

## Out-of-Sequence Data Processing for Track-before-Detect using Dynamic Programming

Felix Govaers\*, Yang Rong<sup>+</sup>, Lai Hoe Chee<sup>+</sup>, Teow Loo Nin<sup>+</sup>, Ng Gee Wah<sup>+</sup>, Wolfgang Koch\*

\*Fraunhofer FKIE, Wachtberg, Germany  
{felix.govaers, wolfgang.koch}@fkie.fraunhofer.de

<sup>+</sup>DSO National Laboratories, Singapore  
{yrong, lhoechee, tloonin, ngeewah}@dso.org.sg

*The results of this paper are also published in [GYL<sup>+</sup>ar].*

**Abstract:** In the recent past, tracking applications increasingly develop towards distributed sensor scenarios. In many cases, such schemes must cope with low observable targets in cluttered environments. Furthermore, such a setup suffers from communication delays and timely delayed sensor data. However, Track-before-Detect methodologies are not suitable for processing time delayed data yet. In this paper, we propose a new extension to a Dynamic Programming Algorithm (DPA) approach for Track-before-Detect in distributed sensor systems. This extension enables the DPA to process time delayed sensor data directly. Such delay might appear because of varying delays in the installed communication links. The extended DPA is identical to the recursive standard DPA in case of all sensor data appear in the timely correct order. In an experimental study, we show that the derived algorithm can compensate all occurring time delays very well.

**Keywords:** Track-Before-Detect, Out-of-Sequence, Dynamic Programming Approach

### 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 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 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) [GK10]. 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 already during a track initialization phase, we propose a new algorithm which is able search for new targets in raw sensor data even if it is timely misaligned.

**Structure** This paper is structured as follows. In the next section 2, an overview to related work is given. The main contribution of this paper is a TBD algorithm which is able to process OoS data sets. In section 3, we recall the basic dynamic programming approach. This algorithm is modified for timely disordered data in section 4. Some experimental results on a data set of a 2D-radar system is presented in section 5. We close the paper with a conclusion given in section 6.

## 2 Related work

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 [Ric96, WZF10]. 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 [RRG05]. 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 for TBD in [DOR08] algebraically calculates the posterior of the emitter's position given the sensor data with respect to properties of the antenna. 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 [KVK97], it calculates the probability of a track existence without

using an estimated spatial covariance matrix of the target state [ASP93]. 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 [OER10].

### 3 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  $\theta_i, i = 1, \dots, N$ . 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 [ASP93], 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. Where as 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

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

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 [KVK97] 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 logarithmic likelihood ratio (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.

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

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 in order to estimate the signal dependent log-term of  $h_{k-1}(\cdot)$  (see [BP99] 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

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

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

## 4 Out-of-Sequence DPA

As stated in the introduction of this paper, 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. out of sequence (OoS) data. In this section, we propose an extension to the DPA algorithm described in section 3 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  $\hat{\theta}_n, \hat{\theta}_{n+1}, \dots, \hat{\theta}_k$ .

**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  $\theta_k$  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  $Z^{k,m} := Z^k \cup \{z_m\}$ . Similar to the scheme above, 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

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

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

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 at most in a decent maneuvering state. 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 in order to obtain an exact result on the conditional density of  $\theta_m$ . For the sake of simplicity, we only incorporate the system dynamics from the recent processing step  $k$ . Using this approximation, we obtain a score update by the auxiliary function

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

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  $t_{k+1}$ .

**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 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 scores at the most recent instant of time belongs to distinct sensor bins.

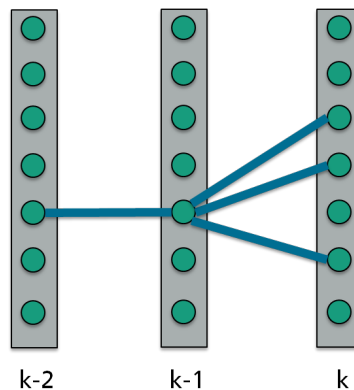


Figure 1: Time-Bin diagram for three DPA paths.

Let us now consider the OoS case. If the threshold is exceeded by the score of bin  $\hat{\theta}_m$  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).

## 5 Experimental Results

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,

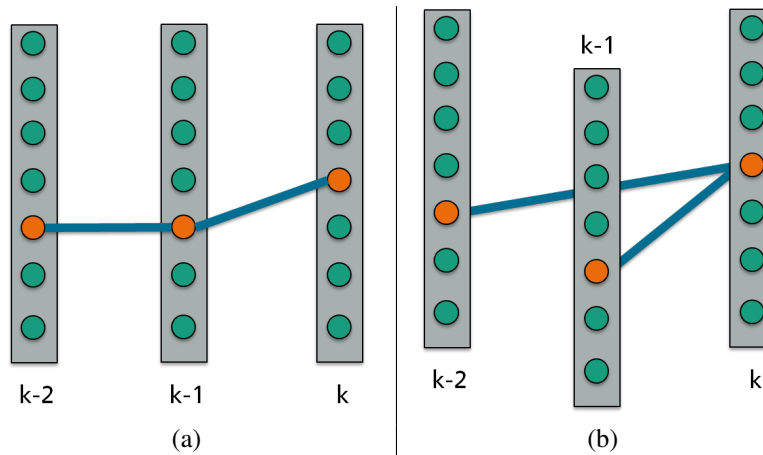


Figure 2: Links obtained by some  $\theta_m$  where  $m = k$  (a) and  $m = k - 1$  (b).

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  $400 \times 372$  sensor bins are taken into account, the accuracy performance test concentrates on a small  $10 \times 10$  bins subset. This subset is purposely chosen such that there is supposed to appear exactly one target within the treated period of time.

Figure 3 presents the results of processing speed for both algorithms, the reprocessing DPA and the OoS-DPA. All values plotted are the mean results of 1000 Monte-Carlo simulations in the time stamp according to a Poisson distributed delay. The delay variance was set to be between  $1s$  and  $5s$ , as shown on the  $x$ -axis. 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 linear growth in time consumption by the mean time delay corresponds to expectations.

In the sequel, we have a look at the DPA output. The question is whether and in how far the obtained tracks by the OoS-DPA are equal to those obtained by an ordered processing of the sensor data. To get an answer to this question, we study a single target scenario with both algorithms 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 axis. At a range bin size of  $60m$ , this corresponds to  $180m$  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  $0.5 \text{ deg}$ , 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 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

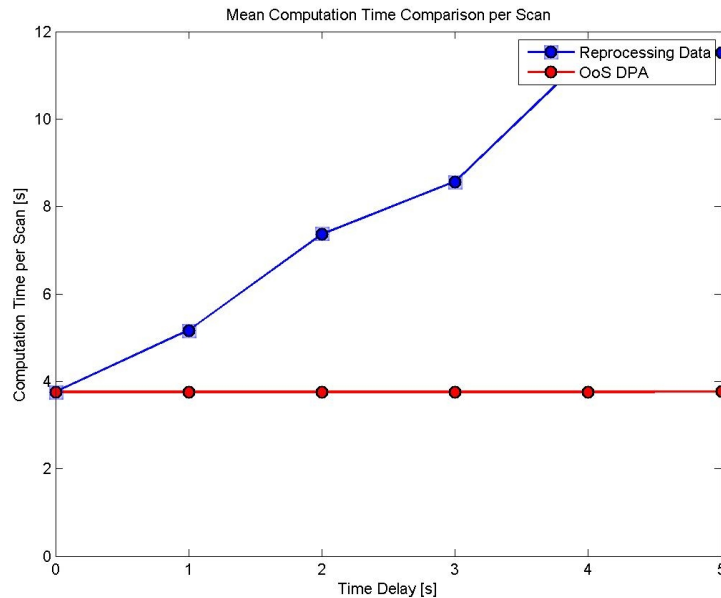


Figure 3: Mean processing time per scan over increasing mean time delay

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  $10 \times 10$  bins was considered for this evaluation. However, the red line shows, that there was no run with a second (false) target detection in it.

## 6 Conclusion

In this paper, we proposed a new extension to the Dynamic Programming Algorithm (DPA) approach for Track-before-Detect 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 in order 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.



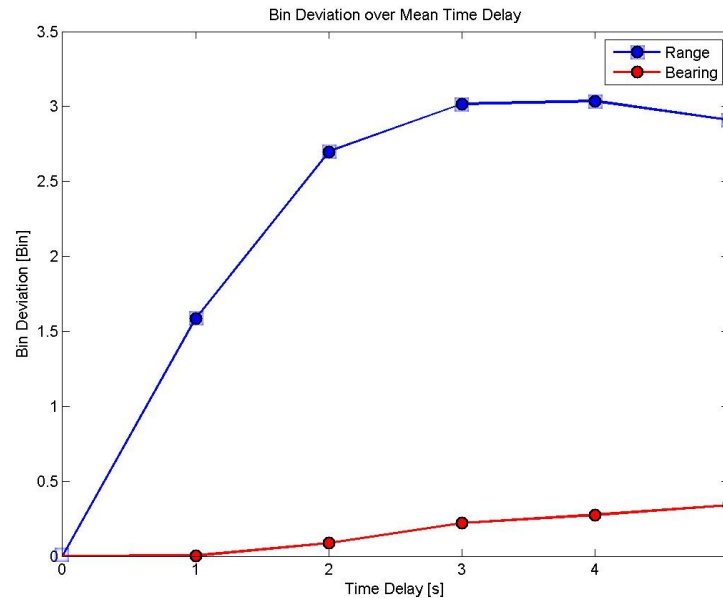


Figure 4: Mean track deviation for OoS DPA

## References

- [ASP93] J. Arnold, S. W. Shaw, and H. Pasternack. Efficient target tracking using dynamic programming. 29(1):44–56, 1993.
- [BP99] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
- [DOR08] B. Demissie, M. Oispuu, and E. Ruthotto. Localization of multiple sources with a moving array using subspace data fusion. In *Information Fusion, 2008 11th International Conference on*, pages 1–7, 302008-july3 2008.
- [GK10] F. Govaers and W. Koch. Distributed Kalman Filter Fusion at Arbitrary Instants of Time. In *Proc. 13th Int. Conf. Information Fusion FUSION '10*, 2010.
- [GYL<sup>+</sup>ar] Felix Govaers, Rong Yang, Hoe Chee Lai, Loo Nin Teow, Gee Wah Ng, and Wolfgang Koch. Track-Before-Detect in Distributed Sensor Applications. *EURASIP Journal on Advances in Signal Processing*, to appear.
- [KVK97] W. Koch and G. Van Keuk. Multiple hypothesis track maintenance with possibly unresolved measurements. 33(3):883–892, 1997.
- [OER10] D. Orlando, F. Ehlers, and G. Ricci. Track-before-detect algorithms for bistatic sonars. In *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, pages 180–185, 2010.

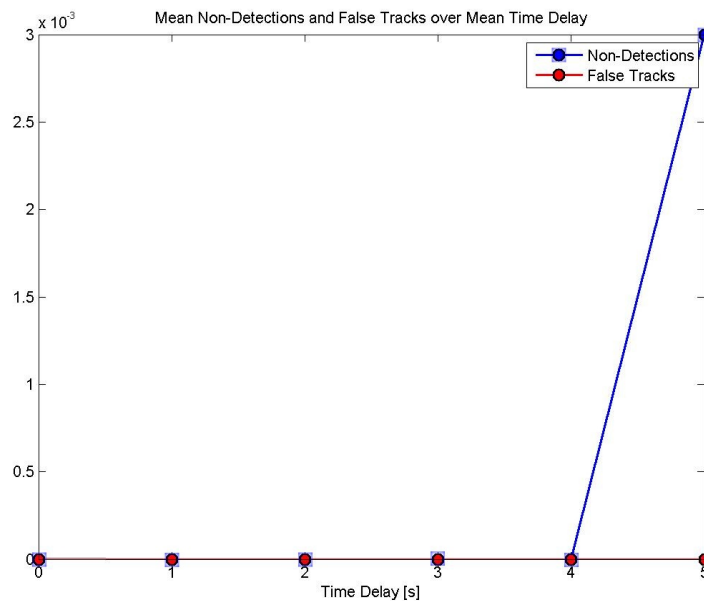


Figure 5: Mean number of false tracks and non-detections

- [Ric96] G.A. Richards. Application of the Hough transform as a track-before-detect method. In *Target Tracking and Data Fusion (Digest No: 1996/253), IEE Colloquium on*, pages 2/1–2/3, November 1996.
- [RRG05] M.G. Rutten, B. Ristic, and N.J. Gordon. A comparison of particle filters for recursive track-before-detect. In *Information Fusion, 2005 8th International Conference on*, volume 1, page 7 pp., 2005.
- [WZF10] Lanling Wei, Xiaoling Zhang, and Ling Fan. A TBD algorithm based on improved Randomized Hough Transform for dim target detection. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 2, pages V2–241–V2–245, 2010.