Target Motion Analysis via Hard and Soft Data Fusion

Yuthika Punchihewa, Ba-Tuong Vo, Ba-Ngu Vo, Amanda Bessell, Sanjeev Arulampalam
Jessica Irons, and Sam Davey

***Abstract*—Target Motion Analysis (TMA) requires the online fusion of multiple hard and soft data sources for target tracking. This paper proposes a Bayesian filtering solution for multi-source fusion with hard and soft data. Appropriate models for various types of hard and soft data are developed so that they can be fused in a consistent manner under the Bayesian framework. The resulting Bayes filter is highly non-linear and non-Gaussian. Hence, a parallel particle filter is developed to facilitate a user adjustable trade-off between computation time and tracking accuracy. Numerical studies on realistic scenarios are also presented.**

***Index Terms*—Target Tracking, Hard Soft Data, Particle Filter, Multi-sensor Fusion.**

I. Introduction

Using multiple data sources in state estimation reduces uncertainty and hence, improves estimation accuracy [6], [13], [14]. For applications such as Target Motion Analysis (TMA) a single data source typically results in limited observability, and a comprehensive solution fundamentally requires multiple data sources [4]. TMA refers to tracking algorithms that use data derived from passive sensors [4], which can be categorized as: *hard data* comprising observations from various arrays of receivers; or *soft data* requiring some form of human interpretation prior to being provided to the TMA operator. All received hard and soft observations are paired with a new/existing track before being fed to the TMA subsystem.

In passive underwater TMA, the adverse signal environment means that hard data alone is not sufficient and soft data needs to be integrated for better tracking. Further, building an accurate and timely tactical picture requires the capability to exploit every possible source of data. For underwater applications, the TMA process is typically human operator guided, and requires manual fusion of soft data. However, performing TMA for a high number of targets within a short time period can result in high cognitive load for human operators. TMA is largely a computational process, and many aspects are naturally amenable to automation that can significantly reduce human operator workload. The challenge lies in devising a principled target tracking solution that fuses hard and soft data in a timely manner.

This paper proposes a Bayesian filtering solution that fuses hard and soft data for passive underwater TMA. The dynamic Bayesian estimation framework allows for a recursive or online computation of the filtering or posterior density, effectively encapsulating all modeling and observation information on the target. Appropriate models for various types of hard and soft data are developed so that they can be fused in a statistically consistent manner. The resulting Bayes filter is highly non-linear and non-Gaussian to which no analytical solutions exist. As a trade-off between computation time and estimation accuracy, we developed a parallel particle filter to propagate the filtering density. Numerical studies and bench-marks on datasets involving experienced TMA operators are presented.

The authors are with Curtin University, Australia, and DST Group, Australia (email: {yuthika.gardiyaw, ba-tuong.vo, ba-ngu.vo}@curtin.edu.au, {amanda.bessell, sanjeev.arulampalam, jessica.irons, samuel.davey}@defence.gov.au).

II. Problem Formulation

In this section, we formulate the target tracking problem from hard and soft data. Typically, the hard data sources are SONAR bearing and bearing rate, while the soft data sources involve information regarding speed, bearing, range, angle-to-bow, and classification from SONAR and/or Optronics (OPTIX).

A. Bayes Filter

Let $x_k \in \mathbb{X}$ and $z_k \in \mathbb{Z}$ denote, respectively, the state and observation of the target at time k. It is assumed that the state space $\mathbb{X}$ and observation space $\mathbb{Z}$ are finite dimensional. Hereon, we use the notation $y_{m:n}$ to denote the sequence $y_m, y_{m+1}, ..., y_n$.

In Bayesian filtering we model the state of the target by a state space model. Specifically, the evolution of the target state is modeled as a discrete-time Markov process [1], [21]. This can be described by a Markov transition density $f_{k|k-1}(\cdot\,|\cdot\,)$, where $f_{k|k-1}(x_k|x_{k-1})$ is the probability density of the state x_k at time k given the state x_{k-1} at the previous time. In addition, the observation of the target is described by the likelihood function $g_k(\cdot\,|\cdot\,)$, where $g_k(z_k|x_k)$ is the probability density of the observation z_k given the target state x_k.

For target tracking we are interested in the *filtering density* $\pi_k(\cdot|z_{1:k})$, where $\pi_k(x_k|z_{1:k})$ is the probability density of state x_k at time k given the observation history $z_{1:k}$. Under the hidden Markov model assumptions, the filtering density can be computed using the Bayes (filtering) recursion [1], [21]:

$$\pi_{k|k-1}(x_k|z_{1:k-1}) = \int f_{k|k-1}(x_k|\xi)\pi_{k-1}(\xi|z_{1:k-1})d\xi, \quad (1)$$

$$\pi_k(x_k|z_{1:k}) = \frac{g_k(z_k|x_k)\pi_{k|k-1}(x_k|z_{1:k-1})}{\int g_k(z_k|\xi)\pi_{k|k-1}(\xi|z_{1:k-1})d\xi}. \quad (2)$$

The filtering density $\pi_k(\cdot|z_{1:k})$ contains all statistical information about the target state at time k given the observation

history $z_{1:k}$. Optimal estimates of the state can be obtained from the filtering density via the *mean* or *mode* [14].

B. Motion Model

Maritime vessels often follow a fixed course (usually at a constant speed), but occasionally undergo different maneuvers such as changes in direction or speed. On the other hand, warships are especially known to maneuver for evading enemies. To describe both types of motion, we employ a Jump Markov System (JMS) model [18]–[20], where the state vector contains a mode variable that corresponds to the motion model of the target. Specifically, the state vector $x_k = (\zeta_k, m_k)$ consists of the kinematics vector ζ_k and mode m_k, with state transition density of the form

$$f_{k|k-1}\left(x_k|x_{k-1}\right) = f_{k|k-1}^{(m_k)}\left(\zeta_k|\zeta_{k-1}\right)\vartheta\left(m_k|m_{k-1}\right), \quad (3)$$

where $f_{k|k-1}^{(m_k)}\left(\cdot|\cdot\right)$ is the kinematics transition density from time $k-1$ to time k, under model m_k, and $\vartheta\left(m_k|m_{k-1}\right)$ is the probability of switching from mode m_{k-1} at time $k-1$ to mode m_k at time k. In this work, the kinematics vector $\zeta_k = (\varsigma_k, \omega_k)$ consists of the target's Cartesian coordinates and velocities $\varsigma_k = (p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k})$, and turn rate ω_k.

To accommodate target maneuvers, we consider two different constant turn models [12] with different noise parameters, to capture the fixed course mode and turning mode, i.e.,

$$\begin{aligned} f_{k|k-1}^{(m_k)}\left(\varsigma_k, \omega_k|\varsigma_{k-1}, \omega_{k-1}\right) &= \mathcal{N}\left(\omega_k; \omega_{k-1}, \sigma_\omega^2\right) \\ &\times \mathcal{N}\left(\varsigma_k; F\left(\omega_{k-1}, T_{k-1}\right)\varsigma_{k-1}, Q(m_k, T_{k-1})\right), \end{aligned} \quad (4)$$

where σ_ω is the standard deviation of the turn rate noise, T_k is the (irregular) sampling interval for time k,

$$F\left(\omega, T\right) = \begin{bmatrix} 1 & \frac{sin(\omega T)}{\omega} & 0 & -\frac{(1-cos(\omega T))}{\omega} \\ 0 & cos\left(\omega T\right) & 0 & -sin\left(\omega T\right) \\ 0 & \frac{(1-cos(\omega T))}{\omega} & 1 & \frac{sin(\omega T)}{\omega} \\ 0 & sin\left(\omega T\right) & 0 & cos\left(\omega T\right) \end{bmatrix},$$

$$Q(m, T) = v^2(m) \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^2 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^2 \end{bmatrix},$$

$$v(m) = \begin{cases} 10^{-5}, & \text{if } m = 1 \\ 1, & \text{if } m = 2 \end{cases}.$$

Note that the model for fixed course motion has very small process noise on the position and velocity. On the other hand, the model with larger process noise is designed to capture significant changes in velocity. More complex JMS models can be developed at the expense of computational load, and a good trade-off is needed.

C. Observation Model

Let (o_x, o_y) denote the position of the ownship at time-step k. Table I summarizes the physical features of the state relevant to the observation models.

In the following, we present the observation likelihood functions for two typical hard data sources and five soft data sources in TMA systems. We adopt the standard assumption that, conditional on the target state, observations from different sources are independent of each other. Hence, the joint likelihood function at a given time is the product of the individual likelihoods of the observations received at that time. For simplicity, the time index k is omitted in this section.

Name	Definition
Position	$P\left(x\right) = \left(p_x, p_y\right)$
Velocity	$V\left(x\right) = \left(\dot{p}_x, \dot{p}_y\right)$
Speed	$S(x) = \left\|\left(\dot{p}_x, \dot{p}_y\right)\right\|$
Course	$\Gamma\left(x\right) = \arctan\left(\frac{\dot{p}_x}{\dot{p}_y}\right)$
Bearing	$B\left(x\right) = \arctan\left(\frac{p_x - o_x}{p_y - o_y}\right)$
Bearing Rate	$\dot{B}\left(x\right) = \frac{\left(p_y - o_y\right)\left(\dot{p}_x - \dot{o}_x\right) - \left(p_x - o_x\right)\left(\dot{p}_y - \dot{o}_y\right)}{\left\|\left(p_x, p_y\right) - \left(o_x, o_y\right)\right\|^2}$
Range	$R\left(x\right) = \left\|\left(p_x, p_y\right) - \left(o_x, o_y\right)\right\|$
Range Rate	$\dot{R}\left(x\right) = \frac{\left(p_x - o_x\right)\left(\dot{p}_x - \dot{o}_x\right) + \left(p_y - o_y\right)\left(\dot{p}_y - \dot{o}_y\right)}{\left\|\left(p_x, p_y\right) - \left(o_x, o_y\right)\right\|}$

TABLE I
RELEVANT FUNCTIONS OF STATE VARIABLES

1) Hard SONAR Bearing: Bearing observations (in radians relative to due North) from SONAR are received roughly at 20s intervals. Let θ denote the received bearing at time k. Then, its observation likelihood is given by

$$g^{(S,B)}(\theta|x) = \mathcal{N}\left(\theta; B\left(x\right), \left(\frac{1}{3} \times \frac{\pi}{180}\right)^2\right). \quad (5)$$

The bearing noise variance is a nominal value based on SONAR operator experience.

2) Hard SONAR Bearing Rate: The bearing rate $\dot{\theta}$ (in radians per second) from SONAR is received at the same time as the bearing, and its observation likelihood is given by

$$g^{(S,BR)}(\dot{\theta}|x) = \mathcal{N}\left(\dot{\theta}; \dot{B}\left(x\right), \left(\frac{1}{60} \times \frac{\pi}{180}\right)^2\right). \quad (6)$$

Again, the noise variance is a nominal value based on SONAR operator experience.

3) Soft SONAR Speed Information: Let s be the speed observation (in yards per second) from SONAR, and $\mathcal{C}$ be the vessel class ('merchant', 'warship', 'fishing', or 'undetermined') reported at time k. Then their joint observation likelihood is given by

$$g^{(S,S)}(s, \mathcal{C}|x) \propto \mathcal{N}\left(s; S(x), \left(sw^{(S,S)}(\mathcal{C})\right)^2\right), \quad (7)$$

where

$$w^{(S,S)}(\mathcal{C}) = \begin{cases} 0.01, & \text{if } \mathcal{C} = \text{'merchant' or 'warship'} \\ 0.03, & \text{otherwise} \end{cases}.$$

4) Soft OPTIX Range Information: Let r be the range report (in yards) from OPTIX, and $\mathcal{C}$ be the vessel class reported at time k. Then their joint observation likelihood is given by

$$g^{(O,R)}(r, \mathcal{C}|x) \propto \mathcal{N}\left(r; R(x), \left(rw^{(O,R)}(\mathcal{C})\right)^2\right), \quad (8)$$

where

$$w^{(O,R)}(\mathcal{C}) = \begin{cases} 0.03, & \text{if } \mathcal{C} = \text{`merchant' or `warship'} \\ 0.06, & \text{otherwise} \end{cases}.$$

5) Soft OPTIX Bearing Information: The likelihood function for bearing report β (in radians) from OPTIX is similar to that for SONAR bearing observation, i.e.

$$g^{(O,B)}(\beta|x) = \mathcal{N}\left(\beta; B(x), \left(\frac{\pi}{180}\right)^2\right). \tag{9}$$

The noise variance is a nominal value based on operator experience.

6) Soft Angle-to-Bow Information: The Angle-to-Bow (ATB) or target angle is given by $\xi - B(x)$, where ξ is the target heading or course, and $B(x)$ is the bearing. In practice the ATBs reported by humans vary widely between operators, depending on their experiences. If the observed target heading falls within 45° of the observed bearing (i.e., $\xi \in \left[B(x) - \frac{\pi}{4}, B(x) + \frac{\pi}{4}\right]$), then the target is said to be *opening* (moving away from the ownship) when it has a positive range rate $\dot{R}(x)$. If the observed target heading falls within 45° of the reverse bearing measurement (i.e., $\xi \in \left[B(x) + \frac{3\pi}{4}, B(x) - \frac{3\pi}{4}\right]$), then the target is said to be *closing* (moving closer to the ownship) when it has a negative range rate $\dot{R}(x)$. Observed target headings outside these course ranges are disregarded, because opening or closing events cannot be determined with reasonable certainty (due to large variations in human operator expertise). The likelihood of the OPTIX target heading observation based on the reported ATBs or closing/opening events is given as follows

$$g^{(O,TH)}(\xi|x) = \begin{cases} 1_{[0,\infty)}(\dot{R}(x)), & \text{if } \xi - B(x) \in \left[-\frac{\pi}{4}, +\frac{\pi}{4}\right] \\ 1_{(-\infty,0]}(\dot{R}(x)), & \text{if } \xi - B(x) \in \left[\frac{3\pi}{4}, -\frac{3\pi}{4}\right] \\ 1, & \text{otherwise} \end{cases}. \tag{10}$$

This likelihood function effectively restricts the state vector to regions of the state space with range rates and bearings that corroborate the observed target heading. Note that regions outside course ranges have no influence on the distribution of the target in the Bayes update (2).

7) Soft SONAR/OPTIX Classification Information: Assuming vessel speed is uniformly distributed in pre-defined ranges for different vessel classes, the likelihood of vessel class based on speed is given by

$$g^{(S,CS)}(\mathcal{C}|x) = \begin{cases} 1_{[3,9]}(S(x)), & \text{if } \mathcal{C} = \text{`merchant'} \\ 1_{[0,15]}(S(x)), & \text{if } \mathcal{C} = \text{`warship'} \\ 1_{[0,5]}(S(x)), & \text{if } \mathcal{C} = \text{`fishing'} \\ 1_{[0,20]}(S(x)), & \text{if } \mathcal{C} = \text{`unclassified'} \end{cases}. \tag{11}$$

This likelihood function effectively restricts the target state to regions of the state space with speeds that corroborate the reported class.

For merchant vessels, the nominal courses c_u and c_d, traveling, respectively, up and down an established shipping lane, are known a priori. A merchant vessel's course can only fall within a limited margin of error, e.g. $\frac{6\pi}{180}$, of the nominal courses. The likelihood function for vessel class based on course, given by

$$g^{(O,CC)}(\mathcal{C}|x) = \begin{cases} 1_{\left[c_u - \frac{6\pi}{180}, c_u + \frac{6\pi}{180}\right] \cup \left[c_d - \frac{6\pi}{180}, c_d + \frac{6\pi}{180}\right]}(\Gamma(x)), & \text{if } \mathcal{C} = \text{`merchant'} \\ 1, & \text{otherwise} \end{cases}, \tag{12}$$

restricts the state of a target reported as 'merchant' to regions of the state space that corroborate merchant vessel courses. Further, the target's position is restricted to the region S_L defined by the boundaries of the shipping lanes (in TMA, the boundaries are defined to be approximately 2000 yards from the closest true shipping lane). The likelihood function for vessel class based on position is given by

$$g^{(O,CP)}(\mathcal{C}|x) = \begin{cases} 1_{S_L}(P(x)), & \text{if } \mathcal{C} = \text{`merchant'} \\ 1, & \text{otherwise} \end{cases}. \tag{13}$$

Thus, for scenarios with shipping lanes, the overall likelihood function for SONAR/OPTIX classification data is given by

$$g^{(SO,C)}(\mathcal{C}|x) = g^{(S,CS)}(\mathcal{C}|x) g^{(O,CC)}(\mathcal{C}|x) g^{(O,CP)}(\mathcal{C}|x). \tag{14}$$

On the other hand, for scenarios without shipping lanes

$$g^{(SO,C)}(\mathcal{C}|x)) = g^{(S,CS)}(\mathcal{C}|x). \tag{15}$$

III. Particle Filtering

The motion and observation models presented above constitute a highly non-linear non-Gaussian filtering problem, for which no analytic solution exists. The particle or sequential Monte Carlo (SMC) method is a class of approximate numerical solutions to the Bayes filter that are applicable to nonlinear non-Gaussian models [2], [8], [9], [21].

A. Sequential Importance Sampling (SIS)

The basis of the particle method is the use of *importance sampling* to approximate the filtering density [2], [8], [9], [21]. More concisely, a weighted point mass approximation to a (probability) density π, by N i.i.d. samples $\{x^{(i)}\}_{i=1}^N$ from a density p, is given by

$$\pi(x) \approx \sum_{i=1}^{N} w^{(i)} \delta_{x^{(i)}}(x), \tag{16}$$

where $\delta_{x^{(i)}}(x)$ denotes a Dirac-delta centered at $x^{(i)}$, and

$$w^{(i)} = \frac{\pi\left(x^{(i)}\right)/p\left(x^{(i)}\right)}{\sum_{j=1}^{N} \pi\left(x^{(j)}\right)/p\left(x^{(j)}\right)}. \tag{17}$$

The density p is usually easy (and inexpensive) to sample from, and is called a *proposal* or *importance function*.

The key operation in particle filtering is the application of sequential importance sampling (SIS) to recursively approximate the filtering density. Let $\{(x_{k-1}^{(i)}, w_{k-1}^{(i)})\}_{i=1}^N$, with

$\sum_{i=1}^{N} w_{k-1}^{(i)} = 1$, denote the set of N weighted samples approximating the filtering density at time time $k-1$, i.e.

$$\pi_{k-1}\left(x_{k-1}|z_{1:k-1}\right) \approx \sum_{i=1}^{N} w_{k-1}^{(i)}\, \delta_{x_{k-1}^{(i)}}\left(x_{k-1}\right). \tag{18}$$

Suppose that $p(\cdot\,|x_{k-1}^{(i)} z_{1:k})$ is a proposal that we can easily sample from. Then, upon receiving a new measurement, the filtering density can be approximated by [2], [8], [9], [21]

$$\pi_{k}\left(x_{k}|z_{1:k}\right) \approx \sum_{i=1}^{N} w_{k}^{(i)}\, \delta_{x_{k}^{(i)}}\left(x_{k}\right), \tag{19}$$

where

$$x_k^{(i)} \quad \sim \quad p(\cdot\,|x_{k-1}^{(i)} z_{1:k}), \tag{20}$$

$$w_k^{(i)} \quad \propto \quad w_{k-1}^{(i)} \frac{g(z_k|x_k^{(i)}) f_{k|k-1}(x_k^{(i)}|x_{k-1}^{(i)})}{p(x_k^{(i)}|x_{k-1}^{(i)}, z_k)}. \tag{21}$$

Note that the propagation and weighting steps of the SIS algorithm are performed independently on each particle, and thus, are well-suited for parallel execution.

B. Resampling

The basic SIS algorithm, however, suffers from particle depletion or degeneracy, i.e., after a number of recursions, nearly all of the particles have negligible weights, which severely degrades the effectiveness of the approximation. This problem can be mitigated by *resampling* the weighted particles $\{(x_k^{(i)},\, w_k^{(i)})\}_{i=1}^{N}$ to generate more offspring of particles with high weights and eliminate those with low weights such that the expected number of offspring of a particle is proportional to its weight [2], [5], [9], [10]. Popular resampling strategies such as stratified sampling, residual sampling and systematic sampling, see e.g. [7], involve a comparison of values drawn from a uniform distribution against the *cumulative sum of particle weights*. This operation cannot be performed independently on each particle, making the resampling non-parallelizable.

For parallel implementation, we employ the *Metropolis* resampler proposed in [11], [17], based on the Metropolis algorithm for sampling from a given unnormalized probability distribution. Our unnormalized probability distribution of interest is given by the unnormalized weights of the particles. Given the i-th (weighted) particle, we sample an index j uniformly from $\{1, ..., N\}$, and assign it to the next iterate of the Markov chain with probability $\alpha(i,j) = \min\{1, w_k^{(j)}/w_k^{(i)}\}$, otherwise the next iterate remains at i. When the Markov chain converges to the discrete weighted distribution of the particles, subsequent iterates are unbiased samples. This algorithm only requires the ratio between pairs of (unnormalized) weights instead of the cumulative sum of weights, and thus permits parallel execution. However, the chain only generates unbiased samples after a certain number iterations. The longer the chain runs, the smaller the bias. Hence, in practice a compromise is needed between execution speed and bias.

C. Improving Particle Diversity

Frequent resampling coupled with small process noise could lead to a rapid loss of particle diversity, resulting in a poor filtering density approximation. Particle regularization and Markov chain Monte Carlo (MCMC) move are two main techniques to combat loss of diversity induced by resampling [8]. While regularization is simple to implement, the regularized particle set is not guaranteed to asymptotically approximate the filtering density. The MCMC move ensures resulting samples asymptotically approximate the filtering density.

The idea behind the MCMC move is to generate, from the resampled particles, a set of N distinct particles that are still distributed according to the filtering density. This can be accomplished (in parallel) by feeding each resampled particle to a Markov chain designed to converge to the filtering density [21]. If an iterate is already a sample from the filtering density, then the next iterate of the Markov chain is also distributed according to the filtering density. In this paper the MCMC move is implemented with a Metropolis-Hastings kernel. Specifically, given particle $x_k^{(i)}$ as the current iterate of the chain, we sample a state $x_k^{(i)*}$ from a proposal distribution $q(\cdot\,|x_k^{(i)})$ and assign it to the next iterate with probability

$$\alpha(x_k^{(i)}, x_k^{(i)*}) = \tag{22}$$
$$\min\left\{1, \frac{g(z_k|x_k^{(i)*}) f_{k|k-1}(x_k^{(i)*}|x_{k-1}^{(i)}) q(x_k^{(i)}|x_k^{(i)*})}{g(z_k|x_k^{(i)}) f_{k|k-1}(x_k^{(i)}|x_{k-1}^{(i)}) q(x_k^{(i)*}|x_k^{(i)})}\right\},$$

otherwise the next iterate remains at $x_k^{(i)}$. Note that the MCMC move only requires one iterate of the chain whereas the Metropolis resampler requires multiple iterates.

D. Implementation details

As alluded to in Section I, all received observations are paired with a new/existing track. For each newly identified track (identified at the point of associating a bearing measurement to that track label for the first time) we initialize a particle filter. The state estimate is taken to be the mean.

The JMS model switching probabilities are chosen to be

$$\vartheta\left(m_k|m_{k-1}\right) = \begin{cases} 0.9, & \text{if } m_k = 1 \text{ and } m_{k-1} = 1 \\ 0.1, & \text{if } m_k = 2 \text{ and } m_{k-1} = 1 \\ 0.5, & \text{if } m_k = 1 \text{ and } m_{k-1} = 2 \\ 0.5, & \text{if } m_k = 2 \text{ and } m_{k-1} = 2 \end{cases}. \tag{23}$$

The particle filter is implemented with proposal

$$p(x_k^{(i)}|x_{k-1}^{(i)}, z_k) = f_{k|k-1}^{(m_k^{(i)})}(\zeta_k^{(i)}|\zeta_{k-1}^{(i)}) \vartheta'(m_k^{(i)}|m_{k-1}^{(i)}), \tag{24}$$

where $\vartheta'(m_k^{(i)}|m_{k-1}^{(i)})$ is the mode proposal, which results in the weight update

$$w_k^{(i)} \quad = \quad w_{k-1}^{(i)}\, g(z_k|x_k^{(i)})\, \frac{\vartheta(m_k^{(i)}|m_{k-1}^{(i)})}{\vartheta'(m_k^{(i)}|m_{k-1}^{(i)})}.$$

Note that the kinematic state of each particle is sampled from the kinematic transition density but the motion model

is sampled from a discrete distribution given the previous motion model. For our experiments we set the mode proposal $\vartheta'(m_k^{(i)}|m_{k-1}^{(i)}) = 0.5$, i.e., each particle is equally likely to retain its current mode or switch to the new mode. Consequently, propagating the 500,000 particles forward through each motion model independently yields a total of 1 million particles.

Resampling is performed after each data update followed by an MCMC move. Applying traditional resampling on 1 million particles results in a computational bottleneck. This is alleviated by the Metropolis resampler implemented in CUDA on the graphics processing unit (GPU) with 1 million parallel threads. The subsequent MCMC move is also implemented in CUDA and executed on the GPU with proposal

$$q(x_k^{(i)*}|x_k^{(i)}) = f_{k|k-1}^{(m_k^{(i)*})}(\zeta_k^{(i)*}|\zeta_{k-1}^{(i)})\vartheta'(m_k^{(i)*}|m_{k-1}^{(i)}), \quad (25)$$

which gives an acceptance probability of

$$\alpha(x_k^{(i)}, x_k^{(i)*}) = \min\left\{1, \frac{g(z_k|x_k^{(i)*})\frac{\vartheta\left(m_k^{(i)*}|m_{k-1}^{(i)}\right)}{\vartheta'\left(m_k^{(i)*}|m_{k-1}^{(i)}\right)}}{g(z_k|x_k^{(i)})\frac{\vartheta\left(m_k^{(i)}|m_{k-1}^{(i)}\right)}{\vartheta'\left(m_k^{(i)}|m_{k-1}^{(i)}\right)}}\right\}.$$

From the resulting 1 million particles, we uniformly draw 500,000 particles to bring the total number of particles back to the original count.

IV. Numerical Experiments

The automated TMA system with the proposed filter is bench-marked on a dataset involving experienced submariners, taken from the *high integration scenario* of the *CRUSE* experiment (referred to as the Gold Standard dataset) carried out by the HuFAC Laboratory at the University of Western Australia [15].

The online tracking performance is measured by the OSPA$^{(2)}$ distance between the set of estimated tracks and set of true tracks [3]. This distance, based on the OSPA metric [22], carries the interpretation of a time-averaged per-track error, and captures both localization and cardinality errors between the set of true and estimated tracks, and penalizes switched tracks or label changes [3]. Our evaluation uses OSPA$^{(2)}$ with order parameter 1 and cutoff parameter 10,000 yards, over a sliding window of 120 time-steps.

Parameter	Initial Observation	Gold Standard dataset
Range	SONAR	$\mathcal{U}(2000, 30000)$
Range	OPTIX	$\begin{cases}\mathcal{U}(1000, 10000), \text{if } \mathcal{C} = \text{'fishing'} \\ \mathcal{U}(2000, 30000), \text{otherwise}\end{cases}$
Bearing	SONAR/OPTIX	$\mathcal{N}\left(\text{initial bearing observation}, \frac{\pi^2}{180^2}\right)$
Course	SONAR/OPTIX	$\mathcal{U}(0, 2\pi)$
Speed	SONAR/OPTIX	$\mathcal{U}(0, 15)$
Turn Rate	SONAR/OPTIX	$\mathcal{N}\left(0, \left(\frac{10^{-8}\pi}{180}\right)^2\right)$

TABLE II
Parameters for filter initialization in Gold Standard dataset.

For each new instance of the filter, the initial 500,000 particles are generated by first sampling the range, bearing, course, speed and turn rate from the densities given in Table II. These values are then used to derive the positions and velocities of the particles, shown in Fig 1. All 500,000 particles are initialized with motion model 1 (smaller noise). Once initiated, bearing observations are received at roughly 20-second intervals.

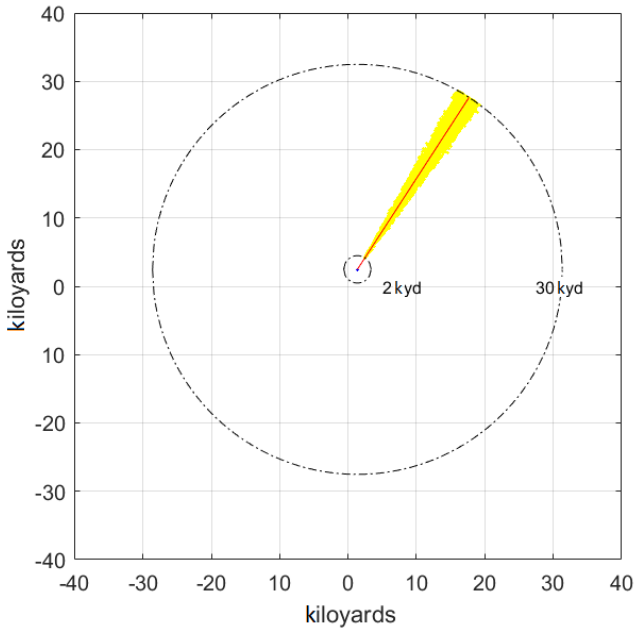

Fig. 1. Gold Standard dataset initial particle distribution (yellow) for a track with a 33° initial bearing (red).

Movemements of ownship and 14 other vessels during the 60-minute duration of the Gold Standard scenario are shown in Fig. 2. This scenario is quite challenging due to several reasons: the ownship performs no maneuvers (i.e., less observability); a relatively high number of tracks (for human operators); a zigging track; no shipping lanes; and tracks appearing on the scene with more diffused ranges. Fig. 3 shows that the automated TMA system's OSPA$^{(2)}$ multi-target tracking error curve stays below that of the experienced human operators for the most part of the 60-minute duration, keeping in mind that the latter had access to additional soft data that has not been integrated into the automated TMA system. This suggests the potential for automated TMA to assist human operators with the overall workload.

Note that current TMA systems assume correct data association, which is not the case in practice. For more reliable operation, the proposed hard-soft data fusion approach can be extended to address data association uncertainty using the multi-sensor Generalized Labeled Multi-Bernoulli (GLMB) filter [24]. Further, given the adverse signal scenario, tracking performance can be improved by using the multi-scan GLMB filter [23] to integrate over multiple data frames, see e.g., [16]. Reliability and better tracking performance, however, come at the expense of increased computational resources.

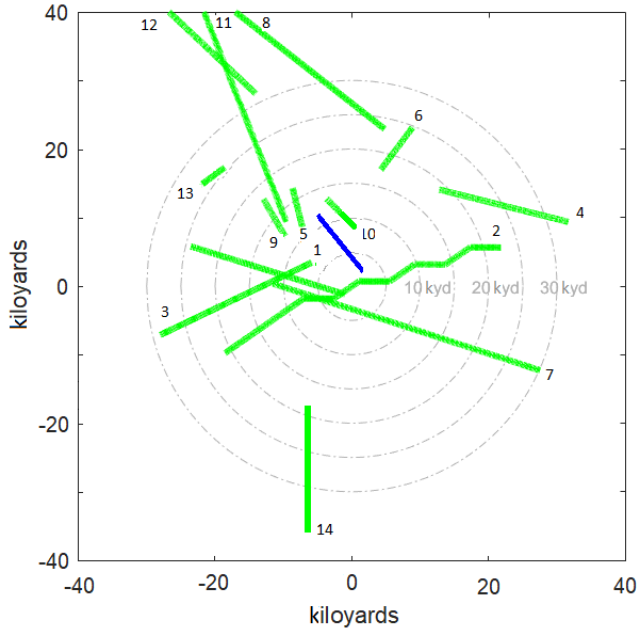

Fig. 2. Gold Standard scenario movements of 14 vessels and ownship (blue). Merchant vessels move unrestricted as there are no defined shipping lanes. Ownship moves at constant velocity on a 321°-course with no maneuvers.

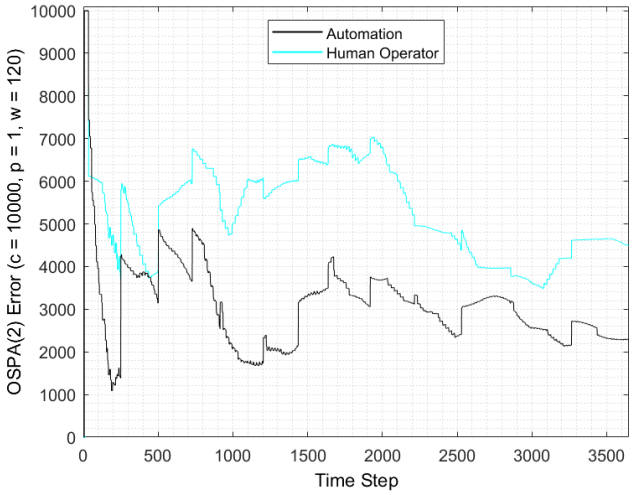

Fig. 3. Gold Standard high resolution scenario multi-target tracking errors.

V. Conclusions

This paper has demonstrated the feasibility of hard and soft data fusion for TMA with promising experimental results. In all experimental settings, track initialization for the automated system used very conservative or very diffuse settings. The prototype implementation, with unoptimized software on standard commercial-off-the-shelf hardware, has a typical processing times of several hundred milliseconds per track and per frame of data. At the scenario peak involving up to 14 tracks, the system had at most several seconds processing delay. These experimental results suggest significant potential for an automated TMA system to assist the crew in reducing their overall workload and improving overall situational awareness.

References

[1] B.D.O. Anderson and J.B. Moore, *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[2] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Nnline Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Processing*, 50(2):174-188, 2002.

[3] M. Beard, B.-T. Vo, and B.-N. Vo, "A Solution for Large-scale Multi-object Tracking," *IEEE Trans. Signal Processing*, 68:2754-2769, 2020.

[4] K. Becker, "Target motion analysis (TMA)", in S. Stergiopoulos (Ed.): *Advanced Signal Processing Handbook*, CRC Press, 2001.

[5] M. Bolic, P.M. Djuric, and S. Hong, "Resampling Algorithms for Particle filters: A Computational Complexity Perspective," *EURASIP J. Applied Signal Processing,* 15:1-11, 2004.

[6] X. Chen, R. Tharmarasa, and T. Kirubarajan, "Multitarget Multisensor Tracking," *Academic Press Library in Signal Processing*, 2:759-812, Elsevier, 2014.

[7] R. Douc and O. Cappe, "Comparison of Resampling Schemes for Particle Filtering," in *Proc. 4th Int. Symp. Image & Signal Processing & Analysis* (ISPA):64-69, 2005.

[8] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics & Computing*, 10(3):197-208, 2000.

[9] N. Gordon, D. Salmond, and A. Smith, "Novel Approach to Non-linear/Non-Gaussian Bayesian State Estimation," in *IEE Proceedings F - Radar & Signal Processing*, 140(2):107-113, 1993.

[10] J.D. Hol, T. Schon, and F. Gustafsson, "On Resampling Algorithms for Particle Filters," in *Proc. IEEE Nonlinear Statistical Signal Processing Workshop*, pp 79-82, Cambridge, UK, September 2006.

[11] A. Lee, C. Yau, M.B. Giles, A. Doucet, and C.C. Holmes, "On the Utility of Graphics Cards to Perform Massively Parallel Simulation of Advanced Monte Carlo methods," *J. Computational & Graphical Statistics*, 19(4):769-789, 2010.

[12] X.R. Li, and V.P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aerospace & Electronic Systems*, 39(4):1333-1364, 2003.

[13] M.E. Liggins, D.L. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice*, Second Edition: CRC Press, 2008.

[14] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Artech House, Inc. Norwood, MA, USA, 2007.

[15] S. Michailovs, S. Pond, M. Schmitt, J. Irons, M. Stoker, T.A. Visser, S. Huf, and S. Loft, "The Impact of Information Integration in a Simulation of Future Submarine Command and Control," *Human Factors*, 2021.

[16] D. Moratuwage, B.-N. Vo, B.-T. Vo, and C. Shim, "Multi-Scan Multi-Sensor Multi-Object State Estimation," *arXiv:2205.15516 [cs.CE]*, 2022.

[17] L. Murray, A. Lee, and P. Jacob, "Parallel Resampling in the Particle Filter," *J. Computational & Graphical Statistics*, 25(3):789-805, 2016.

[18] A. Pasha, B.-N. Vo, H.D. Tuan, and W.-K Ma, "A Gaussian Mixture PHD Filter for Jump Markov System Models," *IEEE Trans. Aerospace & Electronic Systems*, 45(3):919-936, 2009.

[19] Y. Punchihewa, B.-N. Vo, and B.-T. Vo, "A Generalized Labeled Multi-Bernoulli Filter for Maneuvering Targets," in *Proc. 19th Int. Conf. Information Fusion*, Heidelberg, Germany, 2016.

[20] Y. Punchihewa, B.-T. Vo, B.-N. Vo, and D.Y. Kim, "Multiple Object Tracking in Unknown Backgrounds with Labeled Random Finite Sets," *IEEE Trans. Signal Processing*, 66(11):3040-3055, 2018.

[21] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artec House, 2004.

[22] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Trans. Signal Processing*, 56(8):3447-3457, 2008.

[23] B.-N. Vo, and B.-T. Vo, "A Multi-Scan Labeled Random Finite Set Model for Multi-object State Estimation," *IEEE Trans. Signal Processing*, 67(19):4948-4963, 2019.

[24] B.-N. Vo, B.-T. Vo, and M. Beard, "Multi-sensor Multi-object Tracking with the Generalized Labeled Multi-Bernoulli Filter," *IEEE Trans. Signal Processing*, 67(23):5952-5967, 2019.