Abstract—The number of nodes in sensor networks is continuously increasing, and maintaining accurate track estimates inside their common surveillance region is a critical necessity. Modern sensor platforms are likely to carry a range of different sensor modalities, all providing data at differing rates, and with varying degrees of uncertainty. These factors compound the fusion problem as multiple observation models are required, along with a dynamic prediction model. However, the problem is exacerbated when sensors are not registered correctly with respect to each other, i.e. if they are subject to a static or dynamic bias. In this case, measurements from different sensors may correspond to the same target, but do not correlate with each other when in the same Frame of Reference (FoR), which decreases track accuracy. This paper presents a method to jointly estimate the state of multiple targets in a surveillance region, and to correctly register a radar and an Infrared Search and Track (IRST) system onto the same FoR to perform sensor fusion. Previous work using this type of parent-offspring process has been successful when calibrating a pair of cameras, but has never been attempted on a heterogeneous sensor network, nor in a maritime environment. This article presents results on both simulated scenarios and a segment of real data that show a significant increase in track quality in comparison to using incorrectly calibrated sensors or single-radar only.

Index Terms—Sensor fusion, registration, PHD filter, radar, infrared, calibration, tracking, maritime

I. INTRODUCTION

A. State-of-the-Art and Problem Outline

With many advances in sensor suites in recent years, tracking targets from multiple aspects using a range of different sensor modalities is now possible. Sensor fusion is a mature and sophisticated technology that looks to automate the process of combining a number of heterogeneous sources of information. By combining these sources of different information, the result should, in some sense, be better than what would have been possible if the data from each sensor was used individually [1], [2]. Maritime navigation radars typically take a number of seconds to perform a full sweep of the region they are observing, whereas optical systems can potentially have much faster update rates, making the measurements asynchronous. By exploiting the high update rate and fusing these image measurements with the radar, target tracks can be updated and maintained more frequently [3].

The calibration of tracking systems [4] through estimation of their model parameters is an important prerequisite before such systems are deployed out in the field. In particular, when fusing multiple sensor measurements, it is necessary to consider sensor calibration or registration into the same Frame of Reference (FoR). If, in the real world, the set of sensors is incorrectly calibrated, any fusion in the multi-sensor multi-target tracking algorithm could possibly lead to total failure through loss of useful tracking information. Possible calibration or registration errors could come from a number of different sources. These could include incorrect calibration during sensor manufacture, incorrect alignment during installation and setup, and potentially even uncontrollable factors such as harsh weather, sensor drift (Global Positioning System (GPS) drift or Inertial Measurement Unit (IMU) inaccuracies), or misalignment through platform vibrations. Such drift might possibly be dynamic, resulting in the need of repeated, time-consuming re-calibration, or it could be impossible to calibrate the system to a global frame altogether, e.g. in GPS-denied environments. In such cases, it would be advantageous to calibrate sensors relative to each other in an automatic manner to avoid loss in track accuracy due to incorrect data fusion.

Previous work in the fusion of radar and optical imagery data has been shown to be successful when tracking manoeuvring targets [5] and also for the application of avian monitoring [6]. In these articles, the registration problem is either not accounted for, or is treated as a separate process before fusion occurs. Methods for solving the sensor registration problem have been shown in [7]–[10] which use pseudo-measurement approaches to estimate biases in sensor networks, and in [11] which uses deep learning to determine appropriate registration parameters. Machine learning-type methods require vast amounts of realistic training data in order to give reliable results, furthermore these articles only present results on datasets with biases around one order of magnitude smaller than those treated in this article.

The joint method presented below is flexible and allows for varying types of multiple target tracking algorithm to be used, such as Joint Probabilistic Data Association (JPDA) [12], [13], Multiple Hypothesis Tracking (MHT) [14], [15], and Belief Propagation (BP) and message passing [16], [17].
In the single-cluster method, the offset angle \( \phi \) is identified which needs to be accounted for during fusion. The contributions of this paper include:

1) a multi-target tracking (MTT) technique that incorporates sensor calibration in a joint manner, in contrast to existing techniques which solve tracking and calibration separately by using pseudo-measurements [7]–[10]; the proposed technique also avoids the computationally expensive data association problem found in some joint fusion and registration approaches, such as [30];

2) a new predictive model for tracking changes in the dynamic sensor configuration which is included in the parent process computations (Sec. II-A); it also incorporates non-uniform sampling information to overcome the asynchronous aspect of the sensor network (Sec. II-B);

3) non-linear observation models respecting the MTT output states being given in Cartesian coordinates and measurements in polar coordinates; therefore, implementations of two multi-object estimation algorithms are introduced in this paper to solve the non-linear fusion problem, including a novel Extended Kalman Filter (EKF) version of the Panjer filter (Sec. II-B);

4) a comprehensive set of simulations (Sec. III-D), involving a typically challenging tracking scenario where target trajectories cross one another; moreover, the simulations shown in this article consider a much larger angular offset, by an order of magnitude, compared to those simulated in [8], [31].

C. Paper Organization

The remainder of this paper is organized as follows: Sec. II gives an overview of the joint estimation problem, with modelling and implementation information given in Sec. III; results are shown in Sec. IV and conclusions are drawn in Sec. V.

II. MULTIPLE TARGET TRACKING AND FUSION

The main challenge in this work is that the offset angle between the radar and the IRST system, \( \phi \), is unknown and must be estimated recursively along with the target states based on...
on imperfect data. For example, consider the scenario shown in Fig. 1 where a slow moving maritime target follows the trajectory shown in the dark thick line. Both sensors generate accurate measurements in their own FoR; the registration error is only apparent once the measurements are projected onto a common FoR such as WGS84. The radar produces range-azimuth measurements represented by the crosses which are assumed to be close to the true target trajectory.\(^1\) However, the azimuth-onlyIRST measurements shown as the light lines radiating from the origin contain a systematic angular offset, \(\phi\), that is to be estimated in order to perform accurate sensor fusion. The following sections will provide detailed information on the work flow presented in Fig. 2.

A. Parent Process: Sensor Registration

The parent process is responsible for estimating the potentially time-varying sensor registration angle, \(\phi\), in order to project the measurements of one sensor into the FoR of the other sensor for fusion purposes. The densities involved in the parent process are denoted with \(\hat{\theta}\) for the rest of this article. A Bayes recursion is used to propagate the posterior density, \(\hat{P}_k(\phi)\), for the calibration angle:

\[
\hat{P}_{k|k-1}(\phi) = \int \hat{f}_{k|k-1}(\phi | \phi') \hat{P}_{k-1}(\phi') d\phi', \tag{1a}
\]

\[
\hat{P}_k(\phi | Z_k) = \frac{\hat{f}_k(\phi | Z_k) \hat{P}_{k|k-1}(\phi)}{\int \hat{f}_k(\phi' | Z_k) \hat{P}_{k|k-1}(\phi') d\phi'} \tag{1b}
\]

where \(\hat{f}_k(\phi | Z_k)\) is a multi-object likelihood (MOL) function, and \(\hat{f}_{k|k-1}(\phi | \phi')\) is the transition density. Note that the

\(\hat{\theta}^1, \hat{\theta}^2, \hat{\theta}^N, \hat{\theta}_N\)

\(\theta^1, \theta^2, \theta^N, \theta_N\)

MOL, or registration parameter likelihood, is different from a multitarget association likelihood like the one used in the offspring process below as it describes the plausibility of a given sensor registration parameterisation \(\phi\) based on a set of measurements \(Z_k\); more details can be found in [4], [26]. The multi-object/multi-measurement association likelihood, on the other hand, measures the association of the target states given the sensor measurements.

Using a particle filter approach, the registration parameter to be estimated at time-step \(k\) is represented with a particle distribution \(\phi^j_k\) for \(1 \leq i \leq N\), where each \(i\) represents a different sensor geometry. Each particle has a corresponding weight \(w^j_k\) and an underlying set of multi-target estimation statistics \(\theta^i_k\) that are dependent on the offspring process. The particle weight \(w^j_k\) essentially encapsulates the belief that registration parameter \(\phi^j_k\) best represents the true sensor calibration. With the particles fixed to a grid, the particle weights can be computed using grid-based methods [32, p. 9]. Using this method, the recursion for predicting and updating the parent process weights is

\[
w^j_{k|k-1} = \sum_{i=1}^{N} w^i_{k-1} \hat{f}_{k|k-1}(\phi^j_k | \phi^i_k) \tag{2a}
\]

\[
w^j_k = \frac{w^j_{k|k-1} \hat{f}_k(\phi^j_k | Z_k)}{\sum_{i=1}^{N} w^i_{k|k-1} \hat{f}_k(\phi^i_k | Z_k)} \tag{2b}
\]

where \(\hat{f}_{k|k-1}(\phi^j_k | \phi^i_k) = \hat{f}^j_{k|k-1}\) is a discrete density over the difference in angle-indices and \(\hat{f}_k(\phi^j_k | Z_k)\) is the MOL function evaluated for a given sensor geometry at \(\phi^j_k\). Equation (2a) can be written as the convolution of weights \(w^j_{k-1}\) and a kernel \(\hat{f}^j_{k|k-1}\). The transition is modelled as a perturbation with a discretised wrapped Gaussian distribution. This is approximated using a finite-support shifted binomial distribution where \(u \sim B(n, p)\) and \(i = j + u - n/2\) is the relationship between the predicted angle-index, \(i\), and the particle angle-index, \(j\). Heuristically, the values \(n = 6, p = 0.5\) are used, such that Equation (2a) can in practice be calculated as the convolution of \(w^j_{k-1}\) and the kernel \(B(6, 0.5) = \{0.0156; 0.0938; 0.2344; 0.3125; 0.2344; 0.0938; 0.0156\}\). The parent process is initialized with a flat prior distribution, where all sensor geometries are equally likely.

The information contained inside \(\theta^i_k\) at time-step \(k\) is dependent on the type of offspring filter that is chosen, i.e. it either just contains the intensity, \(\mu_k\), of the multi-target distribution in case of the PHD filter, or both \(\mu_k\) and the variance \(\text{var}_k\) in case of the Panjer filter. The filters and their respective MOL functions are described below.

B. Offspring Process: Sensor Fusion

Let \(X\) and \(Z\) denote the state and measurement spaces, respectively. The offspring process estimates the time-varying multi-object state with \(n_k\) targets, \(\psi \in X^{n_k}\), dependent on a certain sensor configuration \(\phi\). This process is assumed to evolve with a Markov transition function \(f_{k|k-1}(\psi | \psi')\), which in this case will follow the near constant velocity (NCV) motion model [33], with a dynamic value for state transition.
time $\Delta_k$. This dynamic value is calculated at the measurement buffer stage as shown in Fig. 2, using the equation $\Delta_k = t_k - t_{k-1}$ where $t_k$ is the known current measurement time, and $t_{k-1}$ is the previous measurement time. The multi-measurement/multi-target likelihood $l_k(\psi|\phi, Z_k)$ describes the association likelihood of targets and measurements. The following Bayes recursion is used to propagate the law $P_k$ of the target process at time $k$:

$$P_{k|k-1}(\psi|\phi) = \int f_{k|k-1}(\psi'|\phi) P_{k-1}(\psi'|\phi)d\psi'$$

(3a)

$$P_k(\psi|\phi, Z_k) = \int_{l_k(\psi'|\phi, Z_k)} l_k(\psi|\phi, Z_k) P_{k|k-1}(\psi'|\phi) d\psi'$$

(3b)

For this work, the offspring process will be fulfilled using either the original PHD or the Panjer PHD filter as described below, and concrete model choices are described in Sec. III-B. When a measurement arrives from the reference sensor, $\theta_i^k$ are updated for all $i = 1, \ldots, N$, and when a measurement arrives from an uncalibrated sensor, both $\theta_i^k$ and the registration parameter weights $w_i^k$ in (2b) are updated using the MOL in Eqns (9) or (13), respectively.

1) PHD Filter: The PHD filter is now a mature method of performing multi-target tracking. It was first developed in 2003 [18], [19] and the common Gaussian mixture implementation was introduced in 2006 [20]. A full derivation of the filter can be found in [18], [20]. This first-order filter propagates only the mean $\mu_2$ of the point process and assumes the predicted number of targets and false alarms are both Poisson distributed [25]. The implementation used in this work can be found in [20, Table IV].

2) Panjer PHD Filter: The Panjer PHD filter was first introduced in [24] as a useful extension to the first-order PHD filter defined earlier. This process propagates both the mean and variance of the point process, using the assumption that the number of predicted targets and the false alarms are Panjer distributed. The Panjer distribution is characterised by two parameters $\alpha$ and $\beta$ which closely correspond to its mean and variance.

Case 1: $0 < \{\alpha, \beta\} \in \mathbb{R}^+ \times \mathbb{R}^+$: Represents a negative binomial distribution, where the variance is greater than the mean. This would be useful, for example, in situations where there may be sudden large influxes of measurements such as strong returns on a rough sea.

Case 2: $0 > \{\alpha, \beta\} \in \mathbb{Z}^+ \times \mathbb{R}^-$: Represents a binomial distribution where the variance is less than the mean. This could be used in a situation where a very consistent number of false alarms is expected, e.g. a static or slow-changing environment.

Case 3: $\{\alpha, \beta\} \to \infty$: The limit case where the ratio stays constant, resulting in the Poisson distribution. This represents the standard false alarm model used in many pieces of target tracking literature.

Having these three different cases available gives more flexibility in modelling the number of targets and false alarms. A full mathematical derivation and pseudo-code is available elsewhere [24], and is omitted here.

3) Extended Kalman Filter PHD and Panjer filters: An EKF version of both offspring filters is required to overcome the non-linearity between the Cartesian state space and the Polar observation space. EKFs use a Jacobian matrix to linearize the non-linear function around the current state estimate. This Jacobian is found by performing partial differentiation of the observation model equations with respect to the variables in the state vector, such that the elements of the matrix are

$$J_{pq} = \frac{\partial h_p}{\partial x_q}, \ p \in \{r, \phi\}, \ q \in \{1, \ldots, 4\}$$

(4)

where $h_p$ is the conventional Cartesian to Polar transformation

$$h_r = r = \sqrt{x^2 + y^2}, \ h_\phi = \phi = \tan^{-1}(x, y),$$

and where $\tan^{-1}(x, y)$ is the four-quadrant inverse tangent function and $X = [x, \dot{x}, y, \dot{y}]^T$. During an offspring update with radar measurements, the Jacobian matrix used is

$$J_R = \begin{bmatrix} \frac{x}{\sqrt{x^2 + y^2}} & 0 & -y \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{-y}{\sqrt{x^2 + y^2}} & 0 & x \frac{-y}{\sqrt{x^2 + y^2}} \end{bmatrix}$$

(5)

and for an IRST update, the Jacobian becomes

$$J_I = \begin{bmatrix} \frac{-y}{x^2 + y^2} & 0 & x \frac{-y}{x^2 + y^2} \\ x \frac{-y}{x^2 + y^2} & 0 & -y \frac{x}{x^2 + y^2} \end{bmatrix}.$$  

(6)

C. The multi-object likelihood functions

In this section, the PHD and the Panjer version of the MOL are revisited, and a short version of their derivations is given in a supplementary document. More detailed explanations are found in [22] and [34]. Note that we will omit an implicit time index $k$ in all notations for the sake of brevity.

1) Notations: In the following, all equations referring to the PHD filter or Poisson distributed phenomena are marked with the symbol $\bullet$ and any reference to the Panjer filter or the Panjer distribution is indicated with the symbol $\circ$. Dependent on the sensor configuration $\phi$, we can write down the spatial distribution of the false alarms as $s_c(\psi|\phi)$. Furthermore, suppose that $\mu_{p|k-1}(\cdot|\phi)$ is the predicted intensity of the PHD or Panjer filter, $l(x|z)$ denotes the single-target single-measurement association likelihood, and $p_d(\cdot|\phi)$ is the (state-dependent) probability of target detection. For arbitrary regions $B \subseteq \mathcal{X}$, the association terms $\mu_{p|k-1}(\cdot|\phi)$ are given by

$$\mu_{z|k-1}^\circ(B|\phi) = \int_B p_d(x) l(x|z) \mu_{p|k-1}(x) dx.$$  

(7)
2) The PHD filter likelihood [22]: For the original PHD filter, the number of false alarms is assumed to be Poisson distributed, in which case \( \lambda_c \) shall denote the Poisson false alarm rate and \( \mu^* (\phi) = \lambda_s c^*(\phi) \) is its intensity function (cf. Eq. (28) in the supplementary document).

**Theorem II.1** (MOL of the PHD filter [22]). Given the short-hand notation

\[
\mu^*_d (X) = \int_X p_d (x|\phi) \mu^*_pr (x|\phi) dx, \tag{8}
\]

the likelihood function of the PHD filter for a given sensor state \( \phi \) is found to be

\[
\hat{\lambda}_s (Z) = \frac{\prod \mu^*_s (z|\phi) + \mu^*_c (X|\phi)}{\exp \left[ \int \mu^*_c (z|\phi) dz + \mu^*_d (X) \right]} \tag{9}
\]

3) The Panjer filter likelihood [25], [34]: Let us define the Pochhammer symbol or rising factorial \( x_{n}^{\uparrow} \) by

\[
x_{n}^{\uparrow} := \prod_{i=0}^{n-1} (x + i), \quad x_{0}^{\uparrow} := 1. \tag{10}
\]

As the name suggests, the Panjer filter approximates Panjer distributed false alarms, in which case we write \( \alpha_c, \beta_c \) for the two Panjer clutter parameters and \( s_z^c \) for the spatial distribution of the false alarms.

**Theorem II.2** (MOL of the Panjer PHD filter). Write \( \alpha = \alpha_{pr}, \beta = \beta_{pr} \) and \( s_z^c \) for the Panjer parameters and the spatial distribution of the predicted process, and let

\[
F_{d,\phi} = 1 + \frac{1}{\beta} \int_X p_d (x|\phi) s_{pr}^o (x|\phi) dx, \tag{11}
\]

\[
F_c = 1 + \frac{1}{\beta_c} \tag{12}
\]

be two auxiliary functions for a given sensor state \( \phi \). The multi-object likelihood function of the Panjer PHD filter for \( \phi \) is found to be

\[
\hat{\lambda}_s (Z) = \sum_{j=0}^{[Z]} \frac{\alpha_j \mu^c (|Z| - j) \mu^c (|Z| - j)}{\beta^c (\beta_c + 1)^{|Z| - j} F_{d,\phi} - \alpha - |Z| - j} \cdot \sum_{Z' \subseteq Z, z \in Z'} \mu^c (X|\phi) \prod_{|Z'| = j} s_z^c (z|\phi). \tag{13}
\]

### III. Modelling, Data, and Scenarios

#### A. Implementation

Much like the parent process in [35], the parent process in this application is represented with a one-dimensional even spread of particles. In this case, particles are distributed on a fixed grid between angles \( \pm 10^\circ \) from the centre of the field of view. The number of particles \( N \) has been chosen to be 201 such that the angular resolution is 0.1° between consecutive filters. With this even spread of particles, and a consistent test between \( \pm 10^\circ \), there is no need for a particle resampling step [36] in this algorithm, reducing the computational effort required. In this work we only consider estimating one parameter, however several parameters could be estimated simultaneously if necessary at the cost of using more particles. A high-level pseudocode for this solution is given in Algorithm 1.

Both offspring processes are implemented using a Gaussian-Mixture representation of the target population like in [20], [24], but using an EKF as described earlier. A measurement-driven birth process will be used as described in [37], and the Hellinger distance is used for component merging [38]; components with low weights below a threshold \( \tau_{prune} \) are removed at every update.

#### B. Model Definitions

1) Dynamical Model: The measurement buffer will have access to the raw range-bearing radar measurements and the bearing-only IRST measurements that are recorded at a given iteration \( k \) and physical time \( t_k \). The MTT routine will be performed using a 4-D Cartesian state vector with elements

\[
x_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^T \tag{14}
\]

where \( x_k, y_k \) are the \( x \) and \( y \) positions of a target, and \( \dot{x}_k, \dot{y}_k \) are the \( x \) and \( y \) velocities of a target. For this maritime surveillance-based scenario, it is assumed that each and every target follows a near-constant velocity (NCV) model [39], [40] that is described by

\[
x_k = F_k x_{k-1} + w_k \tag{15}
\]

where \( F_k \) is the state transition matrix

\[
F_k = \begin{bmatrix}
1 & \Delta_k & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & \Delta_k \\
0 & 0 & 0 & 1
\end{bmatrix}, \quad \Delta_k = t_k - t_{k-1}, \tag{16}
\]
and $w_k$ represents zero-mean white Gaussian process noise with covariance

$$Q_k = \begin{bmatrix} q\Delta_k^3/3 & q\Delta_k^2/2 & 0 & 0 \\
q\Delta_k^2/2 & q\Delta_k & 0 & 0 \\
0 & 0 & q\Delta_k^3/3 & q\Delta_k^2/2 \\
0 & 0 & q\Delta_k^2/2 & q\Delta_k \end{bmatrix}$$

(17)

and $q$ is the acceleration noise value in both the $X$ and $Y$ directions.

2) Measurement Models: The radar measurement model is defined as

$$z_k^R = h^R(x_k) + \eta_k^R,$$

(18)

with

$$h^R(x_k) = \begin{bmatrix} r_k \\
\phi_k \end{bmatrix} = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\
\tan^{-1}(x_k, y_k) \end{bmatrix},$$

(19)

where $r_k > 0$, $\tan^{-1}(x_k, y_k)$ is the four-quadrant inverse tangent function, and the resulting $\phi_k$ lies within $[0, 2\pi)$. The additive noise term $\eta_k^R$ is defined by

$$\eta_k^R \sim \mathcal{N}(\eta_k^R; 0, \text{diag}(\sigma_{r_k}^2, \sigma_{\phi_k}^2))$$

(20)

where $\sigma_{r_k}$ and $\sigma_{\phi_k}$ are the radar’s range and azimuth standard deviations respectively.

The camera measurement model is described by

$$z_k^C = h^C(x_k) + \eta_k^C,$$

(21)

where

$$h^C(x_k) = \phi_k = \tan^{-1}(x_k, y_k).$$

(22)

The additive noise term $\eta_k^C$ is defined by

$$\eta_k^C \sim \mathcal{N}(\eta_k^C; 0, \sigma_{\phi_c}^2)$$

(23)

where $\sigma_{\phi_c}$ is the camera azimuth standard deviation.

C. Sensors

The maritime navigation radar used as a part of the data collection was a Kelvin Hughes SharpEye system [41], shown in the background of Fig. 3. It uses a number of different radar techniques to detect a wide-range of targets in a maritime environment. During the trial, the system was operating in a full 360° sweep mode, with a full scan taking approximately 2.5 seconds to complete. The low-profile antenna gives an azimuthal beam-width of less than 1° at the 3 dB point. Techniques such as monopulse to further improve the angular accuracy of the radar were not in use; accurate target localisation inside the beam was not possible.

The IRST system, shown in the foreground of Fig. 3, was a research platform which limited the amount of information available. IRST is one method for detecting and tracking targets that give off infra-red signatures. The wavelength of this sensor tends to be shorter than that of a conventional maritime radar, thus giving much better angular resolution. One drawback however is that it can be affected by atmospheric conditions and the weather, both shortening its effective range. The reasons for performing sensor fusion in this case are to exploit the better angular accuracy in the IRST measurements and the higher update rate available from the IRST sensor.

D. Simulations

In order to test the algorithm using simulated data, a challenging scenario involving crossing target trajectories has been created. Two targets move around inside the IRST field-of-view and their trajectories cross around 60 iterations into the scenario. Crossing targets often make the data association problem difficult for tracking algorithms; a sensor with a slow update rate will not generate estimates quickly enough, and therefore there will be a loss in track resolution. Sensor properties and update rates are close to those defined in the accompanying data sheets and manuals [41].

In this first scenario, we consider two potential false alarm models for a maritime tracking scenario; the common Poisson model, and also a negative binomial model [42]. The latter will allow for a large variance in the number of false alarms generated from crests of waves or from rough sea conditions [23]. Simulations have been created to test the filters on each of those models. The parameters used for the simulations and the filters are shown in Table I and are typical for multiple target tracking simulations using PHD filters [20], [24]; any variation in parameters for different experiments will be stated later. For the first two experiments (Sec. IV-A and IV-B) where the IRST is calibrated onto the radar FoR, a misalignment of 3° has been simulated in the IRST measurements. Due to the different characteristics and sampling rates of radar and an infrared sensor, a third simulated scenario is considered (Sec. IV-C) where the radar measurements contain an angular bias and the IRST measurements have no bias. This alternative scenario will show that it is possible to still estimate this angular bias, even when exploiting the infrequent radar measurements, rather than the high-frequency IRST measurements.

E. Real Scenario

In order to further test the algorithm developed, a short segment of data from a real scenario was used. Both sensors were located on a sea wall at Fort Blockhouse in Gosport overlooking Portsmouth harbour. The trial coordinators had instructed a number of instrumented targets to be present during the trial. However, there was a large amount of background
traffic such as ferries and cargo vessels passing through, which is actually an advantage: having more targets present in the scene is preferable when attempting to calibrate using this method as more measurement-to-track associations can be made, and therefore increase the MOL.

In the segment of real data used, a target crosses the field of view at approximately 2 kilometres away from the sensors. The segment lasts for approximately 50 seconds and then the target disappears from view. For the sensor fusion aspect, a subset of the IRST measurement sets are used to improve the update rate, but not so much as to heavily rely on bearings-only track updates; a typically challenging problem in MTT. Approximately three sets of IRST measurements are used between consecutive radar scans.

### IV. Results

All results have been averaged over 50 Monte Carlo runs, and the registration angle estimate is taken as the Maximum A Posteriori (MAP) estimate of the parent likelihood function. The tracked output from the PHD filter and Panjer PHD filter will be compared to the simulated trajectories using the Optimal Subpattern Assignment (OSPA) distance [43]. The OSPA distance is a combination of a cardinality error and a localisation error between two sets $X$ and $Y$ with cardinalities $m$ and $n$, which is widely used to determine accuracy in multitarget tracking systems. It is given by [43, Eq. (3)]

$$
d_p^c(X, Y) = \left[ \min_{\pi \in \Pi_n} \sum_{i=1}^{m} d(c, Y_{\pi(i)})^p + c^p(n - m) \right]^{\frac{1}{p}},
$$

using an order parameter $p$ and a cut-off distance $c$. Here, the distance function $d(c, Y_{\pi(i)}) = \min(c, d(x, y))$ is an appropriate distance measure, e.g. the Euclidean distance, cut off at $c$, and $\Pi_n$ denotes the set of all possible permutations of the numbers $1, \ldots, n$. As the order parameter increases, the metric penalises estimates that lie further away from the ground truth more harshly. From [43, Sec. III-D], $p = 2$ is a good practical choice for the order parameter, as it usually gives smoother distance curves, and is consistent with other metrics that use a $p$-th order average construction. The cut-off distance $c$ determines the trade-off between penalising cardinality errors as opposed to localisation errors. For all OSPA results shown, a cut-off parameter of $c = 100$ m and order parameter $p = 2$ will be used.

In Figs. 5 through to 9, where the OSPA distance and IRST pointing angle are plotted, these are the averages taken across all Monte-Carlo trials. We use a consistent target ground truth in all trials and use this to generate a different set of sensor measurements for each individual trial. The legend shown in Fig. 5a is consistent across Figs. 5, 6, 7, and 10a. The red dashed plots represent the case where only the radar is used and no data fusion is performed. The blue dotted plots show the case where the radar and IRST are incorrectly registered and no correction attempt is made, whereas the dashed-dotted black plots show the results for the joint estimation method presented in this work. Finally, as a benchmark to compare our result to, we have also simulated the case where only the radar is used and no data fusion is performed. The dotted black plots show the case where the radar and IRST are correctly registered, giving an optimal result. This is shown as the solid green plot on each figure.

#### A. Poisson Distribution

We start by considering the Poisson distribution as the underlying false alarm model. Here, the mean number of false alarms is equal to the variance, giving a reasonably consistent number per scan. The average false alarm rates for each sensor are shown in Table I. From Figs. 5 and 6, the importance of having the correct registration between sensors is clear. Performing tracking with only radar measurements...
is accurate in itself, however the tracking accuracy can be improved by performing sensor fusion between a radar and a calibrated IRST sensor. Without taking the registration into account, and performing fusion with an IRST sensor that is uncalibrated, tracking performance is decreased and the OSPA distance increases significantly. Comparing Fig. 5a to Fig. 6a, and Fig. 5b to Fig. 6b there is an improvement in tracking performance by using the Panjer PHD filter. In the \( p_d = 0.99 \) case, the \( k = 95 \) OSPA distances for the PHD filter and Panjer PHD filters with estimated registration are 21.72 m and 20.04 m respectively, and in the \( p_d = 0.85 \) case, these are 32.56 m and 30.72 m. This is due to propagation of the variance of the cardinality distribution, as well as the mean.

As with many tracking scenarios, the probability of detection \( p_d \) is an important factor and as shown in Figs. 8a and 9a, performance improves in both the target tracking and the registration estimation as \( p_d \) increases from 0.7 to 0.99.

**B. Negative Binomial Distribution**

For this set of simulations, the variance in the number of false alarms is no longer constrained to be equal to the mean. The included variance parameters are shown in Table I. The PHD filter assumes the false alarm distribution is Poisson, and does not take account of the variance information. The Panjer PHD filter however allows for variance information to be included. It can be seen in Fig. 7 that the Panjer PHD filter does outperform the PHD filter in all cases. At \( k = 95 \) in the estimated registration case, the OSPA distance for the PHD filter is 35.71 m, whereas for the Panjer PHD filter, this is 30.61 m. Again these results show it is important to take account of the registration between the sensors. A larger performance gain can be seen in Fig. 9. We see that even with lower values of \( p_d \) (e.g. 0.7), the Panjer PHD filter is able to give more consistent and accurate estimates of the IRST pointing angle. We require a higher \( p_d \) value (> 0.9) to get accurate registration estimates from the PHD filter.

**C. Alternative Frame of Reference**

We now consider the alternative representation of the sensors, where instead, the radar measurements are assumed to contain some angular bias and the IRST measurements contain no bias. Firstly in Fig. 10a, it can be seen that the tracking accuracy using a single radar has now decreased as expected, and the result where fusion is performed between the unregistered sensors is still poor. The estimated result is close to the unbiased and correctly registered result, with an OSPA distance at \( k = 95 \) of 27.04 m. In Figs. 10b and 10c, as the probability of detection \( p_d \) increases, there is a decrease in the OSPA distance, and the angle estimate tends towards the true registration configuration, following the same trends as the results given in Figs. 8a and 9a.

**D. Real Scenario**

Fig. 11 shows that after the initialisation to 10° and a transition period, a 1.5° to 2° registration error is estimated...
Fig. 8. OSPA Distance over $p_d$. The average OSPA Distance is taken from between Iteration 80 and Iteration 100.

(a) Poisson Distributed

(b) Negative Binomial Distributed

Fig. 9. IRST Pointing Angle over $p_d$. The average pointing angle is taken from between Iteration 80 and Iteration 100.

(a) Poisson Distributed

(b) Negative Binomial Distributed

Fig. 10. Results for alternative FoR scenario.

(c) Estimated Alignment Angle w.r.t. IRST frame

Fig. 11. Estimated correction angle for real scenario
between the radar and the IRST system. After the transition, it can be seen that the angle only deviates from this a small amount, which could be due to some platform vibration, or gusts of wind for example. This error could account for the fact that the installation was done only by eye and therefore subject to human error. The advantage of performing sensor fusion is apparent in the comparison in Fig. 12. The radar track shown in Fig. 12a is accurate, however there are considerable gaps between consecutive estimates. These gaps are filled using fusion of the IRST information, see Fig. 12b. With range information only available in the radar measurements, we see the two larger "jumps" in the track at (−2000, 400) and (−1850, 450). These are points where radar measurements become available again and range information can be updated.

V. CONCLUSIONS

We have successfully demonstrated a suitable method for performing joint sensor registration and fusion using asynchronous and heterogeneous sensors. The registration parameter is estimated on the fly, based on the performance of the underlying target tracker, which has never been performed in this context before. The simulation results on a challenging scenario clearly highlight the importance of calibrating the network before performing fusion across the sensors, even when using an alternative false alarm model. When applying this method to a real data set, a plausible offset angle was found. For this sensor setup of radar and IRST, the well-documented bearings-only tracking problem was evident when using IRST measurements. By not having a range measurement available, tracking accuracy is reduced. This was compensated for by using only a subset of the IRST images available, giving more weight to the sparse range-bearing measurements of the radar.

Taking into consideration the simulated scenario outlined in this work, it is of vital importance that sensor fusion is used. Using only the slow update rate of the radar system will leave longer periods where no track estimates are given, and therefore track resolution is lost. Exploiting the higher update rate of the IRST system majorly improves the performance where the calibration is accurate, especially at the point where the targets cross. Having more track estimates gives better situational awareness, and helps to resolve potential issues with estimate-to-track association.

REFERENCES

The Multi-Object Likelihood for the PHD and Panjer Filters

A supplement supporting the publication:

*Joint Registration and Fusion of an Infra-Red Camera and Scanning Radar in a Maritime Context*

David Cormack, Isabel Schlangen, James R. Hopgood and Daniel E. Clark

I. MOTIVATION

The MOLs of the PHD and Panjer filter (Eq. (9) and (13)) can be proved with the mathematical framework of point processes, which is a general form of the Finite Set Statistics (FISST) developed by Mahler [19]. In Sec. II, we will introduce some mathematical tools that will be necessary to conduct the proofs in Sec. III.

II. A SHORT INTRODUCTION TO POINT PROCESSES

A. Probability Generating Functionals

A useful stochastic concept to describe populations of objects is the notion of a point process \( \Phi \) on \( \mathcal{X} \), which is a random variable on the space \( \mathcal{X} = \bigcup_{n \geq 0} \mathcal{X}^n \) of finite sequences in \( \mathcal{X} \). A realisation of \( \Phi \) is a sequence \( \varphi = (x_1, \ldots, x_n) \in \mathcal{X}^n \), where both the number \( n \in \mathbb{N} \) and the states \( x_i \in \mathcal{X} \) of the objects are random. Each point process \( \Phi \) can be characterised by its Probability Generating Functional (PGFL)

\[
G_{\Phi}(h) = \sum_{n \geq 0} \int \left[ \prod_{i=1}^{n} h(x_i) \right] P^{(n)}_{\Phi}(x_1, \ldots, x_n) \, dx_1 \ldots dx_n,
\]

(15)

where the \( P^{(n)}_{\Phi} \) are permutation-invariant probability measures on \( \mathcal{X}^n \) for all \( n \geq 0 \), and \( h \) is a real-valued function. This type of functional is a useful mathematical representation of a point process since it makes it possible to model the number of objects on the one hand (via considering all cardinalities \( n \) in the sum) and their stochastic properties on the other (via the probabilities \( P^{(n)}_{\Phi} \) for every fixed \( n \)). Let us look at a few common examples of PGFLs which will also be used later on.

The most simple PGFL is that of a Bernoulli process with parameter \( p \), and spatial distribution \( s_\bullet \) given by

\[
G_\bullet(h) = (1 - p) + p \int h(x) s_\bullet(x) \, dx.
\]

(16)

This process is a suitable choice for modelling survival and detection processes since those can have only two outcomes; either, the object dies (or gets miss-detected) with probability \( (1 - p) \), or it survives (or is detected) with probability \( p \), in which case it evolves according to the spatial distribution \( s_\bullet(\cdot) \).

The PGFL of a Poisson point process with parameter \( \lambda \) and intensity \( \mu(\cdot) = \lambda s_\bullet(\cdot) \) is of the form [44]

\[
G_\bullet(h) = \exp \left( \int [h(x) - 1] \mu_\bullet(x) \, dx \right).
\]

(17)

\( 1 \)If instead of sequences, one assumes that realisations of \( \Phi \) are finite sets, we arrive at the Random Finite Set (RFS) theory derived by Mahler [19]. Sets do not allow state repetitions by definition, whereas sequences generally do.

This compact form is found by inserting the probability mass function of a Poisson distribution in Eq. (15) and using the series expansion of the exponential function to remove the infinite sum. The Poisson point process gained its popularity from the fact that while describing common phenomena in many applications, it only depends on one parameter and its exponential form makes it easy to work with.

A less known concept is that of the Panjer distribution, which is common in actuarial mathematics but only made its way to the engineering community very recently. The PGFL of a Panjer point process with spatial distribution \( s_\circ \) and parameters \( \alpha, \beta \) is found to be [44], [34]

\[
G_\circ(h) = \left( 1 + \frac{1}{\beta} \int [1 - h(x)] s_\circ(x) \, dx \right)^{-\alpha}.
\]

(18)

One can show that by taking the limit \( \alpha \to \infty \) while keeping the ratio \( \frac{\alpha}{\beta} \) constant, the PGFL (18) reduces to that of the Poisson process (17). Furthermore, a binomial (or negative binomial) process is obtained by choosing negative (or positive) values for both \( \alpha \) and \( \beta \), which makes this formulation versatile and useful in cases where the Poisson assumption is too restrictive.

B. PGFLs and functional differentiation

Many properties of a point process are encoded in its statistical moments, which are obtained via differentiation of its PGFL. Two particularly important and useful properties are:

- Calculating the expected value [47]:

\[
\mathbb{E}[G_\bullet] = \delta G_\bullet(h; \eta)|_{h=1}
\]

(19)

- Extracting the probability measure of having exactly \( n \) objects:

\[
P^{(n)}_\Phi(x_1, \ldots, x_n) = \delta^n G_\Phi(h; \delta x_1, \ldots, \delta x_n)|_{h=0},
\]

(20)

where \( \delta x \) denotes the Dirac delta function, being nonzero at \( x \) only.

The derivation of the MOL heavily relies on Eq. (20) since we want to express the likelihood of having exactly \( m \) measurements \( z_1, \ldots, z_m \), given the current state of the system. The filter equations of the Probability Hypothesis Density (PHD) filter and its variations propagate the expected value which is found via the rule (19). For the purpose of computing explicit formulae for the Multi-Object Likelihood (MOL) functions, let us first write down the so-called chain differential which admits a higher order product and chain rule [46]; other common differential operators are the Gâteaux and Fréchet differential [45].
Definition II.1 (Chain differential [45], [46]). Let \((\eta_n : \mathcal{X} \rightarrow \mathbb{R}^+ \}_{n \in \mathbb{N}}\) be a sequence of positive, bounded functions converging pointwise to a function \(\eta : \mathcal{X} \rightarrow \mathbb{R}^+\), and let \((\varepsilon_n)_{n \in \mathbb{N}}\) be a sequence of positive real values converging to 0. The chain differential of a functional \(G\) with respect to its functional argument \(h : \mathcal{X} \rightarrow \mathbb{R}^+\) in the direction of \(\eta\) is defined as

\[
\delta G(h; \eta) = \lim_{n \to \infty} \frac{G(h + \varepsilon_n \eta_n) - G(h)}{\varepsilon_n}.
\]

If the limit exists, it is unique for any sequence \((\varepsilon_n)_{n \in \mathbb{N}}\) and \((\eta_n : \mathcal{X} \rightarrow \mathbb{R}^+ \}_{n \in \mathbb{N}}\) with the above properties.

The chain differential yields an \(n\)-fold product rule [44]

\[
\delta^n (F \cdot G)(h; \eta_1, \ldots, \eta_n) = \sum_{\omega \subseteq \{1, \ldots, n\}} \delta^{[\omega]} F(h; \eta_{i \in \omega}) \delta^{[\omega]} G(h; \eta_{j \in \omega}),
\]

with \(\omega\) being the set complement \(\{1, \ldots, n\} \setminus \omega\), and an \(n\)-fold chain rule2 [46]

\[
\delta^n (F \circ G)(h; \eta_1, \ldots, \eta_n) = \sum_{\pi \in S_n} \delta^{[\pi]} F(G(h); \eta_{i \in \pi}) \delta^{[\pi]} \eta_{i \in \pi},
\]

where \(\pi \in S_n\) is a permutation of \(\{1, \ldots, n\}\).

C. Concatenation of point processes

Of course, most problems in Bayesian filtering cannot be described by one single point process only since several processes are involved in both the prediction and the update. Joint PGFLs can be written down in analogy to (15) but involving joint probabilities of the form \(P_{\Psi}^{(n,m)}(x_1 \ldots x_n, z_1 \ldots z_m)\). In general, these do not simplify, however we will be interested in two special cases, namely superposition and branching.

1) Superposition: If two processes are independent, their joint PGFL decomposes into a product of the form

\[
G_{\Psi, \Phi}(h, g) = G_{\Phi}(h) G_{\Psi}(g).
\]

This is used, for example, for superimposing spontaneous clutter which is assumed to be independent of the target process.

2) Branching: If each point in process \(\Phi\) creates a new process \(\Psi\), its resulting PGFL is the concatenation of the individual functionals:

\[
G_{\Psi, \Phi}(h, g) = G_{\Phi}(h) G_{\Psi}(g|\cdot)).
\]

Such a structure is necessary to describe the detection process of the current targets; each target is either detected or missed, hence one would choose a Bernoulli process for the inner and the predicted process for the outer functional.

With the above considerations in mind, the general form of the PGFL describing the joint target and measurement processes, dependent on sensor state \(\phi\), is of the form

\[
G_d(h, g|\cdot, \phi) = G_{pr}(h G_d(h|\cdot, \phi)) G_c(g|\cdot, \phi).
\]

Here, \(G_c(g|\cdot, \phi)\) is the Bernoulli detection process for a given target state \(x\) and sensor state \(\phi\) with detection probability \(p_d(x|\phi)\) and single-object single-measurement association likelihood \(s_\phi(x|\cdot, \phi)\), i.e.

\[
G_d(g|x, \phi) = 1 - p_d(x|\phi) + p_d(x|\phi) \int g(z)|l(x|z, \phi)dz.
\]

III. Derivation of the MOLs

The MOL of each filter is obtained by first setting \(h = 1\) and then taking the \(|Z|\)-fold derivative of (26) w.r.t. \(g\) (in the manner of (20)), i.e. differentiating once for each measurement \(z\) in the set \(Z\). The only difference between the two forms is the model assumption regarding the predicted and clutter processes.

A. Proof of Theorem II.1

In case of the PHD filter, both \(G_{pr}\) and \(G_c\) are Poisson PGFLs with predicted intensity \(\mu_{pr}^*\) and clutter intensity \(\mu_c^*\), such that (26) takes the form

\[
G^*_d(g, 1|\phi) = \exp \left[ \int (g(z) - 1) \mu_c^*(z|\phi) + \int \left(G_d(g|x, \phi) - 1\right) \mu_{pr}^*[x|\phi]dx \right].
\]

Differentiation by \(g\) requires repeated applications of the chain rule (23) which pulls out one multiplicative term \(\mu_c^*(z|\phi) + \int p_d(x|\phi)|l(z|z, \phi)\mu_{pr}^*[x|\phi]dx\) for each \(z \in Z_k\).

The final result is obtained by setting \(g = 0\).

An alternative derivation of the MOL of the PHD filter is found in [22] where the FISST framework is used.

B. Proof of Theorem II.2

The Panjer filter assumes that \(G_{pr}\) and \(G_c\) are Panjer processes with spatial distributions \(s_{pr}^o\) and \(s_c^o\), respectively, and corresponding parameters \(\alpha_{pr}, \beta_{pr}\) and \(\alpha_c, \beta_c\). The explicit form of (26) becomes in this case

\[
G^*_d(g, 1|\phi) = \left[ 1 + \frac{1}{\beta_c} \int (1 - g(z)) s_c^o(z|\phi)dz \right]^{-\alpha_c}
\]

\[
\cdot \left[ 1 + \frac{1}{\beta_{pr}} \int \left( 1 - G_d(g|x, \phi) \right) s_{pr}^o(x|\phi)dx \right]^{-\alpha_{pr}}
\]

which needs to be differentiated using the general product rule (22). Using the notations \(G_{pr}^o(g|\phi) := F_c(g|\phi)^{-\alpha_c}\) and \(G_{pr}^o(g|\phi) := F_d(g|\phi)^{-\alpha_{pr}}\) for the two right-hand-side terms of (29), the corresponding derivatives are

\[
\delta^{|Z|} G_{pr}^o(g; (\delta_{z})_{z \in Z}|\phi) = F_c(g|\phi)^{-\alpha_c - |Z|} \left( \frac{\alpha_c}{\beta_c} Z \right)^\uparrow \prod_{z \in Z} s_c^o(z)
\]

and

\[
\delta^{|Z|} G_{pr}^o(g; (\delta_{z})_{z \in Z}|\phi) = F_d(g|\phi)^{-\alpha_{pr} - |Z|} \left( \frac{\alpha_{pr}}{\beta_{pr}} Z \right)^\uparrow \prod_{z \in Z} \int p_d(x|\phi)l(z|x, \phi) s_{pr}^o(x|\phi)dx.
\]

Including (30) and (31) into (22), switching the summations and rearranging the terms leads to the desired result.