18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Bernoulli Filtering on a Moving Platform

Simon J. Julier
Department of Computer Science
University College London
Gower Street London
WC1E 6BT UK
Email: s.julier@ucl.ac.uk

Amadou Gning
Department of Computer Science
University College London
Gower Street London
WC1E 6BT UK
Email: e.gning@ucl.ac.uk

***Abstract*—This paper considers the problem of tracking a target — which might or might not exist — from a platform whose position is not known perfectly and might contain substantial time dependencies.**

Most single and multi-target tracking algorithms implicitly or explicitly assume that the location of the sensor platform system is known perfectly. However, in practice the location of sensing platforms is often estimated, usually by fusing a set of sensor measurements from different sources. As a result, the error in the platform estimates could be significant and time correlated. These difficulties are compounded in single and multi-target tracking problems when the existence of a target is not guaranteed.

In this paper, we consider the problem of tracking at most a single target from a poorly-localized UAV. We develop a formulation of the Bernoulli filter which incorporates both the target state and the state of the platform. However, because the dimension of the state is relatively large, we develop a suboptimal algorithm which, through neglecting the use of track information to improve the quality of the platform estimate, scales in a manner very similar to that of a conventional Bernoulli filter.

The implementations of the different algorithms are tested in a simulation scenario of a UAV performing safety monitoring of a convoy.

I. Introduction

In many operational contexts, understanding the location of potential threats is extremely important. By understanding where threats might be found, commanders can plan and execute missions to minimise hazards, either by neutralising or by avoiding threats. Consider, for example, the problem illustrated in Figure 1 — a convoy has been tasked to move through an urban environment, avoiding potential threats. One way to develop situation awareness is to deploy a low altitude, agile UAV such as a quadrotor which could explore the area immediately around the convoy to ensure that the area is free of potential hazards. To achieve this goal, the system must be able to carry out multi-target tracking from a moving platform.

The challenges of target-tracking are well-known: uncertainties in the numbers and locations of the targets, together with the association between target and sensor returns, leads to a problem which scales factorially. Many approaches have been developed, including those based on multiple hypothesis tracking [3] and those based on Random Finite Sets (RFS) [4]. In this work, we concentrate on the RFS formulation. An RFS encodes the uncertainty in both the numbers and locations of the targets. Although the full RFS formulation is intractable, tractable solutions for the single target case exist [4], [5]. In addition, many suboptimal algorithms — including the PHD Filter [6], the CPHD Filter [7], the Bernoulli Filter, the multi-Bernoulli [8] Filter and the Labelled Bernoulli Filter [9] — have been developed. These have been shown to be highly effective in some cases, especially when the clutter rate or the density of targets is relatively high.

Fig. 1. The reference scenario, implemented in a high-resolution quadrotor simulator [1]. The brown boxes are footprints of buildings. The blue square is a simulation of a likelihood-based observation model for detecting targets [2].

In our reference scenario, the tracking sensor (a camera) is fixed to a mobile platform. When the state of the platform, its effects can be modelled as a parameter or control input which typically enters into the observation model [4] where it can be treated in a similar manner to a bias parameter [10]. However, in many situations, the state of the platform itself must be estimated from a sequence of sensor measurements using some kind of recursive filter. This introduces time correlated errors which will directly enter into the target estimates. As a result the estimates of different target tracks are not independent of one another. Potential issues include loss of covariance consistency and challenges with data association [11], [12]. Therefore, approaches to overcome the effects of time correlated platform errors must be addressed.

Within target tracking, most work has tended to focus on estimating static errors, such as sensor biases, which do not change over time. Approaches include the use of maximum likelihood [5], [13]–[15], as well as pseudomeasurements [16]. Within the robotics community, dynamic errors due to platform uncertainty are widely studied the context of Simultaneous Localisation and Mapping (SLAM) in which a mobile robot estimates the structure of the environment and uses this estimate to improve its estimate of its own position. Although SLAM traditionally considered the problem of just stationary features, it has been extended to handle both stationary and moving objects as the so-called Detection and Tracking of Moving Objects (DATMO) problem.

Although these works are highly relevant for the problem we seek to address, these algorithms were largely developed by extending tractable approximations to RFS to include the effects of moving platforms. In this paper, we consider the problem from a more fundamental point of view directly in terms of the elements of random sets. Rather than tackle the full multi-target problem directly, in this paper we investigate the problem for the case of at most a single target. Although this could be argued to be a trivial problem, we believe it is both practically useful in its own right, and lays the theoretical foundation for considering how platform movement can be incorporated in more general formulations of multi-target tracking, such as the multi-Bernoulli filter.

The structure of the paper is as follows. In Section II, the problem statement is described and two key problems — dependent tracking of targets and tracking with occlusions — are identified. Section III develops the formulation for the Bernoulli filter to account for correlations in the targets and proposes a full particle implementation of the algorithm. However, the resulting state space is rather large because the target and platform states have to be estimated simultaneously. To overcome this limitation, in Section IV we propose a decoupled implementation in which correlations from the platform estimate are used to propagate the track information, but the track information is not used to improve the quality of the platform estimate. Section V evaluates the solution in a simulation scenario. A summary and conclusions are presented in Section VI.

II. Problem Statement

A. Scenario Description

We consider the problem illustrated in Figure 1. The following assumptions are made:

1) The environment can contain at most one target.
2) The birth, death and time evolution of the target is independent of the platform.
3) All measurements can be classified into two types: those which provide information about just the platform, and those that provide information about the platform and the target.

Given these assumptions, we now analyse the behaviour of the platform and the target.

B. Platform System Description

The state of the platform at time k is $\overset{*}{\mathbf{x}}_k$. It evolves according to the discrete time process model

$$\overset{*}{\mathbf{x}}_k = \overset{*}{\mathbf{f}}\left[\overset{*}{\mathbf{x}}_{k-1}, \overset{*}{\mathbf{u}}_{k-1}, \overset{*}{\mathbf{v}}_{k-1}\right], \tag{1}$$

where $\overset{*}{\mathbf{u}}_k$ are the control inputs and $\overset{*}{\mathbf{v}}_k$ is the process noise, which is not necessarily injected in a linear manner. The associated state transition probability is $\overset{*}{\phi}_{k+1|k}\left(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{x}}'\right)$.

The state of the platform is directly measured by onboard sensors such as a GPS, an Altitude Heading and Reference System (AHARS) or potentially even a visual mapping system. In all cases, the sensor likelihood model is of the form

$$\overset{*}{\mathbf{z}}_k = \overset{*}{\mathbf{h}}\left[\overset{*}{\mathbf{x}}_k, \overset{*}{\mathbf{u}}_k, \overset{*}{\mathbf{w}}_k\right], \tag{2}$$

where $\overset{*}{\mathbf{w}}_k$ is the observation noise which is not necessarily additive. The important thing to note is that this type of observation is *only* a function of the platform's state and has no dependency on the state of the target. The corresponding likelihood model is $\overset{*}{\varphi}\left(\overset{*}{\mathbf{z}}_k|\overset{*}{\mathbf{x}}_k\right)$.

C. Target System Description

1) Probability Distribution: The environment is populated by a time-varying set of targets. At time k, suppose there are T_k. This can be represented by the random set $\mathbf{X}_k$ [4],

$$\mathbf{X}_k = \{\mathbf{x}_1, \ldots, \mathbf{x}_{T_k}.\} \tag{3}$$

In this paper we consider the case that there is at most a single target at any given time. Therefore, the FISST PDF is

$$\pi(\mathbf{X}_k) = \begin{cases} 1 - r_k & \text{if } \mathbf{X}_k = \emptyset \\ r_k \cdot s(\mathbf{x}_k) & \text{if } \mathbf{X}_k = \{\mathbf{x}_k\} \\ 0 & \text{if } |\mathbf{X}_k| \geq 2. \end{cases} \tag{4}$$

where r_k is the probability that the target exists and $s(\cdot)$ is the spatial distribution if it does exist. $s(\cdot)$ is a standard probability distribution, and so

$$\int s(\mathbf{x}_k) \mathrm{d}\mathbf{x}_k = 1. \tag{5}$$

2) State Transition Equations: The transition equations account for target birth, target death and the time propagation of persistent targets. The state transition densities are[1]

$$\begin{aligned} \phi_{k|k-1}\left(\mathbf{X}|\emptyset\right) &= \begin{cases} 1 - p_b & \text{if } \mathbf{X} = \emptyset \\ p_b \cdot b_{k|k-1}\left(\mathbf{x}\right) & \text{if } \mathbf{X} = \{\mathbf{x}\} \\ 0 & \text{if } |\mathbf{X}| \geq 2. \end{cases} \\ \phi_{k|k-1}\left(\mathbf{X}|\left\{\mathbf{x}'\right\}\right) &= \begin{cases} 1 - p_s & \text{if } \mathbf{X} = \emptyset \\ p_s \cdot \pi_{k|k-1}\left(\mathbf{x}\right) & \text{if } \mathbf{X} = \{\mathbf{x}\} \\ 0 & \text{if } |\mathbf{X}| \geq 2, \end{cases} \end{aligned} \tag{6}$$

where p_b is the birth probability, p_s is the survival probability, $b_{k|k-1}\left(\mathbf{x}\right)$ is the birth process and $\pi_{k|k-1}\left(\mathbf{x}\right)$ is the state transition of the target if it persists.

[1]For simplicity, we present the cases where probability of survival, probability of birth, probability of detection and clutter rate are all independent of target state and the environment. However, in general this is not true [2], [17]. The approach can be readily extended to include these dependencies.

3) Measurement Likelihood Equations Model: At each update time step, the camera on the platform acquires an image, a detection algorithm is run, and a set of M_k measurements are extracted. Each measurements consists of the position where it was discovered in the image [2]. The measurement set is

$$\overline{\mathbf{Z}}_k = \{\mathbf{z}_{k,i}, \ldots, \mathbf{z}_{k,M_k}\} \subset \mathcal{Z}. \tag{7}$$

The measurement model specifies that the observations originate from two sources: the targets and the background. The detector output observation process is

$$\mathbf{Z} = \mathbf{C}(\overset{*}{\mathbf{x}}) \bigcup \mathbf{W}(\mathbf{X}, \overset{*}{\mathbf{x}}), \tag{8}$$

where $\mathbf{C}(\overset{*}{\mathbf{x}})$ is the clutter and $\mathbf{W}(\mathbf{X}, \overset{*}{\mathbf{x}})$ is the observation RFS. The clutter likelihood is given by

$$\varphi(\mathbf{Z}|\emptyset) = \kappa(\mathbf{Z}) = e^{-\lambda} \prod_{\mathbf{z} \in \mathbf{Z}} \lambda c(\mathbf{z}). \tag{9}$$

The measurement likelihood for $\mathbf{W}(\mathbf{X}, \overset{*}{\mathbf{x}})$ is found by summing over all assignments of the observations to either clutter or to the target using the relationship

$$\eta(\mathbf{W}|\{\mathbf{X}\}) = \begin{cases} 1 - p_D(\overset{*}{\mathbf{x}}, \mathbf{x}) & \text{if } \mathbf{W} = \emptyset \\ p_D(\overset{*}{\mathbf{x}}, \mathbf{x}) \cdot g(\mathbf{z}|\overset{*}{\mathbf{x}}, \mathbf{x}) & \text{if } \mathbf{W} = \{\mathbf{z}\} \end{cases} \tag{10}$$

The observation likelihood $\varphi(\mathbf{Z}|\{\mathbf{x}\})$ is then found by assigning one measurement at a time to the target and all the rest to clutter [5] to give

$$\begin{aligned} \varphi(\mathbf{Z}|\{\mathbf{x}\}) = (1-r)\kappa(\mathbf{Z}) + \\ + r\kappa(\mathbf{Z}) p_D(\mathbf{x}) \left[\sum_{\mathbf{z} \in \mathbf{Z}} \frac{g(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})} - 1 \right]. \end{aligned} \tag{11}$$

Note that the platform and target states are coupled in two ways. First, the probability of detection, $p_D(\mathbf{x}, \overset{*}{\mathbf{x}})$, is affected by the state of both. For example, in Figure 1, the UAV can observe targets which lie within the view frustum of the camera. Second, the likelihood $g(\mathbf{z}|\overset{*}{\mathbf{x}}, \mathbf{x})$ also depends upon the relative transformation between the target and state.

D. Discussion and Related Work

The target and platform states are coupled through the target observation model described in (8). When the state of the platform is well-known, its distribution is effectively a delta function and the regular Bernoulli filter can be applied without modification. However, when the platform state is not well-known, the effect is to introduce uncertainty into the observation RFS (10).

As explained in the introduction, random set based solutions have begun to be developed to address issues with uncertainty in platform position. Ristic considered the problem of calibration and alignment of a pair of static sensing systems [15] by finding the maximum likelihood overlap between the two systems. Üney proposed to use the Hellinger distance between the intensities computed at different platforms [14]. Both approaches used nonlinear optimisation which could be run over multiple time steps. However, we seek a formulation in which real-time movement of the platform is supported. The closest work we are aware of relates to the use of random sets in Simultaneous Localisation and Mapping (SLAM) [18], [19]. In the SLAM problem, a mobile platform moves through an environment and constructs a map of stationary landmarks. This information is used, in turn, to improve the estimate of the robot's location. Critical to these algorithms is the fact that the correlations are maintained. To achieve this, Mullane et al. developed a solution which uses a proxy map to evaluate the dependency structure [18]. This was extended by Moratuwage et al. to include moving objects [20]. Lee, on the other hand, uses a hierarchical model based on a single cluster process [19] which directly includes moving objects. Recently, Deusch et al. applied the Labelled Multi-Bernoulli Filter to the SLAM problem [21] and demonstrated significant improvements over Mullane et al.'s PHD solution. Although their work does not explicitly include moving objects, these can be readily added using techniques applied in other random set formulations.

Although these approaches yield impressive results, they have been developed by first deriving tractable approximations to RFS, and then applying

III. Target Tracking from a Moving Platform

To fully account for the dependencies between the platform and the target state, our goal is to maintain the FISST PDF

$$f_{k|k}(\overline{\mathbf{X}}|\overline{\mathbf{Z}}_{1:k}) = f_{k|k}(\overset{*}{\mathbf{x}}, \mathbf{X}|\overset{*}{\mathbf{z}}_{1:k}, \mathbf{Z}_{1:k}). \tag{12}$$

To address this problem we begin by noting that, using the chain rule of probability,

$$\begin{aligned} f_{k|k-1}(\overline{\mathbf{X}}|\overline{\mathbf{Z}}_{1:k}) &= f_{k|k-1}(\overset{*}{\mathbf{x}}, \mathbf{X}|\overline{\mathbf{Z}}_{1:k}) \qquad (13) \\ &= f_{k|k-1}(\mathbf{X}|\overset{*}{\mathbf{x}}, \overline{\mathbf{Z}}_{1:k}) f_{k|k-1}(\overset{*}{\mathbf{x}}|\overline{\mathbf{Z}}_{1:k}). \end{aligned}$$

Therefore, substituting from (4), the FISST PDF will be of the form

$$\pi(\overline{\mathbf{X}}) = \begin{cases} (1-r) \cdot s_0(\overset{*}{\mathbf{x}}) & \mathbf{X} = \emptyset \\ r \cdot s_1(\overset{*}{\mathbf{x}}, \mathbf{x}) & \mathbf{X} = \{\mathbf{x}\} \end{cases} \tag{14}$$

where $s_0(\overset{*}{\mathbf{x}})$ is the localization distribution when the target does not exist, and $s_1(\overset{*}{\mathbf{x}}, \mathbf{x})$ is the localization distribution when it does exist. Note that, from this definition, the probability of existence of the target is given by

$$r = 1 - \int \pi\left(\overset{*}{\mathbf{x}}, \emptyset\right) \mathrm{d}\overset{*}{\mathbf{x}}. \tag{15}$$

In a conventional Bernoulli filter, the probability of existence is a scalar quantity which can be propagated separately from the particle distribution which represents the spatial distribution of the target [5]. However, this is not the case here. The reason is that, even if the target does not exist, the filter must still maintain a distribution over the location of the platform.

We now consider the prediction and update steps.

A. Prediction

The prediction is given by

$$f_{k|k-1}(\overline{\mathbf{X}}|\overline{\mathbf{Z}}_{1:k-1}) = \int \overline{\phi}_{k|k-1}\left(\overline{\mathbf{X}}|\overline{\mathbf{X}}'\right) f_{k|k-1}(\overline{\mathbf{X}}'|\overline{\mathbf{Z}}_{1:k})\delta\overline{\mathbf{X}}'. \tag{16}$$

Given our assumptions, the time evolution of the platform and the target are independent of one another. Therefore,

$$\overline{\phi}_{k|k-1}\left(\overline{\mathbf{X}}|\overline{\mathbf{X}}'\right) = \overset{*}{\phi}_{k|k-1}\left(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{x}}'\right)\phi_{k|k-1}\left(\mathbf{X}|\mathbf{X}'\right). \tag{17}$$

In Appendix A, we show that the probability that a target exists evolves according to

$$r_{k|k-1} = p_b(1 - r_{k-1|k-1}) + p_s r_{k-1|k-1}. \tag{18}$$

This is exactly the same as the form of the time evolution of the survival of the target for a conventional Bernoulli filter.

B. Update Step

The update is carried out using Bayes' Rule,

$$f_{k|k}(\overline{\mathbf{X}}_k|\overline{\mathbf{Z}}_{1:k}) = \frac{\overline{\varphi}\left(\overline{\mathbf{Z}}_k|\overline{\mathbf{X}}_k\right)\cdot f_{k|k-1}(\overline{\mathbf{X}}_k|\overline{\mathbf{Z}}_{1:k-1})}{f_k(\overline{\mathbf{Z}}_k|\overline{\mathbf{Z}}_{1:k-1})}. \tag{19}$$

The updated probability of existence, computed from (15), is given by

$$r_{k|k} = 1 - \frac{\int \overline{\varphi}\left(\overline{\mathbf{Z}}_k|\overset{*}{\mathbf{x}}_k,\emptyset\right)\cdot f_{k|k-1}(\overset{*}{\mathbf{x}}_k,\emptyset|\overline{\mathbf{Z}}_{1:k-1})\mathrm{d}\overset{*}{\mathbf{x}}_k}{f_k(\overline{\mathbf{Z}}_k|\overline{\mathbf{Z}}_{1:k-1})}. \tag{20}$$

We consider the case where we update with each type of observation in turn.[2]

1) Update with a Platform Observation: In this case,

$$\overline{\mathbf{Z}}_k = \overset{*}{\mathbf{z}}_k. \tag{21}$$

Substituting for the sensor likelihood model, (19) becomes

$$f_{k|k}(\overline{\mathbf{X}}_k|\overline{\mathbf{Z}}_{1:k}) = \frac{\overset{*}{\varphi}\left(\overset{*}{\mathbf{z}}_k|\overset{*}{\mathbf{x}}_k\right)\cdot f_{k|k-1}(\overline{\mathbf{X}}_k|\overline{\mathbf{Z}}_{1:k-1})}{f_k(\overset{*}{\mathbf{z}}_k|\overline{\mathbf{Z}}_{1:k-1})}. \tag{22}$$

The normalization constant is

$$\begin{aligned} f_k(\overset{*}{\mathbf{z}}_k|\overline{\mathbf{Z}}_{1:k-1}) &= \int \overset{*}{\varphi}\left(\overset{*}{\mathbf{z}}_k|\overset{*}{\mathbf{x}}_k\right) f_k(\overline{\mathbf{X}}|\overline{\mathbf{Z}}_{1:k-1})\delta\overline{\mathbf{X}} \\ &= \int \overset{*}{\varphi}\left(\overset{*}{\mathbf{z}}_k|\overset{*}{\mathbf{x}}_k\right) f_{k|k-1}(\overset{*}{\mathbf{x}}_k,\emptyset|\overline{\mathbf{Z}}_{1:k-1})\mathrm{d}\overset{*}{\mathbf{x}}_k \\ &\quad + \int\int \overset{*}{\varphi}\left(\overset{*}{\mathbf{z}}_k|\overset{*}{\mathbf{x}}_k\right) f_{k|k-1}(\overset{*}{\mathbf{x}}_k,\mathbf{x}_k|\overline{\mathbf{Z}}_{1:k-1})\mathrm{d}\mathbf{x}_k\mathrm{d}\overset{*}{\mathbf{x}}_k \\ &= \Delta_0 + \Delta_1. \end{aligned} \tag{23}$$

Using (20), the new probability of target existence is

$$r_{k|k} = \frac{\Delta_1}{\Delta_0 + \Delta_1} \tag{24}$$

In general, $r_{k|k} \neq r_{k|k-1}$. The reason is that when a target observation is used to perform an update, both the target and platform states are updated. If a track is not present but the platform updates the state as if it is, the effect is to introduce noise into the platform estimate. Direct observations of the pose of the platform can help to identify if this case has arisen.

2) Update with a Set of Target Observations: Now consider the case that just the target observations are available. In this case, (19) becomes

$$f_{k|k}(\overline{\mathbf{X}}_k|\overline{\mathbf{Z}}_{1:k}) = \frac{\varphi\left(\mathbf{Z}_k|\overline{\mathbf{X}}_k\right)\cdot f_{k|k-1}(\overline{\mathbf{X}}_k|\overline{\mathbf{Z}}_{1:k})}{f_k(\mathbf{Z}_k|\overline{\mathbf{Z}}_{1:k-1})}. \tag{25}$$

The target likelihood $\varphi\left(\mathbf{Z}_k|\overline{\mathbf{X}}_k\right)$ is given by extending (48) and (49) in [5] to include the platform state. Specifically,

$$\begin{aligned} \varphi\left(\mathbf{Z}|\overset{*}{\mathbf{x}},\emptyset\right) &= \kappa(\mathbf{Z}) \\ &= e^{-\lambda}\prod_{z\in Z}\lambda c(z,\overset{*}{\mathbf{x}}), \end{aligned} \tag{26}$$

$$\begin{aligned} \varphi\left(\mathbf{Z}|\overset{*}{\mathbf{x}},\{\mathbf{x}\}\right) &= (1-r)\kappa(\mathbf{Z}) + \\ &\quad r\kappa(\mathbf{Z})p_D(\mathbf{x},\overset{*}{\mathbf{x}})\left[\sum_{\mathbf{z}\in\mathbf{Z}}\frac{g(\mathbf{z}|\overset{*}{\mathbf{x}},\mathbf{x})}{\lambda c(\mathbf{z})} - 1\right]. \end{aligned} \tag{27}$$

The normalization constant is

$$\begin{aligned} f_k(\mathbf{Z}_k|\overline{\mathbf{Z}}_{1:k-1}) &= \int \varphi\left(\mathbf{Z}_k|\overset{*}{\mathbf{x}}_k,\mathbf{x}_k\right) f_k(\overline{\mathbf{X}}|\overline{\mathbf{Z}}_{1:k-1})\delta\overline{\mathbf{X}} \\ &= \int \varphi\left(\mathbf{Z}_k|\overset{*}{\mathbf{x}}_k,\emptyset\right) f_{k|k-1}(\overset{*}{\mathbf{x}}_k,\emptyset|\overline{\mathbf{Z}}_{1:k-1})\mathrm{d}\overset{*}{\mathbf{x}}_k \\ &\quad + \int\int \varphi\left(\mathbf{Z}_k|\overset{*}{\mathbf{x}}_k,\mathbf{x}_k\right) f_{k|k-1}(\overset{*}{\mathbf{x}}_k,\mathbf{x}_k|\overline{\mathbf{Z}}_{1:k-1})\mathrm{d}\mathbf{x}_k\mathrm{d}\overset{*}{\mathbf{x}}_k \end{aligned} \tag{28}$$

In Appendix B, we show that the probability of the target existence is similar to that of the Bernoulli filter, and can be written in the form

$$r_{k|k} = \frac{1-\Delta_k}{1 - r_{k|k-1}\Delta_k} r_{k|k-1}. \tag{29}$$

However, the term Δ_k includes the effects of both the platform and target state.

IV. Suboptimal Approximation

The last section described a filter which jointly estimates the (Bernoulli) state of a target and the state of the platform. As a result, the state space consists of both the the platform and target states. These can be high dimensional. For example, in many UAV applications, nine states are used to represent the platform (position, velocity and orientation in 3D). If the target dimension is four (position and velocity in the x and y directions), the overall dimension of the system is 13. Although Gaussian mixture approximations can be efficiently used with such high-dimensional spaces, the nonlinear nature of the detection region of the camera means that we prefer to use a particle filter. However, it is well-known that the number of particles required increases exponentially with dimension, and we seek a simple methods to reduce the overall computational cost of the filter.

[2] If the filter were presented with both types of observation simultaneously, each type of measurement could be fused sequentially.

As explained earlier, there are three reasons why we maintain both the UAV and target states in the state space at the same time:

1) To properly maintain the dependency structure between the platform and the target state estimates.
2) To use the observations of the platform to improve the quality of the estimate of both the platform and the target.
3) To use observations of the target state to improve the quality of the estimate of both the target and the platform.

Of these advantages, the one which is likely to provide the least benefit is to update the platform state based on the track. There are two reasons for this. First, platforms such as UAVs are often equipped with many types of sensors (such as GPS, AHARS and vision-based mapping systems). As a result, the UAV often has access to a range of high-quality information and the marginal information provided by the target estimate is likely to be very small. Second, even though the movement of the target relative to the platform might be measured accurately (for example via a camera) the target dynamics are often highly uncertain, meaning that the target provides relatively little information about the platform estimate.

Therefore, we propose to approximate the joint distribution as follows:

$$\begin{aligned} f_{k|k-1}(\overline{\mathbf{X}}|\overline{\mathbf{Z}}_{1:k}) &= f_{k|k-1}(\overset{*}{\mathbf{x}}, \mathbf{X}|\mathbf{Z}_{1:k}, \overset{*}{\mathbf{z}}_{1:k}) \\ &\approx f_{k|k-1}(\mathbf{X}|\overset{*}{\mathbf{x}}, \mathbf{Z}_{1:k}) f_{k|k-1}(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{z}}_{1:k}) \end{aligned} \tag{30}$$

Specifically, we have conditioned the platform state *only* on the platform observations. As a result, the updated platform information can be exploited by the target filter. However, the target information is not exploited by the platform. This decomposition is very similar to the approximation which is used to derive the Schmidt Kalman Filter (SKF) [22]. To reduce the state space in a Kalman filter, Schmidt proposed "locking" a set of states so that their values do not change when new sensor information arises. However, crucially, the filter maintains the correlations between locked and unlocked states. Similarly here, the flow of information from the platform to the target filter means that the dependency structure is maintained.

We now describe an implementation of this algorithm which we call the decoupled filter.

A. Implementation of the Decoupled Filter

The decoupled filter can be implemented in a very straightforward manner. It consists of the two filters as illustrated in Figure 2. The first filter is the platform filter. It maintains $f_{k|k-1}(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{z}}_{1:k})$. Any preferred estimation algorithm — including a particle filter, Gaussian mixture model, or a particle filter — can be used. The second filter is the Bernoulli target filter which maintains the probability of existence of the target, together with its spatial distribution. We use the particle implementation [5]. To ensure proper conditioning on the platform state, for each particle in the Bernoulli filter we draw a sample from $f_{k|k-1}(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{z}}_{1:k})$ and and condition that this sample is the actual state of the platform.

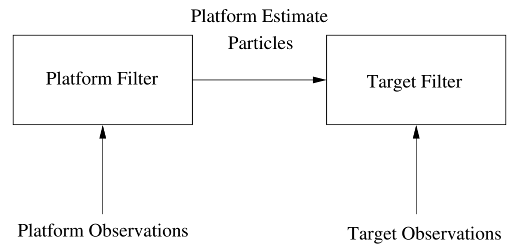

Fig. 2. The decoupled filter consists of two hierarchically arranged filters.

The computational cost of running the algorithm is equal to the cost of running the platform estimator, together with the cost of running the Bernoulli filter. The only additional overhead comes from the need to sample the platform for each Bernoulli particle.

V. Experimental Evaluation

A. Simulation Scenario

We evaluated the performance of the different configurations of the algorithm in simulation using a subset of a high-fidelity quadrotor simulator known as QRSim [1]. QRSim is a high-resolution simulation system for modelling both white and non-white noise sources in realistic UAV models. These correlations can be very important [23]. In this simulation we used a baseline version of QRSim's simulation models.

The state of the platform is

$$\overset{*}{\mathbf{x}} = \begin{bmatrix} x & y & z & u & v & w & \phi & \theta & \psi \end{bmatrix}^\top, \tag{31}$$

where the position is x, y, z the velocity is u, v, w and the attitude is expressed by the ϕ, θ, ψ Tait-Bryan angles using a rotation about z, followed about y and finally by x.

The platform sensing system consists of the following:

1) A GPS system which measures position with standard deviation of 1m, 1m, and 2meter.
2) An AHARS system which measures attitude with 2 degrees standard deviation in all angles.
3) A camera system. The camera system uses full perspective projection, view frustum culling, and lens distortion. Although extensive models of probability of detection of cameras exist [17], we used a simple approach in which the probability of detection was 0.999. In addition, noise was generated as a Poisson process with 10^{-6}.

The state of the target is

$$\mathbf{x} = \begin{bmatrix} x & y & u & v \end{bmatrix}^\top, \tag{32}$$

where the position is x, y and the velocity is u, v.

The parameters of the simulation are found in Table I.

Figure 3 lays out the scenario. A UAV flies at a constant altitude of 30m over flat ground. Three targets pass through the field-of-view of the sensor at different times.

Parameter	Value	Parameter	Value
σ_{GPS}	1m^2	σ_{AHARS}	2°
σ_{PX}	1px	Image size	3264×2248 px
Persistent particles	4000	New born particles	40
Probability of birth	0.3935	Probability of survival	0.9
Probability of detection	0.999	λ	10^{-5}

TABLE I
THE PARAMETER VALUES USED IN THE SIMULATION.

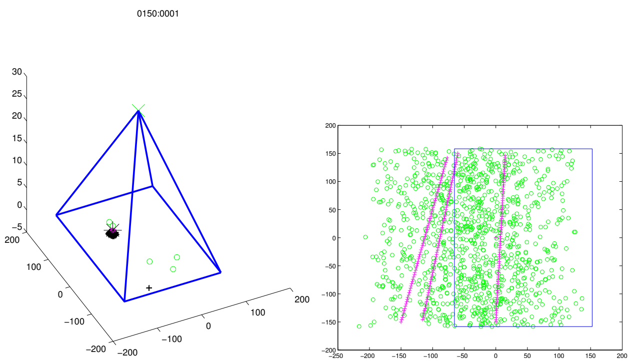

Fig. 3. The scenario. The 3D plot shows the configuration of the scenario. The blue tetrahedron is the 3D view frustum of the camera. The target location is shown as the purple circle, and clutter are hollow circles. The second shows the measurements overlaid with clutter, projected into the ground plane. The track of the 3 targets is visible. The view frustum of the camera is the blue rectangle.

B. Algorithms Tested

To compare the performance of the different algorithms, we implemented the following:

1) **Ideal.** This has full knowledge of the UAV's location, as if it were provided by an oracle. This shows the best possible results.
2) **Raw.** This uses the GPS and AHARS measurements directly in the Bernoulli filter as if they were ground truth. In this example we do not have correlated noise. Rather, this illustrates the effects of large magnitude noise on the filter.
3) **Decoupled.** This implements the decoupled filter.
4) **Decoupled MAP.** This takes the output from the UAV Kalman filter and treats it as an "ideal" measurement, similar to the Raw algorithm.

For the Decoupled and Decoupled MAP filter, we use a standard linear Kalman filter for both position and attitude.[3] Since the filter used ground truth noise values from the simulator, empirical tuning was not required.

Two sets of results were collected — the computed vs. the ground truth probability that a target exists, and the OSPA metric [24] using the settings $c = 25$ and $p = 1$. A target was declared to exist if its probability of existence is at least 0.5.

All results were computed for 100 Monte Carlo runs.

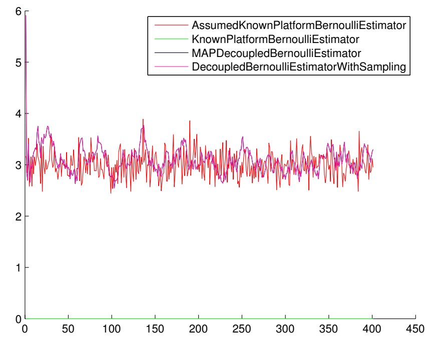

Fig. 4. The normalised estimation error squared for the estimated state of the platform.

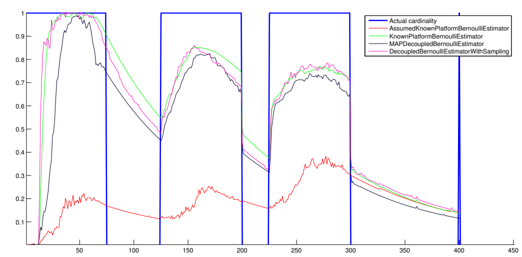

Fig. 5. The probability existence of the target.

C. Results

Figure 4 shows the Normalised Estimation Error Squared (NEES) for the position and orientation states for the platform estimated by each filter. From the properties of NEES, if a filter operates in a covariance consistent manner, the mean value of the NEES should be the same as the dimension of the state. For all the algorithms illustrated here, this is the case. Therefore, in all algorithms the platform is estimated in a covariance consistent manner.

Figure 5 shows the estimated probability of existence of the target for each filter. As can be seen, the expected probability of existence of a target rises when a target is present. However, the Raw filter has a much lower cardinality. The reason is that the large errors in the platform movement cause large apparent movements in the target relative to the platform. As a result, the movement appears to be far more similar to noise. The Decoupled and DecoupledMAP filters, however, use a filtered version of the platform pose. Even though this contains temporal correlations from the Kalman filter, the overall magnitude of the platform noise is greatly reduced. As a result, the filter is able to greatly improve the quality of the estimates. Figure 6 plots the results for the filters. Given the uniformly low value, it shows that the localisation error in

[3] This is acceptable in this application because both the errors and magnitudes of the angles are very small.

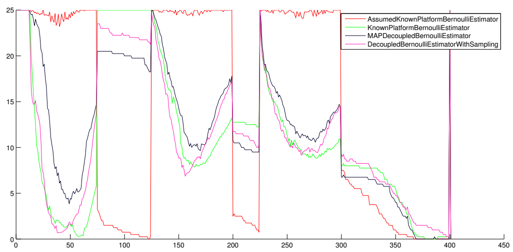

Fig. 6. The OSPA metric for the different filters.

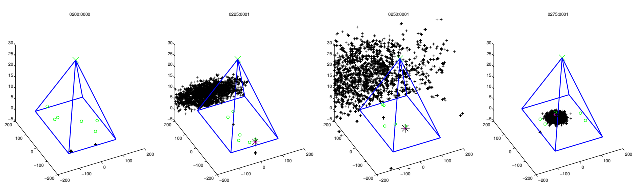

Fig. 7. Frames from the scenario 25 time steps apart. The black particles are from the decoupled filter. False tracks, created by clutter, can exit the detection region of the sensor for a prolonged period of time.

the target is small.

However, all the filters exhibit two problems: first, they can be slow in initializing the location of a target. Second, the probability of target existence is relatively high even when a target is not present. These two results can be explained by a combination of the (relatively high) clutter in the camera together with its limited detection region. Figure 7 shows a series of frames from the scenario. The particles from the decoupled filter are shown. The detection region is shown as a quadrilateral. However, clutter can generate targets which, due to the random walk associated with the target dynamics, can leave the clutter region. Because the probability of detection becomes zero, the weights on these particles only decline slowly due to the non-unit probability of survival. Although the filter could be tuned to reduce the magnitude of these effects, this is inherent for any system in which the detection region of the sensor is finite.

VI. Summary and Conclusions

In this paper, we have considered how to formulate the problem of tracking a single target from a platform. We have developed an extension of the Bernoulli filter to handle the dependency upon the platform. We have also developed a sub-optimal strategy for implementing the combined algorithm, which cascades a Bernoulli filter onto a platform localization algorithm. Our experimental results show that the filter can successfully track the target, when present, with a high degree of accuracy. However, the results also show that the limited field of view of the sensor leads to a number of problems in which clutter can lead to false tracks which, when they fall outside of the field of view of the sensor, can persist for long periods of time.

Our future work will concentrate on two main themes. First, we will apply the formulation to the multi-Bernoulli algorithm to support multiple target tracking. Second, we will investigate more efficient ways to manage occlusion and limited detection regions of sensors.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) (Grant number EP/H011990/1) and the MOD University Defence Research Centre on Signal Processing (UDRC).

Appendix A
Predicted Probability of Target Existence

In this appendix, we prove (18). From (15), we only need to consider $f_{k|k-1}(\overset{*}{\mathbf{x}}, \emptyset|\overline{\mathbf{Z}}_{1:k-1})$. Substituting from (16) and (17),

$$\begin{aligned} f_{k|k-1}(\overset{*}{\mathbf{x}}, \emptyset|\overline{\mathbf{Z}}_{1:k-1}) = & \\ \int \overset{*}{\phi}_{k|k-1} \left(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{x}}'\right) \phi_{k|k-1}\left(\emptyset|\emptyset\right) f_{k-1|k-1}(\overset{*}{\mathbf{x}}', \emptyset|\overline{\mathbf{Z}}_{1:k-1}) \mathrm{d}\overset{*}{\mathbf{x}}' & \\ + \int\int \overset{*}{\phi}_{k|k-1} \left(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{x}}'\right) \phi_{k|k-1}\left(\emptyset|\{\mathbf{x}'\}\right) & \\ \times f_{k-1|k-1}(\overset{*}{\mathbf{x}}', \{\mathbf{x}'\}|\overline{\mathbf{Z}}_{1:k-1} \mathrm{d}\overset{*}{\mathbf{x}}') \mathrm{d}\mathbf{x}'. & \quad (33) \end{aligned}$$

Substituting from (6) and (14),

$$\begin{aligned} & f_{k|k-1}(\overset{*}{\mathbf{x}}, \emptyset|\overline{\mathbf{Z}}_{1:k-1}) = \quad (34) \\ & = (1-p_b)(1-r_{k-1|k-1}) \int \overset{*}{\phi}_{k|k-1} \left(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{x}}'\right) s_0(\overset{*}{\mathbf{x}}') \mathrm{d}\overset{*}{\mathbf{x}}' \\ & \quad + (1-p_s) \int\int \overset{*}{\phi}_{k|k-1} \left(\overset{*}{\mathbf{x}}|\overset{*}{\mathbf{x}}'\right) r_{k-1|k-1} s_1(\overset{*}{\mathbf{x}}', \mathbf{x}') \mathrm{d}\overset{*}{\mathbf{x}}' \mathrm{d}\mathbf{x}' \\ & = (1-p_b)(1-r_{k-1|k-1}) s_0'(\overset{*}{\mathbf{x}}) + (1-p_s) r_{k-1|k-1} s_1'(\overset{*}{\mathbf{x}}) \quad (35) \end{aligned}$$

where $s_0'(\overset{*}{\mathbf{x}})$ and $s_1'(\overset{*}{\mathbf{x}})$ are spatial probability distributions of the platform conditioned on the cases that the target does not or does exist. Finally, from (15),

$$\begin{aligned} r_{k|k-1} &= 1 - \int f_{k|k-1}(\overset{*}{\mathbf{x}}, \emptyset|\overline{\mathbf{Z}}_{1:k-1}) \mathrm{d}\overset{*}{\mathbf{x}} \\ &= 1 - (1-p_b)(1-r_{k-1|k-1}) \int s_0'(\overset{*}{\mathbf{x}}) \mathrm{d}\overset{*}{\mathbf{x}} \\ &\quad - (1-p_s) r_{k-1|k-1} \int (1-p_s) s_1'(\overset{*}{\mathbf{x}}) \mathrm{d}\overset{*}{\mathbf{x}} \\ &= 1 - (1-p_b)(1-r_{k-1|k-1}) - (1-p_s) r_{k-1|k-1} \\ &= p_b(1-r_{k-1|k-1}) + p_s r_{k-1|k-1}. \end{aligned}$$

Appendix B
Probability of Existence with Target Observations

In this appendix, we show that (20), in the case of a target observation, has the Bernoulli-type form in (29), the normalization constant can be written as

$$f_k(\mathbf{Z}_k|\overline{\mathbf{Z}}_{1:k-1}) = (1-r_{k|k-1}) \int \varphi\left(\mathbf{Z}_k|\overset{*}{\mathbf{x}}_k, \emptyset\right) s_0(\overset{*}{\mathbf{x}}_k)\mathrm{d}\overset{*}{\mathbf{x}}_k + r_{k|k-1} \int\int \varphi\left(\mathbf{Z}_k|\overset{*}{\mathbf{x}}_k, \mathbf{x}_k\right) s_1(\overset{*}{\mathbf{x}}_k, \mathbf{x}_k)\mathrm{d}\mathbf{x}_k\mathrm{d}\overset{*}{\mathbf{x}}_k. \tag{36}$$

Using the observation likelihood terms (26) and (27), define the quantity

$$\Delta_k = \int\int \left[\left(\sum_{\mathbf{z}\in\mathbf{Z}} \frac{g(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})}\right) - 1\right] \times p_D(\overset{*}{\mathbf{x}}_k, \mathbf{x}_k) s_1(\overset{*}{\mathbf{x}}_k, \mathbf{x}_k)\mathrm{d}\mathbf{x}_k\mathrm{d}\overset{*}{\mathbf{x}}_k \tag{37}$$

Therefore,

$$f_k(\mathbf{Z}_k|\overline{\mathbf{Z}}_{1:k-1}) = \kappa(\mathbf{Z}_k)(1 - r_{k|k-1}\Delta_k). \tag{38}$$

Furthermore,

$$\int \varphi\left(\mathbf{Z}_k|\overset{*}{\mathbf{x}}_k, \emptyset\right) \cdot f_{k|k-1}(\overset{*}{\mathbf{x}}_k, \emptyset|\overline{\mathbf{Z}}_{1:k-1})\mathrm{d}\overset{*}{\mathbf{x}}_k = (1-r_{k|k-1})\kappa(\mathbf{Z}_k). \tag{39}$$

Substituting into (20),

$$1 - r_{k|k-1} = \frac{\kappa(\mathbf{Z}_k)(1-r_{k|k-1})}{\kappa(\mathbf{Z}_k)(1-r_{k|k-1}\Delta_k)}. \tag{40}$$

Rearranging gives (29).

References

[1] R. de Nardi, "The QRSim Quadrotor Simulator," Department of Computer Science, University College London, Gower Street, London UK, Tech. Rep. RN/13/08, March 2013.

[2] A. Gning, W. T. L. Teacy, R. V. Pelapur, H. AliAkbarpour, K. Palaniappan, G. Seetharaman, and S. J. Julier, "The Effect of State Dependent Probability of Detection in Multitarget Tracking Applications," in *Proceedings of SPIE*, ser. Geospatial InfoFusion and Video Analytics IV; and Motion Imagery for ISR and Situational Awareness II, no. 9089, Baltimore, MD, USA, 5th May 2014, p. 9089 05. [Online]. Available: http://spie.org/Publications/Proceedings/Paper/10.1117/12.2053968

[3] S. S. Blackman, "Multiple Hypothesis Tracking for Multiple Target Tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, January 2004. [Online]. Available: http://ieeexplore.ieee.org?arnumber=1263228

[4] R. P. S. Mahler, *Statistical Multisource–Multitarget Information Fusion*. Artech House, 2007.

[5] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, "A Tutorial on Bernoulli Filters: Theory, Implementation and Applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3406–3430, July 2013. [Online]. Available: http://ieeexplore.ieee.org?arnumber=6497685

[6] R. P. S. Mahler, "Multitarget Bayes Filtering via First-Order Multitarget Moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, October 2003. [Online]. Available: http://ieeexplore.ieee.org?arnumber=1261119

[7] ——, "PHD Filters of Higher Order in Target Number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, October 2007. [Online]. Available: http://ieeexplore.ieee.org?arnumber=4441756

[8] B. Vo, B. Vo, and A. Cantoni, "The Cardinality Balanced Multi-Target Multi-Bernoulli Filter and its Implementations," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409–423, February 2009. [Online]. Available: http://ieeexplore.ieee.org?arnumber=466392

[9] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The Labeled Multi-Bernoulli Filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, May 2014. [Online]. Available: http://ieeexplore.ieee.org?arnumber=6814305

[10] R. Mahler and A. El-Fallah, "Bayesian Unified Registration and Tracking," in *Proceedings of SPIE*, ser. Signal Processing, Sensor Fusion, and Target Recognition XX, no. 8050, May 2011. [Online]. Available: http://spie.org/Publications/Proceedings/Paper/10.1117/12.885145

[11] C. G. Hempell, "The Effect of Sensor Registration Error on the Performance of PMHT," in *Proceedings of the 11th International Conference on Information Fusion (FUSION), 2008*, Cologne, Germany, 30th June – 30th July 2008, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org?arnumber=4632453

[12] A. B. Poore and S. Gadaleta, "Some Assignment Problems Arising from Multiple Target Tracking," *Mathematical and Computer Modelling*, vol. 43, no. 9–10, pp. 1074–1091, May 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S089571770500525X

[13] B. D. Kragel, S. Danford, S. M. Herman, and A. B. Poore, "Bias Estimation Using Targets of Opportunity," in *Proceedings of SPIE*, ser. Signal and Data Processing of Small Targets, vol. 6699, 2007.

[14] M. Üney, D. E. Clark, and S. J. Julier, "Distributed Sensor Registration Based on Random Finite Set Representations," in *Sensor Signal Processing for Defence (SSPD 2012)*, London, UK, 25–27 September 2012, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org?arnumber=6552163

[15] B. Ristic, D. E. Clark, and N. Gordon, "Calibration of Multi-Target Tracking Algorithms Using Non-Cooperative Targets," *IEEE Journal of Selected Topics in Information Processing*, vol. 7, no. 3, pp. 390–398, June 2013. [Online]. Available: http://ieeexplore.ieee.org?arnumber=6494261

[16] X. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Multisensor Multitarget Bias Estimation for General Asynchronous Sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 899–921, July 2005. [Online]. Available: http://ieeexplore.ieee.org?arnumber=1541438

[17] W. T. L. Teacy, S. J. Julier, R. de Nardi, A. Rogers, and N. R. Jennings, "Observation Modelling for Vision-Based Target Search by Unmanned Aerial Vehicles," in *4th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Istanbul, Turkey, 4–8 May 2015.

[18] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A Random-Finite-Set Approach to Bayesian SLAM," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 268–282, February 2011. [Online]. Available: http://ieeexplore.ieee.org?arnumber=5710428

[19] C. S. Lee, D. E. Clark, and J. Salvi, "SLAM with Dynamic Targets via Single-Cluster PHD Filtering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 543–552, March 2013. [Online]. Available: http://ieeexplore.ieee.org?arnumber=6475147

[20] D. Moratuwage, B.-N. Vo, and D. Wang, "Collaborative Multi-Vehicle SLAM with Moving Object Tracking," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 6–10 May 2013, pp. 5702–5708. [Online]. Available: http://ieeexplore.ieee.org?arnumber=6631397

[21] H. Deusch, S. Reuter, and K. Dietmayer, "The Labeled Multi-Bernoulli SLAM Filter," *IEEE Signal P*, vol. 22, no. 10, pp. 1561–1565, October 2015. [Online]. Available: http://ieeexplore.ieee.org?arnumber=7062004

[22] S. F. Schmidt, "Applications of State-Space Methods to Navigation Problems," in *Advances in Control Systems*. Academic Press, 1966, vol. 3, pp. 293–340.

[23] A. Symington, R. D. Nardi, S. J. Julier, and S. Hai, "Simulating Quadrotor UAVs in Outdoor Scenarios," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, Chicago, IL, USA, 14–18 September 2014, pp. 3382–3388.

[24] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, August 2008. [Online]. Available: http://ieeexplore.ieee.org?arnumber=4567674