A track-before-detect algorithm with thresholded observations and closely-spaced targets

Emanuele Grossi, Member, IEEE, Marco Lops, Senior Member, IEEE, and Luca Venturino, Member, IEEE

Abstract

In this letter, we consider the detection architecture in [1], where a track-before-detect processor elaborates the plot-lists provided on a scan-by-scan basis by the detector and plot-extractor of a radar system. We derive a novel track formation procedure in order to provide improved performance in the presence of multiple, closely-spaced targets. Numerical examples are provided to assess the detection capabilities and the accuracy in the estimation of the target position.

Index Terms

Radar systems, multi-frame detection (MFD), track-before-detect (TBD), successive track cancellation (STC).

I. INTRODUCTION

Track-before-detect (TBD) has been proposed for joint target detection and tracking of dim targets, where the signal-to-disturbance ratio (SDR) is small: a number of consecutive scans are jointly processed with no (or very low) intermediate thresholding taking place, and the estimated trajectory is returned at the same time as detection is declared. This technique includes particle filtering [2], [3], dynamic

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

Authors are with the DIEI, Università degli Studi di Cassino e del Lazio Meridionale, Italy 03043. E-mail: e.grossi@unicas.it, lops@unicas.it, l.venturino@unicas.it.

This works was carried out under a research contract sponsored by Selex Galileo, Nerviano (MI), Italy.
programming (DP) [4]–[6], and maximum-likelihood [7], [8] based methods. See also [9] for a survey. In order to limit complexity (which may become quite intense in long-range surveillance systems), a two-step approach has been considered in [1], [4], [10], where a DP-based TBD Processor is used on a short time scale to in order to obtain more reliable detections (or at most short tracklets), which are sent to a true, distinct tracking stage employing Kalman filtering or more sophisticated tracking methods, such as multiple hypotheses tracking, probabilistic data association, etc. [11]. In this context, the authors in [1] focused on the TBD Processor and proposed a novel and efficient DP algorithm for track formation, which does not need discretization of the state space but but directly operates on the plot-lists (provided by the Detector and Plot-Extractor of a standard single-frame system, as shown in Fig. 1). The procedure in [1], however, shows some drawbacks in the presence of multiple, closely-spaced targets, in that strong targets may shadow weak ones. A straightforward way to cope with this situation is to enlarge the search space to all the $t$-uples of admissible trajectories, $t$ being the number of closely spaced targets. This solution, however, is hardly implementable, for it requires large computational resources and prior knowledge of $t$.

In this work, we adapt the successive track cancellation (STC) principle in [12], [13] to the TBD Processor in [1]. Specifically, we derive a novel procedure, which successively estimates target trajectories and removes the corresponding measurements from the input plot-lists, so that enlargement of the search space is not required, and prior knowledge of the number of closely-spaced targets is not needed. Different from [12], [13], we do not need a discretization of the measurement space but directly operate on the plot-lists, thus taking advantage of the fact that the number of candidate detections (or plots) can be much smaller than the number of resolution elements. Numerical results show that our procedure is amenable to real-time implementation and is able to improve the system performance in terms of PD and root mean square error (RMSE) on the estimated position whenever closely-spaced targets are present, while providing the same performance of the procedure in [1] for widely spaced targets.
The reminder of the paper is organized as follows. The system description is provided in the next section, while the track formation procedure with STC is described in Section III. Experimental results are discussed in Section IV, and concluding remarks are given in Section V.

II. SYSTEM DESCRIPTION

At each scan \( n \), the Detector and Plot-Extractor in Fig. 1 receives the raw data from the sensor and produces a list of candidate detections, say \( S_n \). The plot-list \( S_n \) is stored in a matrix, whose \( k \)-th row is the \( k \)-th plot at scan \( n \), defined as \( s_{k,n} = (t_{k,n} \ p_{k,n} \ A_{k,n} \ N_{k,n}) \), with \( t_{k,n} \) the time instant at which the plot has been taken, \( p_{k,n} \) the position measurement (usually the pair range-azimuth, but it may also include range-rate and/or elevation measurements if available), \( A_{k,n} \) the amplitude of the received signal, and \( N_{k,n} \) the power of the disturbance (thermal noise plus clutter). The number of plots at scan \( n \) is \( D_n \), and \( S_n \) is not defined if \( D_n = 0 \). The threshold \( \gamma_1 \) of the Detector and Plot-Extractor is set lower than that used in the single-frame detector, causing an increment in both the probability of detection (PD) and the false alarm rate (FAR), which is the average number of false alarms per minute. The TBD processor correlates the plots in the current plot-list with those in the past \( L - 1 \) plot-lists and confirms or discards each plot in \( S_n \) through a secondary threshold \( \gamma_2 \); the goal is to restore the FAR to a desired level while maintaining part (if not all) of the PD gain.

The TBD processor considered in [1] (and also in [14], [15]) consists of four blocks, as shown in Fig. 2: the Track Formation stage, which correlates the plots acquired in different scans, the Track Pruning stage, which solves possible data association problems, the Plot Confirmation stage, which compares the decision statistics with \( \gamma_2 \), and the Track Smoothing stage, which improves the measurement accuracy.

\( ^1 \)This scheme is general enough to subsume both standard single-frame detection (when \( \gamma_1 = -\infty \) and \( L = 1 \)), and classical multi-frame detection with raw input data (when \( \gamma_1 = -\infty \) and \( L \geq 2 \)).
of the confirmed plots. The track Pruning stage is key since some estimated trajectories at the output of the Track Formation stage may share a common root. This is deleterious, in that target echoes may be responsible for the confirmation of the alarms they caused and of the false alarms in their proximity; moreover, the estimated trajectories of weak targets may erroneously contain plots of strong, close targets. The Track Pruning stage in Fig. 2 attempts to solve these data association problems a posteriori, i.e., after all candidate trajectories have been formed. Here we merge the tasks of these two stages and derive a procedure to solve any data association problem while trajectories are formed. Specifically, a trajectory is first estimated for each plot at the current scan; then, the dominant trajectory (i.e., the one with largest metric) is extracted from each bundle of trajectories sharing a common root, and the corresponding plots are removed from the input data; the process is iterated until there are no trajectories sharing a common root. This procedure is implemented by Algorithm 1 and is referred to as Track Formation with STC; the resulting TBD processor is shown in Fig. 3.

III. TRACK FORMATION WITH STC

Here we illustrate in detail the operations performed by Algorithm 1. Without loss of generality, assume that the current scan is \( n = L \). If a target is present, its trajectory can be uniquely specified by an \( L \)-dimensional index vector, say \( \xi \), defined as follows:\(^2\) if \( \xi_\ell = k \), with \( k \in \{1, \ldots, D_\ell\} \), the target has been observed at scan \( \ell \), and its plot is \( s_{k,\ell} \), while if \( \xi_\ell = 0 \), the target has been missed at scan \( \ell \). Following [1], the metric associated with the trajectory indexed by \( \xi \) is \( \sum_{\ell=1}^{L} z_{\xi_\ell,\ell} \), where \( z_{k,\ell} = A_{k,\ell}^2 / N_{k,\ell} \), if \( k \in \{1, \ldots, D_\ell\} \), and \( z_{k,\ell} = \eta \), if \( k = 0 \), \( \eta \) being a parameter accounting for a missing observation. This metric is related to the energy back-scattered by a prospective target in \( L \) scans.

\(^2\)Here polar (or sensor) coordinates are used for track formation, and velocities are not considered to limit complexity. Therefore, a trajectory is a sequence of consecutive plots taken during \( L \) scans.

Figure 3. TBD Processor with successive track cancellation.
Algorithm 1 Track Formation with STC

Input: \( \{S_{\ell}\}_{\ell=1}^L \) \hspace{1cm} Output: \( \{\mathbf{T}_{k,L}, F_{k,L}\}_{k=1}^{D_L} \)

1: \( A = \emptyset \)
2: \( H_{\ell}^{(1)} = \{1, \ldots, D_{\ell}\}, \quad \ell = 1, \ldots, L \)
3: \( i = 1 \)
4: while \( \mathcal{H}_{\ell}^{(i)} \neq \emptyset \) do
5: \( \{\mathbf{T}_{k,L}, F_{k,L}\}_{k \in \mathcal{H}_{\ell}^{(i)}} = \text{Track-Formation}(\{H_{\ell}^{(i)}, S_{\ell}\}_{\ell=1}^L) \)
6: \( \{\mathbf{K}_{\ell}^{(i)}\}_{\ell=1}^L, \{\mathbf{\hat{T}}_{k,L,\ell}^{(i)}\}_{k \in \mathcal{K}_{\ell}^{(i)}} = \text{Track-Extraction}(\{\mathbf{T}_{k,L,\ell}^{(i)}, F_{k,L,\ell}^{(i)}\}_{k \in \mathcal{H}_{\ell}^{(i)}}) \)
7: \( A = A \cup \{\mathbf{T}_{k,L,\ell}^{(i)}\}_{k \in \mathcal{K}_{\ell}^{(i)}} \)
8: \( H_{\ell}^{(i+1)} = H_{\ell}^{(i)} \setminus \mathcal{K}_{\ell}^{(i)} \), \quad \ell = 1, \ldots, L \)
9: \( i = i + 1 \)
10: end while
11: \( \{\mathbf{T}_{k,L}, F_{k,L}\}_{k=1}^{D_L} = \text{Track-Validation}(A, \{S_{\ell}\}_{\ell=1}^L) \)

consecutive frames. With these definitions, \( \mathcal{H}_{\ell}^{(i)} \) (lines 2 and 8) is the set containing the indexes of the plots available at scan \( \ell \) and iteration \( i \) to form prospective target tracks.

1) Track-Formation: Starting from the plots indexed by \( \{H_{\ell}^{(i)}\}_{\ell=1}^L \), this function (line 5) implements a DP algorithm to estimate the best trajectory ending in each alarm at the current scan \( L \) and compute the corresponding metric. Specifically, it returns

\[
\mathbf{T}_{k,L}^{(i)} = \arg \max_{\xi \in \cal{R}_{k,L}^{(i)}} \sum_{\ell=1}^L z_{\xi,\ell} \quad \text{and} \quad F_{k,L}^{(i)} = \max_{\xi \in \cal{R}_{k,L}^{(i)}} \sum_{\ell=1}^L z_{\xi,\ell}
\]

for all \( k \in H_{\ell}^{(i)} \), where \( \cal{R}_{k,L}^{(i)} \) is the set containing the vectors indexing all trajectories ending in \( s_{k,L} \) that can be formed with the plots indexed by \( \{H_{\ell}^{(i)}\}_{\ell=1}^L \). These trajectories must satisfy the constraints on the maximum target speed (which is set at the design stage) and on the maximum number of consecutive misses (i.e., of consecutive zeros in \( \xi \)), say \( P \), between two detections (needed to avoid large holes in the trajectory). This is a modified version of Algorithm 1 in [1] to account for the fact that at each iteration the set of input plots indexed by \( \{H_{\ell}^{(i)}\}_{\ell=1}^L \) changes, and its implementation is reported in Function 2. This function recursively computes \( \{\mathbf{T}_{k,\ell}^{(i)}, F_{k,\ell}^{(i)}\}_{k \in \mathcal{H}_{\ell}^{(i)}} \) for \( \ell = 1, \ldots, L \). In order to compute \( F_{k,\ell}^{(i)} \), it searches in the set \( \cal{M}_{k,\ell}^{(i)} \) for the best plot that can be linked with alarm \( s_{k,\ell} \), i.e., that satisfies the velocity constraint. If \( \cal{M}_{k,\ell} = \emptyset \), \( F_{k,\ell}^{(i)} \) is set equal to the current measurement \( A_{k,\ell}^2/N_{k,\ell} \) incremented by \( (\ell - 1)\eta \), and the index vector of the corresponding trajectory has \( \ell - 1 \) trailing zeros and \( k \) as the
Function 2 Track-Formation

Input: \( \{H^{(i)}_\ell, S^{(i)}_\ell\}_{\ell=1}^L \)  
Output: \( \{\tau^{(i)}_{k,L}, F^{(i)}_{k,L}\}_{k\in H^{(i)}_L} \)

12: for \( k \in H^{(i)}_1 \) do
13: \( F^{(i)}_{k,1} = A^2_{k,1}/N_{k,1} \)
14: \( \tau^{(i)}_{k,1} = k \)
15: end for
16: for \( \ell = 2, \ldots, L \) do
17: for \( k \in H^{(i)}_\ell \) do
18: \( M^{(i)}_{k,\ell} = \{ (j,p) : p \in \{ \max\{1, \ell - P - 1\}, \ldots, \ell - 1\}, j \in H^{(i)}_p, \text{ and } (s^{(i)}_{j,p}, s^{(i)}_{k,\ell}) \text{ satisfies the velocity constraint} \} \)
19: if \( M^{(i)}_{k,\ell} \neq \emptyset \) then
20: \( (h,m) = \arg \max_{(j,p) \in M^{(i)}_{k,\ell}} F^{(i)}_{j,p} \)
21: \( F^{(i)}_{k,\ell} = F^{(i)}_{h,m} + (\ell - m - 1)\eta + A^2_{k,\ell}/N_{k,\ell} \)
22: \( \tau^{(i)}_{k,\ell} = (\tau^{(i)}_{h,m} 0 \cdots 0 k) \)
23: else
24: \( F^{(i)}_{k,\ell} = (\ell - 1)\eta + A^2_{k,\ell}/N_{k,\ell} \)
25: \( \tau^{(i)}_{k,\ell} = (0 \cdots 0 k) \)
26: end if
27: end for
28: end for

last entry (lines 24 and 25). Otherwise the best admissible past past tracklet (indexed by \( \tau^{(i)}_{h,m} \)) is linked with \( s^{(i)}_{k,\ell} \): \( F^{(i)}_{k,\ell} \) is computed by adding the statistic \( A^2_{k,\ell}/N_{k,\ell} \) to the largest previous metric, stored in \( F^{(i)}_{h,m} \), incremented by \( (\ell - m - 1)\eta \) (to account for \( \ell - m - 1 \) misses, see line 21), and the corresponding trajectory indexed by \( \tau^{(i)}_{k,\ell} \) is updated accordingly (line 22). Function 2 terminates when \( \ell = L \).

2) Track-Extraction: This function (line 6) selects, among all trajectories sharing the same root, the one with the largest metric (i.e., the dominant one). The dominant trajectories extracted at iteration \( i \) are included in the set \( A \) (line 7), while the indexes of their plots are stored in \( \{K^{(i)}_\ell\}_{\ell=1}^L \). The set containing the indexes of plots available at iteration \( i \) is updated in line 8. The while-loop ends when \( H^{(i)}_L = \emptyset \).

3) Track-Validation: Since short trajectories may be unreliable, this function (line 11) shrinks all estimated trajectories with less than a specified minimum value of plots (say \( Q \)), maintaining only the plot at the current scan, and updates their metric accordingly.
Once the algorithm is terminated, each plot \( s_{k,L} \), along with the associated trajectory (indexed by \( \tau_{k,L} \)) and test statistic \( F_{k,L} \), is sent to the Plot Confirmation stage in Fig. 3. The complexity of Algorithm 1 is ruled by that of Function 2, which is linear in the number of integrated scans and quadratic in the average number of alarms in the input data-set [1]. Since the size of the input-data set progressively reduces at each iteration, the computational burden required by Function 2 become negligible after few iterations.

IV. NUMERICAL RESULTS

We discuss a numerical example, where a Swerling I fluctuation model is assumed. Each variable \( A_{k,\ell}^2 / N_{k,\ell} \) is exponentially distributed, and the constant \( \eta \) is set equal to zero. Range and azimuth measurement errors are Gaussian, with standard deviation 20 m and 0.5°, respectively (for simplicity, they are independent of the SDR). The scan period is 1 s, and the search area is ±60° and 40 – 140 km. Four, widely-spaced triplets of targets are generated, and their positions are randomly chosen in ±50° and 50 – 130 km. The targets in each triplet are referred to as Target 1, Target 2, and Target 3, and they follow a constant acceleration model. The separation among targets in each triplet at the last scan (i.e., the current one) is \( d \), and the corresponding velocity and acceleration vectors are random and independent of each other. The SDR of each target evolves with range according to the radar equation, and the value at the last scan is reported in all plots. In all triplets Target \( i \) has the same SDR value, for \( i = 1, 2, 3 \). The thresholds \( \gamma_1 \) and \( \gamma_2 \) in Fig. 1 are set to have FAR equal to \( 10^3 \) and 1 per minute at the input and the output of the TBD processor, respectively. The Track Formation with STC operates with \( L = 10 \), \( v_{\text{max}} = 300 \text{ m/s} \), \( P = 4 \), and \( Q = 3 \), and has no prior information as to the presence of the targets, their number, and their positions; the Track Smoothing stage uses a standard linear regression. For comparison, the performance of the scheme in Fig. 2 and that of the single-frame detector (\( L = 1 \)) are also reported. Finally, the system performances have been estimated by averaging over \( 10^6 \) Monte Carlo runs and over all target groups.

Fig. 4 shows PD and RMSE on the estimated position of the targets versus \( d \), when all targets in the triplet have the same SDR, equal to 9 or 15 dB.\(^3\) It is seen that the scheme in Fig. 2 presents a poor detection capability in the presence of closely-spaced targets, and its performance may even fall below the level of the single-frame detector. The STC strategy, instead, is able to restore the PD gain with respect to the single-frame detector when targets are closely-spaced, making PD almost independent of

\(^3\)The measured target positions are different with probability one due to the presence of measurement errors, even when \( d \to 0 \) (notice that this limiting case may well represent two targets with same azimuth and range separation equal to \( 2\sigma_r = 40 \text{ m} \)).
Figure 4. Probability of detection and root mean square error on the estimated target position versus $d$. All targets have the same SDR.

Moreover, the proposed scheme guarantees an accuracy in the estimated position better than that of the single-frame architecture, performing similarly to the scheme in Fig. 2. Interestingly, for the considered scenario, the estimation error achieves its maximum value around $d = 2 - 3$ km, while decreases for larger and smaller distances. Clearly, in the former case, the estimated trajectory of Target $i$ is unlikely to contain plots from Target $j \neq i$. In the latter case, instead, the linear regression is not dramatically affected by the presence of spurious plots from other targets in the estimated trajectory of Target $i$.

Fig. 5 reports PD and RMSE on the estimated position of the targets versus SDR of Targets 2 and 3, when SDR of Target 1 is 12 dB and $d = 1$ km. Again, the proposed scheme is superior to the scheme in Fig. 2, irrespectively of the SDR imbalance. As to Target 1, it is interesting to notice that the performance of the scheme in Fig. 2 significantly degrades when the SDR of Targets 2 and 3 increases, becoming even worse than that of the single frame detector. For the proposed scheme, instead, both detection and estimation capabilities are almost insensitive to the SDR variation of Target 2 and 3.

Finally, Fig. 6 shows the average execution time versus $d$, when all targets in the triplet have the same
Figure 5. Probability of detection and root mean square error on the estimated target position versus the SDR of Targets 2 and 3 when \( d = 1 \) km.

SDR, equal to \(-\infty\) (i.e., no targets), 9, or 15 dB. For this scenario and our specific code implementation, the running time of the proposed scheme increases with \( d \) and at most doubles that of the scheme in Fig. 2, yet remaining well below the scan duration.

V. CONCLUSION

In this work we have elaborated on the detection architecture in [1] and proposed a novel track formation procedure with successive track cancellation. The proposed solution provides improved performance in the presence of closely-spaced targets with limited increment of the computational load.

REFERENCES

Figure 6. Average execution time vs $d$. All targets have the same SDR.


