TRELLIS STRUCTURE APPROACH TO MULTITARGET TRACKING

Richard Perry Anand Vaddiraju Kevin Buckley

Department of Electrical and Computer Engineering
Villanova University
Villanova, PA 19085
E-mail: `perry(anand, buckley)@ece.vill.edu`

ABSTRACT

In this paper we address sequential multitarget tracking for Radar applications. Specifically, we consider the problem of associating location measurement data over time to form estimates of multiple tracks. The algorithmic approach we employ is sequential and very flexible in that it can handle missed detections, false alarms, track initiation, and number-of-track estimation, while employing maximum likelihood or Bayesian cost functions. The approach is based on a trellis diagram which depicts the possible progressions of sequences of location measurements over time. Computation requirements are managed using a generalized K-path, L-list Viterbi algorithm to prune candidate track sets which have prohibitively high costs. Here we focus on a Bayesian solution to the joint problem of estimating both the number-of-tracks and the tracks themselves.

1. INTRODUCTION

Multitarget tracking is an important and challenging problem in radar signal processing [1] which has received intense attention for well over two decades now. Starting from a time evolving set of noisy measurements from detected targets and false alarms, the problem is to associate the detections over time to form multiple target tracks. Some early work focussed on associating time evolving target measurement data, that includes both real detections and false alarms, to one or more target tracks using a likelihood function formulation. Smith and Buechler [2] selected the single track that maximized a likelihood cost – an approach which Morefield [3] extended to maximum likelihood (ML) estimation of multiple tracks. Realizing the need to incorporate prior detection and false alarm probabilities, Singer et al. [4] and Reid [5] developed, respectively, single track and multitrack Bayesian estimators. Both ML and Bayesian multitrack estimators require evaluation of the cost of every possible (candidate) set of tracks through the measurements. The number of candidate track sets increases exponentially with time since at a given time each candidate track can extend to any of the next measurements. To reduce this computation burden, proposed methods typically only evaluate "feasible tracks" in "gating volumes", i.e. only measurements which are in some sense close to a predicted measurement are considered when extending a candidate track to the next measurement time. A result of this approach is that the track which is optimum over the entire processing interval may not be found, and in particular in sequential track estimation valid tracks can be dropped after track-to-measurement association fails for several measurement times.

This research was supported by ONR under Grant N00014-98-1-0892.

In [6], an alternative approach to computation reduction is proposed. Therein, a K-path extension of the Viterbi algorithm [7] is used to find the best K nonintersecting tracks according to a deterministic energy cost function. A trellis diagram is defined where at each stage (measurement time) states represents all combinations of measurements taken K at a time. Paths through the trellis represent K track sets. The K-path Viterbi algorithm is used to prune the number of paths through the trellis that must be considered, thereby reducing computation. Globally optimum solutions are computed, thus avoiding the problems associated with feasible tracks and gating volumes. Use of this K-path extension of the Viterbi algorithm is predicated on a "finite state machine structure" or "Markov condition" in the cost function, which is satisfied with the particular deterministic energy cost function used in [6]. However, an ML or Bayesian cost function will not satisfy this requirement since track measurements are correlated throughout the history of a track. (One way to think of this is that, in implementing an ML or Bayesian multitrack estimator, Kalman filters are used in the cost computation to generate measurement innovations. These Kalman filters are recursive and thus have infinite memory.) As a result direct use of K-path Viterbi for multitarget tracking is limited.

For the data communication application, another modification of the Viterbi algorithm, termed list Viterbi [8], has been proposed. This algorithm provides, at each stage through the trellis, a list of best paths through the trellis diagram. In [9], a novel K-track, L list Viterbi algorithm approach was proposed for multitarget tracking. By pruning the trellis to a list of best K-tracks instead of to a single best K-track, this approach overcomes the shortcomings of K-path Viterbi tracking [6] while controlling computation without resorting to the use of gating volumes. It also provides a ranked list of K-track estimates which may, according to the optimality criterion, have very similar costs but represent dissimilar trajectories. Such a list can be valuable for post processing classification and/or data fusion.

The algorithm described in [9]: is based on a Maximum Likelihood (ML) cost function; assumes the number of tracks, K, is known; and requires that the probability of target detection is unity. In this paper we employ a more effective Bayesian cost, describe an algorithm for joint estimation of K and the tracks, and account for missed detections.

Below we first introduce the K target tracking problem. We then develop a Bayesian cost function for estimating K and the tracks, and then marginalize over the tracks to derive an alternative estimator of K alone. The trellis diagram representation of the problem is then constructed, and followed by description of the K-path L-list Viterbi algorithm for Bayesian multiple track estimation. Finally, we illustrate this algorithm with several numerical examples.

2. PROBLEM FORMULATION

In this section we introduce the tracking problem, and we develop notation required to describe the Bayesian estimation approach and the trellis structure upon which the proposed generalized Viterbi algorithm is based.

In the Radar multitarget surveillance problem, the objective is to track multiple target trajectories over time. At each measurement time, measurements are provided from multiple target detections. The measurements for each detection consist of a set of D estimated location and/or velocity parameters. We assume that K, the number of targets, is unknown and constant over the processing time. Detections can correspond to either targets or false alarms. Missed detections will be accounted for. The probability of detection P_d is assumed known, as is the density function for the number of false alarms in the surveillance volume per measurement time. The problem we address here is to estimate K, as $\hat{K}$, while associating the measurements into $\hat{K}$ tracks. The algorithm proposed here, is sequential in that at each measurement time m an estimate $\hat{K}$ and $\hat{K}$ tracks (up to time m) are computed utilizing results from computations of estimates at previous times.

We denote as $\mathbf{z}_{m,j}$ the j^{th} the 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. Candidate Tracks

A single candidate or postulated 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 $\boldsymbol{\theta}_n^l$. The vector of measurements for this track is

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

where the subscript $j_m(l)$ is the measurement index at time m for the l^{th} track. 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. Note that a $\mathbf{z}_{m,J_m+1}$ 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 candidate tracks: $\boldsymbol{\theta}_n^l; l = 1, 2, \cdots, L_n^{'}$.

For a given candidate 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 [1]

$$\mathbf{x}_{m+1}^l = \boldsymbol{\Phi}_m \mathbf{x}_m^l + \mathbf{w}_m^l \tag{2}$$

$$\mathbf{z}_{m,j_m(l)} = \mathbf{H}_m \mathbf{x}_m^l + \mathbf{u}_m^l \tag{3}$$

where at time m and for candidate 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.

[1]To simplify the discussion we assume linear state/measurement equations.

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 $\mathcal{Z}(\boldsymbol{\theta}_n^l)$. The Kalman filter equations are:

Kalman Prediction Equations

$$\begin{aligned}
\hat{\mathbf{x}}_{m+1,m} &= \boldsymbol{\Phi}_m \hat{\mathbf{x}}_{m,m} \\
\hat{\mathbf{z}}_{m+1,m} &= \mathbf{H}_{m+1} \hat{\mathbf{x}}_{m+1,m} \\
\mathbf{v}_{m+1} &= \mathbf{z}_{m+1} - \hat{\mathbf{z}}_{m+1,m} \\
\hat{\mathbf{x}}_{m+1,m+1} &= \hat{\mathbf{x}}_{m+1,m} + \mathbf{G}_{m+1}\mathbf{v}_{m+1}
\end{aligned} \tag{4}$$

Kalman Gain Equations

$$\begin{aligned}
\boldsymbol{\Xi}_{m+1,m} &= \boldsymbol{\Phi}_{m+1} \boldsymbol{\Xi}_{m,m} \boldsymbol{\Phi}_{m+1}^{\mathbf{T}} + \mathbf{Q}_{m+1} \\
\mathbf{S}_{m+1} &= \mathbf{H}_{m+1} \boldsymbol{\Xi}_{m+1,m} \mathbf{H}_{m+1}^{\mathbf{T}} + \mathbf{R}_{m+1} \\
\mathbf{G}_{m+1} &= \boldsymbol{\Xi}_{m+1,m} \mathbf{H}_{m+1}^{\mathbf{T}} \mathbf{S}_{m+1}^{-1} \\
\boldsymbol{\Xi}_{m+1,m+1} &= \boldsymbol{\Xi}_{m+1,m} - \mathbf{G}_{m+1} \mathbf{H}_{m+1} \boldsymbol{\Xi}_{m+1,m}
\end{aligned} \tag{5}$$

where $\mathbf{G}$ is the Kalman gain matrix, $\boldsymbol{\Xi}$ is the measurement error's covariance matrix, and dependence on the track index l is not shown for notational convenience.

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 in the Kalman filter computation, 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 innovations, 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).

2.2. Candidate 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 candidate track sets is [2]

$$I_n^{'} = \left(\sum_{j=0}^{K} \begin{pmatrix} J_1 \\ j \end{pmatrix} \right) \left(\prod_{m=2}^{n} \left(K! \sum_{j=0}^{K} \frac{\begin{pmatrix} J_m \\ j \end{pmatrix}}{(K-j)!} \right) \right) \tag{6}$$

Let the i^{th} candidate K-track set be represented as

$$\boldsymbol{\tau}_n^i = \{\boldsymbol{\theta}_n^{l_1(i)}, \boldsymbol{\theta}_n^{l_2(i)}, \cdots, \boldsymbol{\theta}_n^{l_K(i)}\} \quad , \tag{7}$$

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

3. BAYESIAN TRACK ESTIMATION

First we will assume that K is known and consider the problem of estimating K tracks at time n from the measurement data $\mathcal{Z}^n$. We will then extend this to the unknown K problem.

[2]$\begin{pmatrix} A \\ B \end{pmatrix}$ represents the combinations of A things, taken B at a time.

3.1. Estimation of K Tracks

Using a Bayesian formulation which incorporates prior track set probabilities, at time n we have the following optimization problem for the selection from $\boldsymbol{\tau}^i_n; i = 1, 2, \cdots I^{'}_n$

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

$p\left(\mathcal{Z}^n\right)$ is a normalizing factor which serves to make the possible values of $p\left(\boldsymbol{\tau}^i_n|\mathcal{Z}^n\right)$ sum to unity, and can be ignored since it is independent of $\boldsymbol{\tau}^i_n$. $p\left(\boldsymbol{\tau}^i_n\right)$ is the apriori probability of the set of tracks, and is derived from the known probability of missed detections and the false alarm rate. Letting $J_m = T^i_m + F^i_m$, where T^i_m is the number of measurement vectors at time m used in the K track set $\boldsymbol{\tau}^i_n$ and F^i_m is the number of assumed false alarms, we have that

$$p\left(\boldsymbol{\tau}^i_n\right) = \prod_{m=1}^{n} P_d^{T^i_m} \left(1 - P_d\right)^{K - T^i_m} p(F^i_m) \quad , \tag{9}$$

where $p(F^i_m)$ is the Poisson distributed probability of F^i_m false alarms.

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

$$p\left(\mathcal{Z}^n|\boldsymbol{\tau}^i_n\right) = p\left(\mathcal{Z}(\boldsymbol{\tau}^i_n)|\boldsymbol{\tau}^i_n\right) \cdot p\left(\overline{\mathcal{Z}}(\boldsymbol{\tau}^i_n)|\boldsymbol{\tau}^i_n\right) \quad , \tag{10}$$

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

$$p\left(\overline{\mathcal{Z}}(\boldsymbol{\tau}^i_n)|\boldsymbol{\tau}^i_n\right) = \left(\frac{1}{Y}\right)^{\sum_{m=1}^{n} F^i_m} \quad . \tag{11}$$

In terms of the Kalman filter generated measurement innovations, for the track measurements,

$$p\left(\mathcal{Z}(\boldsymbol{\tau}^i_n)|\boldsymbol{\tau}^i_n\right) = \prod_{p=1}^{K} p(\mathcal{Z}(\boldsymbol{\theta}^{l_p(i)}_n)) = \prod_{p=1}^{K}\prod_{m=1}^{n} p(\mathbf{v}_{m,j_m(l_p(i))}) \tag{12}$$

$$= d^i_n \exp\left\{-\frac{1}{2}\sum_{p=1}^{K}\sum_{m=1}^{n} \mathbf{v}^T_{m,j_m(l_p(i))}\mathbf{S}^{-1}_{m,j_m(l_p(i))}\mathbf{v}_{m,j_m(l_p(i))}\right\} \tag{13}$$

where for each track p the product over m does not include any missed detections since as noted earlier measurements and innovations for missed detections don't exist. Keeping this in mind,

$$d^i_n = \prod_{p=1}^{K}\prod_{m=1}^{n} \frac{1}{(2\pi)^{D/2} \cdot \sqrt{\det(\mathbf{S}_{m,j_m(l_p(i))})}} \quad . \tag{14}$$

The Bayesian cost for estimating the best K tracks is described by (8), along with (9), (11), (13) and (14).

To derive an equivalent cost from which a time recursive trellis structure representation can be derived, take the negative natural log of (8), ignoring $p\left(\mathcal{Z}^n\right)$. For the i^{th} track set, the following equivalent Bayesian optimization problem is obtained:

$$\min_{\boldsymbol{\tau}^i_n} \quad \Lambda^n\left(\boldsymbol{\tau}^i_n\right) = \sum_{m=1}^{n} \Lambda_m(\boldsymbol{\tau}^i_n) \tag{15}$$

where the time incremental cost is

$$\Lambda_m(\boldsymbol{\tau}^i_n) = \lambda_m(\boldsymbol{\tau}^i_n) + \sum_{p=1}^{K} \lambda_m(\boldsymbol{\theta}^{l_p(i)}_n) \tag{16}$$

with

$$\begin{aligned} \lambda_m(\boldsymbol{\tau}^i_n) &= F^i_m \ln(Y) - T^i_m \ln(P_d) \\ &- (K - T^i_m)\ln(1 - P_d) - \ln(p(F^i_m)) \quad , \end{aligned} \tag{17}$$

and the p^{th} track incremental cost at time m is

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

Although from (15), the time recursion

$$\Lambda^n(\boldsymbol{\tau}^i_n) = \Lambda^{n-1}(\boldsymbol{\tau}^i_n) + \Lambda_n(\boldsymbol{\tau}^i_n) \tag{19}$$

can be formed, to solve this Bayesian problem directly, at time n $I^{'}_n$ incremental costs must be computed. In the following sections we develop an algorithmic approach which reduces this computational requirement.

3.2. Joint Estimation of K and the Tracks

For the estimation of both the number of tracks K and the tracks themselves $\boldsymbol{\tau}^i_n$, (8) can be considered a function of unknown K, resulting in the following joint estimation problem

$$\max_{K,\boldsymbol{\tau}^i_n} \quad p\left(K, \boldsymbol{\tau}^i_n|\mathcal{Z}^n\right) = \frac{p\left(\mathcal{Z}^n|K, \boldsymbol{\tau}^i_n\right) \cdot p\left(\boldsymbol{\tau}^i_n|K\right) \cdot p\left(K\right)}{p\left(\mathcal{Z}^n\right)} \quad . \tag{20}$$

If we assume that K is uniformly distributed from $K = 0$ to some maximum value $K = K_{max}$, and that $p\left(\boldsymbol{\tau}^i_n|K\right) = p\left(\boldsymbol{\tau}^i_n\right)$, and that $p\left(\mathcal{Z}^n|K, \boldsymbol{\tau}^i_n\right) = p\left(\mathcal{Z}^n|\boldsymbol{\tau}^i_n\right)$. This results in the estimator

$$\hat{K}, \hat{\boldsymbol{\tau}}^i_n = \arg \min_{K,\boldsymbol{\tau}^i_n} \left\{\Lambda^n\left(\boldsymbol{\tau}^i_n\right) = \sum_{m=1}^{n} \Lambda_m(\boldsymbol{\tau}^i_n)\right\} \tag{21}$$

where all values are as defined for (15), and now the cost is searched over all $\boldsymbol{\tau}^i_n; i = 1, 2, \cdots I^{'}_n(K)$ for each $K = 0, 1, \cdots, K_{max}$. (Referring to (6), note that the number of candidate track sets for a given K is a function of K.)

3.3. Marginalization for Number of Track Estimation

If estimation of K is of primary interest, it can be estimated without explicit estimation of $\boldsymbol{\tau}^i_n$ by marginalizing $p\left(K, \boldsymbol{\tau}^i_n|\mathcal{Z}^n\right)$ over $\boldsymbol{\tau}^i_n$. Compared to (20), the resulting estimator,

$$\hat{K} = \arg \max_{K} \left\{p\left(K|\mathcal{Z}^n\right) = \sum_{i=1}^{I^{'}_n(K)} p\left(K, \boldsymbol{\tau}^i_n|\mathcal{Z}^n\right)\right\} \quad , \tag{22}$$

can have better performance characteristics, which can be better for small number-of-measurement cases. has different performance characteristics, which can be better for small number-of-measurement cases [10].

4. TRELLIS DIAGRAM FORMULATION

Consider the trellis diagram in Figure 4 which depicts the possible progressions of sequences of location measurements over time for $K = 1$. Each stage of the trellis corresponds to a measurement time. For $K = 1$, at the m^{th} stage there are $J_m + 1$ states, one for each measurement and one to account for a missed target. A branch is a connection from a state at stage $m-1$ to another state at stage m. At time n, the $L_n^{'}$ candidate tracks $\boldsymbol{\theta}_n^l; l = 1, 2, \cdots, L_n^{'}$ correspond to the $L_n^{'}$ possible paths through the trellis from stage 1 to n. For each track, we assign an incremental cost to each branch. For example, for the l^{th} candidate track $\boldsymbol{\theta}_n^l$, the branch cost from stage $m-1$ to m (i.e. from state $j_{m-1}(l)$ to $j_m(l)$) is $\lambda_m(\boldsymbol{\theta}_n^i)$. It is important to note that since for each candidate track a Kalman filter (which has memory back to the 1^{st} stage through the Kalman states) is being used to compute the incremental costs, the cost associated with a branch depends on what track is being considered. That is, each branch has multiple costs assigned to it, one for each track through it. The cost of the path $\boldsymbol{\theta}_n^l$ is the sum of its branch costs. The Bayesian track estimation problem is now one of finding the minimum-cost path through this trellis.

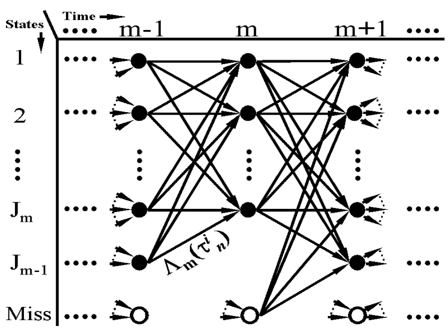

Figure 1: Trellis Diagram for $K = 1$.

For K track estimation, each state of the trellis in Figure 4 represents a set of K measurements and/or missed measurements. So, for stage m with J_m measurements, there are

$$M_m = \sum_{j=0}^{K} \begin{pmatrix} J_m \\ j \end{pmatrix} \tag{23}$$

states. Each branch from stage $m-1$ to m has a set of $K!/(K - T_m^i)!$ permutations associated with it, specifying the possible orders in which the K measurements and/or missed measurements are used. Now for each candidate K-track set, the incremental cost assigned to a branch is the sum of the costs of K candidate tracks. So, for candidate K-track set $\boldsymbol{\tau}_n^i$, the branch cost from stage $m-1$ to m is $\Lambda_m(\boldsymbol{\tau}_n^i)$. As with the $K = 1$ case, the Bayesian K-track estimation problem is now one of finding the minimum-cost path through this trellis.

5. A LIST VITERBI TRACKING ALGORITHM

Based on the trellis diagram problem formulation described above, we now describe a multitarget track estimation algorithm which provides a list of L best K-tracks as well as estimates of the number of tracks. The algorithm is sequential, self-initiating and can handle false alarms and missed detections. In this section a K-track algorithm is described, so that $K = 1$ is treated within this context.

The Bayesian tracking problem described in Section 3 would, if solved directly, be computationally prohibitive for large n and even moderate values of K and the $J_m^{\prime}s$. Here we propose to reduce or prune the number of candidate tracks by effectively, at each time m, eliminating from further consideration the candidate tracks that have too large (i.e. too unlikely) total costs at that time. In the trellis problem formulation, this will be implemented by only keeping the L best sets of tracks into each state.

The problem is to find the lowest cost path through the trellis. The Viterbi algorithm [7] can be used to sequentially prune the paths through the trellis for certain ML optimization problems (as well as for some other problems). At any stage in the trellis, it keeps only one path to each state. So it is only applicable to problems for which all other paths can not be optimum. It has been used extensively in digital communication and other application problems for which the sequence can be modeled as a Markov process. It has also been proposed for multitarget tracking based on a deterministic energy cost [6], where the incremental path costs from a state $j_{m-1}(l)$ to a state $j_m(l)$ is a function of only the $j_{m-1}(l)$ and $j_m(l)$ measurements.

In general, multitarget tracking problems, couched in a trellis framework, can not be solved using the Viterbi algorithm. For the Bayesian estimation problem considered here, the need for (infinite memory) Kalman filters for each candidate track precludes its use, so that to solve the problem at time n an exhaustive search of all $I_n^{'}$ 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 $I_{n-1}^{'}$ paths into stage $n-1$ must be extended to the $I_n^{'}$ paths into stage n, even though the costs of some of these paths will indicate that the corresponding K-track sets are highly unlikely. Alternatively, 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 [3].

Generically, the list Viterbi algorithm [8] does this. The multitarget tracking algorithm we propose incorporates the list Viterbi algorithm, along with the K-track trellis formulation of the multitarget tracking problem [6], and Bayesian cost computation based on Kalman filter innovations from Section 3. The algorithm determines a list of L feasible K-tracks sets as a list of L paths through the trellis. In general, paths are evaluated according to a defined cost function, where the incremental costs in progressing from stage to stage are computed as branch costs from states to states.

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 (17) for the apriori probabilities, since no Kalman fil-

[3]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.

ter 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 (16) from the L-best of each previous state to the total previous costs, and temporarily store these candidate new costs.

   3.2) Select the L-best of the candidate new costs and update the associated Kalman filters.

4) *Results:*

   4.1) Output the final L-best sets of tracks.

   4.2) Output estimates of $\hat{K}$, using just the lowest cost results for a joint estimation (21), or marginalization over the L-best tracks as an approximation to (22).

The algorithm uses a fixed value of K, and can extract k-track sets from the L-best K-track results for $k = 1, 2, \cdots K$.

Although the algorithm is described above in block form, it is time-recursive, and at any time during the iterations can produce current estimates of the L-best tracks and $\hat{K}$. As usual with the Viterbi algorithm, the trellis can be truncated periodically, to limit storage to a fixed size.

6. ILLUSTRATIVE NUMERICAL EXAMPLES

First, two parabolic target tracks consisting of $n = 21$ $\mathcal{X}$ and $\mathcal{Y}$ position values were generated. Gaussian measurement noise with a variance of 0.01 was added to the target tracks. "False detect" events were 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]. Probability of a missed detection was assumed to be 0.3, and for the purpose of illustrating that our algorithm can "regain" a target even after a series of missed detections, the lower track was forced to have missed detections from $\mathcal{X} = 1.5$ to 2.5. Other simulation parameters were K=2 and L=16. Figure 2 shows the $\mathcal{X}$ and $\mathcal{Y}$ position values for the true tracks, and the noisy measurements used as input to the algorithm.

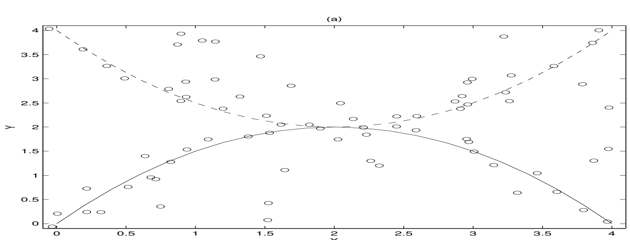

Figure 2: True Tracks and Measurements

Figure 3 shows the best paths, the total cost of which is 117.35. Notice that the one track starts out following the lower target, but switches to the upper target around the region where the true tracks intersect. Figure 4 shows the fourth best paths, which have a total cost of 118.41, just slightly higher than the cost of the best paths, and which correctly follow both the targets. Notice that during missed-detect events, the lower track follows a path tangential to the true track, but "regains" the target again when detections are recorded.

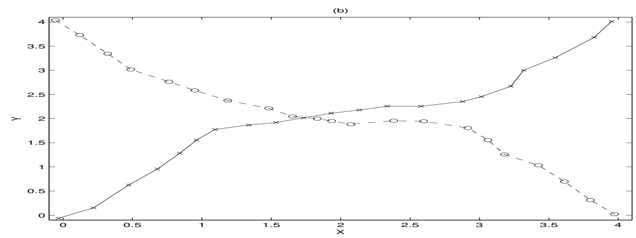

Figure 3: Best set of two tracks, cost=117.35

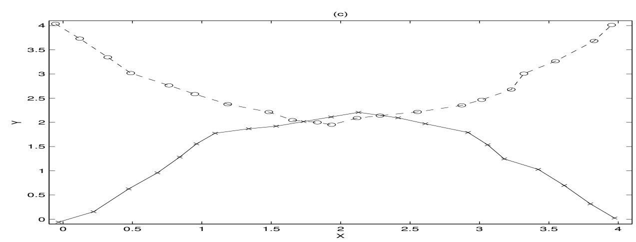

Figure 4: 4^{th} best set of two tracks, cost=118.41.

Second, Monte Carlo simulations were conducted to compare the Bayesian (21) and marginalization (22) estimates of K. N=25 trials were run per measurement noise value. F $K = 2$ linear tracks were simulated, with $n = 6$ measurement times. Measurement noise variance was 0.01. $P_d = 0.7$. 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]. $K_{max} = 4$ and $L = 16$. Figure 5 shows estimator performance for varying measurement noise variance. On the basis of these simulations, it can be concluded that estimation by marginalization can be advantageous over Bayesian estimation.

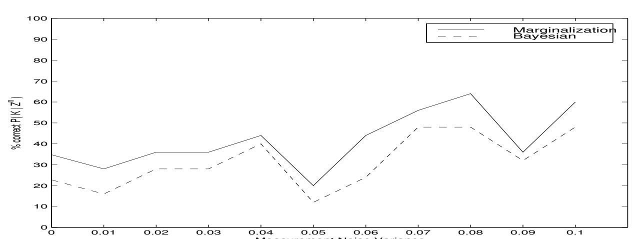

Figure 5: Percentage correct estimates of K.

Last, 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. Poisson distribution with a false alarm rate of 1, and the false detections themselves were generated using a uniform distribution over the range [0,4]. $K_{max} = 3$ and $L = 32$. 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: Percent correct for the marginalized K estimator:

7. CONCLUSION

In this paper we describe a sequential K-track L-list Viterbi algorithm for Bayeasian measurement /track association and estimation of the number of target tracks K. This algorithm can handle missed detections, false alarms and track initialization. Compared to direct computation of the solution of the Bayesian problem, this algorithm manages computational cost by sequentially eliminating from further consideration candidate K-track sets which have too great a Bayesian cost. The trellis diagram framework onto which the algorithm is built is generally applicable in that: it can accommodate other Bayesian an ML tracking problem solutions, as will as practical approximations to these; and it can be used for situations other than the one addressed here (e.g. for sequential track initiation and elimination).

We plan to further develop this generalized Viterbi tracking algorithm approach, by extending the algorithm to implement sequential track initiation and elimination, by considering further computation reduction via additional candidate K-track set pruning, and by incorporating practical Bayesian/ML motivated techniques such as joint probabilistic data association filtering (JPDAF) [11], probabilistic multi-hypothesis tracking (PMHT) [12] and interactive multiple model (IMM) [13].

8. REFERENCES

[1] Y. Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Trans. on Automatic Control*, Vol. 23, No. 4, pp. 618-626, August 1978.

[2] P. Smith and G. Buechler. A branching algorithm for discriminating and tracking multiple objects. *IEEE Trans. on Automatic Control*, Vol. 20, No. 1, pp. 101-104, February 1975.

[3] C. Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Trans. on Automatic Control*, Vol 22, No. 3, pp. 302-312, June 1977.

[4] 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*, Vol. 20, No. 4, pp. 423-432, July 1974.

[5] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, Vol. AC-24, No. 6, pp. 843-854, Dec. 1979.

[6] A. M. Viterbi J. K. Wolf and G. S. Dixon. Finding the best set of k paths through a trellis with application to multitarget tracking. *IEEE Trans. on Aerospace & Elect. Systems.*, Vol. 25, No. 2, pp. 287-296, March 1989.

[7] G. D. Forney. The viterbi algorithm. *Proc. of the IEEE*, Vol. 61, No. 3, pp. 268-278, March 1973.

[8] N. Seshadri and C. Sundberg. List viterbi decoding algorithms with applications. *IEEE Trans. on Communications*, Vol. 42, No. 2/3/4, pp.313-323, Feb/March/April 1994.

[9] R. Perry, A. Vaddiraju and K. Buckley, "Multitarget list Viterbi target tracking," *Proc. Asilomar Conf. on Circuits, Systems and Computers,* Nov. 1998.

[10] B. Radich and K. Buckley, "Bayesian marginal model selection for low rank sources," *Proc. IEEE Workshop on SSAP,* Sept. 1998.

[11] T. Fortmann, Y. Bar-Shalom and M. Scheffe, "Sonar tracking multiple targets using joint probabilistic data association," *IEEE Jour. of Oceanic Eng.*, Vol. 8, No. 3, pp. 173-183, July 1983.

[12] R. Streit and T. Luginbuhl, "Maximum likelihood method for probabilistic multi-hypothesis tracking," *Proc. SPIE*, Vol. 2335, pp. 394-405, April 1994.

[13] H. Blom and Y. Bar-Shalom, "The interactive multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. on Automatic Control*, Vol. AC-33, No. 8, pp. 780-783, Aug. 1988.