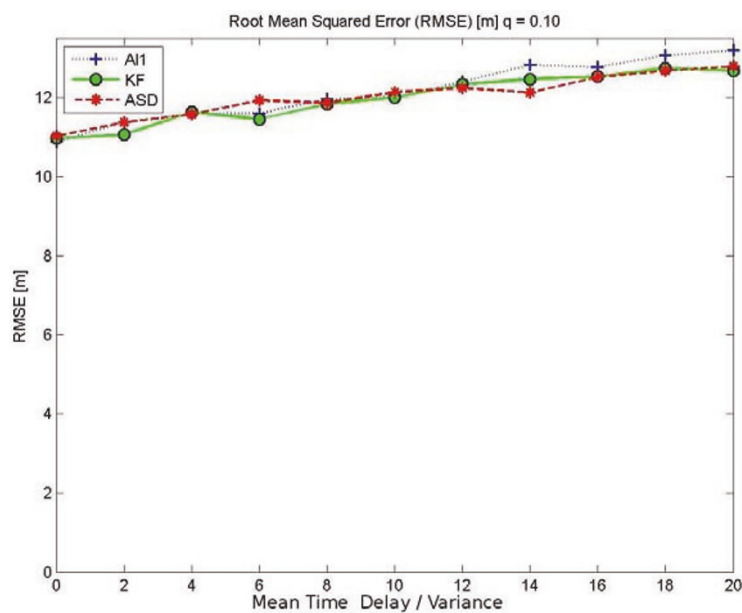


Figure 10 Simulative results for $q = 0.1$.

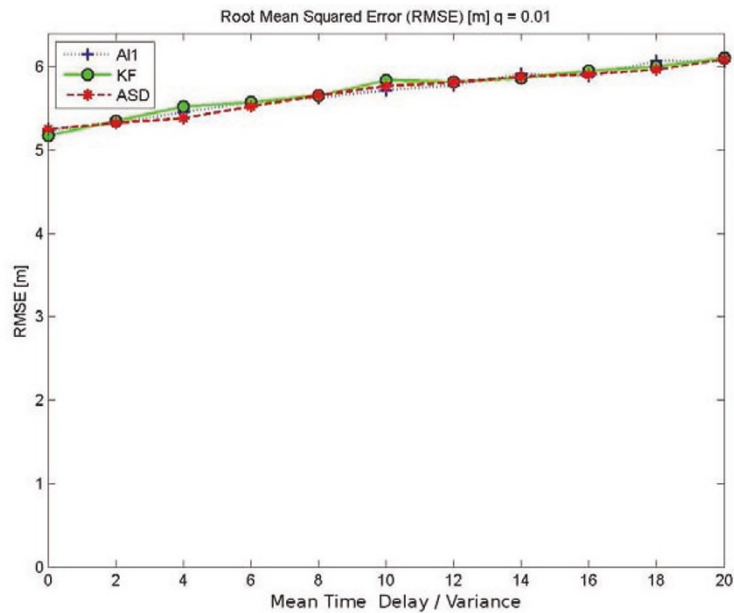


Figure 11 Simulative results for $q = 0.01$.

gained tracking results as *situation pictures* at an arbitrarily chosen but fixed instant of time.

4.3.3. Scenario one

At first, we have a look at the results of scenario one. Here, a Poisson distributed delay for both, measurements and DPA output of a single sensor, is inserted with a mean delay $\bar{\Delta}_k$ of 0, 10, and 20 s, respectively. Figure 17(a) - 17(c) and 17(d) - 17(f) shows all tracks occurring at the 300th scan, i.e., at time $t = 660$ s, in

range-bearing and x - y space, respectively. On the left hand, a star indicates a position of track initialization, while a diamond is set at the track deletion. A green line in between shows the trace of a target. It includes all positions obtained by processing the data arrived up to the chosen scan. Due to possible delays for $\bar{\Delta}_k > 0$, information contained in scans of later instants of time might be processed already. This explains why some tracks appear advanced further in comparison to the

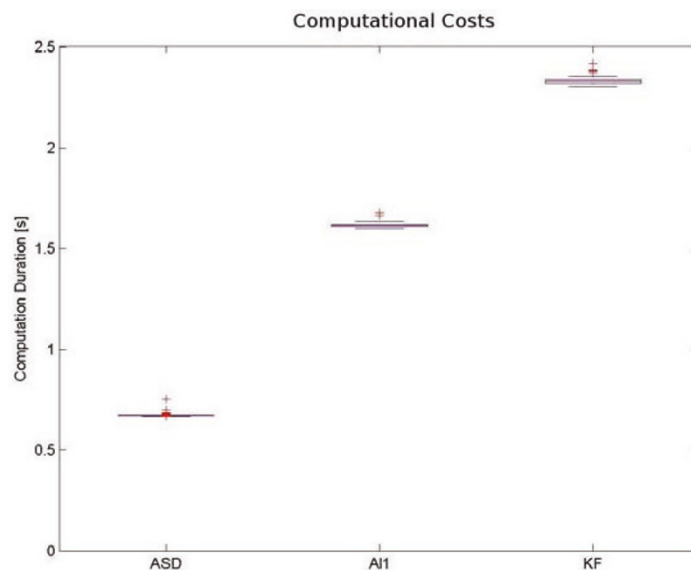


Figure 12 Processing time for ordered case, $\bar{\Delta}_k = 0$ s.

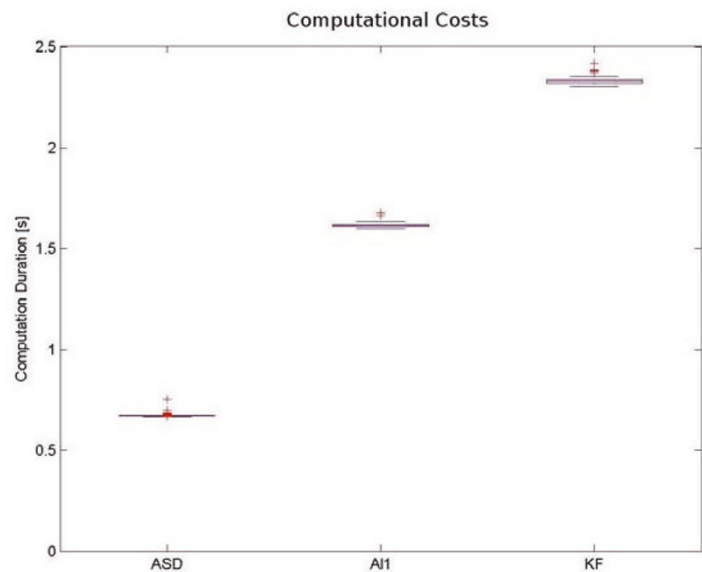


Figure 13 Processing time for OoS case, $\bar{\Delta}_k = 20$ s.

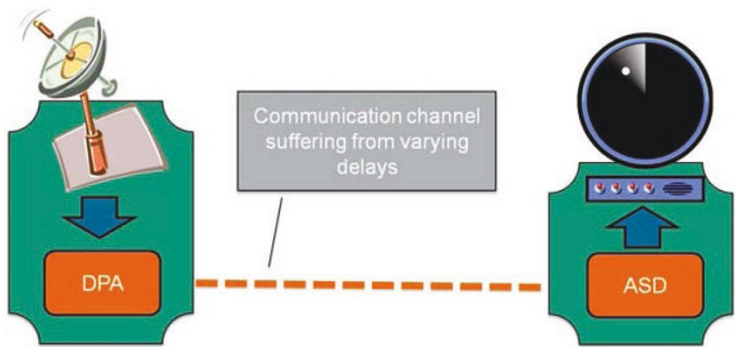


Figure 14 One sensor, TBD, and ASD.

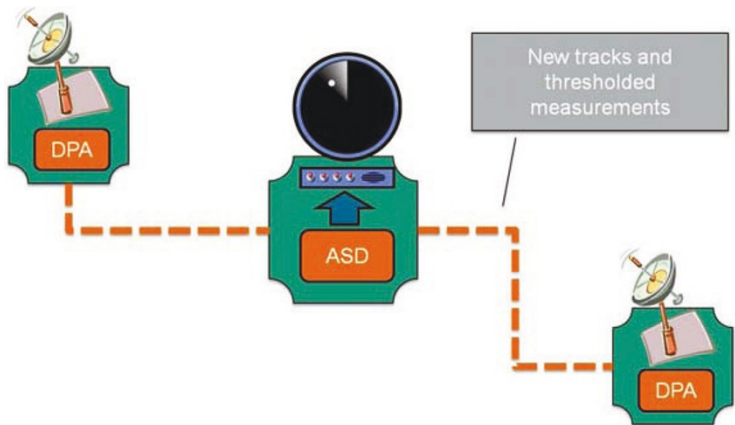


Figure 15 Multiple sensors and TBD, single ASD.

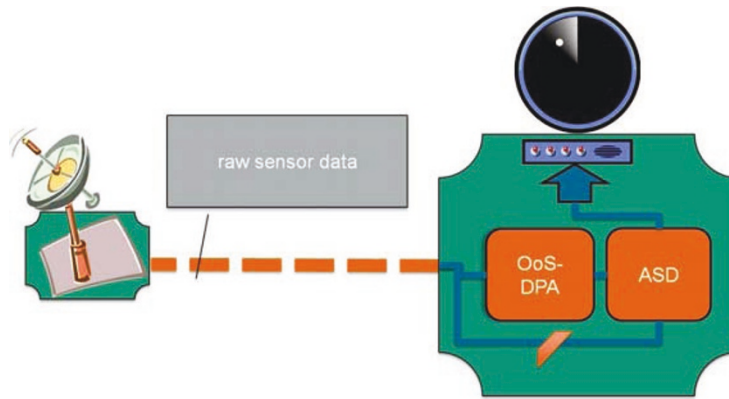


Figure 16 Multiple sensors, single TBD, and ASD.

ordered case ($\bar{\Delta}_k = 0$ s). A similar effect can be observed in the x - y space on the right hand. Here, the red rectangle indicates the sensor position and estimated positions in an ASD are plotted by blue crosses. With increasing mean delay $\bar{\Delta}_k$, the accumulated state positions are stretched further apart. This is due to the fact that the ASD contains states corresponding to a wider range of points in time.

4.3.4. Scenario two

For an evaluation of the second scenario, the data set was split up. Two sensors were simulated, the first using all even scans, the second using all odd scans. Furthermore, in order to obtain a gain of the sensor fusion on common tracks, each sensors Field of View (FoV) was restricted to a fixed part of a full circle. This is shown in Figure 18, where the filled area corresponds to the FoV.

The tracking results are given in Figure 19. As one can see, the all tracks of the optimal case, where $\bar{\Delta}_k = 0$, can be recovered in the OoS cases. To this end, the fusion center benefits from both sensors without the need of storing and reprocessing data. The thresholded measurements from sensor 1 can be seen in Figure 19

(d)-19(f) as thin black dots. The effects described for scenario one apply in this situation, too.

5. Conclusion

In this article, we proposed a new extension to the DPA approach for TBD challenges. This extension enables the DPA to process time-delayed sensor data directly. These might appear because of delays in communication networks. The extended DPA is identical to the standard DPA for case all sensor data appears in the timely correct order. Therefore, one might speak of a *natural* extension. However, some approximations to the exact solution are necessary to prevent dramatically increasing costs on numerical power. In an evaluation on a data set of a real radar system, this approximation was shown to have at most marginal effect on the results. This is supposed to hold whenever the observed targets are in a non- or moderate-maneuvering state during the track extraction phase.

Then we numerically proved that the intensity of the process noise can have severe impact on the estimation error when using approximations. The results clearly

Table 1 Parameter overview for TM

Parameter	Value	Explanation
P_D	0.6	Detection probability for a target by thresholded sensor data.
$ \text{FoV} $	$\pi \cdot (60 \text{ m} \times 400)^2 \approx 1.8 \times 10^9 \text{ m}^2$	Size of sensors Field of View (FoV)
ρ_F	$100/ \text{FoV} $	Clutter density
A	$(1 - P_D)/(1 - \rho_F)$	Track deletion threshold
B	P_D/ρ_F	Track confirmation Threshold
C	$10^{30} \cdot B$	Maximum LR-score of tracks. This prevents a numerical overflow of LR-score
LR_0	$1/10 \cdot B$	Initial value of LR-score for new tracks
μ_{tm}	5×10^3	Threshold for track merging
μ_{hm}	1×10^2	Threshold for hypotheses merging
μ_p	0.01	Threshold for hypotheses dumping
ASD length	15	Number of states in the ASD filter

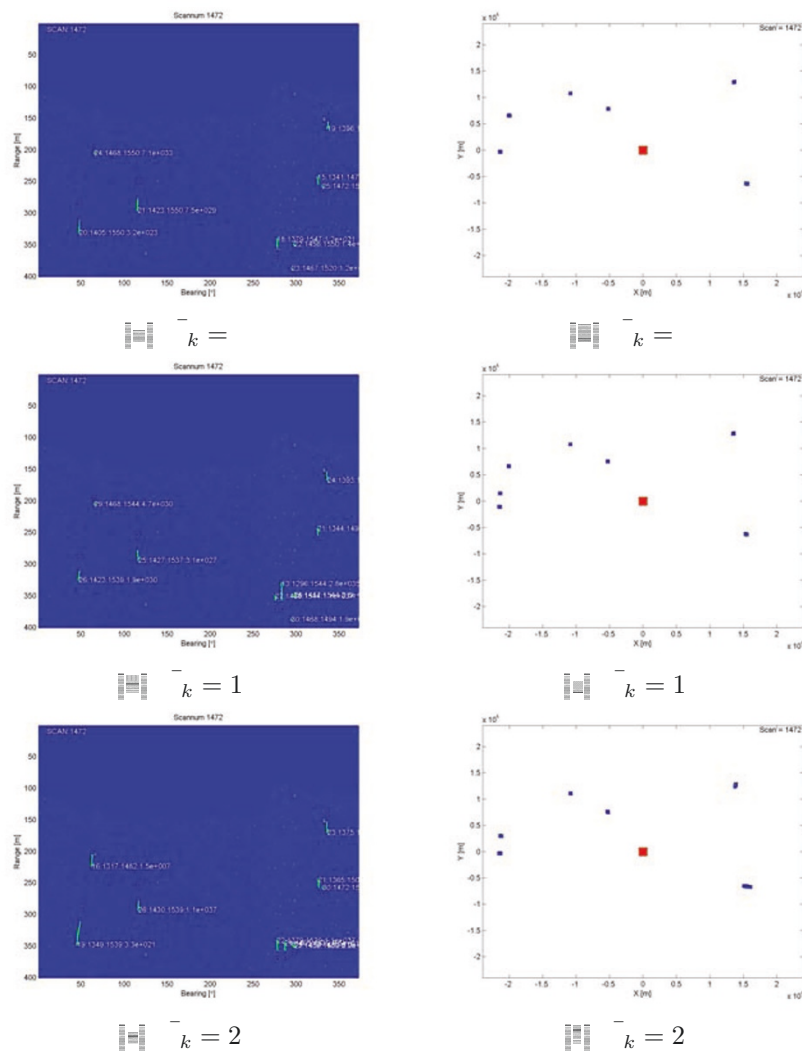


Figure 17 Track visualization in range-bearing space (a)-(c) and x-y space (d)-(f) after 300 scans for various mean delays $\bar{\Delta}_k$ (a) $\bar{\Delta}_k = 0$ s, (b) $\bar{\Delta}_k = 10$ s, (c) $\bar{\Delta}_k = 20$ s, (d) $\bar{\Delta}_k = 0$ s, (e) $\bar{\Delta}_k = 10$ s, (f) $\bar{\Delta}_k = 20$ s.

show that an ASD filter processes OoS measurements optimally in terms of a mean squared error, i.e., equivalent to a KF reprocessing outdated data.

Last but not least, we applied the combination of a DPA and an ASD on data of a real radar sensor system. As no ground-truth is available, we cannot derive an

estimation error. However, the situation pictures give an impression of how seamless this combination worked out. The experienced process time might be higher than a KF using Global Nearest Neighbor (GNN) [24], but it is still applicable for real time processing. Furthermore, it is clear that tracking results of an MHT is superior to the GNN scheme, because it involves all measurements within the selected gate [27]. A more profound analysis of this consideration about tracking performance and processing time is left as future work at this point.

Endnote

^aWe prefer the term *retrodiction* to *smoothing*, as it combines the process of *prediction* and a *retrospective* view.

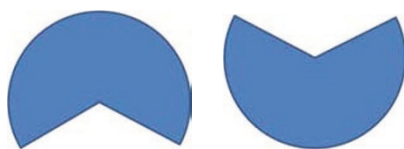


Figure 18 Field of view of sensor 1 (left) and sensor 2 (right).

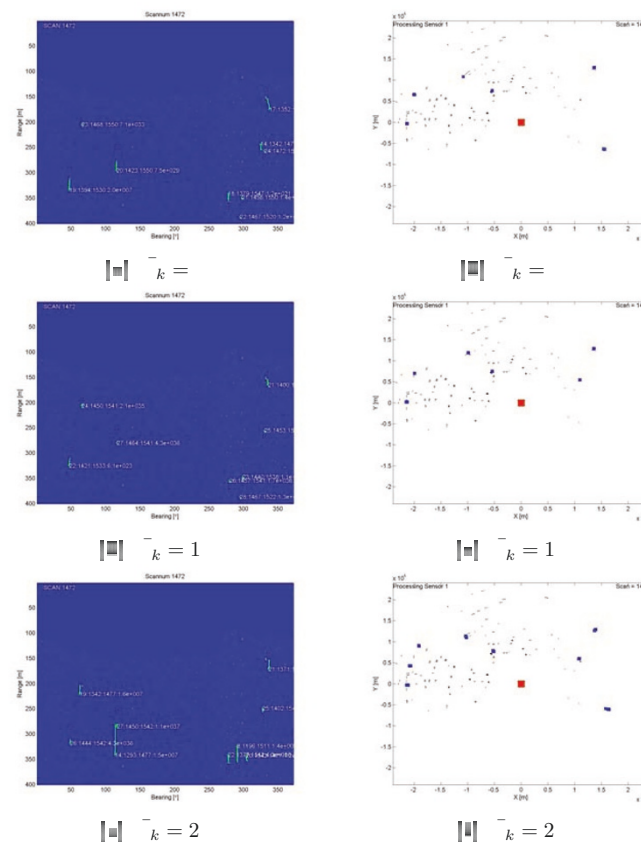


Figure 19 Track visualization for two overlapping sensors in range-bearing space (a)-(c) and x-y space (d)-(f) after 300 scans for various mean delays Δ_k (a) $\Delta_k = 0$ s. (b) $\Delta_k = 10$ s. (c) $\Delta_k = 20$ s. (d) $\Delta_k = 0$ s. (e) $\Delta_k = 10$ s. (f) $\Delta_k = 20$ s

Abbreviations

ASD: Accumulated State Density; DPA: Dynamic Programming Algorithm; FC: Fusion Center; FoV: Field of View; GNN: Global Nearest Neighbor; KL: Kalman filter; LSE: Least Squared Errors; LLR: Log-Likelihood-Ratio; LOTS: low-observable targets; MHT: Multiple-Hypotheses-Tracker; OoS: Out-of-Sequence; RMSE: root mean squared error; SNR: signal-to-noise ratio; TBD: Track-before-Detect; TM: track maintenance; T2TF: Track-to-Track Fusion.

Author details

¹Fraunhofer-FKIE, Wachtberg, Germany ²DSO National Laboratories, Singapore, Singapore

Competing interests

The authors declare that they have no competing interests.

Received: 30 November 2010 Accepted: 13 July 2011

Published: 13 July 2011

References

1. F Govaers, W Koch, Distributed Kalman filter fusion at arbitrary instants of time, in *Proceedings of 13th International Conference Information Fusion, FUSION '10* (2010)
2. W Koch, On accumulated state densities with applications to out-of-sequence measurement processing, in *Proceedings of 12th International Conference Information Fusion, FUSION '09*, pp. 2201–2208 (2009)
3. F Govaers, W Koch, Out-of-sequence processing of cluttered sensor data using multiple evolution models, in *Proceedings of 13th International Conference Information Fusion, FUSION'10* (2010)
4. W Koch, F Govaers, On accumulated state densities with applications to out-of-sequence measurement processing. *IEEE Trans Aerosp Electron Syst.* **38**(3), 769–777 (2002). doi:10.1109/TAES.2002.1039398
5. Y Bar-Shalom, Update with out-of-sequence measurements in tracking: exact solution. *IEEE Trans Aerosp Electron Syst.* **38**(3), 769–777 (2002). doi:10.1109/TAES.2002.1039398
6. Y Bar-Shalom, H Chen, M Mallick, One-step solution for the multistep out-of-sequence-measurement problem in tracking. *IEEE Trans Aerosp Electron Syst.* **40**(1), 27–37 (2004). doi:10.1109/TAES.2004.1292140
7. JM Covino, BJ Griffiths, A new estimation method for multisensor data fusion, in *Proceedings of SPIE Conference on Sensor and Sensor Systems for Guidance and Navigation* (1991)
8. O Drummond, Track fusion with feedback, in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets* (1996)
9. M Mallick, Out-of-sequence track filtering using the decorrelated pseudo measurement approach, in *Proceedings of SPIE conference on Signal and Data Processing for Small Targets.* **5428**, 154–166 (2004)
10. Y Bar-Shalom, H Chen, IMM estimator with out-of-sequence measurements. *IEEE Trans Aerosp Electron Syst.* **41**(1), 90–98 (2005). doi:10.1109/TAES.2005.1413749
11. S Maskell, R Everitt, R Wright, M Briers, W Malvern, Multi-target out-of-sequence data association, in *International Conference on Information Fusion* (2004)
12. Z Jia, A Balasuriya, S Challa, Sensor fusion-based visual target tracking for autonomous vehicles with the out-of-sequence measurements solution. *Robot Autonon Syst.* **56**(2), 157–176 (2008). doi:10.1016/j.robot.2007.05.014
13. S Challa, RJ Evans, X Wang, A Bayesian solution and its approximations to out-of-sequence measurement problems. *Inf Fusion.* **4**(3), 185–199 (2003). doi:10.1016/S1566-2535(03)00037-X

14. M Mallick, S Coraluppi, C Carthel, Advances in asynchronous and decentralized estimation, in *Aerospace Conference, 2001, IEEE Proceedings*. **4** (2001), pp. 4/1873-4/1888
15. M Arulampalam, S Maskell, N Gordon, T Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans Signal Process.* **50**(2), 174–188 (2002). doi:10.1109/78.978374
16. M Orton, A Marrs, Particle filters for tracking with out-of-sequence measurements. *IEEE Trans Aerosp Electron Syst.* **41**(2), 693–702 (2005). doi:10.1109/TAES.2005.1468758
17. G Richards, Application of the Hough transform as a track-before-detect method, in *Target Tracking and Data Fusion (Digest No:1996/253), IEE Colloquium* (Nov. 1996). pp. 2/1-2/3
18. L Wei, X Zhang, L Fan, A tbd algorithm based on improved randomized hough transform for dim target detection, in *2nd International Conference on Signal Processing Systems (ICSPS)*. **2**, pp. V2-241–V2-245 (2010)
19. M Rutten, B Ristic, N Gordon, A comparison of particle filters for recursive track-before-detect, in *8th International Conference on Information Fusion*. **1**, 7 (2005)
20. B Demissie, M Oispuu, E Ruthotto, Localization of multiple sources with a moving array using subspace data fusion", in *11th International Conference on Information Fusion, 30 June-3 July*, 1–7 (2008)
21. W Koch, G Van Keuk, Multiple hypothesis track maintenance with possibly unresolved measurements. *IEEE Trans Aerosp Electron Syst.* **33**(3), 883–892 (1997)
22. J Arnold, SW Shaw, H Pasternack, Efficient target tracking using dynamic programming. *IEEE Trans Aerosp Electron Syst.* **29**(1), 44–56 (1993). doi:10.1109/7.249112
23. D Orlando, F Ehlers, G Ricci, Track-before-detect algorithms for bistatic sonars, in *2nd International Workshop on Cognitive Information Processing (CIP)*, 180–185 (2010)
24. SS Blackman, R Popoli, *Design and analysis of modern tracking systems* (Artech House, New York, 1999)
25. Y Bar-Shalom, X Li, T Kirubarajan, *Estimation with Applications to Tracking and Navigation* (Wiley-Interscience, New York, 2001)
26. D Salmond, Mixture reduction algorithms for target tracking in clutter, in *SPIE Signal and Data Processing of Small Targets*, 434–445 (1990)
27. SS Blackman, Multiple hypothesis tracking for multiple target tracking. *IEEE Aerosp Electron Syst Mag.* **19**(1), 5–18 (2004)

doi:10.1186/1687-6180-2011-20

Cite this article as: Govaers et al.: Track-before-detect in distributed sensor applications. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:20.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com