Time-Recursive Number-of-Tracks Estimation for MHT

Jessica Bradley, Kevin Buckley and Richard Perry

ECE Department, Villanova University, Villanova PA 19085

ABSTRACT

In this paper we address the issue of measurement-to-track association within the framework of multiple hypothesis tracking (MHT). Specifically, we generate a maximum a posteriori (MAP) cost as a function of the number of tracks K. This cost is generated, for each K, as a marginalization over the set of hypothesized track-sets. The proposed algorithm is developed based on a trellis diagram representation of MHT, and a generalized list-Viterbi algorithm for pruning and merging hypotheses. Compared to methods of pruning hypotheses for either MHT or Bayesian multitarget tracking, the resulting Viterbi MHT algorithm is less likely to incorrectly drop tracks in high clutter and high missed-detection scenarios. The proposed number-of-tracks estimation algorithm provides a time-recursive estimate of the number of tracks. It also provides track estimates, allows for the deletion and addition of tracks, and accounts for false alarms and missed detections.

Keywords: Multiple Hypothesis Tracking, MHT, multiple target tracking, number-of-track estimation, Maximum A Posterior estimation, MAP

1. INTRODUCTION

In multitarget tracking, we start with time evolving sets of noisy measurements of detected targets and false alarms, and we sequentially associate the detections over time to form multiple target tracks. With one class of track estimators, termed *Multiple Hypothesis Trackers (MHT's)*,[1,2] this is accomplished by considering all hypothesized track sets. A *Maximum A Posterior (MAP)* criterion, which is commonly used to compare hypotheses, can account for missed detections and false alarms. MAP costs are computed using Kalman filter generated innovations and a priori track set probabilities. The principal disadvantage of MHT is that the number of hypothesized track sets to be evaluated increases exponentially with time. An effective alternative approach is based on *conditional mean* estimation of the target states. The class of resulting tracking algorithms is referred to as *Bayesian*. Single[3] and multiple[4] target Bayesian algorithms have been developed. Since the conditional mean is computed as a weighted sum of the means conditioned on each hypothesized track set, Bayesian algorithms incur the same computational burden as MHT. MHT and Bayesian tracking algorithms are sometimes referred to as measurement and track based, respectively, since the former algorithms associate measurements while the latter tend to extend existing tracks.

Numerous approaches to hypothesis *pruning* and *merging* have been developed to reduce the computational burden of MHT and Bayesian trackers. Two basic approaches are measurement and likelihood gating, which have been proposed in both MHT,[1,2] and Bayesian,[3,4] algorithms. A shortcoming of both gating and likelihood based pruning is that the optimum track can be dropped, for example, during a target maneuver or after a few missed detections. Track merging is implemented in practical, suboptimal Bayesian single track estimators. For a single track, Singer et al.[3] suggest merging all tracks that share measurements for the past N times. The Probabilistic Data Association Filter (PDAF),[5] an approximate Bayesian tracker, has been interpreted as the $N = 0$ case. Joint PDAF (JPDAF),[6] which extends PDAF to multitarget tracking, time-recursively extends multiple existing tracks which share measurements in their pruning gates. JPDAF, like other suboptimum trackers, can generate diverged tracks during target maneuvers or after a few missed detections. The Viterbi MHT algorithm[7] uses a trellis diagram formulation and a generalized Viterbi algorithm to manage pruning and merging, in the context of multitrack MAP based MHT, so as to keep a certain number of less likely tracks which could later become optimum. This approach can reduce the occurrence of dropped and diverged tracks.

Allowance for an unknown and possibly time-varying number of tracks can be made within the context of either MHT or Bayesian multitarget tracking. For a fixed number of tracks, within an optimum MHT framework, this has been accomplished[2] by employing a MAP cost for hypothesized track-sets which represent different numbers of tracks.

Further author information: K.B.: E-mail: buckley@ece.vill.edu; 610-519-5658, FAX: (610)-519-4436
This research was supported by ONR under Grant N00014-98-1-0892.

Although in that reference the number of tracks was assumed constant over the processing interval, this approach can be generalized to allow for a changing number of tracks. Generally, the number-of-tracks estimation problem can be viewed as a model order selection problem,[8] and methods from the statistical and signal processing literature can be considered for adaptation for multitarget tracking. In this paper we consider what is termed a "Bayesian" model selection approach, which leads to algorithms that extend the MAP MHT approach.[2] Furthermore, we employ the Viterbi MHT algorithm to control computational requirements through managed pruning and merging.

2. MULTITARGET TRACKING PROBLEM

At each measurement time, noisy measurements are provided from multiple target detections. The measurements for each detection consist of a set of D estimated location and/or velocity parameters. Detections can correspond to either targets or false alarms. Missed detections will be accounted for. The probability of detection P_d will be assumed known, as will the density function for the number of false alarms per measurement time in the surveillance volume.

We assume that K, the number of targets, is unknown. For theoretical development, we assume that K is constant over the processing interval. We then develop a processing algorithm for the estimation of K which allows variation of K over time.

We denote as $\mathbf{z}_{m,j}$ the j^{th} measurement vector at time m. Then, $\mathbf{Z}_m = \{\mathbf{z}_{m,j}; j = 1, 2, ..., J_m\}$ is the set of J_m measurement vectors at time m, and $\mathcal{Z}^n = \{\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_n\}$ denotes the set of all measurement vectors up to time n.

2.1. Hypothesized Track Sets

A single hypothesized track is characterized by a measurement vector or a missed detection for each measurement time. At time n let one such track, the l^{th} track, be denoted by the measurement-to-track association vector

$$\boldsymbol{\theta}_n^l = \{j_1(l), j_2(l), \cdots, j_n(l)\} \tag{1}$$

where the subscript $j_m(l)$ is the measurement/missed-detection index at time m for the l^{th} track. Note that we account for the possibility of a missed detection by letting $j_m(l)$ range from 1 to $J_m + 1$, with $j_m(l) = J_m + 1$ indicating a missed detection. The vector of measurements for the l^{th} track is

$$\mathcal{Z}(\boldsymbol{\theta}_n^l) = \{\mathbf{z}_{1,j_1(l)}, \mathbf{z}_{2,j_2(l)}, \cdots, \mathbf{z}_{n,j_n(l)}\} \ . \tag{2}$$

$\mathbf{z}_{m,J_m+1}$ is not an actual measurement, but indicates a missed detection at time m, and is a notational convenience. There are $L_n' = \prod_{m=1}^{n}(J_m + 1)$ of these hypothesized tracks: $\boldsymbol{\theta}_n^l; l = 1, 2, \cdots, L_n'$. Association of the measurements into a single track is the problem of selecting the best $\boldsymbol{\theta}_n^l$, which is equivalent to estimating the discrete parameter $l \in \{1, 2, ..., L_n'\}$.

For a given hypothesized track $\boldsymbol{\theta}_n^l$, measurement noise is assumed additive, Gaussian and temporally white. The trajectory and measurements are assumed to evolve in time according to the state/measurement equations*

$$\mathbf{x}_{m+1}^l = \boldsymbol{\Phi}_m \mathbf{x}_m^l + \mathbf{w}_m^l \tag{3}$$

$$\mathbf{z}_{m,j_m(l)} = \mathbf{H}_m \mathbf{x}_m^l + \mathbf{u}_m^l \tag{4}$$

where at time m and for hypothesized track $\boldsymbol{\theta}_n^l$, $\mathbf{x}_m^l$ is the state vector, and $\boldsymbol{\Phi}_m$ and $\mathbf{H}_m$ are the state transition and output matrices respectively. The $\mathbf{w}_m^l$ and $\mathbf{u}_m^l$ vectors are zero mean, mutually independent, white and Gaussian with known covariance matrices $\mathbf{Q}_m^l$ and $\mathbf{R}_m^l$ respectively.

A Kalman filter can be applied to this track to smooth the measurements, to provide minimum variance state vector estimates, and to compute the innovations for the measurement sequence $\mathcal{Z}(\boldsymbol{\theta}_n^l)$. For each hypothesis, the measurement innovations sequence $\{\mathbf{v}_{1,j_1(l)}, \mathbf{v}_{2,j_2(l)}, \cdots, \mathbf{v}_{n,j_n(l)}\}$ and corresponding covariance matrices $\{\mathbf{S}_{1,j_1(l)}, \mathbf{S}_{2,j_2(l)}, \cdots, \mathbf{S}_{n,j_n(l)}\}$ generated using a hypothesis-specific Kalman filter are to be used in optimum track estimation. Note that a $\mathbf{v}_{m,j_m(l)}$ corresponding to $\mathbf{z}_{m,J_m+1}$ is not an actual innovation, but is a result of a missed detection at time m, and again is a notational convenience. In such a case, the Kalman state is just the predicted state (i.e. no measurement is available to update the state).

*To simplify the discussion we assume linear state/measurement equations.

2.2. Hypothesized K-Track Sets

Now consider a set of K tracks. Assume tracks can not share detections, and again assume missed detections are possible.† It can be shown that the number of hypothesized track sets is‡

$$I_n^{'}(K) = \left(\sum_{j=0}^{K} \left(\begin{array}{c} J_1 \\ j \end{array} \right) \right) \left(\prod_{m=2}^{n} \left(K! \sum_{j=0}^{K} \frac{\left(\begin{array}{c} J_m \\ j \end{array} \right)}{(K-j)!} \right) \right) \quad . \tag{5}$$

Remember that K is unknown (i.e. it is an unknown variable). For a hypothesized K, let the index $i(K)$ denote the $i(K)^{th}$ hypothesized K-track set. This K-track set is represented by the $i(K)^{th}$ set of measurement-to-track associations:

$$\boldsymbol{\tau}_n^{i(K)} = \{\boldsymbol{\theta}_n^{l_1(i(K))}, \boldsymbol{\theta}_n^{l_2(i(K))}, \cdots, \boldsymbol{\theta}_n^{l_K(i(K))}\} \quad , \tag{6}$$

where the superscript $l_k(i(K))$ denotes the k^{th} track of the $i(K)^{th}$ K-track set. $\mathcal{Z}(\boldsymbol{\tau}_n^{i(K)})$ will be used to denote the measurement data corresponding to the $i(K)^{th}$ K-track set.

For known K, Multiple Hypothesis Tracking (MHT) algorithms aim to determine at time n the best from the $I_n^{'}(K)$ hypothesized track sets. $I_n^{'}(K)$ grows exponentially with n, with a multiplicative increase of

$$K! \sum_{j=0}^{K} \frac{\left(\begin{array}{c} J_n \\ j \end{array} \right)}{(K-j)!} \tag{7}$$

hypothesized tracks at each time n. So, $I_n^{'}(K)$ can be prohibitively large for even moderate values of n, K and numbers of measurements.

Below, for the estimation of K, we introduce a MAP based approach which treats K as an additional variable to be hypothesized over. This increases, relative to MHT with known K, the number of hypotheses to consider. We address this algorithmic issue in Section 4.

3. MAP BASED ESTIMATION OF THE NUMBER OF TRACKS

Here we introduce two MAP based procedures to estimate the number of tracks K: one based on joint estimation of K and the tracks; and one in which for each K the hypothesized tracks are marginalized over. In Section 5 we illustrate that the latter approach can outperform the former in limited data situations.

3.1. Joint Estimation of K and the Tracks

We begin with a MAP formulation, which incorporates prior track set probabilities, for the estimation of both the number of tracks K and the tracks themselves (i.e. the track set associations $\boldsymbol{\tau}_n^{i(K)}$). At time n we have the following optimization problem. For $K = 0, 1, \cdots, K_{max}$ and $\boldsymbol{\tau}_n^{i(K)}; i(K) = 1, 2, \cdots I_n^{'}(K); K = 0, 1, \cdots, K_{max}$

$$\max_{K, \boldsymbol{\tau}_n^{i(K)}} \quad p\left(K, \boldsymbol{\tau}_n^{i(K)} | \mathcal{Z}^n\right) = \frac{p\left(\mathcal{Z}^n | K, \boldsymbol{\tau}_n^{i(K)}\right) \cdot p\left(\boldsymbol{\tau}_n^{i(K)} | K\right) \cdot p\left(K\right)}{p\left(\mathcal{Z}^n\right)} \quad . \tag{8}$$

$p\left(\mathcal{Z}^n\right)$ is a normalizing factor which serves to make the possible values of $p\left(\boldsymbol{\tau}_n^{i(K)} | \mathcal{Z}^n\right)$ sum to unity, and can be ignored since it is independent of $\boldsymbol{\tau}_n^{i(K)}$ and K. Concerning $p(K)$, any prior distribution can be incorporated. However, here we will assume that K is uniformly distributed from $K = 0$ to some maximum value $K = K_{max}$. The term $p\left(\boldsymbol{\tau}_n^{i(K)} | K\right)$ is the *a priori* probability of the $i(K)^{th}$ set of tracks, and is derived from knowledge of the sensors,

---

†These assumptions allow for representation of unresolved targets as a combination of detections and missed detections.

‡$\left(\begin{array}{c} A \\ B \end{array} \right)$ represents the number of combinations of A things taken B at a time.

targets, noise, clutter, jammers and preprocessors, as manifested for example as the probability of missed detections and the false alarm rate. In Section 5 a specific $p\left(\boldsymbol{\tau}_n^{i(K)}|K\right)$ will be used for illustration purposes.

$p\left(\mathcal{Z}^n|K,\boldsymbol{\tau}_n^{i(K)}\right)$ is the joint probability density function of the measurements $\mathcal{Z}^n$ conditioned on the number of tracks K and the track set $\boldsymbol{\tau}_n^{i(K)}$. For a given K and $\boldsymbol{\tau}_n^{i(K)}$, we can partition $\mathcal{Z}^n$ into the data associated with the tracks, $\mathcal{Z}(\boldsymbol{\tau}_n^{i(K)})$, and data associated with false alarms, $\overline{\mathcal{Z}}(\boldsymbol{\tau}_n^{i(K)})$. We then have

$$p\left(\mathcal{Z}^n|K,\boldsymbol{\tau}_n^{i(K)}\right) = p\left(\mathcal{Z}(\boldsymbol{\tau}_n^{i(K)})|K,\boldsymbol{\tau}_n^{i(K)}\right)\cdot p\left(\overline{\mathcal{Z}}(\boldsymbol{\tau}_n^{i(K)})|K,\boldsymbol{\tau}_n^{i(K)}\right) \quad , \tag{9}$$

where, under the assumption that false alarms are uniformly distributed over the surveillance volume Y,

$$p\left(\overline{\mathcal{Z}}(\boldsymbol{\tau}_n^{i(K)})|K,\boldsymbol{\tau}_n^{i(K)}\right) = \left(\frac{1}{Y}\right)^{\sum_{m=1}^{n} F_m^{i(K)}} \quad , \tag{10}$$

where $F_m^{i(K)}$ is the number of false alarms at time m given hypothesized track set $\boldsymbol{\tau}_n^{i(K)}$. In terms of the Kalman filter generated measurement innovations, for the track measurements[§],

$$p\left(\mathcal{Z}(\boldsymbol{\tau}_n^{i(K)})|K,\boldsymbol{\tau}_n^{i(K)}\right) = \prod_{p=1}^{K} p(\mathcal{Z}(\boldsymbol{\theta}_n^{l_p(i(K))})) = \prod_{p=1}^{K}\prod_{m=1}^{n} p(\mathbf{v}_{m,j_m(l_p(i(K)))}) \tag{11}$$

$$= d_n^{i(K)} \exp\left\{-\frac{1}{2}\sum_{p=1}^{K}\sum_{m=1}^{n} \mathbf{v}_{m,j_m(l_p(i(K)))}^T \mathbf{S}_{m,j_m(l_p(i(K)))}^{-1} \mathbf{v}_{m,j_m(l_p(i(K)))}\right\} \tag{12}$$

where for each track p the product (or sum) over m does not include any missed detections since as noted earlier measurements and innovations for missed detections do not exist. Keeping this in mind,

$$d_n^{i(K)} = \prod_{p=1}^{K}\prod_{m=1}^{n} \frac{1}{(2\pi)^{D/2}\cdot\sqrt{\det(\mathbf{S}_{m,j_m(l_p(i(K)))})}} \quad . \tag{13}$$

Note that, for the $K=0$ hypothesis, there are no measurements $\mathcal{Z}(\boldsymbol{\tau}_n^i)$ associated with tracks. Then the first term on the right of (9) is not present. All measurements are considered false alarms, resulting in a simple MAP probability computation composed of (10) and a single $p\left(\boldsymbol{\tau}_n^{i(K)}|K=0\right)$ (i.e. for $i(0)=1$).

The resulting joint MAP estimator of K and $\boldsymbol{\tau}_n^{i(K)}$ is

$$\hat{K},\hat{\boldsymbol{\tau}}_n^{i(K)} = \arg\max_{K,\boldsymbol{\tau}_n^{i(K)}} \left\{p\left(\mathcal{Z}(\boldsymbol{\tau}_n^{i(K)})|K,\boldsymbol{\tau}_n^{i(K)}\right)\cdot p\left(\overline{\mathcal{Z}}(\boldsymbol{\tau}_n^{i(K)})|K,\boldsymbol{\tau}_n^{i(K)}\right)\cdot p\left(\boldsymbol{\tau}_n^{i(K)}|K\right)\right\} \tag{14}$$

where the propability is searched over all $\boldsymbol{\tau}_n^{i(K)}; i=1,2,\cdots I_n^{'}(K)$ for each $K=0,1,\cdots,K_{max}$. (Referring to (5), note that the number of hypothesized track sets for a given K is a function of K.)

3.2. Marginalization for Number of Track Estimation

If estimation of K is of primary interest, it can be accomplished without explicit estimation of $\boldsymbol{\tau}_n^{i(K)}$ by marginalizing $p\left(K,\boldsymbol{\tau}_n^{i(K)}|\mathcal{Z}^n\right)$ over the $\boldsymbol{\tau}_n^{i(K)}$. Compared to (8), the resulting estimator,

$$\hat{K} = \arg\max_{K}\left\{p\left(K|\mathcal{Z}^n\right) = \sum_{i(K)=1}^{I_n^{'}(K)} p\left(K,\boldsymbol{\tau}_n^{i(K)}|\mathcal{Z}^n\right)\right\} \quad , \tag{15}$$

can have better performance characteristics, particularly for small number-of-measurement cases.

[§]For $m=1$, there are no Kalman predictions, so the innovations are just the measurements.

4. MHT VITERBI ALGORITHM FOR NUMBER-OF-TRACK ESTIMATION

In this section we describe an approach to controlling the computational load for the MHT methods described above. This approach starts with a trellis diagram representation on hypothesized track sets, and uses a generalized Viterbi algorithm approach to hypothesis pruning and merging. A principal advantage of this approach to hypothesis reduction is that at any time n it systematically keeps some less likely hypotheses, which reduces the probability of dropping hypotheses that may become most likely at a later time.

In the first two subsections we assume that K is known and constant over the processing time. In Subsection 4.3 we expand that approach to unknown but constant K, and in Subsection 4.4 we discuss time varying unknown K.

4.1. Trellis Diagram Formulation

To derive an equivalent cost from which a time recursive trellis structure representation can be developed, take the negative natural log of (14). The following equivalent cost is obtained:

$$\Lambda^n\left(K, \boldsymbol{\tau}_n^{i(K)}\right) = \sum_{m=1}^{n} \Lambda_m(K, \boldsymbol{\tau}_n^{i(K)}), \tag{16}$$

where the time m incremental cost is

$$\Lambda_m(K, \boldsymbol{\tau}_n^{i(K)}) = \lambda_m(K, \boldsymbol{\tau}_n^{i(K)}) + \sum_{p=1}^{K} \lambda_m(\boldsymbol{\theta}_n^{l_p(i(K))}), \tag{17}$$

where $\lambda_m(K, \boldsymbol{\tau}_n^{i(K)}) = \ln\{p\left(\boldsymbol{\tau}_n^{i(K)}|K\right)\}$ and the p^{th} track incremental cost at time m is

$$\begin{aligned}\lambda_m(\boldsymbol{\theta}_n^{l_p(i(K))}) &= \frac{1}{2}\ln(\det(\mathbf{S}_{m,j_m(l_p(i(K)))})) + \frac{D}{2}\ln(2\pi) \\ &+ \frac{1}{2}\mathbf{v}^T_{m,j_m(l_p(i(K)))}\mathbf{S}^{-1}_{m,j_m(l_p(i(K)))}\mathbf{v}_{m,j_m(l_p(i(K)))}\ .\end{aligned} \tag{18}$$

For now, assume that K is known. Assume that tracks can not share detections, and that missed detections (as many as K at time m) are possible. With these assumptions we define a trellis representation of all hypothesized K track sets. This trellis is depicted in Figure 1. At each stage (time), each state represents a set of K measurements and/or missed detections. (This is the trellis proposed by Wolf, et al.[9])

For stage m with J_m measurements, there are

$$M_m = \sum_{j=0}^{K} \binom{J_m}{j} \tag{19}$$

states (nodes). A branch is a connection from a state at some time $m-1$ to one at time m. Each branch from stage $m-1$ to m has a set of $K!/(K_{max} - J_m + F_m^{i(K)})!$ permutations associated with it, specifying the possible orders in which the K measurements and/or missed detections represented by the state at stage m are used.[¶] Each permutation corresponds to one or more hypothesized K-track set.

Each incremental cost assigned to a branch is the sum of a prior cost $\lambda_m(K, \boldsymbol{\tau}_n^{i(K)})$ and the costs of the K hypothesized tracks. So, for hypothesized K-track set $\boldsymbol{\tau}_n^{i(K)}$, the branch cost from stage $m-1$ to m is $\Lambda_m(\boldsymbol{\tau}_n^{i(K)})$. Note that the problem of identifying the best K-track set is now one of finding the minimum-cost path through this trellis.

[¶]Because each state represents all permutations of the corresponding K measurements/missed-events, each path through the trellis represents a number of hypothesized track sets.

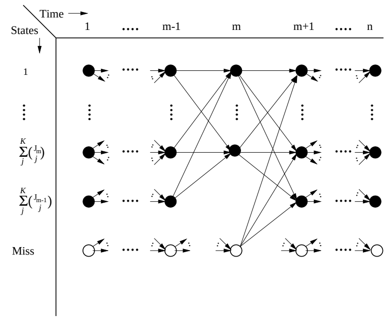

Figure 1. Trellis Diagram for general K.

4.2. The MHT Viterbi Algorithm for Known K

Based on the trellis diagram problem formulation described above, we now describe a multitarget track estimation algorithm which provides a list of L track sets. The algorithm is sequential, self-initiating and can handle false alarms and missed detections. In this subsection we assume K is known and constant, and we are interested in identifying the MAP estimate of the K-track set.

The problem is to find the lowest cost path through the trellis described in Subsection 4.1. For the MAP estimation problem considered here, the need for (infinite memory) Kalman filters for each hypothesized track precludes the use of the Viterbi algorithm, so that to solve the problem at time n an exhaustive search of all $I_n^{'}(K)$ paths through the trellis appears necessary. At any stage n in the trellis, this requires that we consider all paths into each state at stage $n-1$, each extended to all states at stage n. That is, the $I_{n-1}^{'}(K)$ paths into stage $n-1$ must be extended to the $I_n^{'}(K)$ paths into stage n, even though the costs of some of these paths will indicate that the corresponding K-track sets are highly unlikely. We propose to keep, at each stage in the trellis, a "list" of only the best (lowest cost) L paths to each state, where L is selected to assure that no feasible paths are pruned[‖]. Generically, the list Viterbi algorithm[10] does this.

The multitarget tracking algorithm we employ[7] incorporates the list Viterbi algorithm, along with the K-track trellis formulation of the multitarget tracking problem,[9] and MAP cost computation based on Kalman filter innovations and prior tract set probabilities. The algorithm determines a list of L feasible K-tracks sets as a list of L paths through the measurement set trellis.

Algorithm

1) *Setup:* For each time $m = 1, 2, \cdots, n$, allocate arrays of size M_m-by-L for the predecessor state indices, predecessor state L-best indices, permutation number of the current state measurement indices to track associations, and current total minimum negative log costs. Allocate Kalman filters for each track in each L-best sets of tracks for each state.

2) *Initialization:* For $m = 1$, $l = 1$, the current costs are set using the *a priori* probabilities, since no Kalman filter innovations are available initially. The Kalman filters are initialized using the available measurements for time $m = 1$, and using predicted values for time $m = 2$.

3) *Iteration:* For each time $m = 2, 3 \cdots, n$, and for each state $j = 1, 2 \cdots, M_m$

   3.1) Add incremental costs (17) from the L-best of each previous state to the total previous costs, and temporarily store these hypothesized new costs.

   3.2) Implement additional hypothesis merging and pruning as discussed below.

‖Selection of L will depend both on the distribution of false alarm measurements and on the variance of true target measurements. This issue is not addressed here.

3.3) Select the L-best of the hypothesized new costs** and update the associated Kalman filters.

4) *Results:* Output the final L-best sets of tracks.

Although the algorithm is described above in block data form, it is time-sequential, and at any time during the iterations can produce current estimates of the best tracks.

Additional Hypothesis Merging and Pruning

The list-Viterbi implementation prunes the number of hypotheses out of stage m to $L \cdot M_m$. The value of L required to achieve a certain level of performance can be reduced by merging similar paths into a state prior to pruning the number of paths to L. This way, the list of L will contain diverse hypotheses, which is the objective (i.e. we want to keep some less-likely, significantly different hypotheses on the chance that they will become part of more likely hypotheses later in time).

In order to merge similar paths, a criteria must be established to determine whether paths are similar enough to be merged. The criteria used in our simulation code follows. Tracks are merged if:

1. they end on the same trellis state and state permutation, or if some of the K tracks end in a sequence of missed target events and the last actual measurements used for these tracks (at some previous trellis stage) are the same while the rest of the K tracks have common measurements back to the this previous stage; and

2. the trellis states used for the track sets coincide for at least two stage indices, and differ for no more than one stage index.

At each trellis state, the similar paths are first identified. For each set of similar paths, the paths are probabilistically averaged,[3] i.e. the associated Kalman filter states are averaged, weighted by their normalized probabilities. The resulting averaged Kalman states then replace the states for the most likely path in the set, and the other similar paths are pruned.

4.3. Number-of-Track Estimation

Exact implementation of either the joint $K, \boldsymbol{\tau}_n^{i(K)}$ estimator or the marginalized K estimator requires calculation of costs for all hypothesized track sets $\boldsymbol{\tau}_n^{i(K)}$ for all possible K. This in not practical. Alternatively we use the MHT Viterbi algorithm to prune and merge.

Assume that K_{max} is the largest possible value of K. For the trellis and algorithm described in Subsections 4.1 and 4.2 above, consider a trellis for a fixed value of K_{max}. Embedded in the branch and path costs of this trellis are all of the costs required to evaluate all K-track set hypotheses for all $K = 0, 1, \cdots, K_{max}$. So, we can use this single trellis structure to solve the K estimation problem. The number-of-tracks estimator we have implemented:

1) Constructs the trellis for $K = K_{max}$.

2) Employs the Viterbi MHT algorithm to prune hypothesized track sets.

3) Uses track costs stored in the trellis to generate track set costs for $K = 0, 1, \cdots, K_{max}$.

4) Estimates K using (a merged/pruned version of) either (14) or (15).

**For this selection process, we have developed of an "N-best" type algorithm,[11] based an a modification of the Karp[12] linear assignment programming algorithm.

4.4. Time Varying Number-of-Track Estimation

For the tracking scenario described in Section 2 and number-of-tracks estimation problems formulated in Section 3, the number of tracks K was assumed fixed over the processing interval. This lead to the following set of hypothesized track-sets:

$$\boldsymbol{\tau}_n^{i(K)} \ ; \ K = 0, 1, \cdots, K_{max} \tag{20}$$

where the $i(K)$ are the hypothesis indices for the different K. Although this can be a formidable number of hypotheses, as we have described Viterbi MHT can be employed to manage this, though as with any pruning/merging strategy this results in a suboptimal estimator.

Here we generalize the scenario by partitioning the processing interval into measurement blocks of temporal length T, and by allowing K to vary from block to block. For time $n = tT$, where t is the block index, the set of hypothesized track-sets is now:

$$\boldsymbol{\tau}_n^{i(K_1, K_2, \cdots, K_t)} \ ; \ K_t = 0, 1, \cdots, K_{max} \ ; \ \ t = 1, 2, \cdots, T \ . \tag{21}$$

Now $i(K_1, K_2, \cdots, K_t)$ are the hypothesis indices for the number-of-tracks sequences $K_1, K_2, \cdots, K_t$. The number of hypotheses is even more formidable, but again Viterbi MHT can be used to manage this.

The Viterbi MHT algorithm as described above is modified by "truncating the trellis", a technique commonly used in digital communication applications of the Viterbi algorithm. We propose the following. Consider processing the t^{th} measurement block to estimate K_t. We assume that the estimates for previous blocks, i.e. $\hat{K}_1, \cdots, \hat{K}_{t-1}$, are correct. For the current block, t, the hypotheses are:

$$\boldsymbol{\tau}_n^{i(\hat{K}_1, \hat{K}_2, \cdots, \hat{K}_{t-1}, K_t)} \ ; \ K_t = 0, 1, \cdots, K_{max} \ , \tag{22}$$

where $i(\hat{K}_1, \hat{K}_2, \cdots, \hat{K}_{t-1}, K_t)$ are the hypothesis indices for $\hat{K}_1, \hat{K}_2, \cdots, \hat{K}_{t-1}$ fixed and K_t varying from 0 to K_{max}. Note that $n = tT$.

In the Viterbi MHT algorithm, at time $(t-1)T$, $\hat{K}_{t-1}$ is computed, the costs for the hypotheses $\boldsymbol{\tau}_{(t-1)T}^{i(\hat{K}_1, \hat{K}_2, \cdots, \hat{K}_{t-1})}$ are computed, and the trellis paths (i.e. the hypotheses) are merged and pruned. Starting at time $n = (t-1)T + 1$, at each time n Viterbi MHT is run as described above. That is, costs for hypotheses for all $K_t = 1, 2, \cdots, K_{max}$ are computed and trellis paths are merged and pruned. At time $n = tT$, the trellis is truncated. That is, $\hat{K}_t$ is computed (using either joint estimation or merging) and fixed for the next block.

5. NUMERICAL EXAMPLES

For a given hypothesis $\boldsymbol{\tau}_n^{i(K)}$ let $J_m = T_m^{i(K)} + F_m^{i(K)}$, where $T_m^{i(K)}$ is the number of measurement vectors at time m used in the K track set $\boldsymbol{\tau}_n^{i(K)}$ and $F_m^{i(K)}$ is the number of false alarms. Here we assume that

$$p\left(\boldsymbol{\tau}_n^{i(K)}\right) = \prod_{m=1}^{n} \ P_d^{T_m^{i(K)}} \ (1 - P_d)^{K - T_m^{i(K)}} \ p(F_m^{i(K)}) \ \ , \tag{23}$$

where $p(F_m^{i(K)})$ is the Poisson distributed probability of $F_m^{i(K)}$ false alarms. Then the incremental *a priori* cost for (17) is

$$\lambda_m(\boldsymbol{\tau}_n^{i(K)}) \ = \ F_m^{i(K)} \ln(Y) \ - \ T_m^{i(K)} \ln(P_d) \ - \ (K - T_m^{i(K)}) \ln(1 - P_d) \ - \ \ln(p(F_m^{i(K)})) \ . \tag{24}$$

We have demonstrated[7] how, with proper selection of the list length L in conjunction with merging and pruning, the Viterbi MHT algorithm can recover form a diverged track after several missed detections. We also show that Viterbi MHT can generate lists of track-set estimates, each with a close to optimum cost, so that if other information becomes available the track-set that is most consistent can be selected. Here we focus on number-of-track estimation. Four sets of simulation results are presented here. For the first two, we assume the number of tracks is fixed over the processing. For the last two we use the Viterbi MHT algorithm which allows for variation in the number of tracks.

First we conducted a Monte Carlo simulation to compare the joint K/track estimator (referred to below as the MAP estimator) and the marginalization estimator. $K = 2$ tracks were simulated which were linear in the $\mathcal{X}$ and $\mathcal{Y}$ position values values. $N = 500$ trials were conducted, with $n = 6$ measurement times per trial. Gaussian measurement noise was added to each target measurement. The number of "false detect" events was generated

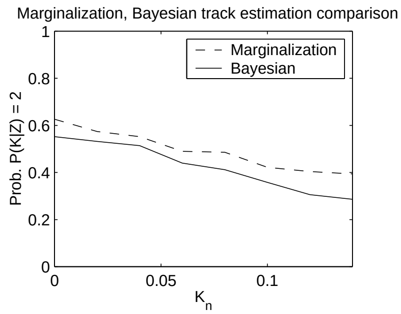

Figure 2. Number-of-track estimation comparison: MAP vs. marginalization.

using a Poisson distribution with a false alarm rate of 2, and the false detections themselves were generated using a uniform distribution over the range [0,4] in both $\mathcal{X}$ and $\mathcal{Y}$. Probability of a missed detection was assumed to be 0.3. K_{max}=3 and L=16. Figure 2 shows estimator performance for varying maesurement and model noise variance. On the basis of these simulations, it can be concluded that estimation by marginalization can be advantageous over MAP estimation. Note, however, that this advantage is not significant for all scenarios. The advantage is more pronounced with limited data situations.

Second, Monte Carlo simulations were run to study K estimator performance for varying probability of detection P_d. N=100 trials were run per P_d value. $K = 2$ linear tracks were simulated, with $n = 21$ measurement times. Measurement noise variance was 0.01. A Poisson distribution with a false alarm rate of 1 was used for the number of false detections, and the false detections themselves were generated using a uniform distribution over the range [0,4] in both $\mathcal{X}$ and $\mathcal{Y}$. $K_{max} = 3$ and $L = 32$ were used. Table 1 shows the percentages for different estimates of K vs. P_d.

$\longrightarrow K$ $\downarrow P_d$	0	1	2	3
1.00	0	7	93	0
0.95	0	0	96	4
0.90	0	0	91	9
0.85	0	3	85	12
0.80	1	6	72	21

Table 1. Number-of-track estimation results: estimated number of tracks vs. P_d.

For the third simulation two parabolic target tracks consisting of $n = 20$ $\mathcal{X}$ and $\mathcal{Y}$ position values were generated, where $\mathcal{X}$ varied linearly over time from $\mathcal{X} = 0$ to $\mathcal{X} = 4$. The two targets were present over the entire processing interval. Gaussian measurement noise with a variance of 0.01 was added to the target tracks. The number of "false detect" events was generated using a Poisson distribution with a false alarm rate of 2, and the false detections themselves were generated using a uniform distribution over the range [0,4] in both $\mathcal{X}$ and $\mathcal{Y}$. Probability of a missed detection was assumed to be 0.2. K_{max}=3 and L=8. We partitioned the data into 4 blocks of $T = 5$ measurement times. Figure 3(a) shows the $\mathcal{X}$ and $\mathcal{Y}$ position values for the true tracks (solid lines), along with the noisy measurements (boxed and circled x's) and false alarms (x's) used as input to the algorithm. Two tracks were detected for each block (i.e. $\hat{K}_t = 2$; $t = 1, 2, 3, 4$). Figure 3(b) shows the resulting best 2-track set estimate.

For the fourth simulation two parabolic target tracks consisting of $n = 20$ $\mathcal{X}$ and $\mathcal{Y}$ position values were again generated, where $\mathcal{X}$ varied linearly over time from $\mathcal{X} = 0$ to $\mathcal{X} = 4$. This time, one target was turned off over the middle 10 measurement times. Again, Gaussian measurement noise with a variance of 0.01 was added to the target tracks. The number of "false detect" events was generated using a Poisson distribution with a false alarm rate of 2, and the false detections themselves were generated using a uniform distribution over the range [0,4] in both $\mathcal{X}$ and

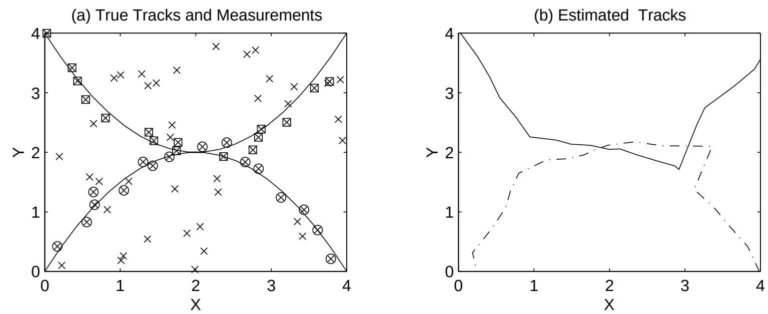

Figure 3. Two parabolic tracks estimated with the time-varying K algorithm: (a) true track and measurements with false detections; (b) estimated tracks.

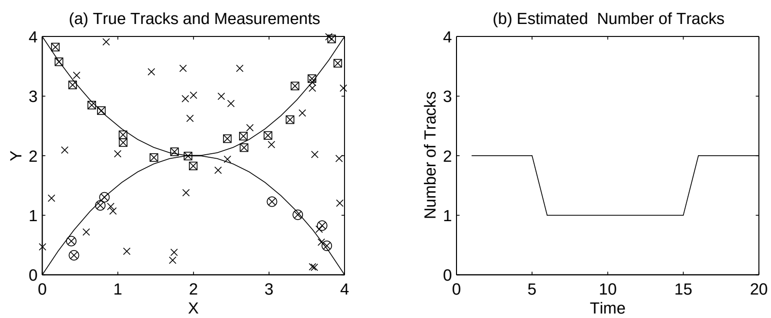

Figure 4. Two parabolic tracks estimated with the time-varying K algorithm: (a) true track and measurements with false detections; (b) estimated number of tracks vs. time.

$\mathcal{Y}$. Probability of a missed detection was assumed to be 0.2. K_{max}=3 and L=8. We partitioned the data into 4 blocks of $T = 5$ measurement times. Figure 4(a) shows the $\mathcal{X}$ and $\mathcal{Y}$ position values for the true tracks (solid lines), along with the noisy measurements (boxed and circled x's) and false alarms (x's) used as input to the algorithm. This time the algorithm correctly detected two tracks for the first and last block, and one track for the middle two (i.e. $\hat{K}_1 = \hat{K}_4 = 2$, $\hat{K}_2 = \hat{K}_3 = 1$). Figure 4(b) shows $\hat{K}$ vs. time.

6. CONCLUSION

In this paper, we introduced two number-of-tracks estimation algorithms, which employ the Viterbi MHT algorithm for MAP based multiple hypothesis multitarget tracking. One algorithm jointly estimates both the number-of-tracks and the measurement-to-track associations, while the other marginalizes over the associations to provide an estimate of the number-of-tracks only. Simulation results illustrate that this approach can be effective, and that marginalization can offer some advantage over the joint estimator.

REFERENCES

1. P. Smith and G. Buechler, "A branching algorithm for discriminating and tracking multiple objects," *Trans. on Automatic Control* **20**, pp. 101–104, Feb 1975.
2. C. Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *Trans. on Automatic Control* **22**, pp. 302–312, June 1977.
3. R. A. Singer, R. G. Sea, and K. M. Housewright, "Derivation and evaluation of improved tracking filters for use in dense multitarget environments," *IEEE Trans. on Info. Theory* **20**, pp. 423–432, July 1974.
4. D. B. Reid, "An algorithm for tracking multiple targets," *Trans. on Automatic Control* **AC-24**, pp. 843–854, Dec 1979.

5. Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica* **11**, pp. 451–460, Sept 1975.
6. K.-C. Chang and Y. Bar-Shalom, "Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers," *IEEE Trans. on Automatic Control* **29**, pp. 585–594, July 1984.
7. K. Buckley, A. Vaddiraju, and R. Perry, "A new pruning/merging algorithm for MHT multitarget tracking," *Radar–2000* , May 2000.
8. B. Radich and K. Buckley, "Single-snapshot DOA estimation and source number detection," *Signal Processing Letters* **4**, pp. 109–111, Apr 1997.
9. J. K. Wolf, A. M. Viterbi, and G. S. Dixon, "Finding the best set of K paths through a trellis with application to multitarget tracking," *Trans. on Aerospace and Elect. Sys.* **25**, pp. 287–296, Mar 1989.
10. N. Seshadri and C. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. on Communications* **42**, pp. 313–323, Feb/Mar/Apr 1994.
11. R. Danchick and G. Newnam, "A fast method for finding the exact N-best hypotheses for multitarget tracking," *Trans. on Aerospace and Elect. Sys.* **29**, pp. 555–560, Apr 1993.
12. R. Karp, "An algorithm to solve the m x n assignment problem in expected time O(mn log n)," *Networks* **10**, pp. 143–152, 1980.