Proceedings of the SPIE
Conference on Signal and Data Processing of Small Targets
Vol. 6236, April 2006

Complexity Reduction in MHT/MFA Tracking: Part II, Hierarchical Implementation and Simulation Results

Benjamin J. Slocumb and Aubrey B. Poore

Numerica Corporation

ABSTRACT

The MHT/MFA approach to tracking has been shown to have significant advantages compared to single frame methods. This is especially the case for dense scenarios where there are many targets and/or significant clutter. However, the data association problem for such scenarios can become computationally prohibitive. To make the problem manageable, one needs effective complexity reduction methods to reduce the number of possible associations that the data association algorithm must consider. At the 2005 SPIE conference, Part I of this paper[1] was presented wherein a number of "gating algorithms" used for complexity reduction were derived. These included bin gates, coarse pair and triple gates, and multiframe gates. In this Part II paper, we provide new results that include additional gating methods, describe a hierarchical framework for the integration of gates, and show simulation results that demonstrate a greater than 95% effectiveness at removing clutter from the tracking problem.

Keywords: MHT/MFA Tracking, Complexity Reduction, Clutter Rejection, Gating Algorithms, Hierarchical Framework.

1. INTRODUCTION

In surveillance radar applications, it is essential that the sensor provide an accurate track picture of the surveillance area. Estimates of position and heading are important, but more critical is the maintenance of continuous tracks through clutter, target maneuvers, and other challenging scenarios. The track picture serves as input to higher-level decision functions that need to assess target ID, allocate resources, and schedule intercept tasks. For these decisions to be reliable, it is critical that a minimum amount of track breaks and swaps occur. Thus, pre-filter techniques that can improve tracking performance relative to breaks and swaps, and can reduce the computational loading on the tracking system, are critical to the overall mission. The objective of this paper is to develop a complexity reduction pre-filter system that can be deployed with MHT/MFA tracking systems to significantly reduce the tracker processing load as well as improve tracking performance.

The concept of a complexity reduction pre-filter is as follows. Measurements that are generated on one radar scan must associate with measurements on following scans to properly associate to a target track. If a measurement cannot feasibly associate to other measurements on subsequent scans, then there is no point in processing this data because it will be discarded later and will slow the processor down. Thus, the goal of the complexity reduction pre-filter is to use efficient (computationally cheap) methods to identify measurements that could feasibly associate together to form a target track before passing the data to the tracker where more expensive association techniques are employed.

The importance of effective pre-filtering can be seen from a simple example that illustrates the potential computational complexity of forming these sequences. If one has M measurements on each of two scans, then the number of comparisons that one must perform is M^2. Over three frames of data, the number is M^3. A small problem is $M = 100$ which implies $M^2 = 10,000$ over two frames and $M^3 = 1,000,000$ comparisons over three. The task of the pre-filter is to reduce the number of measurements sent to the tracker as much as possible with the golden rule that "true associations should never be ruled out!"

The underlying data management process within the pre-filter builds "strings" or "arcs" of measurements*, one from each scan, that are dynamically feasible parts of a target track and that can be used by the MHT/MFA tracker to feed an established track or initiate a new track. By maintaining a list of feasible arcs, we maintain an organized database that identifies which measurements are potentially target-oriented.

*We will use the nomenclature of "arc" to correspond to a dynamically feasibly linked set of measurements, one from each successive frame of data being considered. The arc is a graphically motivated description of connectivity.

To make the arc concept clear, consider an example where the current scan or "frame" of data is identified by index $k = 1$, the prior frame has index $k = 2$, etc. Suppose the pre-filter operates with $N = 3$ scans of data (N is a system parameter), thus the *window* holds three scans at any one time. An arc is then denoted by the triple $[i_1, i_2, i_3]$ of measurement indices where i_1 corresponds to the measurement index in frame-1, i_2 corresponds to the measurement index in frame-2, and i_3 corresponds to the measurement index in frame-3. If $i_k = 0$, then *no measurement* is used for the kth frame. This allows for a possible missed detection over the window. Notice that we build all possible arcs with feasible associations and we include the missed detection possibility. The complexity reduction pre-filter database is therefore a list of arcs of the type $[i_1, i_2, \ldots, i_N]$ that denote the feasibly connected measurements.

The complexity reduction pre-filter described in this paper combines efficient gating algorithms with an efficient data management process. In Section 2, we describe a collection of gating methods; in particular, we present new gates based on range and range-rate data. In Section 3, we describe the underlying data management process and the integrated hierarchical gate application system. Section 4 presents simulation results that demonstrate the effectiveness of the pre-filter in an airborne surveillance application. Section 5 gives a summary of the paper.

2. GATING ALGORITHMS

Gating is a term that refers to a collection of algorithms that rule out highly infeasible measurement-to-measurement, measurement-to-track, or track-to-track pairings and/or sequences. Such algorithms are an important step in the complexity reduction for all tracking algorithms, especially the MHT/MFA techniques. Starting with N "frames of data," the general procedure is to generate a sequence of measurements (including missed detections) across N frames. To build up these strings, one starts with a pair of measurements from different frames. From there, one builds triples from two pairs, and so forth to n-tuples from $(n-1)$-tuples. Thus, the first step is to explain how pairs are formed in an efficient manner; the second is to extend these to the aforementioned strings.

The application of measurement gates is best accomplished in a hierarchy of steps that include: (i) bin or cell gating; (ii) dynamic feasibility gate tests for either two or three measurements; (iii) multiple frame gating wherein one puts n-tuples together from two $(n-1)$-tuples; (iv) filter prediction gates for track extension; (v) outlier detection for batch estimation, especially for track initiation; and (vi) fine gating based on likelihood ratios. We address only the first three gates in this paper.

In this section, we show the derivation of the important gates used in a pre-filter system for MHT/MFA tracking. In Section 2.1, we summarize the gates that were developed in the prior paper.[1] Then in Section 2.2, we describe some new gates that are based on range-rate measurements. The problem is generally posed within the context of a "frames of data" because the MHT/MFA data association problem is posed over N data sets or frames of data. In this development, we assume that biases are removed and any residual biases are addressed by correctly inflating and shaping the measurement covariances; otherwise, the derivation would need to be modified to account for potential biases directly.

2.1. Summary of gates previously presented

In this section, we summarize the gates that were derived in Part I of this paper.[1]

2.1.1. Dynamic pair gating for measurements

Given two measurements p_1 and p_2, the dynamic pair gate declares the pairing of the two measurements to be dynamically feasible if a target can move from one measurement to the other at a maximum velocity and to within the noise in the two measurements over the time interval $\Delta t = t_2 - t_1$ between the two measurements. The derivation of this gate is given in Part I of the paper,[1] and the gate result is

$$\begin{aligned} |p_{2x} - p_{1x}| &\le V_{\max}|t_2 - t_1| + \chi_{\alpha,n}\sqrt{C_{1xx}} + \chi_{\alpha,n}\sqrt{C_{2xx}} \\ |p_{2y} - p_{1y}| &\le V_{\max}|t_2 - t_1| + \chi_{\alpha,n}\sqrt{C_{1yy}} + \chi_{\alpha,n}\sqrt{C_{2yy}} \quad \text{(Cuboid Gate)} \\ |p_{2z} - p_{1z}| &\le V_{\max}|t_2 - t_1| + \chi_{\alpha,n}\sqrt{C_{1zz}} + \chi_{\alpha,n}\sqrt{C_{2zz}} \end{aligned}$$

where $V_{\max}$ is the maximum target velocity, $\mathrm{cov}(p_1) = C_1$ and $\mathrm{cov}(p_2) = C_2$, and $\chi^2_{\alpha,n}$ is a Chi-square parameter with certainty parameter α and n degrees of freedom. An alternate dynamic pair gate is the Spheroid test,

$$\|p_2 - p_1\| \le V_{\max}\,|t_2 - t_1| + \chi_{\alpha,n}\sqrt{\mathrm{tr}C_1} + \chi_{\alpha,n}\sqrt{\mathrm{tr}C_2} \quad \text{(Spheriod Gate)}$$

2.1.2. A predictive pair gate

Now consider two frames of data. One could test each measurement p_1 in the first frame with each measurement p_2 in the second frame. If there are M measurements in the first frame and N measurements in the second frame, then there are $M \times N$ tests that must be performed. One can considerably reduce this number of comparisons through using bin gating prior to the comparisons to reduce the number of potential parings. The first step is to develop a prediction gate from the measurement p_1 in the first frame to those in the second.

Let $C_{\max}$ denote the three by three matrix obtained as the maximum over all covariances in the second frame, component by component, and let p denote an arbitrary measurement in the second frame. Then,

$$
\begin{aligned}
|p_x - p_{1x}| &\leq V_{\max}|t - t_1| + \chi_{\alpha,n}\sqrt{C_{1xx}} + \chi_{\alpha,n}\sqrt{C_{\max\, xx}} \\
|p_y - p_{1y}| &\leq V_{\max}|t - t_1| + \chi_{\alpha,n}\sqrt{C_{1yy}} + \chi_{\alpha,n}\sqrt{C_{\max\, yy}} \\
|p_z - p_{1z}| &\leq V_{\max}|t - t_1| + \chi_{\alpha,n}\sqrt{C_{1zz}} + \chi_{\alpha,n}\sqrt{C_{\max\, zz}}
\end{aligned}
$$

defines a cuboid centered at p_1 at time t. Next define $\Delta T_{\max}(t_1) = \text{Maximum}\,\{|t - t_1|\,|\}$ where the maximum runs over time t of each measurement in the second frame. Then

$$
\begin{aligned}
|p_x - p_{1x}| &\leq V_{\max}\Delta T_{\max}(t_1) + \chi_{\alpha,n}\sqrt{C_{1xx}} + \chi_{\alpha,n}\sqrt{C_{\max\, xx}} \\
|p_y - p_{1y}| &\leq V_{\max}\Delta T_{\max}(t_1) + \chi_{\alpha,n}\sqrt{C_{1yy}} + \chi_{\alpha,n}\sqrt{C_{\max\, yy}} \\
|p_z - p_{1z}| &\leq V_{\max}\Delta T_{\max}(t_1) + \chi_{\alpha,n}\sqrt{C_{1zz}} + \chi_{\alpha,n}\sqrt{C_{\max\, zz}}
\end{aligned}
$$

defines the cuboid centered at p_1 that contains a measurement that is dynamically consistent with p_1.

2.1.3. Combined bin and dynamic pair gating

If there are M measurements on frame-1, and N measurements on frame-2, then when applying the dynamic pair gate above one would perform $M \times N$ tests/comparisons. If N is "large," then in order to reduce this number to something like $M \times k$ where $k \ll N$, one generally performs bin or cell gating (see Figure 3 in the Part I paper). The general procedure for bin gating is as follows. One first forms bins in the second frame. For each measurement in the first frame, say p_1, one uses the prediction gate to determine which bins in the second frame are contained in or intersected by the prediction gate. To each of the measurements in the identified bins, one then applies the pair tests (i.e., Spheriod or Cuboid Gate Test) to determine which pairs can be ruled out. At the conclusion of this procedure, one has pairs of measurements that might emanate from the same target and has ruled out those that cannot.

The choice of the size of the bin can be determined adaptively or *a priori*. Bins should not be too small nor too large and can be based on the size of the prediction gate. Once the prediction gate is defined relative to the measurement p_1, one needs an efficient method for returning all measurements from those bins that the prediction gate contains or intersects. We have found that fixed bin sizes and hash tables work extremely well. Other choices might be octrees or quadtrees.

Finally, we stress that if M and N are "small," then one should bypass bin gating because the number $M \times N$ is small. Also, if M or N is large and the other is small, then one would generally bin the frame with the largest number of measurements. Thus, we generally assume $M \leq N$; otherwise, the frames should be interchanged.

2.1.4. Multi-frame gating

As discussed in the introduction, strings (arcs) of measurement sequences are used in the pre-filter process. These strings can be directly used by the MHT/MFA tracker as potential extensions of a track or to initiate new tracks. This is accomplished as follows. Let i_k denote the index to a measurement from frame k. A triple $[i_k, i_m, i_n]$ is built from two pairs $[i_k, i_m]$ and $[i_m, i_n]$ that contain an overlap of one measurement i_m. A quadruple $[i_k, i_m, i_n, i_p]$ is built from two triples $[i_k, i_m, i_n]$ and $[i_m, i_n, i_p]$ with the requirement of an overlap of two measurements $[i_m, i_n]$. An N-tuple is built from two $(N-1)$-tuples with the requirement of an overlap of $(N-2)$ measurements. If one has a dynamic K-tuple test, one can apply the test after the construction of the K-tuple to eliminate the K-tuple and its further use in the construction of strings.

2.2. New gates based on range-rate data

We now present three gates that utilize range-rate data. The gates in Sections 2.2.2 and 2.2.3 are extensions to the prior work.[1]

2.2.1. A range and range-rate dynamic pair gate: fixed platform case

In this section, we develop a range and range-rate dynamic pair gate for the case where the sensor is stationary in an inertial reference frame. The case of a moving platform is more complex and is discussed in Section 2.2.2, but the general approach and goals are the same. Let $r(t)$ be the range to the target at time t, and $\dot{r}(t)$ be the range-rate. The fundamental theorem of calculus and the rectangle and trapezoid integration rules yield

$$\begin{aligned} r(t_2) - r(t_1) &= \int_{t_1}^{t_2} \frac{dr(s)}{ds} ds \\ &= \frac{dr(c)}{dt}\Delta t_1 \quad \text{(mean value theorem)} \\ &= \frac{dr(t_1)}{dt}\Delta t_1 + \frac{1}{2}\frac{d^2r(c)}{dt^2}(\Delta t_1)^2 \quad \text{(rectangle rule)} \\ &= \frac{1}{2}\left(\frac{dr(t_1)}{dt} + \frac{dr(t_2)}{dt}\right)\Delta t_1 - \frac{1}{12}\frac{d^3r(c)}{dt^3}(\Delta t_1)^3 \quad \text{(trapezoid rule)} \end{aligned}$$

for some c between t_1 and t_2, and where $\Delta t_1 = t_2 - t_1$. Now, one can use each of these to derive gates. First, consider the *trapezoid rule* case. Using $r(t_i) = r_i + \nu_i$ and $\dot{r}(t_i) = \dot{r}_i + \mu_i$ where $\nu_i \sim \mathcal{N}(0, \sigma_{r_i})$ and $\mu_i \sim \mathcal{N}(0, \sigma_{\dot{r}_i})$, one can derive

$$\begin{aligned} r_2 - r_1 &= \frac{1}{2}(\dot{r}_1 + \dot{r}_2)\Delta t_1 - \nu_1 + \nu_2 + \frac{1}{2}(\mu_1 + \mu_2)\Delta t_1 - \frac{1}{12}\frac{d^3r(c)}{dt^3}(\Delta t_1)^3 \\ |r_2 - r_1 - \frac{1}{2}(\dot{r}_1 + \dot{r}_2)\Delta t_1| &\le |\nu_1| + |\nu_2| + \frac{1}{2}(|\mu_1| + |\mu_2|)|\Delta t_1| + \frac{1}{12}|\frac{d^3r(c)}{dt^3}(\Delta t_1)^3| \\ &\le 3\sigma_{r_1} + 3\sigma_{r_2} + \frac{3}{2}(\sigma_{\dot{r}_1} + \sigma_{\dot{r}_2})|\Delta t_1| + \frac{1}{12}J_{\max}|\Delta t_1|^3 \end{aligned}$$

where we have used a three σ-value as an upper bound on the errors and where $J_{\max}$ is an upper bound on $\max\{\, \frac{d^3r(t)}{dt^3} \mid t_1 \le t \le t_2 \,\}$.

In a similar fashion, one can derive a gate based on the *rectangle rule*. The result is

$$\begin{aligned} |r_2 - r_1 - \frac{1}{2}(\dot{r}_1)\Delta t_1| &\le |\nu_1| + |\nu_2| + |\mu_1||\Delta t_1| + \frac{1}{2}|\frac{d^2r(c)}{dt^2}(\Delta t_1)^2| \quad (1) \\ &\le 3\sigma_{r_1} + 3\sigma_{r_2} + 3\sigma_{\dot{r}_1}|\Delta t_1| + \frac{1}{2}A_{\max}|\Delta t_1|^2 \quad (2) \end{aligned}$$

where we have used a three σ-value as an upper bound on the errors and where $A_{\max}$ is an upper bound on $\max\{\, \frac{d^2r(t)}{dt^2} \mid t_1 \le t \le t_2 \,\}$.

Likewise, one can derive a gate based on the *mean value theorem*. The result is

$$\begin{aligned} |r_2 - r_1| &\le |\nu_1| + |\nu_2| + |\frac{dr(c)}{dt}(\Delta t_1)| \\ &\le 3\sigma_{r_1} + 3\sigma_{r_2} + V_{\max}|\Delta t_1| \end{aligned}$$

where we have used a three σ-value as an upper bound on the errors and where $V_{\max}$ is an upper bound on $\max\{\, |\frac{dr(t)}{dt}| \mid t_1 \le t \le t_2 \,\}$ over the interval $[t_1, t_2]$.

2.2.2. A range and range-rate dynamic pair Gate: moving platform case

We now extend the results in the previous section to the case of a moving platform. The difficulty with a moving platform is that the estimates on the derivatives of $r(t)$ are more complicated. The gate in (1) is the one that we develop here for the range and range-rate gate test for a moving platform. Let $R_t(t)$ and $R_p(t)$ denote position vectors to the target and platform, respectively, and define $R(t) = R_{targ}(t) - R_p(t)$ to be the position vector from the platform to the target. Then $r(t) = \sqrt{R(t) \cdot R(t)}$ is the distance to the target and $\dot{r}(t)$ denotes the range rate. Both $r(t)$ and $\dot{r}(t)$, as well as the azimuth,

are measured by the sensor. In deriving gates in this case, the only difference is in the estimates for $\dot{r}(t)$ and $\ddot{r}(t)$. Observe now the following:

$$\dot{r}(t) = \frac{\dot{R}(t) \cdot R(t)}{r(t)} = \frac{\dot{R}_t(t) \cdot R(t)}{r(t)} - \frac{\dot{R}_p(t) \cdot R(t)}{r(t)}$$
$$\ddot{r}(t) = \frac{\ddot{R}(t) \cdot R(t)}{r(t)} + \frac{\dot{R}(t) \cdot \dot{R}(t)}{r(t)} - \frac{\dot{r}^2(t)}{r^2(t)}$$

The big difference between the stationary and moving platform is in the bounding of $\dot{r}(t)$ and $\ddot{r}(t)$. Here are the bounds.

$$\begin{aligned}
|\dot{r}(t)| &= |\frac{\dot{R}(t) \cdot R(t)}{r(t)}| = |\frac{\dot{R}_t(t) \cdot R(t)}{r(t)} - \frac{\dot{R}_p(t) \cdot R(t)}{r(t)}| \\
&\leq \|\dot{R}_t(t)\| + \|\dot{R}_p(t)\| \\
|\ddot{r}(t)| &= \frac{|\ddot{R}(t) \cdot R(t)|}{r(t)} + \frac{|\dot{R}(t) \cdot \dot{R}(t)|}{r(t)} + \frac{\dot{r}^2(t)}{r^2(t)} \\
|\ddot{r}(t)| &\leq \|\ddot{R}(t)\| + \frac{\|\dot{R}(t)\|^2}{r(t)} + \frac{\dot{r}^2(t)}{r^2(t)} \\
&\leq \|\ddot{R}_t(t) - \ddot{R}_p(t)\| + \frac{\|\dot{R}_t(t) - \dot{R}_p(t)\|^2}{r(t)} + \frac{\dot{r}^2(t)}{r^2(t)} \\
&\leq \|\ddot{R}_t(t)\| + \|\ddot{R}_p(t)\| + \frac{(\|\dot{R}_t(t)\| + \|\dot{R}_p(t)\|)^2}{r(t)} + \frac{\dot{r}^2(t)}{r^2(t)} \\
&\leq (\|\ddot{R}_t(t)\| + \|\ddot{R}_p(t)\|) + \left(\|\dot{R}_t(t)\| + \|\dot{R}_p(t)\|\right)^2 \left(\frac{1}{r(t)} + \frac{1}{r^2(t)}\right)
\end{aligned}$$

Starting from the previously derived estimate

$$|r_2 - r_1 - \frac{1}{2}(\dot{r}_1)\Delta t_1| \leq |\nu_1| + |\nu_2| + |\mu_1||\Delta t_1| + \frac{1}{2}|\frac{d^2 r(c)}{dt^2}(\Delta t_1)^2|$$

for some c between t_1 and t_2. Thus, we have

$$\begin{aligned}
|r_2 - r_1 - \frac{1}{2}(\dot{r}_1)\Delta t_1| &\leq |\nu_1| + |\nu_2| + |\mu_1||\Delta t_1| + \frac{1}{2}|\frac{d^2 r(c)}{dt^2}(\Delta t_1)^2| \\
&\leq |\nu_1| + |\nu_2| + |\mu_1||\Delta t_1| + \\
&\frac{1}{2}(\Delta t_1)^2 \left((\|\ddot{R}_t(c)\| + \|\ddot{R}_p(c)\|) + \left(\|\dot{R}_t(c)\| + \|\dot{R}_p(c)\|\right)^2 \left(\frac{1}{r(c)} + \frac{1}{r^2(c)}\right)\right)
\end{aligned}$$

for some c between t_1 and t_2. Now, we bound, approximate, and evaluate terms as follows.

$$\begin{aligned}
V_{tmax} &= \text{Maximum } \{\|\dot{R}_t(c)\| \mid c \text{ between } t_1 \text{ and } t_2\} \\
A_{tmax} &= \text{Maximum } \{\|\ddot{R}_t(c)\| \mid c \text{ between } t_1 \text{ and } t_2\} \\
V_{pmax} &= \text{Maximum } \{\|\dot{R}_p(c)\| \mid c \text{ between } t_1 \text{ and } t_2\} \\
A_{pmax} &= \text{Maximum } \{\|\ddot{R}_t(c)\| \mid c \text{ between } t_1 \text{ and } t_2\}
\end{aligned}$$

Now V_{tmax} and A_{tmax} are parameters that are used to bound the motion of the targets. V_{pmax} can be evaluated by knowing the velocity of the platform over the time interval connecting t_1 and t_2. We can approximate V_{pmax} and A_{pmax}

$$\begin{aligned}
V_{pmax} &= \text{Maximum } \{\|\dot{R}_p(t_1)\| + \text{Error in } \|\dot{R}_p(t_1)\|, \|\dot{R}_p(t_2)\| + \text{Error in } \|\dot{R}_p(t_2)\|\} \\
A_{pmax} &= \frac{\|\dot{R}_p(t_2) - \dot{R}_p(t_1)\|}{|\Delta t_1|} + \frac{\text{Error in } \|\dot{R}_p(t_2)\| + \text{Error in } \|\dot{R}_p(t_1)\|}{|\Delta t_1|} \\
r_{min} &= \text{Maximum } \{r_1 - |\nu_1|, r_2 - |\nu_2|\}
\end{aligned}$$

wherein the errors in the data are incorporated into the bounds. Using the bounds $|\nu_i| \leq 3\sigma_{r_i}$ and $|\mu_i| \leq 3\sigma_{\dot{r}_i}$, one has

$$|r_2 - r_1 - \frac{1}{2}\dot{r}_1\Delta t_1| \leq 3\sigma_{r_1} + 3\sigma_{r_2} + 3\sigma_{\dot{r}_1}|\Delta t_1| + \frac{1}{2}(\Delta t_1)^2\left(A_{tmax} + A_{pmax} + \left(V_{tmax} + V_{pmax}\right)^2\left(\frac{1}{r_{min}} + \frac{1}{r_{min}^2}\right)\right) \quad (3)$$

Note that an alternate gate test is

$$|r_1 - r_2 + \frac{1}{2}\dot{r}_2\Delta t_1| \leq 3\sigma_{r_1} + 3\sigma_{r_2} + 3\sigma_{\dot{r}_2}|\Delta t_1| + \frac{1}{2}(\Delta t_1)^2\left(A_{tmax} + A_{pmax} + \left(V_{tmax} + V_{pmax}\right)^2\left(\frac{1}{r_{min}} + \frac{1}{r_{min}^2}\right)\right) \quad (4)$$

Given two measurements $(r_i, \dot{r}_i)$, one would apply both tests.

2.2.3. A range and range-rate multiframe gate

We now develop a range and range-rate multiframe measurement gate that evaluates a correlation hypothesis of a sequence of measurements for their dynamic feasibility. The test is implemented as a linear Kalman filter with a state of range and range-rate. The measurement state used in the filter is also the range and range-rate. The inclusion of process noise ensures that the gate is robust against target maneuvers.

We show the derivation of the gate for the case of three frames, but the approach easily generalizes to an N-frame gate. Without loss of generality, let the three-frame measurements be denoted as $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$, where the measurement times are $\{t_1, t_2, t_3\}$. We assume that the three measurements are time-ordered, i.e., $t_1 \leq t_2 \leq t_3$. Each measurement is the vector $\mathbf{z}_i = [r_i, \dot{r}_i]^T$ with covariance

$$\mathbf{R}_i = \begin{bmatrix} \sigma_{r_i}^2 & 0 \\ 0 & \sigma_{\dot{r}_i}^2 \end{bmatrix}$$

The bearing measurement b_i is not used in this three-point gate.

The multiframe gate is implemented as a linear Kalman filter with the initial filter state and covariance being the first measurement,

$$\mathbf{x}_1 = \mathbf{z}_1 \text{ and } \mathbf{P}_1 = \mathbf{R}_1 \quad (5)$$

Let $\Delta t_{ij} = t_j - t_i$, and the state transition matrix and process noise covariance be denoted as

$$\mathbf{F}(\Delta t) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{Q}(q, \Delta t) = q\begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix} \quad (6)$$

Next, the filter state and covariance are predicted to t_2, the time of the second measurement,

$$\mathbf{x}_{2|1} = \mathbf{F}(\Delta t_{12})\mathbf{x}_1, \quad (7)$$

$$\mathbf{P}_{2|1} = \mathbf{F}(\Delta t_{12})\mathbf{P}_1\mathbf{F}(\Delta t_{12})^T + \mathbf{Q}(q, \Delta t_{12}), \quad (8)$$

where we use a process noise q that consists of a fixed user-defined part q_0 that models target acceleration, and an adaptive part $A_{p_{\max}}$ that incorporates the own-ship motion:

$$q = q_0 + A_{p_{\max}} \quad (9)$$

The process noise for the own-ship acceleration is estimated as

$$A_{p_{\max}} = \min\left\{A_{p_0},\ \hat{A}_p + \varepsilon_{A_p}\right\}, \quad (10)$$

$$\hat{A}_p = \frac{1}{\Delta t_{12}}\|\mathbf{v}_p(t_2) - \mathbf{v}_p(t_1)\| \quad (11)$$

where $\mathbf{v}_p(t_i)$ is the own-ship Cartesian velocity vector. The acceleration estimate $\hat{A}_p$ in (11) is only necessary if the aircraft navigation system is not able to provide the own-ship acceleration; otherwise the estimate is determined as $A_p = \max\{\|\mathbf{a}_p(t_1)\|, \|\mathbf{a}_p(t_2)\|\}$, where $\mathbf{a}_p(t_i)$ is the own-ship Cartesian acceleration vector. The term ε_{A_p} is the 3σ acceleration

error in the reported own-ship acceleration, and A_{p_0} is an upper bound on the platform acceleration that prevents the estimate from becoming unreasonable in case of faulty own-ship acceleration reports.

Now, applying the regular Kalman filter equations, we compute the innovation matrix,

$$\mathbf{S}_2 = \mathbf{R}_2 + \mathbf{H}\mathbf{P}_{2|1}\mathbf{H}^T$$

with measurement residual $\nu_2 = \mathbf{z}_2 - \mathbf{x}_{2|1}$, and trivial observation matrix $\mathbf{H} = \mathbf{I}$. The first gate test is now

$$\nu_2^T \mathbf{S}_2^{-1} \nu_2 < \chi^2_{\alpha,n} \tag{12}$$

If the inequality (12) does not hold, the measurements $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ will be rejected as a feasible triple, otherwise, the computation continues with the next Kalman filter step:

$$\mathbf{x}_{2|2} = \mathbf{x}_{2|1} + \mathbf{K}_2\nu_2, \;\; \mathbf{P}_{2|2} = \mathbf{P}_{2|1} - \mathbf{K}_2\mathbf{S}\mathbf{K}_2^T$$

with the Kalman gain

$$\mathbf{K}_2 = \mathbf{P}_{2|1}\mathbf{H}^T\mathbf{S}_2^{-1}$$

Next we repeat the Kalman filter prediction and compute $\mathbf{x}_{3|2}$, the covariance $\mathbf{S}_3$, and the normalized innovation $\nu_3^T\mathbf{S}_3^{-1}\nu_3$. We repeat the Chi-square test as in Eqn. (12), and accept or reject the triple as feasible based on the outcome.

3. PRE-FILTER DATA MANAGEMENT

At the center of the complexity reduction pre-filter system is an efficient data management mechanism that organizes the measurements so that tests (the gating algorithms described in Section 2) can be applied. For each scan, or frame, of radar measurements the data are held in a data structure that enables "arcs" between possibly feasible associations to be established. Strings of measurements connected by arcs are then held in the data structure; the length of the string is determined by the window size N set in the pre-filter system. As these strings are established by initial gating tests, subsequent gating tests are applied that may remove some arcs. In this way, the gates are applied in an *hierarchical* manner using the computationally cheapest gates first and the more expensive gates last. In this section, we explain the methodology for establishing arcs, applying the moving window data management system, and applying the hierarchical gating process.

3.1. Arc addition process: arc extension

To describe the arc formation process, let the measurements obtained in the most recent scan be identified by the set $\mathcal{Z}_1$ and the ith measurement in this scan be identified as $\mathbf{z}_1^i$. Thus, $\mathcal{Z}_k = \{\mathbf{z}_k^{i_m}\}_{m=1}^{M_k}$, for M_k measurements in the kth most recent scan. The measurement $\mathbf{z}_k^i$ is used to represent a vector of parameters such as range, bearing, elevation, range-rate, and possibly others. Let the window be of size N, thus the frames of data in the window are $\{\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_N\}$.

If measurement $\mathbf{z}_1^{i_1}$ on frame-1 associates with $\mathbf{z}_2^{i_2}$ on frame-2, then we say that an arc $[i_1,\ i_2]$ exists between the two measurements. The feasible association is established through gating tests. Extending the concept, if both measurements gate with measurement $\mathbf{z}_3^{i_3}$ in frame-3, then the arc $[i_1,\ i_2,\ i_3]$ is established. Continuing in this way over all N frames, an arc defined by $[i_1,\ i_2,\ i_3, \ldots,\ i_N]$ is established when all pairwise gating tests indicate association feasibility. We call this an N-scan arc, or string, and this data is what is stored within the pre-filter data structure. One can envision a $K \times N$ matrix of indices where K is the total number of feasible arcs stored within the pre-filter processor.

When a new frame of data is obtained (say frame-n for *new*), the bin gate algorithm is applied for each measurement in this frame, $\mathbf{z}_n^{i_m},\ m \in \{1, \ldots, M_n\}$. Each measurement is gated with the measurements in the prior frames $\{\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_{N-1}\}$. Note that as some measurements are eliminated, some arcs are also removed from consideration, which removes some measurements from consideration in the prior frames. For measurement $\mathbf{z}_n^{i_m}$ in frame-n, if we find, for example, that the following pairs are feasible via the pair gate tests:

$$[\mathbf{z}_n^{i_m}, \mathbf{z}_1^{i_1}], \quad [\mathbf{z}_n^{i_m}, \mathbf{z}_2^{i_2}], \quad \ldots \quad [\mathbf{z}_n^{i_m}, \mathbf{z}_{N-1}^{i_{N-1}}] \tag{13}$$

then we would form the extended arc $[i_m, i_1,\ i_2,\ i_3, \ldots,\ i_{N-1}]$ if the the arc $[i_1, i_2,\ i_3, \ldots,\ i_{N-1}]$ existed in the data structure. While the initial pair tests in (13) are accomplished with the bin gate, each pair is subsequently tested with a

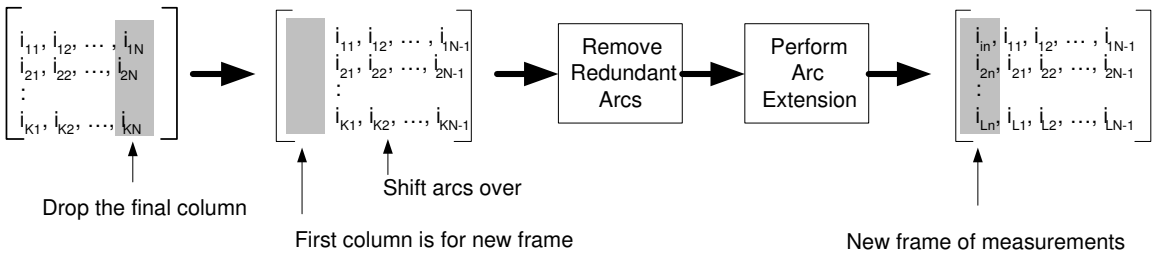

Figure 1. Description of the moving window process within the complexity reduction pre-filter.

sequence of dynamical pair gates before the feasible association is allowed. After the arc is established, then additional multiframe gates may also be applied. This is described in more detail in Section 3.3.

To summarize, feasible measurement strings, or data arcs, are established in the pre-filter using the following steps:

- *Pair Gating*. Each measurement in the new data frame is gated with the prior $N-1$ frames of data for N-scan processing. First, a prediction gate using the bin gate process is applied to identify feasible associations. For each feasible association identified by the bin gate, additional dynamic pair gates are applied. The pre-filter allows any number of dynamic pair gates to be applied in a successive order (a hierarchy).
- *Arc Extension*. Using the $N-1$ sets of feasible associations (i.e, lists of measurements in the prior frames that pass the pair gates), we find arcs in the data structure that have measurement indices in these sets. For each arc that is identified, we extend the arc with the index of the candidate measurement from the current frame.
- *Missed Detection Extension*. To allow for possible missed detections within the window, we also create extension arcs with the "miss" in the appropriate arc column. For all arcs that existed in the window, we create an extension arc with "0" in the current frame column. These arcs have the form $[0,\ i_1,\ i_2,\ \ldots,\ i_{N-1}]$. We also create new arcs of the form $[i_m,\ 0,\ 0, \ldots,\ 0]$, for $i_m = 1, \ldots, M_n$, to support the possible initiation of new targets.

3.2. Window shift process: the moving window data structure

Prior to the arc addition process described in the previous section, the complexity reduction pre-filter will first conduct a *window shift* process to allow the new frame of data to be added to the window. The process is outlined in Figure 1. We start with a data array of size $K \times N$, for K arcs in the list and a window size of N. First, the indices in the earliest frame of data in the window are dropped from the arc. In effect, we slide the window and leave behind the associations from the earliest frame of data in the window. The arc indices that remain are shifted to the right to make room for the indices in the new frame of data. In performing the drop-and-shift process, some number of arcs will become redundant because the dimension of the arcs has been reduced. To see this, consider the following example. Suppose the following two arcs are initially present in the data structure:

$$[i_1,\ i_2\ , \ldots, i_{N-1},\ i_N] \tag{14}$$

$$[i_1,\ i_2\ , \ldots, i_{N-1},\ i'_N], \quad i_N \neq i'_N \tag{15}$$

The two arcs differ only in the final indices, i_N and i'_N. When the final column of indices is dropped during the shift process, the two arcs become identical. Thus, one of the two redundant arcs must be removed. Through the use of efficient software data structures, the redundant arcs can be automatically dropped with a minimal number of operations.

After the redundant arcs are removed, the next step is to conduct arc extension (see Section 3.1). When this is complete, a new arc array of dimension $L \times N$ will exist as shown in Figure 1. In general, L will be different from K since some redundant arcs were dropped and then new arcs were added with the new frame.

3.3. Gating hierarchy

We explain now how a gating hierarchy can be used to make the complexity reduction pre-filter system efficient. As mentioned previously, the concept of using a hierarchy is to apply the computationally cheapest gates first, and then apply the more expensive gates last. Using this sequence, one can achieve the highest efficiency in the process. There are two parts to the hierarchy: the bin/pair gate component, and the multiframe component.

Bin/Pair Gate Component. We first apply the bin gate between measurements in the current frame-n and all the prior frames $\{1, 2, \ldots, N-1\}$. The bin gate uses a predictive gate to form an initial set of feasible pairs. Suppose the candidate measurement index on frame-n is identified as i_n^m. Suppose on frame-1 the bin gate identifies the set $\{i_1^{j_1}, \ldots, i_1^{j_P}\}$ as feasible associations. The pair gate hierarchy operates by stepping through the combinations $[i_n^m, i_1^{j_p}]$, where $p = 1, \ldots, P$, and applying the sequence of specified dynamical pair gates. Whenever a gate rules out a feasible association, the measurement $i_1^{j_p}$ is no longer considered and all remaining gating tests are bypassed. In the end, some subset of $\{i_1^{j_1}, \ldots, i_1^{j_P}\}$ will remain and these measurements are passed along to the arc extension step.

Multiframe Gate Component. After pair gating has been applied, and the arcs have been extended with the feasible associations, then the multiframe gates can be applied. To apply the multiframe gates, we step through the arc list in the data structure. For each arc in the list, we apply a sequence of multiframe gates (cheapest first). Whenever one gate rules out an arc, the arc is removed and the application of gates to that arc is stopped. If K is the size of the multiframe gate, and the window size is N, then if $K < N$ there are several options for applying the multiframe gate. One is to apply the gate only to the first K non-zero indices in the arc (i.e., to the first K measurements in the window, skipping over missed detections). A second and more expensive option is to apply the K-tuple gate to all possible combinations in the arc that involve the candidate (first frame) measurement. The second approach obviously provides a more complete test, but is also more computationally expensive.

3.4. Pre-filter output to the tracker

The output of the pre-filter is the arc list stored in the data management system. This list identifies feasible strings of measurements that could correspond to actual targets. The MHT/MFA will directly use the arc list within its track management system. For each string where measurements in the prior $N-1$ frames were used in a track hypothesis, then the MHT/MFA tracker may use the current measurement to update that track if the filter and score gates successfully pass. Also, depending on the tracker's extension/initiation logic, if the number of non-zero (i.e., not a miss) measurements in the arc is M out of N, then the tracker can attempt to initiate a track on these measurements.

Finally, it should be pointed out that the N-scan pre-filter process can be applied to a traditional single-frame tracking system, in addition to the MHT/MFA tracker. The tracker extension/initiation process just needs to be integrated with the arc list data. If this is not possible, then the alternative is to apply a measurement pruning process to the arc list based on an M of N criteria. If an arc is defined by $[i_1,\ i_2\ , \ldots,\ i_N]$, and at least M of the $\{i_m\}$ are non-zero, then the complexity reduction pre-filter would pass the measurement with index i_1 on to the tracker. Otherwise, the measurement i_1 is considered clutter and is discarded. This pruning process cuts out all the clutter measurements that do not feasibly form arcs with a "substantial" number of measurements in it. One drawback of this pruning approach is that the initiation of a track will be delayed until at least M measurements from the target have been obtained. The delay is avoided if the tracker can receive the entire arc list in one message (i.e., handle prior frame measurements) and applies the initialization process to those measurements.

4. SIMULATION RESULTS

To evaluate the performance of the complexity reduction pre-filter, a prototype was developed that implements the moving window structure, the gating hierarchy, and the specific gating algorithms (bin, predictive pair, dynamic pair, and dynamic triple). The prototype has been integrated with an airborne (moving platform) surveillance radar simulation model, and gating performance metrics have been established.

4.1. Simulation scenario description

The sensor model used in the simulation study is one typical of an airborne surveillance radar. The beamwidth of the radar is modeled to be 6°, and the range resolution is modeled as 30 m. The scan period is modeled as 10 sec. The radar possesses a Doppler processing capability and can generate target range rate measurements. A tracker processes the range, bearing, and range-rate measurements. It is a conventional single hypothesis (global nearest neighbor) tracker that employs a two-model IMM filter with low-noise and high-noise nearly-constant-velocity models. The tracker employs conventional filter gate processing to identify feasible measurements for association.

In the scenario, there is one maneuvering target in clutter. The target pulls two 2 g maneuvers and one 4 g maneuver within a 1200 sec scenario. For this scenario, we varied the clutter density (discussed below) to assess the effectiveness of the pre-filter. The trajectory duration lasts 1200 sec in the simulation; with a radar scan period of 10 sec, there are on average 120 target measurements over the trajectory. We ran 10 Monte Carlo runs.

A random clutter model was established for the sensor model that allowed the complexity reduction pre-filter to be stressed. We set up the clutter generation so that all returns were restricted to a defined box around the target. The size of the box, and the number of clutter returns per box, were set by parameters; thus, the clutter density around the targets could be controlled. The clutter measurements were uniformly placed in the box, and then the coordinates were transformed to range and bearing to form the clutter measurements. For the clutter range-rate measurement, we accounted for the fact that the clutter returns would be spread over the Doppler spectrum when the airborne platform was moving. Three principle components exist in the airborne radar clutter spectrum[2]: the mainbeam clutter, the altitude clutter, and the sidelobe clutter. Typically the mainbeam clutter, which has a known Doppler value, is removed with a "Doppler notch" in the radar signal processor. The altitude clutter spectrum is broader than the mainbeam clutter spectrum but is always centered on zero Doppler. Meanwhile the sidelobe clutter, which results from returns obtained through the antenna sidelobes in all directions, has a clutter spectrum that is uniform over the Doppler range $[-2v/\lambda, 2v/\lambda]$, where v is the platform velocity and λ is the radar wavelength. Thus, any measurements obtained from sidelobe clutter have a range rate value that is uniformly distributed over $[-v, v]$.

To study the impact of clutter on the complexity reduction pre-filter system, we will assume the clutter is the result of sidelobe clutter. Thus, each clutter measurement will have a range rate value that is obtained from a random number that is uniformly distributed on $[-v, v]$. This is a reasonable assumption since the mainbeam clutter will have been removed by the radar Doppler processor. The addition of altitude clutter could be considered as well.

4.2. Results

To evaluate the pre-filter in a standalone mode, we set the window size to $N = 3$. We required $M = 3$ measurements in an arc, i.e., 3-of-3, before the measurement in the first (most current) frame was allowed to be passed to the tracker. In scenarios with missed detections, we would need a 3-of-4 or 4-of-5 rule, but we did not simulate missed detections in this study. By checking the source of those measurements (target or clutter) that were passed out of the pre-filter, we could score the clutter suppression capability. We performed this analysis for varying levels of clutter density.

The gating hierarchy employed in the study was as follows: (i) a bin gate with an xy prediction gate; (ii) an xy dynamic pair gate; (iii) a range dynamic pair gate; (iv) a range and range-rate dynamic pair gate; and (v) a range and range-rate three-point gate. These gates were formulated in Section 2.

Table 1 below shows the statistics for the case where the clutter density was 5 returns in the 20×20 km box near the target on each scan. This should be viewed as a "severe" clutter density case, which may or may not be realistic, but was created specifically to stress the gates. The table shows how each of the gates progressively remove more clutter. The pair gates each knock out about 35% of the arcs (all containing associated clutter returns), while the triple gate knocks out 58% of the clutter arcs. When taken in whole, the net effect is that more than 95% of the clutter has been removed from the input. Other statistics of interest in the table are as follows. The "gates falsely removed" corresponds to cases where truth arcs are removed; as shown, no cases of the removal of truth occurred. The "missed gate opportunities" corresponds to cases where arcs contained both truth and clutter and should be removed; as shown, only 48 of these arcs survive. Within these arcs, 26 unique clutter measurements exist, and these are allowed to pass through (thus the 95.63% of 595 clutter measurements). The "target returns removed" corresponds to the first two returns in the scenario; this happens because of the 3-of-3 window specified for this study, and is an artifact that results from a simulator interface limitation (prior frame measurements could not be sent to the tracker).

Table 1. Complexity reduction pre-filter gating statistics for the clutter density case of 5 per 20×20 km.

Gate	Statistic
xy_dynamicPairGate	
calls to gate	7667
gates passed	4922 (64.20%)
gates falsely failed	0 (0.00%)
missed gate opportunities	49 (1.00%)
Percentage arcs ruled out	35.80% (2745 of 7667)
range_dynamicPairGate	
calls to gate 7 6782	
gates passed	4491 (66.22%)
gates falsely failed	0 (0.00%)
missed gate opportunities	49 (1.09%)
Percentage arcs ruled out	37.69% (2291 of 6078)
range_rangeRate_dynamicPairGate	
calls to gate	4181
gates passed	2809 (67.18%)
gates falsely failed	0 (0.00%)
missed gate opportunities	49 (1.74%)
Percentage arcs ruled out	36.23% (1372 of 3787)
range_rangeRate_threePointGate	
calls to gate	394
gates passed	165 (41.88%)
gates falsely failed	0 (0.00%)
missed gate opportunities	48 (29.09%)
Percentage arcs ruled out	58.12% (229 of 394)
Clutter removed	95.63% of 595 clutter reports
Targets removed	2 target returns

Table 2. Gating statistics for various clutter densities

Clutter Density Case	Clutter removed
5 clutter returns per 20×20 km	95.63% of 595 clutter reports
5 clutter returns per 30×30 km	98.66% of 595 clutter reports
5 clutter returns per 50×50 km	99.66% of 595 clutter reports

Table 2 shows the clutter removed statistics for three different clutter densities. In the most severe case (5 returns in 20×20 km) we obtain 95% removal effectiveness. As we relax the density to 5 returns in 50×50 km, we achieve better than 99% clutter removed. This broader distribution of clutter is more realistic, thus one can expect the complexity reduction pre-filter to contribute to very significant improvements in tracker processing and runtime performance.

In addition to evaluating the gating statistics, we assessed the impact of the pre-filter on the tracker performance. Table 3 summarizes the tracking metrics of interest over the run. As shown, the presence of severe clutter causes the tracker to degrade substantially. However, with the complexity reduction pre-filter removing most of the clutter data, the tracker performance is optimal. This emphasizes the point that, in addition to reducing the tracker's computational load, the pre-filter improves the tracker's performance.

5. SUMMARY

In this Part II paper, we have extended the complexity reduction work presented in our prior paper.[1] First, we developed several new range and range-rate gates. We described a data management system that provides an efficient process for maintaining all feasible arcs within a moving window. We also described a hierarchical gate application process that enables the computationally cheep gates to be applied first, and the more expensive gates to be applied last. Simulation results for a

Table 3. Tracker metrics summary for the clutter density case of 5 per 20×20 km and 10 MC runs.

Track Metric	Without Pre-Filter	With Pre-Filter
Number of tracks produced (on MC run #1)	7	1
Track completeness	60-80%	100
Cumulative switches	2.5	0
Cumulative breaks	4.8	0
Redundant track ratio	1.2 - 1.4	1.0
Spurious track ratio	0.4	0.0

stressing clutter condition showed that the complexity reduction pre-filter system was able to remove a substantial amount of the clutter in a simulated airborne surveillance radar scenario. The impact on tracking was evaluated where it was shown without the pre-filter, the tracking performance was severely diminished, but with the pre-filter employed the performance was optimal.

ACKNOWLEDGMENTS

This work was supported in part by the US Navy under contract N68335-05-C-0223.

REFERENCES

1. A. B. Poore, "Complexity reduction in MHT/MFA tracking," in *Proceedings of the SPIE, Conference on Signal and Data Processing of Small Targets*, O. E. Drummond, ed., **5913**, pp. 59131F–1/12, August 2005.
2. G. W. Stimson, *Introduction to Airborne Radar*, SciTech, Mendham, NJ, 2nd ed., 1998.