December 2024 Volume 19 Number 2 ISSN 1557-6418

JAIF24

Journal of Advances in Information Fusion

A semi-annual archival publication of the International Society of Information Fusion

Regular Papers Page

ISIF

INTERNATIONAL SOCIETY OF INFORMATION FUSION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied INFORMATION FUSION technologies. Technical areas of interest include target tracking, detection theory, applications for information fusion methods, image fusion, fusion systems architectures and management issues, classification, learning, data mining, Bayesian and reasoning methods.

Journal of Advances in Information Fusion (ISSN 1557-6418) is published semi-annually by the International Society of Information Fusion. The responsibility for the contents rests upon the authors and not upon ISIF, the Society, or its members. ISIF is a California Nonprofit Public Benefit Corporation at P.O. Box 4631, Mountain View, California 94040.

On the Observability of General Nonlinear Gaussian State Space Models Using Discrete Distributional Approximations

ARIANE HANEBECK
CLAUDIA CZADO

We consider arbitrary nonlinear stochastic discrete-time state space models (SSMs) with time-invariant parameters and nonadditive Gaussian disturbances. Given an observation trajectory, the goal is to obtain an estimate of the augmented state (the underlying state trajectory and the time-invariant parameters of the model). A numerical approach to checking this type of observability is given, and a quantitative assessment of the degree of observability is provided. In general, no global statements for all observation trajectories can be made on the observability of nonlinear SSMs. However, we can find regions of the state-observation space (consisting of all possible observation trajectories and corresponding state trajectories) in which an estimate of the augmented state can be obtained. This is achieved by approximating the continuous distribution of observation trajectories and state trajectories using an optimal discrete distribution. The associated locations of the point masses are called design values. For these design values, we can then check whether the augmented state can be recovered. We could also use random realizations of the observation trajectory. However, when using design values, a smaller number of considered observation trajectories is required to achieve a good coverage of the space. We illustrate our approach to checking observability for different specifications of discrete-time SSMs in univariate and multivariate settings.

Manuscript received October 9, 2023; revised July 8, 2024; released for publication January 10, 2025

Refereeing of this contribution was handled by Paolo Braca.

A. Hanebeck is with the Applied Mathematical Statistics, Technische Universität München, 85748 Garching, Germany (e-mail: Ariane.Hanebeck@tum.de).

C. Czado is with the Applied Mathematical Statistics and Munich Data Science Institute (MDSI), Technische Universität München, 85748 Garching, Germany (e-mail: czado@tum.de).

Comment: Preliminary considerations of this paper have been presented at the 25th International Conference on Information Fusion (FUSION 2022) in Linköping in [18]. The present paper expands the method from Gaussian models (which express Gaussian random variables as a function of desired parameters and a Gaussian disturbance term) to general nonlinear Gaussian state space models. The algorithm uses the distributional approximations presented in [32].

I. INTRODUCTION

In time series analysis, state space models (SSMs) are a common tool to model the dynamic behavior of observed multivariate data $\mathbf{Z}_t, t = 1, \ldots, T$, using multivariate latent states $\mathbf{X}_t, t = 0, \ldots, T$. As in [12], we consider arbitrary nonlinear stochastic discrete-time SSMs with unknown time-invariant parameters $\boldsymbol{\Omega}$ and nonadditive Gaussian disturbances $\boldsymbol{\epsilon}_t, \boldsymbol{\eta}_t, t = 1, \ldots, T$, of the form

$$\begin{aligned} &\text{Observation equation: } \mathbf{Z}_t = g(\boldsymbol{\Omega}, \mathbf{X}_t, \boldsymbol{\epsilon}_t) \\ &\text{State equation: } \mathbf{X}_t = h(\boldsymbol{\Omega}, \mathbf{X}_{t-1}, \boldsymbol{\eta}_t) \text{ for } t = 1, \ldots, T. \end{aligned} \tag{1}$$

More details on this model can be found in Section II.A.

For a given observation trajectory $\mathbf{z}_1, \ldots, \mathbf{z}_T$, the goal is to obtain an estimate of the augmented state $\boldsymbol{\Theta}$ consisting of initial state $\mathbf{x}_0$, state trajectory $\mathbf{x}_1, \ldots, \mathbf{x}_T$, and parameters $\boldsymbol{\Omega}$. Simulations show that the joint maximum a posteriori (MAP) estimator is biased in this situation. Instead, we choose the marginal MAP estimator to recover the augmented state. To avoid high-dimensional integration for the marginalization, a Bayesian Markov chain Monte Carlo (MCMC) setup is chosen.

In general, no global statements for all observation trajectories can be made on the observability of nonlinear SSMs [11], [13], [24]. This means that an estimate of the augmented state cannot necessarily be recovered for all observation trajectories (see [18, p. 2] for an intuitive example). We call a given SSM observable if the marginal MAP estimator exists, i.e., if the marginal MAP estimate can be recovered for at least one realization of the observation trajectory. Section II.B gives the exact theoretical definition of observability.

Our contribution is a novel approach to checking this type of observability in practice. As no global statements for all observation trajectories can be made, all possible realizations of the observation trajectory and corresponding state trajectory have to be analyzed. This is in general not possible, which is why we only consider a set of appropriately selected disturbance realizations called design disturbances. For obtaining design disturbances, we approximate the distribution of the Gaussian disturbances by a discrete distribution. For that, the approach proposed in [32] is used, which minimizes a distance measure similar to Cramér-von Mises between the continuous and the discrete distribution. The locations of the point masses of this discrete distribution are called the design disturbances.

Since observability properties depend on the parameters $\boldsymbol{\Omega}$, the initial state $\mathbf{x}_0$, and the number of time steps T, observability is always considered for fixed values of $\boldsymbol{\Omega}$, $\mathbf{x}_0$, and T. Each design disturbance leads to a corresponding design state trajectory and observation trajectory by plugging it into the corresponding model. The detailed construction of these design values is given in Section III.A.

The design disturbances are created to cover the space of disturbances homogeneously. Our simulations

show that they allow us to choose a smaller number of considered observation trajectories compared to using random values to obtain a satisfactory coverage of the space. Furthermore, they are determined offline before checking for observability and hence do not lead to additional computational effort.

For all chosen design observation trajectories, we execute two tasks. The first task is used to obtain information about the type of extrema of the posterior density (see Section III.B). In the second task, we then check whether the augmented state can be recovered by considering the marginal MAP estimate of the posterior density (see Section III.C). This tells us in which regions of the state-observation space the marginal MAP estimator is defined for the fixed time-invariant parameters $\mathbf{\Omega}$, initial state $\mathbf{x}_0$, and T.

There are many applications of SSMs in the literature. For example, [6] uses SSMs to describe flight control systems. In [36], the models are used to facilitate biological and biomedical modeling, while [17] use SSMs to analyze epidemiological data. Furthermore, [22] and [23] propose copula-based SSMs to capture the time dynamics in air pollution. Regarding observability, several definitions and approaches to checking it are developed. One approach is to use Kalman's observability matrix [3], [17], [20]. However, this approach is only used for linear SSMs. A definition for nonlinear observability based on the so-called indistinguishability is given in [20]. The models that are considered allow for deterministic input only in the state equation, which is described by a differential equation. The suggested approach to check this kind of observability is an extension of Kalman's observability matrix using Lie derivatives. An algebraic approach for nonlinear polynomial SSMs of this type is proposed in [15]. In [28], it is noted that the existing approaches checking for observability cannot identify for which observation trajectories the underlying parameters can be obtained. For models with uncertainties, which are restricted to bounded intervals, [28] solve this problem by using interval analysis. An overview of different observability definitions found in the literature is given in [37] and more recently [3]. In contrast to our approach, the literature imposes restrictions on the considered models, such as linearity or a specific form. Furthermore, only a deterministic input is allowed, in most cases only in the observation equation.

The remainder of the paper is organized as follows: In Section II, we introduce the type of model we consider and the theoretical definition of observability based on the existence of the marginal MAP estimator. Section III describes our proposal to check this definition of observability in practice using design observation trajectories. In Section IV, we illustrate how this approach can be used to decide upon observability of different examples of Gaussian SSMs. Section V concludes the paper and gives an outlook for possible adaptions and for future work to extend the approach to non-Gaussian SSMs. In Section VI, an overview over the most important definitions is given.

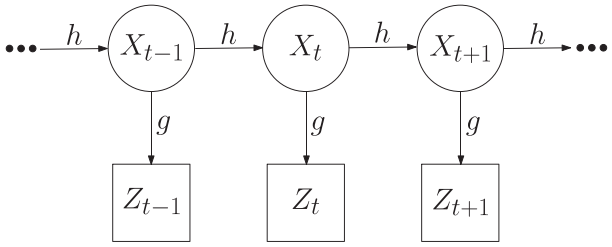


Figure 1. A general SSM for $d = p = 1$.

II. SSMS AND OBSERVABILITY NOTATIONS

A. Stochastic SSMs

We describe the model (1) in more detail now.

1) Definitions: The random variable $\mathbf{Z}_t \in \mathbb{R}^d$ denotes the observation at time t, while $\mathbf{X}_t \in \mathbb{R}^p$ represents the (latent) state at time t. This latent state is an underlying variable driving the behavior of the observations. The state equation describes the evolution of the unobserved latent state over time depending on the random disturbances $\boldsymbol{\eta}_t, t = 1, \ldots, T$, while the observation equation describes how the observation $\mathbf{Z}_t$ at time t is defined, given the latent state $\mathbf{X}_t$ and the random disturbance $\boldsymbol{\epsilon}_t$ at time t. We define the random observation trajectory from time point 1 to T by $\mathcal{Z}_T = (\mathbf{Z}_1^\top, \ldots, \mathbf{Z}_T^\top)^\top \in \mathbb{R}^{T \cdot d}$ with realization $z_T = (\mathbf{z}_1^\top, \ldots, \mathbf{z}_T^\top)^\top$. The random trajectory of the latent state is defined by $\mathcal{X}_T = (\mathbf{X}_1^\top, \ldots, \mathbf{X}_T^\top)^\top \in \mathbb{R}^{T \cdot p}$ with realization $\chi_T = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_T^\top)^\top$. Figure 1 illustrates an SSM for $d = p = 1$.

We assume the initial value $\mathbf{x}_0$ to be fixed and unknown and denote by $\mathbf{x}_0^{true}$ the true underlying value. Furthermore, $\mathbf{\Omega}$ is a vector of unknown time-invariant parameters, where $\mathbf{\Omega}^{true}$ is the true underlying value of $\mathbf{\Omega}$.

For the disturbances $(\boldsymbol{\epsilon}_t)_{t=1,\ldots,T}$ and $(\boldsymbol{\eta}_t)_{t=1,\ldots,T}$, we assume that they are serially independent and independent of each other at all time points [12, Ch. 9] with $\boldsymbol{\epsilon}_t \sim \mathcal{N}_d(\mathbf{0}, R_t)$ and $\boldsymbol{\eta}_t \sim \mathcal{N}_p(\mathbf{0}, Q_t)$, where $R_t \in \mathbb{R}^{d \times d}$ and $Q_t \in \mathbb{R}^{p \times p}$ are known covariance matrices.

2) Underlying Parameters: Consider an SSM of the form (1) with fixed true underlying values $\mathbf{\Omega}^{true}$ and $\mathbf{x}_0^{true}$. Given an observation trajectory z_T from that model, the underlying unknown true parameters are $\mathbf{\Theta}^{true} = (\mathbf{\Omega}^{true}, \mathbf{x}_0^{true}, \chi_T)$. The values $\mathbf{\Omega}^{true}$ and $\mathbf{x}_0^{true}$ always stay the same, but the trajectory χ_T of the state is changing with every realization of the disturbances $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\eta}_t, t = 1, \ldots, T$, and hence every observation trajectory z_T. Consequently, there is not one common true underlying value of χ_T that could be compared for different z_T's. Thus, we treat χ_T as a nuisance parameter vector, and the parameters of interest are $\mathbf{\Omega}$ and $\mathbf{x}_0$. Following [5] and [25], the vector $\mathbf{\Theta} = (\mathbf{\Omega}, \mathbf{x}_0, \chi_T)$ of unknown

parameters is called the *augmented state,* consisting of the time-invariant parameters, the initial state, and the state trajectory. Even though we primarily consider the parameters $\mathbf{\Omega}$ and $\mathbf{x}_0$ in our proposed approach, we are also interested in χ_T in applications to allow for predictive simulation.

For better readability, we write $\mathbf{\Theta} = (\mathbf{\Omega}, \mathbf{x}_0, \chi_T)$ instead of $\mathbf{\Theta} = (\mathbf{\Omega}^\top, \mathbf{x}_0^\top, \chi_T^\top)^\top$. This notation will be used throughout the paper, e.g, also for z_T, $\mathcal{Z}_T$, χ_T, and $\mathcal{X}_T$.

3) Posterior Density: The joint posterior density of (1) is defined by

$$\Pi(\mathbf{\Theta}|z_T) \propto \ell\left(\mathbf{\Theta}|z_T\right) \cdot p(\mathbf{\Theta}) = \left[\prod_{t=1}^{T} f_z\left(\mathbf{z}_t|\mathbf{x}_t, \mathbf{\Omega}\right)\right] \cdot \left[\prod_{t=1}^{T} p_x\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{\Omega}\right)\right] \cdot p_0(\mathbf{x}_0) \cdot p_{\mathbf{\Omega}}(\mathbf{\Omega}). \tag{2}$$

The corresponding log-posterior density is defined by $\pi(\mathbf{\Theta}|z_T) = \log(\Pi(\mathbf{\Theta}|z_T))$. Here, $\ell\left(\mathbf{\Theta}|z_T\right)$ is the likelihood of $\mathbf{\Theta}$ given z_T, $p(\mathbf{\Theta})$ is the prior of the parameter vector $\mathbf{\Theta}$, p_0 is the prior for $\mathbf{x}_0$, $\prod_{t=1}^{T} p_x\left(\cdot|\mathbf{x}_{t-1}, \mathbf{\Omega}\right)$ is the prior for χ_T given $\mathbf{x}_0$, and $p_{\mathbf{\Omega}}$ is the prior for $\mathbf{\Omega}$. The function $f_z\left(\mathbf{z}_t|\mathbf{x}_t, \mathbf{\Omega}\right)$ is the density of $\mathbf{Z}_T|\{\mathbf{X}_t = \mathbf{x}_t, \mathbf{\Omega}\}$ defined by the observation equation.

The prior p_0 might depend on the true underlying value of $\mathbf{x}_0$ if information about it is available. The prior $\prod_{t=1}^{T} p_x\left(\cdot|\mathbf{x}_{t-1}, \mathbf{\Omega}\right)$ of χ_T might either just be the product of the densities $f_x\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{\Omega}\right)$ of $\mathbf{X}_t|\{\mathbf{X}_{t-1} = \mathbf{x}_{t-1}, \mathbf{\Omega}\}$ defined by the state equation, i.e., $p_x = f_x$, or more information might be available on χ_T that can be incorporated. For example, the support of f_x could be restricted to obtain p_x.

Example 1. *If the true underlying value Ω_i^{true} of a time-invariant parameter component $\Omega_i \in (-1, 1)$ of $\mathbf{\Omega}$ is known, we define the restricted prior*

$$p^{res}_{\Omega_i^{true}}(\Omega_i) = \begin{cases} U(0,1) & \text{for } \Omega_i^{true} > 0, \\ U(-1,0) & \text{for } \Omega_i^{true} < 0. \end{cases} \tag{3}$$

If Ω_i^{true} is not known, prior expert knowledge might be available on the sign of Ω_i.

4) Models Linear in State: One type of SSM, which is often studied, is *linear in the states* $\mathbf{X}_t$ and is given by

$$\begin{aligned} \mathbf{Z}_t &= A(\mathbf{\Omega}) \cdot \mathbf{X}_t + B(\mathbf{\Omega}) \cdot \boldsymbol{\epsilon}_t\,, \\ \mathbf{X}_t &= C(\mathbf{\Omega}) \cdot \mathbf{X}_{t-1} + D(\mathbf{\Omega}) \cdot \boldsymbol{\eta}_t \end{aligned} \tag{4}$$

for $t = 1, \ldots, T$, and $A(\mathbf{\Omega}) \in \mathbb{R}^{d\times p}$, $B(\mathbf{\Omega}) \in \mathbb{R}^{d\times d}$, $C(\mathbf{\Omega}) \in \mathbb{R}^{p\times p}$, and $D(\mathbf{\Omega}) \in \mathbb{R}^{p\times p}$ for unknown $\mathbf{\Omega}$. Note that the matrices $A(\mathbf{\Omega})$, etc, do not need to be linear in the parameters $\mathbf{\Omega}$ in order for the model to be linear in the states. The disturbance distributions are defined as before.

We introduce one example of this type now.

Example 2. *The model we consider with $d = p = 1$ is of the form*

$$\begin{aligned} &Z_t = aX_t + \sqrt{1-a^2} \cdot \epsilon_t\,,\ X_t = aX_{t-1} + \sqrt{1-a^2} \cdot \eta_t\,, \\ &t = 1, \ldots, T, \end{aligned} \tag{5}$$

with $\epsilon_t \sim \mathcal{N}(0,1)$ i.i.d. and $\eta_t \sim \mathcal{N}(0,1)$ i.i.d. independent. The parameter $\Omega = a \in (-1,1)$ is unknown, so the parameters of interest are a and x_0 with unknown nuisance parameters $x_1, \ldots, x_T$. Using $Z_t|\{X_t = x_t, a\} \sim \mathcal{N}(ax_t, 1-a^2)$, $X_t|\{X_{t-1} = x_{t-1}, a\} \sim \mathcal{N}(ax_{t-1}, 1-a^2)$, we determine the posterior for $\mathbf{\Theta} = (a, x_0, \chi_T)$ as

$$\Pi(\mathbf{\Theta}|z_T) \propto \left[\prod_{t=1}^{T} \frac{1}{\sqrt{1-a^2}} \cdot e^{-\frac{(z_t - ax_t)^2}{2(1-a^2)}}\right] \cdot \left[\prod_{t=1}^{T} \frac{1}{\sqrt{1-a^2}} \cdot e^{-\frac{(x_t - ax_{t-1})^2}{2(1-a^2)}}\right] \cdot p^{res}_{a^{true}}(a). \tag{6}$$

We use a noninformative prior for x_0. For known a^{true}, the prior of a is chosen by $p^{res}_{a^{true}}(a)$, as defined in (3). For small values of $|a|$, this restriction is necessary as otherwise the posterior becomes bimodal. For larger values of $|a|$, the restriction might not always be necessary. Furthermore, in general, the integral over all parameters $a \in (-1,1), x_t \in (-\infty, \infty), t = 0, \ldots, T$, of (6) with uniform prior on $(-1,1)$ for a might not exist.

The log-posterior is of the form

$$\pi(\mathbf{\Theta}|z_T) = -\frac{1}{2}\frac{1}{1-a^2}\left[\sum_{t=1}^{T}(z_t - ax_t)^2 + \sum_{t=1}^{T}(x_t - ax_{t-1})^2\right] - T\log(1-a^2) + \log(p^{res}_{a^{true}}(a)).$$

B. Theoretical Definition of Observability

For different values of $\mathbf{\Omega}^{true}$, $\mathbf{x}_0^{true}$, and T, the observability properties of a given SSM can vary [20]. For this reason, observability is always investigated for a fixed initial value $\mathbf{x}_0^{true}$ and fixed time-invariant parameters $\mathbf{\Omega}^{true}$. Hence, if certain values of the parameters $(\mathbf{\Omega}, \mathbf{x}_0)$ are of special interest, they can be studied in detail. Additionally, different numbers of time steps T can be considered.

To define our notion of observability, we first consider a fixed but arbitrary observation trajectory z_T with corresponding state trajectory χ_T of model (1) using $\mathbf{x}_0^{true}$ and $\mathbf{\Omega}^{true}$, i.e.,

$$\begin{aligned} \mathbf{Z}_t &= g(\mathbf{\Omega}^{true}, \mathbf{X}_t, \boldsymbol{\epsilon}_t) \text{ for } t = 1, \ldots, T, \\ \mathbf{X}_t &= h(\mathbf{\Omega}^{true}, \mathbf{X}_{t-1}, \boldsymbol{\eta}_t) \text{ for } t = 2, \ldots, T, \\ \mathbf{X}_1 &= h(\mathbf{\Omega}^{true}, \mathbf{x}_0^{true}, \boldsymbol{\eta}_1) \end{aligned}$$

and study the posterior $\Pi(\mathbf{\Theta}|z_T)$ defined in (2) using priors, which are specified specifically for each model. Then, two tasks are carried out. In the first task, we obtain information about the type of extrema of $\Pi(\mathbf{\Theta}|z_T)$

by checking if the joint MAP estimate can be obtained. In the second task, we consider the marginal MAP estimate. The existence of the marginal MAP estimate in the second task depends on the type of extrema of the joint posterior, which is investigated in the first task.

1) Existence of Joint MAP Estimate: We want to investigate if the joint MAP estimate

$$\hat{\boldsymbol{\Theta}}(z_T) = (\hat{\boldsymbol{\Omega}}(z_T), \hat{\mathbf{x}}_0(z_T), \widehat{\chi}_T(z_T)) = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, \Pi(\boldsymbol{\Theta}|z_T) \tag{7}$$

exists, from which we obtain information about the type of extrema of the posterior. For this, we maximize the log-posterior π, i.e., $\hat{\boldsymbol{\Theta}}(z_T) = (\hat{\boldsymbol{\Omega}}(z_T), \hat{\mathbf{x}}_0(z_T), \widehat{\chi}_T(z_T)) = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, \pi(\boldsymbol{\Theta}|z_T)$.

Given an observation trajectory z_T, the ideal case is to have a unique maximum of the corresponding log-posterior $\pi(\cdot|z_T)$. However, there are three further cases to be considered.

To define the three cases, we first have to define the notion of a ridge. For a function $f : \mathbb{R}^N \to \mathbb{R}$, a ridge is a curve consisting of local maxima in $N-1$ dimensions. A point $\mathbf{u}_0 \in \mathbb{R}^N$ is such a local maximum if $f(\mathbf{u}) < f(\mathbf{u}_0)$ for $\mathbf{u} \in \mathcal{S} \subset \mathbb{R}^N$ with $\dim(\mathcal{S}) = N-1$. We define the set of local maxima on a ridge by $\mathcal{S}_R$ and say a ridge is of constant height if $f(\mathbf{u}_1) = f(\mathbf{u}_2)$ for all $\mathbf{u}_1 \neq \mathbf{u}_2 \in \mathcal{S}_R$ and of variable height otherwise. For an example of a variable height ridge, see Figure 4, which shows that a ridge is in general a nonlinear feature of the posterior density. Examples where ridges are encountered are given in [29] and [30].

The three cases are given as follows:

1. No maximum: In the case of no finite maximum, a maximum can only be found on the boundary, and it is not possible to recover the true augmented state. This might happen if we have a ridge of variable height with no local maxima within the boundaries.

2. Finitely many maxima: In the case of two or more distinct maxima with the same log-posterior value, no unique joint MAP estimate $\hat{\boldsymbol{\Theta}}(z_T)$ of $\boldsymbol{\Theta}$ given z_T can be recovered. However, it might be possible to find a unique maximum $\operatorname{argmax}_{\boldsymbol{\Theta}} \pi(\boldsymbol{\Theta}|z_T)$ when constraining the parameter space using restricting priors on the parameters. In general, the value of the posterior in the distinct local maxima is not the same. Then, a constraint of the parameter space around the global maximum is usually carried out. This also leads to a posterior with a unique maximum.

3. Infinitely many maxima: In the case where there is a ridge of constant height of the log-posterior, the function does not have a distinct maximum $\operatorname{argmax}_{\boldsymbol{\Theta}} \pi(\boldsymbol{\Theta}|z_T)$. In that case, it is not possible to recover the true augmented state. However, it might be possible to find estimates for functions of parameters.

Example 3 (Finitely many maxima). *Consider a slightly changed version of the model in (5) with $d = p = 1$ given by*

$$Z_t = aX_t + \sqrt{1-a^2} \cdot \epsilon_t\,,\ X_t = a^2 X_{t-1} + \sqrt{1-a^4} \cdot \eta_t\,,\ t = 1, \ldots, T,$$

with $\epsilon_t \sim \mathcal{N}(0,1)$ i.i.d. and $\eta_t \sim \mathcal{N}(0,1)$ i.i.d. independent for unknown $a \in (-1,1)$. Then, given an observation trajectory z_T with underlying parameter values $(a^{true}, x_0^{true}, \chi_T)$, the parameter values $(-a^{true}, -x_0^{true}, -\chi_T)$ lead to the same posterior value. Hence, we can use the prior $p^{res}_{a^{true}}(a)$ to constrain the parameter space and to find the unique underlying parameter values $(a^{true}, x_0^{true}, \chi_T)$.

Example 4 (No maximum). *For $\epsilon_t \sim \mathcal{N}(0,1)$, $\eta_t \sim \mathcal{N}(0,1)$ i.i.d. independent, consider*

$$Z_t = aX_t + \epsilon_t\,,\ X_t = X_{t-1} + \eta_t\,,\ t = 1, \ldots, T. \tag{8}$$

We are interested in $a \in (-1,1)$, the initial state x_0, and the latent state trajectory χ_T, given an observation trajectory z_T. The log-posterior is given in (13). For a given z_T, the posterior has a ridge of variable height for small values of T. More details are given in Section IV.

A model of the form (4) is often called a linear SSM. However, it is only linear in the states $\mathbf{X}_t$. We call an SSM *linear* if it is linear in both the states $\mathbf{X}_t$ and the time-invariant parameters $\boldsymbol{\Omega}$.

For a linear model and a given observation trajectory z_T, the derivatives $\frac{\mathrm{d}}{\mathrm{d}\Theta_i}\pi(\boldsymbol{\Theta}|z_T)$, $i = 1, \ldots, |\boldsymbol{\Theta}|$, of the log-posterior form a system of linear equations for the parameters in $\boldsymbol{\Theta}$. Hence, it is only possible to have none, one unique, or infinitely many maxima. The second case of having finitely many maxima cannot occur [11], [20].

Consider, for example, the model given in (8), which is of the form (4) and hence linear in the states $X_t, t = 0, \ldots, T$. If the parameter a is known and not of interest, it is also a linear model because it is linear in all unknown parameters $X_0, \ldots, X_T$. However, if a is of interest and $\Omega = a$, the derivative of the log-posterior in (13) with respect to a is $\sum_{t=1}^{T}(z_t x_t - a x_t^2)$, which is obviously not linear in x_t. Hence, this model is nonlinear and only linear in the states X_t.

2) Existence of Marginal MAP Estimate: This approach of considering the marginal MAP estimate is also used in [22], [23] to estimate the unknown parameters. The differences between joint and marginal MAP estimates are discussed in [14, Ch. 13] and [27].

Consider the marginal posterior densities

$$\begin{aligned}\Pi(\Omega_j|z_T) &= \int \Pi(\boldsymbol{\Omega}, \mathbf{x}_0, \chi_T|z_T)\,\mathrm{d}\boldsymbol{\Omega}_{-j}\,\mathrm{d}\mathbf{x}_0\,\mathrm{d}\chi_T,\\ &\qquad j = 1,\dots,|\boldsymbol{\Omega}|,\\ \Pi(x_{0,i}|z_T) &= \int \Pi(\boldsymbol{\Omega}, \mathbf{x}_0, \chi_T|z_T)\,\mathrm{d}\boldsymbol{\Omega}\,\mathrm{d}\mathbf{x}_{0,-i}\chi_T,\\ &\qquad i = 1,\dots,p,\\ \Pi(x_{t,i}|z_T) &= \int \Pi(\boldsymbol{\Omega}, \mathbf{x}_0, \chi_T|z_T)\,\mathrm{d}\boldsymbol{\Omega}\,\mathrm{d}\mathbf{x}_0\,\mathrm{d}\mathbf{x}_{t,-i}\,\mathrm{d}\chi_{T,-t},\\ &\qquad i = 1,\dots,p, t = 1,\dots,T,\end{aligned} \tag{9}$$

where $\mathbf{x}_t = (x_{t,1},\dots,x_{t,p})$, $\mathbf{x}_{t,-i} = (x_{t,1},\dots,x_{t,i-1},x_{t,i+1},\dots,x_{t,p}) \in \mathbb{R}^{p-1}$ for $t = 0,\dots,T$, $\boldsymbol{\Omega}_{-j} = \boldsymbol{\Omega} \setminus \{\Omega_j\} \in \mathbb{R}^{|\boldsymbol{\Omega}|-1}$, $\chi_{T,-t} = (\mathbf{x}_1,\dots,\mathbf{x}_{t-1},\mathbf{x}_{t+1},\dots,\mathbf{x}_T) \in \mathbb{R}^{p(T-1)}$.

We maximize the marginal posterior densities defined in (9) over Ω_j, $j = 1,\dots,|\boldsymbol{\Omega}|$, and $x_{t,i}, t = 0,\dots,T, i = 1,\dots,p$, respectively, and obtain $\hat{\Omega}_j(z_T)_{\mathrm{mar}} = \operatorname{argmax}_{\Omega_j}\Pi(\Omega_j|z_T)$ and $\hat{x}_{t,i}(z_T)_{\mathrm{mar}} = \operatorname{argmax}_{x_{t,i}}\Pi(x_{t,i}|z_T)$. The marginal MAP estimate is then denoted by $\hat{\boldsymbol{\Theta}}(z_T)_{\mathrm{mar}} = (\hat{\boldsymbol{\Omega}}(z_T)_{\mathrm{mar}}, \hat{\mathbf{x}}_0(z_T)_{\mathrm{mar}}, \widehat{\chi_T}(z_T)_{\mathrm{mar}})$. If one of the three cases of (1) no maximum, (2) finitely many maxima, or (3) infinitely many maxima occurs and the joint MAP estimate cannot be recovered, then it might also not be possible to recover the marginal MAP estimate. How to obtain the marginal MAP estimate in practice is discussed in Section III.C.

3) The Definition of Observability: Until now, we have considered a fixed but arbitrary observation trajectory z_T with underlying χ_T. Now, we define the notion of observability for a given SSM of the form (1).

Definition 1. *Define the* state-observation space *as the space of all possible realizations of the observation trajectory $\mathcal{Z}_T$ and corresponding state trajectory $\mathcal{X}_T$, i.e., for $\mathbf{x}_0^{true} \in \mathbb{R}$, $\boldsymbol{\Omega}^{true} \in \mathbb{R}^{|\boldsymbol{\Omega}|}$, define*

$$\begin{aligned}&SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) =\\ &\{\textit{Realizations } (\chi_T, z_T) \textit{ of } (\mathcal{X}_T, \mathcal{Z}_T):\\ &\quad \mathbf{Z}_t = g(\boldsymbol{\Omega}^{true}, \mathbf{X}_t, \boldsymbol{\epsilon}_t), t = 1,\dots,T,\\ &\quad \mathbf{X}_t = h(\boldsymbol{\Omega}^{true}, \mathbf{X}_{t-1}, \boldsymbol{\eta}_t) \textit{ for } t = 2,\dots,T,\\ &\quad \mathbf{X}_1 = h(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, \boldsymbol{\eta}_t),\\ &\boldsymbol{\epsilon}_t \sim \mathcal{N}_d(0, R_t) \textit{ i.i.d., and } \boldsymbol{\eta}_t \sim \mathcal{N}_p(0, Q_t)\\ &\quad \textit{i.i.d. independent}\}.\end{aligned} \tag{10}$$

Remark 1. *The joint MAP estimator and the marginal MAP estimator of a nonlinear SSM are not necessarily defined for all elements of $SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$.*

The phenomenon of Remark 1 can also be found in the literature on SSMs, where the state equation is described by a differential equation, for which the concepts can be transferred to our type of model. In [24], it is stated that *the observability for nonlinear systems is, in general, not only a local property but also depends on the input of the system*. The authors of [13] talk about *bad inputs*, and the same arguments can be found in [11]. These "bad" inputs lead to observation trajectories for which the augmented state cannot be recovered. In the setup of this paper, they correspond to "bad" realizations of the disturbances, inducing realizations in $SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$, for which the augmented state cannot be recovered. This leads to the following definitions:

Definition 2 (Existence of the joint MAP estimator). *If the log-posterior $\pi(\cdot|z_T)$ does not have a distinct maximum for a given z_T, no unique joint MAP estimate $\hat{\boldsymbol{\Theta}}(z_T)$ of $\boldsymbol{\Theta}$ can be recovered. Let $\mathcal{E}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \subset SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \subset \mathbb{R}^{T(p+d)}$ be the set of realizations z_T and corresponding χ_T such that $\hat{\boldsymbol{\Theta}}(z_T) = \operatorname{argmax}_{\boldsymbol{\Theta}}\pi(\boldsymbol{\Theta}|z_T)$ can be uniquely recovered. If $\mathcal{E}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) = \emptyset$, the estimator $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)$ does not exist. If $\mathcal{E}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \neq \emptyset$, the estimator*

$$\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T) = \begin{cases}\hat{\boldsymbol{\Theta}}(z_T), & (\mathcal{X}_T, \mathcal{Z}_T) = (\chi_T, z_T)\\ & \quad \in \mathcal{E}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)\\ \textit{undefined}, & \textit{otherwise}\end{cases}$$

exists but is not necessarily defined for every realization $(\chi_T, z_T) \in SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$.

If the joint MAP estimator exists, it is often biased, and a Bayesian analysis for all parameters jointly is not the best approach [14, Ch. 13.4]. However, the existence of $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)$ and the type of extrema of the posterior provide information for the marginal MAP approach.

Definition 3 (Observability based on marginal MAP estimator). *Let $\mathcal{E}^{\mathrm{mar}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \subset SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$ be the set of realizations (χ_T, z_T) such that the marginal MAP estimate $\hat{\boldsymbol{\Theta}}(z_T)_{\mathrm{mar}}$ can be uniquely recovered. If $\mathcal{E}^{\mathrm{mar}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) = \emptyset$, $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\mathrm{mar}}$ does not exist, and we call the model unobservable. If $\mathcal{E}^{\mathrm{mar}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \neq \emptyset$, we call the model observable. Then,*

$$\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\mathrm{mar}} = \begin{cases}\hat{\boldsymbol{\Theta}}(z_T)_{\mathrm{mar}}, & (\mathcal{X}_T, \mathcal{Z}_T) = (\chi_T, z_T)\\ & \quad \in \mathcal{E}^{\mathrm{mar}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)\\ \textit{undefined}, & \textit{otherwise}\end{cases}$$

exists but is not necessarily defined for every $(\chi_T, z_T) \in SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$. If we constrain our parameter space to obtain a unique marginal MAP estimate (recall case 2. Finitely many maxima), we call our model locally observable.

Remark 2. *In Definition 3, we define a model to be locally observable if the parameter space has to be constrained due to finitely many maxima of the posterior density. This should not be confused with a different type of locality that is described in Remark 1, stating that the (marginal) MAP estimator is not necessarily defined for all realizations $(\chi_T, z_T) \in SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$.*

In the ideal case, the estimator $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\mathrm{mar}}$ is consistent for the parameters of interest, i.e.,

$(\hat{\boldsymbol{\Omega}}(\mathcal{Z}_T)_{\text{mar}}, \hat{\mathbf{x}}_0(\mathcal{Z}_T)_{\text{mar}}) \xrightarrow{P} (\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true})$ for $T \to \infty$. If not, a bias would be detected by the algorithm and could be taken into account.

For an observable model, we do not only want to answer the question of observability but also want to know how well it is observable. As measures, we consider the cardinality of $\mathcal{E}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$ and $\mathcal{E}^{\text{mar}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$, $\mathbb{E}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T))$, $\text{Var}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T))$, and $\text{MSE}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T))$ as well as $\mathbb{E}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\text{mar}})$, $\text{Var}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\text{mar}})$, and $\text{MSE}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\text{mar}})$. We introduce the local variance, which is also used to quantify the degree of observability: Given an observation trajectory z_T, we want to approximate the posterior $\Pi(\cdot|z_T)$ by a Gaussian density in $\boldsymbol{\Theta}$ at the joint MAP estimate $\hat{\tilde{\boldsymbol{\Theta}}}(z_T)$ if it exists, using a second-order Taylor polynomial. For given z_T, the local variances of the parameter estimates $\hat{\Theta}_j(z_T)$ are then defined by the variances of the fitted Gaussian density. Hence, they are given by $\text{LVar}(\hat{\Theta}_j(z_T)) := -\mathbf{H}_\pi^{-1}(\hat{\tilde{\boldsymbol{\Theta}}}(z_T))_{jj}$, $j = 1, \ldots, |\boldsymbol{\Theta}|$, where $\mathbf{H}_\pi(\cdot) \in \mathbb{R}^{|\boldsymbol{\Theta}|\times|\boldsymbol{\Theta}|}$ is the Hessian matrix of $\pi(\cdot|z_T)$. In [18], the local variance is derived in detail for the one-dimensional parameter case.

Hence, we now have two types of variances. The first one is the variance $\text{Var}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T))$ of the estimator $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)$. In contrast to $\text{Var}(\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T))$, the local variance considers how peaked the maximum of $\pi(\cdot|z_T)$ in the joint MAP estimate $\hat{\boldsymbol{\Theta}}(z_T)$ is, given one specific observation trajectory z_T. The maximum of $\pi(\cdot|z_T)$ is getting more peaked if the local variances are decreasing (see [18, Fig. 2]).

III. PROPOSAL TO CHECK OBSERVABILITY IN PRACTICE

From Remark 1 we know that the joint and marginal MAP estimators might not be defined everywhere. In particular, the existence of $\hat{\boldsymbol{\Theta}}(z_T)$ depends on the realization (χ_T, z_T) of $(\mathcal{X}_T, \mathcal{Z}_T)$, i.e., $\mathcal{E}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \neq SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$ (see Definition 2). The same holds for the marginal MAP estimator $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\text{mar}}$, i.e., $\mathcal{E}^{\text{mar}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \neq SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$ (see Definition 3). We want to know for which elements in $SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$ the estimators are defined if they exist. In general, the estimators cannot be determined analytically to see where they are defined. Hence, in theory, we have to try all realizations $(\chi_T, z_T) \in SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$ to check if a joint or marginal MAP estimate of $\pi(\cdot|z_T)$ can be recovered. That is not possible in practice as there are infinitely many elements in $SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$. Instead, for fixed values of $\boldsymbol{\Omega}^{true}$, $\mathbf{x}_0^{true}$, and T, we construct K design observation trajectories $\tilde{z}_T^{(k)} = \tilde{z}_T^{(k)}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true})$ with corresponding design state trajectory $\tilde{\chi}_T^{(k)} = \tilde{\chi}_T^{(k)}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true})$, $k = 1, \ldots, K$, approximately representing all possible elements in $SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$.

For this, the disturbance distribution is approximated by a discrete distribution and the corresponding locations of the point masses are used as designed disturbance realizations, from which we construct the design values $\tilde{z}_T^{(k)}$ and $\tilde{\chi}_T^{(k)}$. Then, given one design observation trajectory $\tilde{z}_T^{(k)}$, the goal is to check whether we can estimate the augmented state $(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, \tilde{\chi}_T^{(k)})$. We discuss the construction of these design observation trajectories in detail now.

A. Construction of the Design Observation Trajectories

The construction of the K design observations $\tilde{\mathbf{z}}_t^{(k)}$ and the corresponding design states $\tilde{\mathbf{x}}_t^{(k)}$, $t = 1, \ldots, T$, $k = 1, \ldots, K$, approximately representing $SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$, given $\boldsymbol{\Omega}^{true}$, $\mathbf{x}_0^{true}$, T, and K, will now be discussed in detail. We only consider $R_t = I_d$ and $Q_t = I_p$. For different covariance matrices, transformations or other approaches to approximate non-standard multivariate normal distributions could be used.

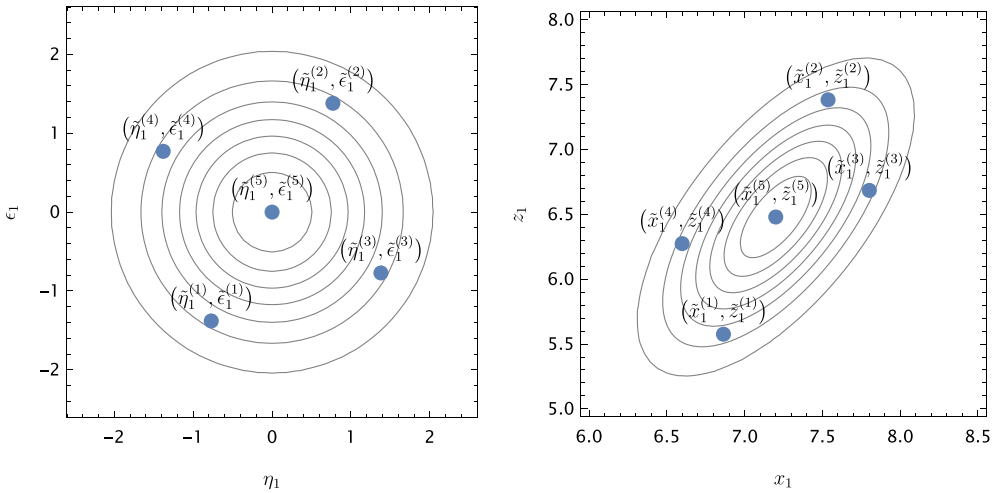


Figure 2. Left: Illustration of the design disturbance vectors $(\tilde{\eta}_1^{(k)}, \tilde{\epsilon}_1^{(k)})$ for $k = 1, \ldots, 5$, together with the contour plot of the standard normal density. Right: Illustration of the corresponding $(\tilde{x}_1^{(k)}, \tilde{z}_1^{(k)})$, $k = 1, \ldots, 5$, together with the contour plot of the theoretical density for $a^{true} = 0.9$, $x_0^{true} = 8$.

1) Case: $d = p = 1$: The parts of Model (1) with $d = p = 1$ on which we have an influence are the disturbances ϵ_t and $\eta_t, t = 1, \ldots, T$, because they are not fixed to a specific value. Hence, we need to find suitable values for them. In order to obtain K design disturbance vectors, we use the fact that $(\eta_1, \ldots, \eta_T, \epsilon_1, \ldots, \epsilon_T) \sim \mathcal{N}_{2T}(\mathbf{0}, I_{2T})$. Then, we approximate the distribution of $\mathcal{N}_{2T}(\mathbf{0}, I_{2T})$ by a discrete distribution with K point mass vectors in $\mathbb{R}^{2T}$ and obtain $(\tilde{\eta}_1^{(k)}, \ldots, \tilde{\eta}_T^{(k)}, \tilde{\epsilon}_1^{(k)}, \ldots, \tilde{\epsilon}_T^{(k)}), k = 1, \ldots, K$. To do this, the algorithm introduced in [32] is used.

The idea of this algorithm is to approximate the continuous distribution of $\mathcal{N}_q(\mathbf{0}, I_q)$ by a discrete distribution that takes on K point-symmetric point masses with equal probability. The locations of the point masses of this discrete distribution are called discrete approximations or deterministic samples. For an even number of values $K = 2N$, the discrete distribution takes on the values $\{\mathbf{s}_1, \ldots, \mathbf{s}_N, -\mathbf{s}_1, \ldots, -\mathbf{s}_N\}$. For an odd number $K = 2N + 1$, we add $\mathbf{0}$. Hence, the vectors we have to find are given by $\mathcal{S}_N^q = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$. To find the optimal values of the vectors in $\mathcal{S}_N^q$, we need a distance measure between a continuous and a discrete distribution. For this, the so-called localized cumulative distribution function of [19] is utilized. Then, a distance measure based on the Cramér-von Mises distance for the localized cumulative distribution function is defined. Minimizing the distance between $\mathcal{N}_q(\mathbf{0}, I_q)$ and the desired discrete distribution over $\mathcal{S}_N^q$ leads to an optimal discrete approximation.

We obtain a matrix with dimensions $2T \times K$ consisting of the design disturbance values. The kth column corresponds to one design disturbance vector $(\tilde{\eta}_1^{(k)}, \ldots, \tilde{\eta}_T^{(k)}, \tilde{\epsilon}_1^{(k)}, \ldots, \tilde{\epsilon}_T^{(k)})$ for $k = 1, \ldots, K$, where the first half belongs to the state disturbance $\eta_t, t = 1, \ldots, T$, and the second half belongs to the observation disturbance $\epsilon_t, t = 1, \ldots, T$. Given these design disturbance vectors, the design observations and corresponding design states are defined by

$$\begin{aligned}
\tilde{z}_t^{(k)} &= g(\boldsymbol{\Omega}^{true}, \tilde{x}_t^{(k)}, \tilde{\epsilon}_t^{(k)}), t = 1, \ldots, T, \\
\tilde{x}_t^{(k)} &= h(\boldsymbol{\Omega}^{true}, \tilde{x}_{t-1}^{(k)}, \tilde{\eta}_t^{(k)}), t = 2, \ldots, T, \\
\tilde{x}_1^{(k)} &= h(\boldsymbol{\Omega}^{true}, x_0^{true}, \tilde{\eta}_1^{(k)})
\end{aligned}$$

for fixed values x_0^{true} and $\boldsymbol{\Omega}^{true}$ of the initial state x_0 and the time-invariant parameters $\boldsymbol{\Omega}$.

Example 5 *.For $T = 1$, we approximate the distribution of $\mathcal{N}_2(\mathbf{0}, I_2)$ by $K = 5$ values. This leads to the approximations of the disturbances (η_1, ϵ_1) shown on the left side of Fig. 2. Then we obtain $\tilde{z}_1^{(k)} = g(\boldsymbol{\Omega}^{true}, \tilde{x}_1^{(k)}, \tilde{\epsilon}_1^{(k)})$ and $\tilde{x}_1^{(k)} = h(\boldsymbol{\Omega}^{true}, x_0^{true}, \tilde{\eta}_1^{(k)})$ for $k = 1, \ldots, 5$. For the model given in 2, this means $\tilde{z}_1^{(k)} = a^{true}\tilde{x}_1^{(k)} + \sqrt{1 - (a^{true})^2} \cdot \tilde{\epsilon}_1^{(k)}$ and $\tilde{x}_1^{(k)} = a^{true}x_0^{true} + \sqrt{1 - (a^{true})^2} \cdot \tilde{\eta}_1^{(k)}$. On the right side of Fig. 2, the result is plotted together with the contour plot of the theoretical density of (X_1, Z_1) for $a^{true} = 0.9, x_0^{true} = 8$.*

2) Case: General d and p: For general d and p, the construction of the design observation trajectories can be done analogously. For $R_t = I_d$ and $Q_t = I_p$, the joint distribution of $(\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_T, \boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T)$ is $\mathcal{N}_{(d+p)T}(\mathbf{0}, I_{(d+p)T})$, which we approximate by a discrete distribution with K values to obtain design observation disturbances that we can insert into the model. We obtain a matrix with dimensions $(d + p)T \times K$ consisting of the design disturbance values. This leads to design observation trajectories $\tilde{z}_T^{(k)} = (\tilde{\mathbf{z}}_1^{(k)}, \ldots, \tilde{\mathbf{z}}_T^{(k)})$, based on the design state trajectories $\tilde{x}_T^{(k)} = (\tilde{\mathbf{x}}_1^{(k)}, \ldots, \tilde{\mathbf{x}}_T^{(k)}), k = 1, \ldots, K$.

B. Finding the Joint MAP Estimate Using Deterministic Numerical Optimization

As described in Section II.B , we want to investigate whether the joint MAP estimator exists to obtain information about the type of extrema of the posterior. Furthermore, we are interested in the properties of the joint MAP estimator, such as mean, variance, and local variance. Using the design observation trajectories $\tilde{z}_T^{(k)}, k = 1, \ldots, K$, we maximize the K posterior densities $\Pi(\cdot|\tilde{z}_T^{(k)}), k = 1, \ldots, K$, defined in (2), numerically over $\boldsymbol{\Theta}$.

The numerical maximization of $\Pi(\cdot|\tilde{z}_T^{(k)})$ is done using a primal barrier method (see [26, Sec. 19.6]) for $k = 1, \ldots, K$. When obtaining the optimization result, we have to perform checks to make sure that we have found a maximum. Due to the phenomenon described in Remark 1, it might be possible that for some design vectors $\tilde{z}_T^{(k)}$ with corresponding $\tilde{x}_T^{(k)}$, no maximum can be found. This leads to the estimates $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)}), k \in \mathcal{K}^{\max} \subset \{1, \ldots, K\}$, with $|\mathcal{K}^{\max}| \leq K$.

The optimization procedure searches for minima; hence, we use the function $-2\pi(\cdot|\tilde{z}_T^{(k)})$ for the optimization. Then, given a design observation trajectory $\tilde{z}_T^{(k)}$, we check the following properties of the result $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)})$ to ensure that the optimizer has converged to a minimum. The thresholds are chosen based on results of numerical studies we executed. For the kth optimization, we choose $(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, \tilde{x}_T^{(k)})$ as initial value.

Numerical Checks. Hessian: *Is the Hessian $H_{-2\pi}(\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)}))$ of $-2\pi(\cdot|\tilde{z}_T^{(k)})$ in $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)})$ positive definite? Then, we have a minimum of $-2\pi(\cdot|\tilde{z}_T^{(k)})$ and hence a maximum of $\Pi(\cdot|\tilde{z}_T^{(k)})$.*

Gradients: *Are the gradients of $-2\pi(\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)}))$ close to zero, i.e., smaller than a threshold? The threshold we use is 10^{-5}.*

Eigenvalues: *Is the result a ridge? The ratio of the smallest eigenvalue to the largest eigenvalue of $H_{-2\pi}(\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)}))$ should not be too small. We want a ratio of $> r_0 = 0.00001$. The ratio we consider is the inverse of the condition number of the Hessian, which is often considered in*

the literature [33], [34] and can be used in sensitivity analysis the way we also use it [16, Sec. 8.3.3]. As the threshold of r_0 is chosen based on results of numerical studies, one more step is applied to ensure that a ridge is present. In the case of a ratio $< r_0$, another starting point for the optimization is chosen. That starting point is close to the found optimum but shifted in the direction of the ridge. If this new optimization leads to a different value than before, we conclude that a ridge is present.

One way to quantify the observability of an SSM is the number $|\mathcal{K}^{\max}|$ of design observation trajectories $\tilde{z}_T^{(k)}$ with corresponding $\tilde{x}_T^{(k)}$, for which a unique estimate $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)})$ can be recovered. We denote the set of values k for which a maximum can be recovered by $\mathcal{K}^{\max} = \mathcal{K}^{\max}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) = \{k \in \{1, \dots, K\} : \operatorname{argmax}_{\boldsymbol{\Theta}} \Pi(\boldsymbol{\Theta} | \tilde{z}_T^{(k)}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}))$ fulfills numerical checks listed above$\} \subset \{1, \dots, K\}$ with $|\mathcal{K}^{\max}| \leq K$. For further analyses, only the results $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)}), k \in \mathcal{K}^{\max}$, are considered.

We say that the degree of observability for the fixed values $\boldsymbol{\Omega}^{true}$, $\mathbf{x}_0^{true}$, and T increases with $|\mathcal{K}^{\max}|$. Given the estimates $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)}), k \in \mathcal{K}^{\max}$, that were recovered, the mean, variance, and MSE of the corresponding estimator $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)$ can be estimated. The local variance can be used as another indicator of the degree of observability. A lower local variance indicates a higher degree of observability as the joint MAP estimator has higher precision.

C. Finding the Marginal MAP Estimate Using MCMC Sampling

In general, the integrals in the marginal posterior densities in (9) cannot be determined analytically. One solution to this is to use MCMC methods to approximately solve the integrals. We use a variant of Hamiltonian Monte Carlo (HMC), more specifically the No-U-Turn-Sampler [21] implemented in `Stan` [8], a platform for statistical modeling. Using `Stan`, we obtain MCMC samples from the posterior. Then, we can determine marginal MAP estimates from univariate kernel density estimates.

The three cases of (1) no maximum, (2) finitely many maxima, and (3) infinitely many maxima of the posterior can lead to convergence problems in the MCMC sampler. According to [1], *ridges in the posterior density [...] wreak havoc with both sampling and inference*. Additionally, multimodal posteriors lead to problems, which can however be avoided by constraining the parameter space if the posterior is not highly multimodal. Furthermore, there are examples where there are *no posterior modes and numerical stability issues can arise as sampled parameters approach constraint boundaries* [1]. That is why we first obtain information about the type of extrema by considering the joint MAP estimate to understand when such situations occur.

Sampling with `Stan` from the log-posterior $\pi(\cdot|\tilde{z}_T^{(k)})$ for one fixed $\tilde{z}_T^{(k)}$ outputs the samples

$$\boldsymbol{\Theta}(\tilde{z}_T^{(k)})^{(r)} = \left(\boldsymbol{\Omega}(\tilde{z}_T^{(k)})^{(r)}, \mathbf{x}_0(\tilde{z}_T^{(k)})^{(r)}, \mathbf{x}_1(\tilde{z}_T^{(k)})^{(r)}, \dots, \mathbf{x}_T(\tilde{z}_T^{(k)})^{(r)}\right), \quad r = 1, \dots, R,$$

of $\boldsymbol{\Theta}$, where R is the number of MCMC samples we obtain after discarding the burn-in phase.

Define $\mathcal{K}^{\texttt{Stan}} = \mathcal{K}^{\texttt{Stan}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) = \{k \in \{1, \dots, K\} :$ `Stan` converges when sampling from $\pi(\cdot|\tilde{z}_T^{(k)}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}))\}$ with $\left|\mathcal{K}^{\texttt{Stan}}\right| \leq K$. To check convergence, no divergent transitions after warmup are allowed. Furthermore, the convergence diagnostics R-hat [14] as well as bulk and tail effective sample size [35] are considered. Additionally, the maximum treedepth and the Bayesian Fraction of Missing Information (BFMI) [7] are checked. For more information on convergence diagnostics and the exact thresholds used in `Stan` to check convergence, see [2]. Only the results $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)})_{\text{mar}}, k \in \mathcal{K}^{\texttt{Stan}}$, are considered for further analyses. Hence, the diagnostics of all considered `Stan` runs are within the desired thresholds.

If $\left|\mathcal{K}^{\texttt{Stan}}\right|$ is close to K, we say that the model has a high degree of observability. If no marginal MAP estimate is recovered, i.e., $\left|\mathcal{K}^{\texttt{Stan}}\right| = 0$, we conclude that the model is not observable for the chosen values of $\boldsymbol{\Omega}^{true}$, $\mathbf{x}_0^{true}$, and T.

With the output of `Stan`, it is possible to approximate the marginal MAP estimate of the parameters as it is done in [22] and [23]. We obtain the marginal MAP estimate of Θ_j by

$$\hat{\Theta}_j(\tilde{z}_T^{(k)})_{\text{mar}} = \operatorname{argmax} \operatorname{kde}((\Theta_j(\tilde{z}_T^{(k)})^{(r)})_{r=1,\dots,R}), \quad j = 1, \dots, |\boldsymbol{\Theta}|, \tag{11}$$

where kde is the kernel density estimate [31]. The values $\hat{\boldsymbol{\Theta}}(\tilde{z}_T^{(k)})_{\text{mar}}, k \in \mathcal{K}^{\texttt{Stan}}$, can be used to estimate the mean and variance of the marginal MAP estimator $\hat{\boldsymbol{\Theta}}(\mathcal{Z}_T)_{\text{mar}}$.

For $j = 1, \dots, |\boldsymbol{\Theta}|$, we can determine 90% credible intervals $(\hat{\Theta}_j(\tilde{z}_T^{(k)})_{q5}, \hat{\Theta}_j(\tilde{z}_T^{(k)})_{q95})$ for Θ_j by

$$\begin{aligned}\hat{\Theta}_j(\tilde{z}_T^{(k)})_{q5} &= \text{empirical 5\%-quantile of } (\Theta_j(\tilde{z}_T^{(k)})^{(r)})_{r=1,\dots,R},\\ \hat{\Theta}_j(\tilde{z}_T^{(k)})_{q95} &= \text{empirical 95\%-quantile of } (\Theta_j(\tilde{z}_T^{(k)})^{(r)})_{r=1,\dots,R}.\end{aligned} \tag{12}$$

For every parameter Θ_j, we determine the percentage of iterations $k \in \mathcal{K}^{\texttt{Stan}}$, for which the true value Θ_j^{true} is in the interval $(\hat{\Theta}_j(\tilde{z}_T^{(k)})_{q5}, \hat{\Theta}_j(\tilde{z}_T^{(k)})_{q95})$, $j = 1, \dots, |\boldsymbol{\Theta}|$, i.e.,

$$\widehat{\text{Cover}}_{90}(\Theta_j^{true}) = \frac{|\{k \in \mathcal{K}^{\texttt{Stan}} : \hat{\Theta}_j(\tilde{z}_T^{(k)})_{q5} < \Theta_j^{true} < \hat{\Theta}_j(\tilde{z}_T^{(k)})_{q95}\}|}{|\mathcal{K}^{\texttt{Stan}}|}.$$

To get an idea how the marginal MAP estimator performs for the latent states $x_{t,i}, t = 1, \dots, T, i = 1, \dots, p$,

we estimate the bias by

$$\widehat{\text{Bias}}(x_{t,i}) = \frac{\sum_{k\in\mathcal{K}^{\texttt{Stan}}}\left(\hat{x}_{t,i}(\tilde{z}_T^{(k)})_{\text{mar}} - \tilde{x}_{t,i}^{(k)}\right)}{|\mathcal{K}^{\texttt{Stan}}|},$$
$$t = 1, \ldots, T, i = 1, \ldots, p,$$

where $\tilde{\mathbf{x}}_t^{(k)} = (\tilde{x}_{t,1}^{(k)}, \ldots, \tilde{x}_{t,p}^{(k)})$, and the MSE by

$$\widehat{\text{MSE}}(x_{t,i}) = \frac{\sum_{k\in\mathcal{K}^{\texttt{Stan}}}\left(\hat{x}_{t,i}(\tilde{z}_T^{(k)})_{\text{mar}} - \tilde{x}_{t,i}^{(k)}\right)^2}{|\mathcal{K}^{\texttt{Stan}}|},$$
$$t = 1, \ldots, T, i = 1, \ldots, p.$$

IV. ILLUSTRATIONS

We consider the different scenarios given in Table I.

In the following, we give an overview of the SSMs we investigate together with the respective definition of $\boldsymbol{\Theta}$ and the log-posterior $\pi(\boldsymbol{\Theta}|z_T)$. The prior $p_{\boldsymbol{\Omega}}(\boldsymbol{\Omega})$ is abbreviated by $p(\boldsymbol{\Omega})$. The disturbances are always assumed to be standard normal i.i.d. and independent and the prior p_0 of $\mathbf{x}_0$ is set to the uniformative prior. For sampling using `Stan`, we always consider 4 chains with 10 000 iterations and a burn-in of 4000, respectively, leading to $R = 24000$. Note here that running `Stan` on $\pi(\cdot|\tilde{z}_T^{(k)})$ can be parallelized for each $k = 1, \ldots, K$. We allow for random initial values in order to investigate the behavior when the true underlying values are not known.

We do not show the detailed results for all the considered models. However, they are available from the authors upon request.

1) Model `(Same,d=p=1)` (Same dynamics for observations and states for d=p=1):

$$Z_t = aX_t + \sqrt{1-a^2}\cdot\epsilon_t\,,\ X_t = aX_{t-1} + \sqrt{1-a^2}\cdot\eta_t\,,$$
$$t = 1, \ldots, T.$$

Log-posterior of $\boldsymbol{\Theta} = (a, x_0, \chi_T)$:

$$\pi(\boldsymbol{\Theta}|z_T) = -\frac{1}{2}\frac{1}{1-a^2}\left[\sum_{t=1}^{T}(z_t - ax_t)^2 + \sum_{t=1}^{T}(x_t - ax_{t-1})^2\right] - T\log(1-a^2) + \log(p^{res}_{a^{true}}(a)).$$

2) Model `(Sep,d=p=1)` (Separate dynamics for observations and states for d=p=1):

$$Z_t = bX_t + \sqrt{1-b^2}\cdot\epsilon_t\,,\ X_t = aX_{t-1} + \sqrt{1-a^2}\cdot\eta_t\,,$$
$$t = 1, \ldots, T.$$

Log-posterior of $\boldsymbol{\Theta} = (a, b, x_0, \chi_T)$:

$$\pi(\boldsymbol{\Theta}|z_T) = -\left[\sum_{t=1}^{T}\frac{(z_t - bx_t)^2}{2(1-b^2)} + \frac{(x_t - ax_{t-1})^2}{2(1-a^2)}\right] - \frac{T}{2}\left[\log(1-a^2) + \log(1-b^2)\right] + \log(p(a,b)).$$

3) Model `(Sep,d=2,p=1)` (Separate dynamics for bivariate observations driven by single states):

$$Z_{t,1} = aX_t + \sqrt{1-a^2}\cdot\epsilon_{t,1}\,,\ Z_{t,2} = bX_t + \sqrt{1-b^2}\cdot\epsilon_{t,2}\,,$$
$$X_t = cX_{t-1} + \sqrt{1-c^2}\cdot\eta_t.$$

Log-posterior of $\boldsymbol{\Theta} = (a, b, c, x_0, \chi_T)$:

$$\pi(\boldsymbol{\Theta}|z_T) = -\frac{1}{2}\frac{1}{1-a^2}\sum_{t=1}^{T}(z_{t,1} - ax_t)^2 - \frac{T}{2}\log(1-a^2) - \frac{1}{2}\frac{1}{1-b^2}\sum_{t=1}^{T}(z_{t,2} - bx_t)^2 - \frac{T}{2}\log(1-b^2) - \frac{1}{2}\frac{1}{1-c^2}\sum_{t=1}^{T}(x_t - cx_{t-1})^2 - \frac{T}{2}\log(1-c^2) + \log(p(a,b,c)).$$

4) Model `(Sep,d=1,p=2)` (Separate dynamics for univariate observations and bivariate states):

$$Z_t = aX_{t,1} + bX_{t,2} + \sqrt{1-a^2-b^2}\cdot\epsilon_t$$
$$X_{t,1} = cX_{t-1,1} + \sqrt{1-c^2}\cdot\eta_{t,1}\,,\ X_{t,2} = dX_{t-1,2} + \sqrt{1-d^2}\cdot\eta_{t,2}\,,\ t = 1, \ldots, T$$

Table I
The Considered Scenarios

Dim. d of observations	Dim. p of state	Linearity	Considered models
1	1	Nonlinear (linear only in state)	`(Same,d=p=1)`, `(Sep,d=p=1)`, `(Sep,d=p=1,ranWalk)`
2	1	Nonlinear (linear only in state)	`(Sep,d=2,p=1)`
1	2	Nonlinear (linear only in state)	`(Sep,d=1,p=2)`
1	1	Nonlinear	`(Sep,d=p=1, Multiplicative)`

Log-posterior of $\boldsymbol{\Theta} = (a, b, c, d, \mathbf{x}_0, \chi_T)$:

$$\begin{aligned}\pi(\boldsymbol{\Theta}|z_T) = &-\frac{1}{2}\frac{1}{1-a^2-b^2}\sum_{t=1}^{T}(z_t - ax_{t,1} - bx_{t,2})^2 \\ &- \frac{T}{2}\log(1-a^2-b^2) \\ &-\frac{1}{2}\frac{1}{1-c^2}\sum_{t=1}^{T}(x_{t,1} - cx_{t-1,1})^2 \\ &- \frac{T}{2}\log(1-c^2) \\ &-\frac{1}{2}\frac{1}{1-d^2}\sum_{t=1}^{T}(x_{t,2} - dx_{t-1,2})^2 \\ &- \frac{T}{2}\log(1-d^2) + \log(p(a,b,c,d)).\end{aligned}$$

5) Model `(Sep,d=p=1,ranWalk)` (Univariate observations with random walk state dynamics):

$$Z_t = aX_t + \epsilon_t\,,\ X_t = X_{t-1} + \eta_t\,,\ t = 1, \ldots, T$$

Log-posterior of $\boldsymbol{\Theta} = (a, \mathbf{x}_0, \chi_T)$:

$$\begin{aligned}\pi(\boldsymbol{\Theta}|z_T) = &-\frac{1}{2}\sum_{t=1}^{T}(z_t - ax_t)^2 - \frac{1}{2}\sum_{t=1}^{T}(x_t - x_{t-1})^2 \\ &+ \log(p^{res}_{a^{true}}(a)).\end{aligned} \tag{13}$$

6) Model `(Sep,d=p=1,Multiplicative)` (Additive state and multiplicative observation equation):

$$Z_t = \sqrt{a}\cdot X_t \cdot \epsilon_t\,,\ X_t = a + X_{t-1} + \eta_t\,,\ t = 1, \ldots, T.$$

Log-posterior of $\boldsymbol{\Theta} = (a, \mathbf{x}_0, \chi_T)$:

$$\begin{aligned}\pi(\boldsymbol{\Theta}|z_T) = &-\frac{1}{2}\sum_{t=1}^{T}\left(\frac{z_t^2}{a\cdot x_t^2} + \log(a\cdot x_t^2)\right) \\ &-\frac{1}{2}\sum_{t=1}^{T}(x_t - a - x_{t-1})^2 + \log(p(a)).\end{aligned}$$

A. Model `(Same,d=p=1)`

To decide on the number of design points K, we consider $\boldsymbol{V}_{K_i} = (\frac{|\mathcal{K}^{\max}|}{K_i}, \hat{\mathbb{E}}(\hat{a}(\mathcal{Z}_T)), \hat{\mathbb{E}}(\hat{x}_0(\mathcal{Z}_T)), \widehat{\mathrm{Var}}(\hat{a}(\mathcal{Z}_T)), \widehat{\mathrm{Var}}(\hat{x}_0(\mathcal{Z}_T)), \overline{\mathrm{LVar}}(\hat{a}(\tilde{z}_T^{(k)})), \overline{\mathrm{LVar}}(\hat{x}_0(\tilde{z}_T^{(k)}))) \in \mathbb{R}^7$ of the joint MAP estimator $(\hat{a}(\mathcal{Z}_T), \hat{x}_0(\mathcal{Z}_T))$ when using $K_i = i \cdot 100$ design observation trajectories. $\hat{\mathbb{E}}$ and $\widehat{\mathrm{Var}}$ denote the empirical mean and variance (see Table II). $\overline{\mathrm{LVar}}(\hat{\Theta}_i(\tilde{z}_T^{(k)}))$ is the mean of the local variances over k.

We want to find the value of i for which $\boldsymbol{V}_{K_i}$ has converged. For that, define the relative difference $RD(x, y) = \frac{|x-y|}{\max(|x|,|y|)}$ between two values $x, y \in \mathbb{R}$. For vectors, RD is defined component-wise. Then, we determine the smallest value i, for which $RD(\boldsymbol{V}_{K_i}, \boldsymbol{V}_{K_j}) < (c_1, c_1, c_1, c_2, c_2, c_2, c_2)$ for all $j > i$ and choose the number of design values as $K = i \cdot 100$. The values $(c_1, c_2) \in \mathbb{R}^2$ are chosen thresholds. We choose two different values because $|\mathcal{K}^{\max}|/K_i$ and the empirical means are converging faster than the empirical and the local variance.

Using the design disturbance values allows us to choose a smaller value of K compared to using random values of the disturbances. Numerical studies show that when using random values of the disturbances, a higher value of i has to be chosen to ensure convergence of $\boldsymbol{V}_{K_i}$. For example, for $(a^{true}, x_0^{true}) = (0.9, 8)$, $T = 4$, we choose $K = 300$ using $(c_1, c_2) = (0.01, 0.01)$ and the design disturbances. When using random values of the disturbances for the same setup, we have to choose

Table II
Top and Middle: The Empirical Mean and Variance of the Joint MAP Estimator (Left) and the Marginal MAP Estimator (Right)

	Joint MAP estimation	Marginal MAP estimation
T	$\left(\hat{\mathbb{E}}(\hat{a}(\mathcal{Z}_T)), \hat{\mathbb{E}}(\hat{x}_0(\mathcal{Z}_T))\right)$	$\left(\hat{\mathbb{E}}(\hat{a}(\mathcal{Z}_T)_{\mathrm{mar}}), \hat{\mathbb{E}}(\hat{x}_0(\mathcal{Z}_T)_{\mathrm{mar}})\right)$
	$(\bar{\mu}, \bar{\nu}) = \frac{\sum_{k\in\mathcal{K}^{\max}}(\hat{a}(\tilde{z}_T^{(k)}), \hat{x}_0(\tilde{z}_T^{(k)}))}{\lvert\mathcal{K}^{\max}\rvert}$	$(\bar{\mu}_{\mathrm{mar}}, \bar{\nu}_{\mathrm{mar}}) = \frac{\sum_{k\in\mathcal{K}^{\mathtt{Stan}}}(\hat{a}(\tilde{z}_T^{(k)})_{\mathrm{mar}}, \hat{x}_0(\tilde{z}_T^{(k)})_{\mathrm{mar}})}{\lvert\mathcal{K}^{\mathtt{Stan}}\rvert}$
20	(0.941, 7.15)	(0.896, 8.07)
50	(0.949, 6.99)	(0.898, 8.05)
T	$\left(\widehat{\mathrm{Var}}(\hat{a}(\mathcal{Z}_T)), \widehat{\mathrm{Var}}(\hat{x}_0(\mathcal{Z}_T))\right)$	$\left(\widehat{\mathrm{Var}}(\hat{a}(\mathcal{Z}_T)_{\mathrm{mar}}), \widehat{\mathrm{Var}}(\hat{x}_0(\mathcal{Z}_T)_{\mathrm{mar}})\right)$
	$\frac{\sum_{k\in\mathcal{K}^{\max}}\left((\hat{a}(\tilde{z}_T^{(k)})-\bar{\mu})^2, (\hat{x}_0(\tilde{z}_T^{(k)})-\bar{\nu})^2\right)}{\lvert\mathcal{K}^{\max}\rvert-1}$	$\frac{\sum_{k\in\mathcal{K}^{\mathtt{Stan}}}\left((\hat{a}(\tilde{z}_T^{(k)})_{\mathrm{mar}}-\bar{\mu}_{\mathrm{mar}})^2, (\hat{x}_0(\tilde{z}_T^{(k)})_{\mathrm{mar}}-\bar{\nu}_{\mathrm{mar}})^2\right)}{\lvert\mathcal{K}^{\mathtt{Stan}}\rvert-1}$
20	(0.0001960, 0.468)	(0.000605, 0.728)
50	(0.0000756, 0.374)	(0.000355, 0.602)
T	$\left(\widehat{\mathrm{MSE}}(\hat{a}(\mathcal{Z}_T)), \widehat{\mathrm{MSE}}(\hat{x}_0(\mathcal{Z}_T))\right)$	$\left(\widehat{\mathrm{MSE}}(\hat{a}(\mathcal{Z}_T)_{\mathrm{mar}}), \widehat{\mathrm{MSE}}(\hat{x}_0(\mathcal{Z}_T)_{\mathrm{mar}})\right)$
20	(0.00184, 1.18)	(0.000621, 0.733)
50	(0.00248, 1.40)	(0.000360, 0.605)

The Empirical Mean is Closest to the True Values $(a^{true}, x_0^{true}) = (0.9, 8)$ When Using the Marginal MAP Estimate. Bottom: The Estimated MSE for the Two Estimators.

$(c_1, c_2) = (0.01, 0.03)$, and it only converges for values between $K = 2500$ and $K = 4500$ or higher depending on the realizations.

In the following, we investigate observability properties for different choices of (a^{true}, x_0^{true}).

$(a^{true}, x_0^{true}) = (0.9, 8)$. For $T = 20$ and $T = 50$, we determine the values of K as described above. For $T = 20$, we choose $K = 1300$ with $(c_1, c_2) = (0.01, 0.03)$, and for $T = 50$, we choose $K = 3100$ with $(c_1, c_2) = (0.01, 0.04)$. K increases with T because more values are necessary to accurately cover a higher-dimensional space.

Maximizing the log-posterior densities $\pi(\boldsymbol{\Theta}|\tilde{z}_T^{(k)})$, $k = 1, \ldots, K$, leads to the left side of Fig. 3, showing the estimates $(\hat{a}(\tilde{z}_T^{(k)}), \hat{x}_0(\tilde{z}_T^{(k)}))$, $k \in \mathcal{K}^{\max}$, of (a, x_0). The joint MAP estimator exists with $|\mathcal{K}^{\max}|/K = 1$ but is biased. The empirical expected value and variance of the corresponding estimator $(\hat{a}(Z_T), \hat{x}_0(Z_T))$ are reported in the left part of Table II. The bias is increasing in T, while the empirical variance is decreasing. As expected, the mean of the K local variances is decreasing for increasing T (not shown here).

Using `Stan` to obtain the marginal MAP estimates as discussed in Section III.C, the right side of Fig. 3 shows that the issue of the bias gets solved. However, there is a trade-off with the variance. This can also be seen on the right side of Table II.

To study the joint behavior of bias and variance, we report estimates for the MSE of a and x_0 for the joint and marginal MAP estimation, respectively (bottom of Table II).

The estimated MSE of the marginal MAP estimator is always smaller. This is due to the fact that the bias is a lot smaller for the marginal MAP estimator compared to the joint MAP estimator, and the variance is not increasing enough to obtain a higher value of the MSE values. The values of the estimated coverage of a, x_0, and $x_1, \ldots, x_T$, as well as the estimated average bias and MSE of x_t, $t = 1, \ldots, T$, are satisfactory (Table III).

$(a^{true}, x_0^{true}) = (0.7, 8)$. We study the behavior of the model for a smaller value of a. We need a larger value of T in order to ensure a good degree of observability. Hence, we consider $T = 100$ and choose $K = 3600$ with $(c_1, c_4) = (0.01, 0.06)$. The joint MAP approach leads again to an obvious bias with $|\mathcal{K}^{\max}| = K$. Compared to $(a^{true}, x_0^{true}) = (0.9, 8)$, the local variance shows that the maximum of the posterior here is on average less peaked.

We use `Stan` to obtain the marginal MAP estimates, leading to a value of $|\mathcal{K}^{\texttt{Stan}}| = 2910$ smaller than $|\mathcal{K}^{\max}|$. As before, using the marginal MAP estimator solves the issue of the bias, but we have a trade-off with the variance. However, the estimated MSE is smaller for the marginal estimator. The average MSE of x_t, $t = 1, \ldots, T$, is slightly higher compared to the case $(a^{true}, x_0^{true}) = (0.9, 8)$. The estimated coverage of a, x_0, and $x_1, \ldots, x_T$ is satisfactory.

B. Model `(Sep,d=p=1)`

For this model, we face the problem of having multiple modes with equal height. Consider

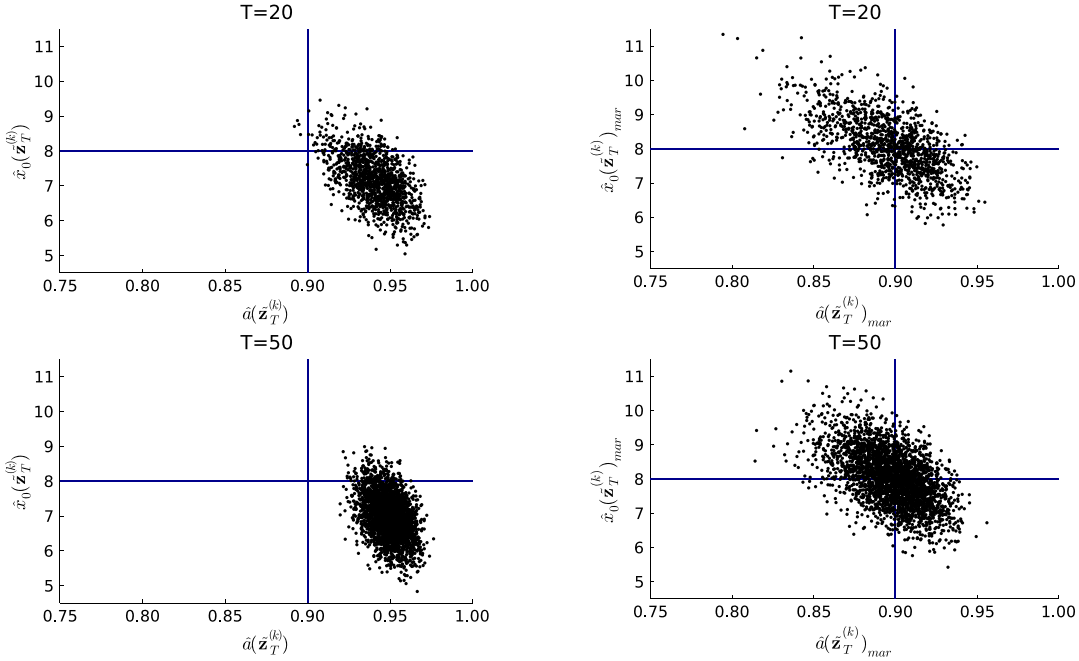


Figure 3. **$(a^{true}, x_0^{true}) = (0.9, 8)$:** Pairwise scatter plots of the joint MAP estimates $(\hat{a}(\tilde{z}_T^{(k)}), \hat{x}_0(\tilde{z}_T^{(k)}))$, $k \in \mathcal{K}^{\max}$, with $|\mathcal{K}^{\max}| = K$ (left) and marginal MAP estimates $(\hat{a}(\tilde{z}_T^{(k)})_{\text{mar}}, \hat{x}_0(\tilde{z}_T^{(k)})_{\text{mar}})$, $k \in \mathcal{K}^{\texttt{Stan}}$, with $|\mathcal{K}^{\texttt{Stan}}| = K$ (right) for different values of T. The true values are indicated by the lines.

Table III
The Estimated Coverage of the Parameters $\boldsymbol{\Theta} = (a, x_0, x_1, \ldots, x_T)$ as well as the Average Estimated Bias and MSE of $x_t, t = 1, \ldots, T$, for the Marginal MAP Estimates

T	$\widehat{\text{Cover}}_{90}(a^{true}, x_0^{true})$	$\frac{1}{T}\sum_{t=1}^{T} \widehat{\text{Cover}}_{90}(x_t^{true})$	$\frac{1}{T}\sum_{t=1}^{T} \lvert\widehat{\text{Bias}}(x_t)\rvert$	$\frac{1}{T}\sum_{t=1}^{T} \widehat{\text{MSE}}(x_t)$
20	0.870, 0.895	0.910	0.00892	0.116
50	0.893, 0.905	0.909	0.00335	0.105

$\boldsymbol{\Theta}_1 = (a, b, x_0, \chi_T)$ and $\boldsymbol{\Theta}_2 = (a, -b, -x_0, -\chi_T)$. As $\pi(\boldsymbol{\Theta}_1|z_T) = \pi(\boldsymbol{\Theta}_2|z_T)$, the posterior has two distinct modes, so we have to constrain our parameter space. One solution is to constrain both a and b while it is also possible to only pose restrictions on b. We consider both options.

Kreuzer *et al.* [22] approach the problem slightly different. The correlation between two consecutive observations Z_{t-1} and Z_t is $\text{Cor}(Z_{t-1}, Z_t) = b^2 \cdot a$. From this formula, it is obvious that there is an identifiability problem in b. As a solution, the model is reduced to a problem with one time-invariant parameter by setting $b = a^c$ for some $c \geq 1$ and the remaining parameter a is restricted to (0,1). Then, $\text{Cor}(Z_{t-1}, Z_t) = a^{2c+1}$, so that the identifiability problem is solved. With $c = 1$, this leads to Model `(Same,d=p=1)`.

$(a^{true}, b^{true}, x_0^{true}) = (0.9, 0.9, 8)$: Restricting both a and b to (0,1). We consider $T = 75$ and $K = 4000$ and choose the prior of a and b by $p(a, b) = p^{res}_{a^{true}}(a) \cdot p^{res}_{b^{true}}(b)$. We obtain $|\mathcal{K}^{\max}| = 1956$, hence the value of $|\mathcal{K}^{\max}|/K$ is significantly smaller than 1. There is an obvious bias of the joint MAP estimator again.

Using the marginal MAP estimates, we obtain $3680 = \left|\mathcal{K}^{\texttt{Stan}}\right| > |\mathcal{K}^{\max}|$. Using the marginal approach solves the issue of the bias again but leads to a trade-off with the variance. However, the MSE is smaller for the marginal approach again. The estimated coverage of a, b, x_0, and $x_1, \ldots, x_T$ as well as the average bias and MSE of $x_t, t = 1, \ldots, T$, are satisfactory.

$(a^{true}, b^{true}, x_0^{true}) = (0.9, 0.9, 8)$: Restricting b to (0,1) only. The prior of a and b is now chosen by $p(a, b) = 1_{(-1,1)}(a) \cdot p^{res}_{b^{true}}(b)$, and we consider again $T = 75$ and $K = 4000$. Compared to the case where we restrict a and b, the value of $\left|\mathcal{K}^{\texttt{Stan}}\right|$ decreases from 3680 to 2109, i.e., the degree of observability is decreasing. However, the values of the marginal MAP approach when only restricting b are very similar to restricting both a and b.

C. Model `(Sep,d=2,p=1)`

As noted by [23], the parameters $\boldsymbol{\Theta}_1 = (a, b, c, x_0, \chi_T)$ and $\boldsymbol{\Theta}_2 = (-a, -b, c, -x_0, -\chi_T)$ give the same posterior value and thus two distinct maxima. Hence, we have to restrict our parameter space. [23] propose to restrict only a to (0,1) and impose no further restrictions. We show now that this leads to an observable model. However, $\left|\mathcal{K}^{\texttt{Stan}}\right|$ increases significantly when also restricting the variables b and c.

$(a^{true}, b^{true}, c^{true}, x_0^{true}) = (0.9, 0.9, 0.9, 8)$: Restricting a, b, and c to (0,1). The prior of a, b, and c is chosen by $p(a, b, c) = p^{res}_{a^{true}}(a) \cdot p^{res}_{b^{true}}(b) \cdot p^{res}_{c^{true}}(c)$. We investigate $T = 50$ and $K = 4000$. We are now in the case of $d = 2$ and $p = 1$. The joint MAP approach leads to a biased estimator again, which is solved by the marginal MAP approach, for which we obtain $3995 = \left|\mathcal{K}^{\texttt{Stan}}\right| > |\mathcal{K}^{\max}| = 1833$. The values of the estimated coverage of a, b, x_0, and $x_1, \ldots, x_T$, as well as the average bias and MSE of $x_t, t = 1, \ldots, T$, are satisfactory.

$(a^{true}, b^{true}, c^{true}, x_0^{true}) = (0.9, 0.9, 0.9, 8)$: Restricting only a. The prior we use is $p(a, b, c) = p^{res}_{a^{true}}(a) \cdot 1_{(-1,1)}(b) \cdot 1_{(-1,1)}(c)$. Then, all the values of the marginal approach are practically identical to the case before. The difference is that $\left|\mathcal{K}^{\texttt{Stan}}\right| = 2972$, which is lower than the value before. Hence, the degree of observability increases when setting more restrictions.

D. Model `(Sep,d=1,p=2)`

As a prior, we use $p(a, b, c, d) = p^{res}_{a^{true}}(a) \cdot p^{res}_{b^{true}}(b) \cdot p^{res}_{c^{true}}(c) \cdot p^{res}_{d^{true}}(d)$. For this model, we have $d = 1$ and $p = 2$, which means that the dimension of the latent variable is higher than the dimension of the observation. This leads to unobservability of the model, i.e., $\left|\mathcal{K}^{\texttt{Stan}}\right| = 0$. As an additional step to setting no initial values for the MCMC sampling, we also set the initial value to the true parameters to make sure that no marginal MAP estimates can be recovered.

E. Model `(Sep,d=p=1,ranWalk)`

As a prior for a, we use

$$p_{a^{true}}(a) = \begin{cases} 1, a \in (0, \infty) & a^{true} > 0 \\ 1, a \in (-\infty, 0) & a^{true} < 0. \end{cases} \tag{14}$$

$(a^{true}, x_0^{true}) = (0.9, 8) : T = 4$. For $T = 4$, this model is not observable. No maximum of $\pi(\boldsymbol{\Theta}|\tilde{z}_T^{(k)})$, $k = 1, \ldots, K$, that fulfills our checks is found on the bounded support when using $K = 2000$ design observation trajectories, i.e., $|\mathcal{K}^{\max}| = 0$. The reason is that the posterior is a ridge of variable height, as shown in Fig. 4 for design observation $\tilde{z}_4^{(1)}$. The figure shows the profile posterior $\pi(a, x_0|\tilde{z}_4^{(1)}) = \max_{x_1,\ldots,x_4} \pi(a, x_0, x_1, \ldots, x_4|\tilde{z}_4^{(1)})$. Fur-

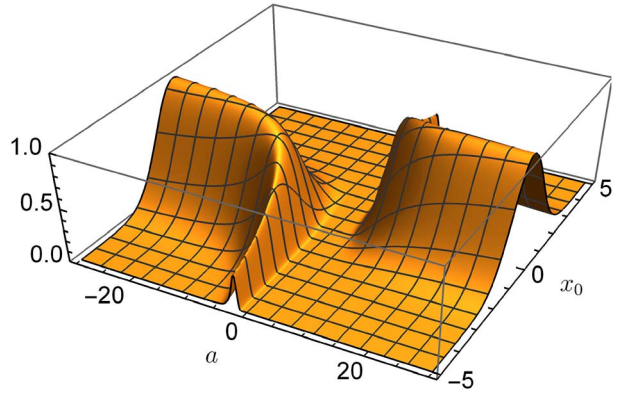


Figure 4. $(a^{true}, x_0^{true}) = (0.9, 8)$: Profile posterior $\pi(a, x_0|\tilde{z}_4^{(1)})$.

thermore, `Stan` does not converge for any $k = 1, \ldots, K$, when sampling from $\pi(\boldsymbol{\Theta}|\tilde{z}_T^{(k)})$, i.e., $\left|\mathcal{K}^{\texttt{Stan}}\right| = 0$. This makes sense with the information we have on the type of the extrema of the posterior.

$(a^{true}, x_0^{true}) = (0.9, 8) : T = 100$. For $T = 100$, the optimization procedure still does not find any joint maximum of the log-posteriors $\pi(\boldsymbol{\Theta}|\tilde{z}_T^{(k)})$, $k = 1, \ldots, 4000$. However, the marginal approach converges for 1575 out of 4000 design observation trajectories. Hence, for $T = 100$, the model is observable, but the degree of observability is low.

F. Model `(Sep,d=p=1,Multiplicative)`

As the prior of a, we choose (14) again and $(a^{true}, x_0^{true}) = (0.9, 8)$. In contrast to the previous examples, we set the initial values for sampling in `Stan` to the true underlying values. We investigate $T = 50$ and $K = 4000$.

For the joint MAP approach, we obtain $|\mathcal{K}^{\max}| = 3995$. Using the marginal MAP approach, we obtain $\left|\mathcal{K}^{\texttt{Stan}}\right| = 2598$ and that the bias as well as the variance and the MSE are higher compared to the other models we considered. The estimated MSE values are better for the marginal MAP approach. While the values of the estimated bias and MSE of $x_t, t = 1, \ldots, T$, are high compared to the other examples, the coverage is satisfactory.

V. CONCLUSIONS AND OUTLOOK

For a statistical SSM, we propose a definition of observability based on the existence or non-existence of a chosen parameter estimator. In particular, we consider the marginal MAP estimator in this paper. We provide an algorithm to check this definition of observability in practice. The algorithm allows us to check the observability properties of very general SSMs with Gaussian disturbances. Furthermore, we do not only answer the question of observability, but we also provide a quantitative observability measure given by the values $|\mathcal{K}^{\max}|$ and $\left|\mathcal{K}^{\texttt{Stan}}\right|$, the local variance, as well as the properties of the joint and marginal MAP estimator.

In general, observability is not a global property for all observation trajectories and has to be checked for every realization of the observation trajectory with corresponding state trajectory. Given this insight, the key idea of the proposed algorithm is to use deterministic approximations of the distribution of observation trajectories and state trajectories. Our simulations show that one advantage over using random observations is that the necessary number of samples is reduced in order to guarantee convergence of the properties of the joint MAP estimator (see Model `(Same, d=p=1)`). To the best of the authors' knowledge, this is the first time that deterministic approximations are used in order to obtain observability properties of stochastic SSMs.

To obtain marginal MAP estimates, the programming language `Stan` is used to sample from the posterior density, which works well for the considered examples. Alternatives could be the iterated batch importance sampling algorithm [9], the particle Markov chain Monte Carlo algorithm [4], or the SMC^2 algorithm [10].

In numerical studies, we check the observability properties of interesting SSMs with random disturbances using the proposed definition of observability and our algorithm. The numerical studies show that the approach works and that using the marginal MAP estimator leads to satisfactory results for the considered examples. Note that the definition of observability can also be adapted to other types of estimators.

In this work, the initial state $\mathbf{x}_0$ is treated as an unknown fixed value. This information might, for example, be given in a scenario where a mobile robot always starts from the same position. However, our approach can be adapted to consider observability for unknown random values of $\mathbf{x}_0$. The next step is to extend this approach to copula SSMs [22], [23], where observability is still an open question. For that, the design values have to be modified.

VI. GLOSSARY OF DEFINITIONS

Notation	Explanation
$\mathcal{Z}_T = (\mathbf{Z}_1^\top, \ldots, \mathbf{Z}_T^\top)^\top \in \mathbb{R}^{T \cdot d}$	Random observation trajectory $\mathcal{Z}_T$ for $t = 1, \ldots, T$, consisting of the observations $\mathbf{Z}_t$ at time point t
$z_T = (\mathbf{z}_1^\top, \ldots, \mathbf{z}_T^\top)^\top$	Realization of $\mathcal{Z}_T$
$\mathcal{X}_T = (\mathbf{X}_1^\top, \ldots, \mathbf{X}_T^\top)^\top \in \mathbb{R}^{T \cdot p}$	Random trajectory $\mathcal{X}_T$ of the latent state, consisting of the latent states $\mathbf{X}_t$ at time point t
$x_T = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_T^\top)^\top$	Realization of $\mathcal{X}_T$
$\mathbf{X}_0$	Initial state
$\mathbf{x}_0^{true}$	True underlying value of $\mathbf{X}_0$
$\boldsymbol{\Omega}$	Vector of unknown time-invariant parameters
$\boldsymbol{\Omega}^{true}$	True underlying value of $\boldsymbol{\Omega}$

Notation	Explanation
$(\boldsymbol{\epsilon}_t)_{t=1,\dots,T}$ and $(\boldsymbol{\eta}_t)_{t=1,\dots,T}$	Disturbances, serially independent and independent of each other at all time points with $\boldsymbol{\epsilon}_t \sim \mathcal{N}_d(\mathbf{0}, R_t)$ and $\boldsymbol{\eta}_t \sim \mathcal{N}_p(\mathbf{0}, Q_t)$, where $R_t \in \mathbb{R}^{d\times d}$ and $Q_t \in \mathbb{R}^{p\times p}$ are known covariance matrices
$\boldsymbol{\Theta}^{true} = (\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, \chi_T)$	Underlying unknown true parameters
$\Pi(\boldsymbol{\Theta}\|z_T)$	Joint posterior density
$\pi(\boldsymbol{\Theta}\|z_T)$	Log-posterior density
$\hat{\boldsymbol{\Theta}}(z_T) = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, \Pi(\boldsymbol{\Theta}\|z_T)$	Joint MAP estimate
$\hat{\boldsymbol{\Theta}}(z_T)_{\text{mar}}$	Marginal MAP estimate
$SO(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T)$	State-observation space
$\mathcal{E}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \subset \mathbb{R}^{T(p+d)}$	Set of realizations z_T and corresponding χ_T such that $\hat{\boldsymbol{\Theta}}(z_T)$ can be uniquely recovered
$\hat{\boldsymbol{\Theta}}(Z_T)$	Joint MAP estimator
$\mathcal{E}^{\text{mar}}(\boldsymbol{\Omega}^{true}, \mathbf{x}_0^{true}, T) \subset \mathbb{R}^{T(p+d)}$	Set of realizations (χ_T, z_T) such that $\hat{\boldsymbol{\Theta}}(z_T)_{\text{mar}}$ can be uniquely recovered
$\hat{\boldsymbol{\Theta}}(Z_T)_{\text{mar}}$	Marginal MAP estimator
$\text{LVar}\left(\hat{\Theta}_j(z_T)\right)$	Local variance of $\hat{\Theta}_j(z_T)$
$(\tilde{\eta}_1^{(k)}, \dots, \tilde{\eta}_T^{(k)}, \tilde{\epsilon}_1^{(k)}, \dots, \tilde{\epsilon}_T^{(k)})$	Design disturbance vectors for $d = p = 1$
$\tilde{z}_T^{(k)} = (\tilde{\mathbf{z}}_1^{(k)}, \dots, \tilde{\mathbf{z}}_T^{(k)})$	Design observation trajectories
$\tilde{\chi}_T^{(k)} = (\tilde{\mathbf{x}}_1^{(k)}, \dots, \tilde{\mathbf{x}}_T^{(k)})$	Design state trajectory
$\mathcal{K}^{\max}$	Set of values k for which a maximum is recovered and $\|\mathcal{K}^{\max}\| \leq K$
$\boldsymbol{\Theta}(\tilde{z}_T^{(k)})^{(r)}$	MCMC samples
$\mathcal{K}^{\texttt{Stan}}$	Set of values k for which `Stan` converges and $\|\mathcal{K}^{\texttt{Stan}}\| \leq K$

REFERENCES

[1] https://mc-stan.org/docs/stan-users-guide/problematic-posteriors.html.

[2] https://mc-stan.org/misc/warnings.html. Accessed: March 10, 2024.

[3] L. A. Aguirre, L. L. Portes, and C. Letellier "Structural, dynamical and symbolic observability: From dynamical systems to networks," *PLOS ONE*, vol. 13, 2018, Art. no. e0206180.

[4] C. Andrieu, A. Doucet, and R. Holenstein "Particle Markov chain Monte Carlo methods," *J. Roy. Stat. Soc. Ser. B: Stat. Methodol.*, vol. 72, pp. 269–342, 2010.

[5] A. Balakrishnan and V. Peterka Identification in Automatic Control Systems, *Automatica*, vol. 5, pp. 817–829, 1969.

[6] A. V. Balakrishnan "Identification and adaptive control: An application to flight control systems," *J. Optim. Theory Appl.*, vol. 9, pp. 187–213, 1972.

[7] M. Betancourt "A. Conceptual Introduction to Hamiltonian Monte Carlo"2018, arXiv:1701.02434.

[8] B. Carpenter et al. "Stan: A probabilistic programming language," *J. Stat. Softw.*, vol. 76, pp. 1–32, 2017.

[9] N. Chopin "A sequential particle filter method for static models," *Biometrika*, vol. 89, pp. 539–552, 2002.

[10] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos "SMC2: An efficient algorithm for sequential analysis of state space models," J. Roy. Statist. Soc., Ser. B, vol. 75, no. 3, pp. 397–426, 2013.

[11] O. Datcu, R. Tauleigne, A. Vlad, and J.-P. Barbot "Observability-singularity manifolds in the context of chaos based cryptography," in *Proc. 3rd Int. Conf. Syst. Control*, 2013, pp. 105–110.

[12] J. Durbin and S. J. Koopman *Time Series Analysis by State Space Methods*, 2nd revised ed. London, U.K.: Oxford University Press, 2012.

[13] J. Gauthier, H. Hammouri, and S. Othman "Simple observer for nonlinear systems applications to bioreactors," *IEEE Trans. Autom. Control*, vol. 37, pp. 875–880, 1992.

[14] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin Bayesian Data Analysis, 3rd ed. London, U.K: Chapman & Hall (Ltd.), 2013.

[15] D. Gerbet and K. Röbenack "'An algebraic approach to identifiability," *Algorithms*, vol. 14, pp. 255, 2021.

[16] P. E. Gill, W. Murray, and M. H. Wright *Practical Optimization.* London, U.K.: Emerald Group Publishing Limited, 1982.

[17] F. Hamelin, A. Iggidr, A. Rapaport, and G. Sallet *Observability, Identifiability Epidemiol. a Surv.*, arXiv:2011.12202 [math], (2021), https://arxiv.org/abs/2011.12202.

[18] A. Hanebeck and C. Czado "On the observability of Gaussian models using discrete density approximations," *Proc. 25th Int. Conf. Inf. Fusion*, 2022, pp. 1–8.

[19] U. Hanebeck and V. Klumpp "Localized cumulative distributions and a multivariate generalization of the Cramér-Von Mises distance," in *IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2008, pp. 33–39.

[20] R. Hermann and A. Krener "Nonlinear controllability and observability," *IEEE Trans. Autom. Control*, vol. 22, pp. 728–740, Oct. 1977.

[21] M. D. Hoffman and A. Gelman "The No-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.*, vol. 15, pp. 1593–1623, 2014.

[22] A. Kreuzer, L. Dalla Valle, and C. Czado "A Bayesian non-linear state space copula model for air pollution in Beijing," *J. Roy. Stat. Society: Ser. C. (Applied Statistics)*, vol. 71, no. 3, pp. 613–638, 2022.

[23] A. Kreuzer, L. Dalla Valle, and C. Czado "Bayesian multivariate nonlinear state space copula models," *Comput. Statist. Data Anal.*, vol. 188, 2023, Art. no. 107820.

[24] K. Langueh, O. Datcu, J.-P. Barbot, G. Zheng, and K. Busawon "Observabilty Singularities and Observer Design: Dual Immersion Approach," *IFAC-PapersOnLine*, vol. 49, pp. 511–516, 2016.

[25] R. Mohler and W. Kolodziej "An overview of bilinear system theory and applications," *IEEE Trans. Syst., Man, Cybern.*, vol. 10, pp. 683–688, Oct. 1980.

[26] J. Nocedal and S. J. Wright Numerical optimization, in *Springer Series in Operations Research* T. V. Mikosch, S. Resnick, B. Zwart, and T. Dieker, Eds., 2nd ed., Berlin, Germany: Springer-Verlag, 2006.

[27] A. O'Hagan "On Posterior Joint and Marginal Modes," *Biometrika*, vol.

63, pp. 329–333, 1976.
[28] T. Paradowski, S. Lerch, M. Damaszek, R. Dehnert, and B. Tibken
"Observability of uncertain nonlinear systems using interval analysis,"
Algorithms, vol. 13, pp. 66, 2020.
[29] L. Polansky, P. de Valpine, J. O. Lloyd-Smith, and W. M. Getz
"Likelihood ridges and multimodality in population growth rate models,"
Ecology, vol. 90, pp. 2313–2320, 2009.
[30] A. E. Raftery and L. Bao
"Estimating and projecting trends in HIV/AIDS generalized epidemics using incremental mixture importance sampling,"
Biometrics, vol. 66, pp. 1162–1173, 2010.
[31] B. Silverman
"Density estimation for statistics and data analysis,"
Biometrical J., vol. 30, pp. 876–877, 1986.
[32] J. Steinbring, M. Pander, and U. D. Hanebeck
"The smart sampling Kalman filter with symmetric samples,"
J. Adv. Inf. Fusion, vol. 11, pp. 71–90, 2016.
[33] J. M. Tabeart, S. L. Dance, A. S. Lawless, N. K. Nichols, and J. A. Waller
"New bounds on the condition number of the Hessian of the preconditioned variational data assimilation problem,"
Numer. Linear Algebra with Appl., vol. 29 2022.
[34] W. N. van Wieringen
"Lecture notes on ridge regression," 2021, arXiv:1509.09169.
[35] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C Bürkner
"Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC (with discussion),"
Bayesian Anal., vol. 16, pp. 667–718, 2021.
[36] A. F. Villaverde
"Observability and structural identifiability of nonlinear biological systems,
Complexity," vol. 2019, 2019, Art. no. e8497093.
[37] E. Walter
Identifiability of State Space Models with Applications to Transformation Systems, no. 46 in
Lecture Notes in Biomathematics, vol. 46. Berlin, Germany: Springer-Verl.

Ariane Hanebeck received the M.Sc. degree in mathematics from Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2020. She is currently pursuing the Ph.D. degree with the Chair of Applied Mathematical Statistics, Technical University of Munich, Germany. Her research interests include copula-based state space models and MCMC sampling for vine copula models. Applications are in the fields of mortality analysis and flight safety.

Claudia Czado received the Ph.D. degree in operations research and industrial engineering from Cornell University, Ithaca, NY, USA, in 1989. Since 1998, she has held an Associate Professorship of Applied Mathematical Statistics at the Technical University of Munich, Germany. Her research interests include statistical dependence modeling using vine copulas (vine-copula.org) and their application to engineering, finance, and health.

Optical Flow and IMU Fusion for Drone Horizontal Velocity Control

DJEDJIGA BELFADEL
JOHN CAIN
DAVID HAESSIG
CHERIF CHIBANE

This paper presents a vision-aided, horizontal velocity control system for unmanned aerial vehicles (UAVs). Angular velocity data provided by an Inertial Measurement Unit (IMU) is combined with pixel displacement data given by an optical flow sensor to produce a measure of horizontal velocity in two dimensions. Robust velocity control is achieved without reliance on the Global Positioning System (GPS). To validate the proposed approach, a series of dynamic simulations and hardware experiments were conducted. The hardware tests were designed to characterize the output of the optical flow sensor as a function of horizontal velocity and altitude. This data was then used to formulate an optical flow sensor mathematical model that, to our knowledge, is new and has not appeared in any prior optical flow sensor-related literature. This model is shown to produce output data that is consistent with the experimentally measured sensor data gathered under equivalent conditions. In addition, this model has been used to create the optical flow sensor model employed when simulation testing a UAV's controlled dynamic motion. Simulation results demonstrate that this method successfully achieves accurate and reliable control of a UAV's velocity in the horizontal plane.

Manuscript received March 28, 2024; revised June 19, 2024; released for publication January 10, 2025

Refereeing of this contribution was handled by Mattia Brambilla.

D. Belfadel and J. Cain are with the Department of Electrical Engineering, Fairfield University, Fairfield, CT 06824 USA (e-mail: dbelfadel@fairfield.edu; john.cain@student.fairfield.edu).

D. Haessig and C. Chibane are with the AuresTech Inc., Bridgewater, NJ 08807 USA (email: dave@aurestech.com; cherif.chibane@aurestech.com).

I. INTRODUCTION

Localization is the process of determining an object's position and possibly also its orientation in space. It is related to navigation and is a process that is critical in aerospace and robotics applications [1]. A similar and less stringent process is that of location stabilization, which involves the control of an object's velocity and stabilization of its location, but does not also require knowledge of the object's location. The process targeted by the work presented in this paper is that of location stabilization.

To perform localization and location stabilization, robots typically estimate their velocity and position by numerically integrating acceleration data provided by an inertial measurement unit (IMU) [2], [3]. This IMU data contains, however, measurement noise and sensor bias errors that can significantly degrade the accuracy of these estimates [4]. The numerical integration processes involved in generating the estimates can also introduce additional error, which further degrades accuracy [5]. Euler integration, for example, causes truncation errors similar to Taylor series approximations [6]. These and other sources of error can lead to the deterioration of the velocity and position signals required for accurate localization and stabilization. Over time, these errors can accumulate, resulting in significant inaccuracies that affect the system's performance [7]. The impact of these drift errors can be mitigated by adding sensors that are able to observe the drift. A wide range of sensors can be used for this purpose, one being the optical flow sensor (OFS).

Cameras in unmanned aerial vehicles (UAVs) can aid in localization through object recognition and tracking, utilizing landmarks. Smart cameras and applicable software libraries will calculate motion from frame changes and use digital signal processing to detect changes in shadow levels in each pixel [8]. These approaches, however, are computationally demanding, and issues like unknown obstacles or lighting changes can cause data inaccuracies. To address these, SLAM (simultaneous localization and mapping) uses multiple cameras and Light Detection and Ranging (LIDAR) sensors to handle environmental lighting changes [9]. SLAM combines multiple sensors to generate a 3D map of the surroundings [10], offering accurate position estimation but at a high computational cost. Self-maneuvering UAVs must also be capable of autonomous landing. Several researchers, such as [8] and [11], have investigated the use of cameras and vision-aided systems for autonomous landing by fusing data from cameras, OFS, and IMUs without the need for altimeters like LIDAR or ultrasonic sensors. These studies demonstrate that vision-based systems can accurately measure both altitude and horizontal velocity, which is expected to improve landing performance. In contrast, our work does not involve a camera and instead relies on an altimeter, showing that horizontal velocity can be measured

effectively using only the OFS, IMU, and altitude sensor. Additionally, this sensor configuration requires significantly fewer computational resources than vision-based systems. OFS provide a low cost and computationally tractable approach for drone location stabilization, platform navigation [12], [13], and to perform position and velocity control, as done in this paper and in [14]. Their effectiveness as a sensor for short-term stabilization has been established [15], but their applicability for long-term positioning is still under active research. Studies assessing the potential of OFS for position estimation, particularly when fused with other sensor systems, are found in [13] and [16].

This paper presents a new mathematical model for an OFS (the Truth Model) and compares this model to experimentally measured sensor data captured over a range of operating conditions that would be typical when the OFS is mounted on a drone, i.e., limited altitude, limited velocity, and limited roll and pitch tilt angles, i.e., less than 25°. This new OFS model is shown herein to provide good results for several test cases that fall within this operating envelope.

This paper also presents a horizontal velocity control system design for rotary wing UAVs based on the new OFS model. This design involves a velocity sensor equation that combines the output of the OFS with angular rate data given by an IMU, and from these produces a measure of the drone platform horizontal translational velocity in two dimensions. A simulation study is performed to assess the impact that OFS pixel quanization has on control system performance. The velocity sensor and control design presented herein assumes that platform's altitude above the ground is known, and available as an input to the velocity sensor. Altitude data is assumed to be provided, for example, by a separate device such as a sonar or radar altimeter. Two single-axis models are applied to generate the 2D output from a single two-axis OFS.

To assess the OFS's response to changes in the physical environment, we constructed a linear rail setup to observe the OFS's output at various sensor velocities and altitudes. Experiments were also conducted to determine the sensor's resilience to environmental fluctuations, varying light levels, and surface textures. These test results are reported herein, and the results are used to validate the new OFS model proposed herein.

A Simulink model of the physical plant (i.e., a fixed-wing UAV), the proposed velocity sensor, the OFS, and the Control Law were constructed to evaluate this system's performance. It is a single-axis, continuous-time dynamic model of a UAV's horizontal motion and includes an IMU model, a hybrid (continuous-time, discrete-time) model of an OFS, and a discrete-time proportional–integral (PI) controller. Performance has been assessed in three scenarios: a step response in commanded velocity, a response to a wind gust, and the tracking of very slow velocity commands. Each scenario demonstrates an important drone control quality that is facilitated by the OFS. The results demonstrate (a) the rapid response rise time of approximately 1 s, (b) the detection and rejection of location perturbations caused by wind, and (c) the control of velocity at very low velocity setpoint levels. These results indicate that the control design approach proposed here is an effective method for controlling a drone's horizontal velocity.

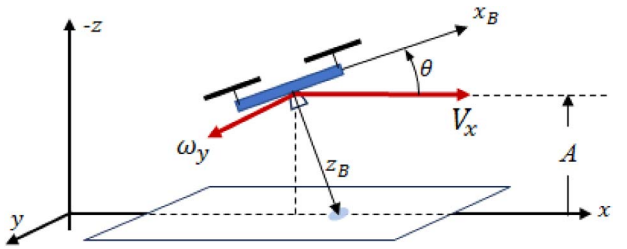


Figure 1. Scene velocity as a function of platform linear and angular velocities.

Section II outlines the problem formulation and fundamental OFS equations, detailing the OFS Truth Model and Velocity Sensor Model for control. Section III, Experimental Setup, covers the linear velocity rail system, IMU, and PMW3901 OFS. Section IV, Simulated and Experimental OFS Results and Comparison, compares experimental results with simulations, including constant velocity tests, oscillatory motion, and additional testing. Section V discusses the control design, plant dynamics, and simulation results, including step response, wind gust response, and slow movement. Finally, Section VI presents conclusions.

II. PROBLEM FORMULATION AND SENSOR MODEL

The problem under consideration is illustrated in Fig. 1, which portrays a drone equipped with an OFS assembly at an altitude A, moving with horizontal velocity V_x, and rotating with angular velocity ω_y about the y axis. The platform's translational motion is limited to motion along the x axis, and the pitch angle θ is assumed to be small, i.e., less than 25° in magnitude. The sensor's field of view (FOV) is centered at the location where the z_B and x axes intersect. The platform body coordinate frame contains axes x_B and z_B (y_B is not shown).

The pinhole model of the image motion observed by an OFS, given in [15], is often used to mathematically characterize the output of this device. This model defines the instantaneous velocity of the image falling on the image plane at a pixel located at (x, y) on the image plane with focal distance f. This pixel velocity field, also called the optical flow field, is represented symbolically as $P(x, y, f) = \{\delta_x, \delta_y\}$, where δ_x and δ_y are the components of the optical flow velocity vector on the image plane at location (x, y). The detailed derivation of these linear, continuous-time velocities is provided in [15] and expressed as:

$$\delta_x = \frac{V_z x - V_x f}{A} - \omega_y f + \omega_z y + \frac{\omega_x xy - \omega_y x^2}{f}, \qquad (1)$$

$$\delta_y = \frac{V_z y - V_y f}{A} + \omega_x f - \omega_z x + \frac{\omega_x y^2 - \omega_y xy}{f}, \quad (2)$$

where A is the distance from the OFS to the ground, i.e., the altitude. As noted in [15], however, an OFS produces a single pair of values at every sample time and not the velocities being observed at each pixel over the field of the image plane. An OFS device and processing algorithm generate, in a sense, the mean of the optical flow field in a central neighborhood of pixels around the center point at pixel $P(0, 0, f)$. Thus, it is sensible to compute the output of an OFS device, as noted in [15], using the velocity values given by (δ_x, δ_y) at the point $P(0, 0, f)$. After substituting this point into equations (1) and (2), we find:

$$\delta_x = -\frac{V_x f}{A} - \omega_y f, \quad (3)$$

$$\delta_y = -\frac{V_y f}{A} + \omega_x f, \quad (4)$$

Noting that δ_x and δ_y are linear velocities of a pixel at an orthogonal distance f, we can divide equations (3) and (4) through by f and use angular velocities $\Omega_x = \frac{\delta_x}{f}$ and $\Omega_y = \frac{\delta_y}{f}$:

$$\Omega_x = -\frac{V_x}{A} - \omega_y, \quad (5)$$

$$\Omega_y = -\frac{V_y}{A} + \omega_x. \quad (6)$$

This allows us to work in terms of the angular velocity of the OFS body, a more natural and physically observable quantity associated with OFS motion rather than the linear translational optical flow present on the image plane. Here, equation (6) applies to Fig. 1, while equation (5) pertains to translational motion along the y axis and is not pictured. The problem formulation for that case is analogous to that of the x axis but involves the assembly's velocity in the y direction and roll angular velocity about the x axis. These are the equations we carry forward into the development of the hybrid continuous-time and discrete-time model of the OFS output. Reference [15] also includes a discrete-time OFS model. A key difference between that model and the one being presented here is that this new model incorporates the finite resolution and quantization associated with the pixels. Additionally, our model computes the quantized angular displacement observed by the sensor and the change in image angle from one image frame to the next. Thus, it does not quantize the instantaneous angular velocity appearing near the center of the image frame, but quantizes the integrated, continuous-time velocity (i.e., accumulated angular displacement up to the time when the most recent image frame is captured).

A. OFS Truth Model

The model created to represent the OFS output is one that involves the integral of the angular rate of the scene. Since both equations (5) and (6) are very similar, we will continue with only equation (5), dropping the subscripts for clarity. Equation (5) is then:

$$\Omega = -\omega - \frac{V}{A}, \quad (7)$$

where Ω is the scene angular velocity, ω is the angular velocity of the sensor body, V is the sensor translational velocity, and A is the sensor distance (or altitude) from the scene being observed (i.e., the ground). Integrating this to produce an angle:

$$\phi(t) = \int_{t_0}^{t} \left(-\omega - \frac{V}{A}\right) dt, \quad (8)$$

where ω, V, and A are functions of the continuous time variable t.

To produce the sensor output, we integrate $\phi(t)$ to the frame time kT at frame index k, quantizing to the quantization step size Q, and generate the difference in the quantized result:

$$\phi(kT) = \int_{t_0}^{kT} \left(-\omega - \frac{V}{A}\right) dt, \quad (9)$$

$$X_k = \text{floor}\left(\frac{\phi_k}{Q}\right), \quad (10)$$

$$\Delta X_k = X_k - X_{k-1}. \quad (11)$$

This value, ΔX_k, is what comes out of the sensor to represent motion of the scene in one direction. It is an integer value and has units of counts. There is an equivalent equations for ΔY_k, representing motion in the other orthogonal direction.

B. Sensor Model for Conversion to Linear Velocity

To convert from the sensor output's change in scene angle ΔX_k to linear velocity, we integrate the scene angle over a single frame time period:

$$\Delta\phi_k = \int_{(k-1)T}^{kT} \left(-\omega - \frac{V}{A}\right) dt. \quad (12)$$

To generate this integrated result, we assume that the inputs are constant over the very short time period T:

$$\Delta\phi_k = -\omega_k T - \frac{V_k}{A_k} T. \quad (13)$$

The angular rate ω_k is the pitch angular velocity and is provided by the IMU. The altitude A_k is assumed to be known, provided by an altitude sensor, which can be a barometric altimeter's measure of the distance above the take-off altitude, assuming that the ground is relatively flat. Equation (11) is converted from units of counts to units of radians (angle) using the quantization step size:

$$\Delta\phi_k = \Delta X_k Q. \quad (14)$$

Table I
Linear Rail Test Bed Hardware Component

Component type	Specifications
PMW3901MB (2-axis optical flow sensor)	Measurement range: 80 mm-infinity, 4-Wire SPI @ 2 MHz
HC-SR04 (Ultrasonic ranging sensor)	Measurement range: 2 cm–400 cm, eight 40 kHz signal transmit and detect.
2× Atmega328P (Arduino)	Low power AVR 8-bit microcontroller, 32K bytes of progammable flash memory, and support of SPI/I2C communication.
DRV8825 (stepper motor driver IC)	Rated for 2.5 A max current, and a resolution of 32 microsteps.
ZD-0029 R REIFENG (limit switch)	Connect VCC (ramps of +); Black line: Connect GND(ramps of -); Green line: Connect SIGNAL(ramps in s).
Nema 17 stepper motor	Bipolar stepper, Step Angle: 1.8deg, with a holding torque of 45 Ncm (63.74oz.in)
AC/DC power supply	110/220 V AC in, adjustable up to 12 V 30 A out.

Combining equations (13) and (14) and solving for velocity V_k:

$$V_k = \left[-\omega_k - \Delta X_k \left(\frac{Q}{T}\right)\right] A_k. \tag{15}$$

This is the equation that will be used to produce a velocity measurement that will serve as the feedback signal for velocity control.

The OFS is an imaging device that generates pixel flow measurements from the scene in view. By comparing pixel content between successive frames of an image, the direction and speed of motion of the image are determined within the sensor. Consider the case in which the scene is moving by several pixels in distance from one frame to the next, with a frame capture time of 20 ms (i.e., 50 Hz). The delta-angle output produced by the sensor is an integer count value representing the angular rotation measured between the two frame images, quantized to an integer representing the pixel size. The fractional part of the motion is removed from the output by quantization. When the motion is slow enough so that it takes more than 20 ms to move a signal pixel, then the output data toggles between 0 and 1 with a duty cycle that depends on velocity. If motion is very slow so that it takes several frames, e.g., 10, to move by a single pixel, then the output will sit at zero (0) counts until the scene finally reaches a distance of one pixel, and at that sample time it will output a change of 1 pixel, then it will drop back to outputting zeros while moving to the next pixel quantum distance.

The OFS's quantization of angular position has an advantage over that of a sensor that measured angular velocity; the OFS does not introduce an accumulating error due to quantization. A velocity sensor, on the other hand, could experience an angular rate that remains within one quantum, and that motion would never be evident in the sensor's output. The OFS instead is sensing angular displacement, or the integral of the angular rate, and detecting the change in that angle from frame to frame, so that even the smallest angular rate input is eventually detected when that motion reaches an angular displacement of one pixel. The sensor outputs a count value of 1 when the distance accumulated exceeds another quantization step size. When the sensor is moving slowly and therefore the scene is moving slowly over the focal plane, the sensor produces a stream of 0's punctuated with a periodic value of 1 count that occurs when the distance traversed reaches a distance equaling one pixel.

III. EXPERIMENTAL SETUP

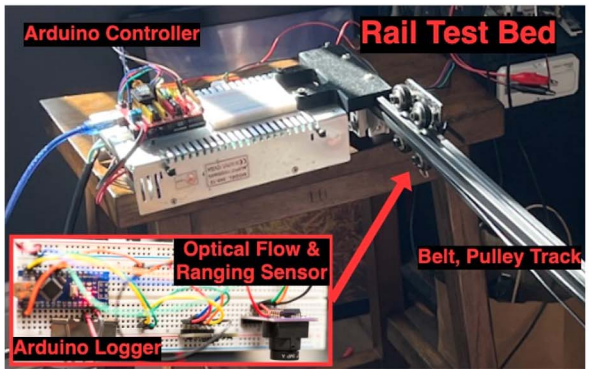
A. Linear Velocity Rail System

We evaluated the PMW3901 sensor's ability to adapt to environmental changes in light intensity, surface texture, and altitude using a linear rail system with 80/20 aluminum extrusion bars, powered by a Nema 17 stepper motor and a DRV8825 stepper driver.

The PMW3901 collects 2D displacement data, and an HC-SR04 ultrasonic sensor is used to measure altitude. This setup is most effective for distances under 5 m. A powered rail sliding mechanism replicates linear flight along a single axis using 80/20 aluminum extrusion rods and a stepper motor-powered belt drive. Two Atmega328P (Arduino) microcontrollers handle data collection and slider motion. The stepper motor, driven by a DRV8825 driver IC and powered by a 12V, 30A power supply, determines the slider position using a limit switch for homing the slider and tracking individual stepper motor steps. The setup can be placed on sawhorses to simulate varying altitudes. The hardware components of the linear rail test bed are detailed in Table I, while the system layout is depicted in Fig. 2.

B. Inertial Measurement Unit

An IMU measures the linear accelerations and angular rotational rates that occur along a coordinate frame that is attached to the IMU body. This body frame rotates with the IMU itself. Since the IMU senses 3 axes of rate and acceleration, it is referred to as a 6-axis sensor. A model of the IMU can be found in [17]. This IMU was not integrated with the experimental test setup but will be used in hardware testing that will involve both



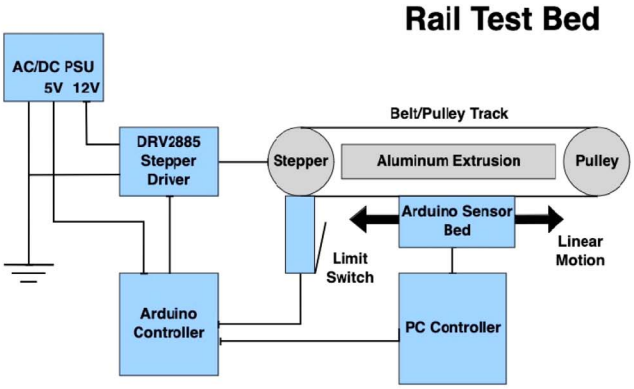


Figure 2. Linear rail system with stepper-driven belt drive.

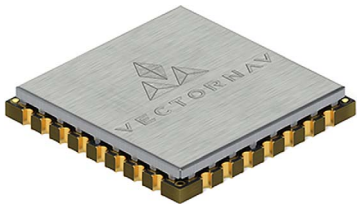


Figure 3. The VN-100 IMU/AHRS by VectorNav technologies.

the OFS and IMU. It is described here, therefore, for reference.

1) VN-100 IMU/AHRS Sensor Specifications: The VN-100, manufactured by VectorNav Technologies, shown in Fig. 3, is an example of an IMU that is commonly used in applications such as UAVs, drones, and robotics. Tables II and III provide details of its performance specifications and attitude/heave capabilities. The VN-100's compact size and low power consumption make it an attractive option for use in small or portable devices where space and power are limited. Linear accelerations measured by the IMU must be transformed from the IMU body to a navigation frame, corrected for gravity, and integrated to produce platform velocity and position. As noted, IMU sensor noise causes the dead-reckoning navigation solution to deviate from the correct solution with a random walk type of error that grows with time and is unbounded. To constraint the growth of these drift errors, an aided inertial navigation algorithm will utilize measurement data provided by additional sensors to improve or aid the solution produced by the IMU alone. Sensor data may include, for example, optical flow, measurement of platform ground speed, air speed, altitude above ground, range to a specific location, etc.

Table III
The VN-100 IMU Attitude and Heave Performance

Specification	Value
Range (heading/yaw, roll)	±180°
Range (pitch)	±90°
Heading (magnetic)	2.0° RMS
Pitch/roll (static)	0.5° RMS
Pitch/roll (dynamic)	1.0° RMS
Heave accuracy	5% or 5 cm
Delayed heave accuracy	2% or 2 cm
Angular resolution	0.001°

C. OFS Model

The Pimoroni PMW3901 OFS, shown in Fig. 4, is an example of an embedded light-tracking optical sensor. It captures a collection of pixel displacement values known as an optical flow field, and then averages this field over a region at the center of the sensor's Field of View, producing what is designated a pixel vector with x and y components. The Pimoroni sensor data sheet indicates that the sensor has a Field of View (FOV) of 42°, (±21°). Detailed specifications of the PMW3901 Optical Flow Sensor are provided in Table IV. From a series of tests at varying velocities during which the sensor pixel vector outputs were captured, it was observed that the PMW3901 must move a minimum of 0.0015 radians to register a new output with a change of 1 pixel count. The flow vectors produced within the sensor in either the x or

Table II
Sensor Performance Specifications

Specification	Accelerometer	Gyroscope	Magnetometer	Barometer
Range	±16 g	±2000°/s	±2.5 Gauss	10–1200 mbar
In-run bias stability	<0.04 mg	<10°/h	-	-
Noise density	0.14 mg/$\sqrt{\text{Hz}}$	0.0035°/s/$\sqrt{\text{Hz}}$	140 μGauss/$\sqrt{\text{Hz}}$	-
Bandwidth	260 Hz	256 Hz	200 Hz	200 Hz
Cross-axis sensitivity	±0.05°	<0.05°	±0.05°	-

Figure 4. The Pimoroni PMW3901 optical flow sensor.

y direction must separately average to this quantization level to produce a new pixel count of 1. A key feature of the sensor is that it is not measuring velocity but is measuring a change in the location of the scene.

IV. SIMULATED AND EXPERIMENTAL OFS RESULTS AND COMPARISON

A. Constant Velocity Tests at Fixed Altitudes

This section contains the results of the experiments performed to assess the PMW3901 sensor's ability to provide useful velocity and displacement measurement data at various speeds and over a range of fixed altitudes. This evaluation was primarily focused on understanding the sensor's performance when altitude is known, with the aim of gauging the reliability of its raw data output. Tests were conducted on the linear rail system described above. In this first set of tests, the sensor was moved at a constant velocity from one end of the rail system to the other. Sensor output data was captured for a brief period before motion began and ended after motion stopped at the other end. Hardware tests and simulations were run at four velocities: 0.05, 0.1, 0.2, and 0.3 m/sec, all with a sampling frequency of 50 Hz and all at a distance of 0.66 m between the OFS and the ground scene. These results are given in Fig. 5.

For an input velocity setpoint of 0.05 m/s on the rail system, the OFS model produces an output having a mean value of 1.016 counts/sample when motion is occurring (4.7–15.4 s). In all four cases, the mean "counts/sample" values shown on each plot were computed over the time periods of OFS motion. Multiplication by the sample rate of 50 samples/s produces the sensor's Output Rate in counts/s, the third column of Table V. For example: 1.016 counts/sample $\times$ 50 samples/s $=$ 50.8 counts/s.

Table IV
Specifications of the PMW3901 Optical Flow Sensor

Specification	Value
Field of view (FOV)	42° (±21°)
Minimum radian change for output	≈ 0.0015 radians
Frame time	20 ms (50 Hz)
Quantization level for output	1 pixel count
Motion detection type	Angular displacement

Table V
Sensor Raw Output Parameters and Derived Quantization Step Size

Velocity (m/s)	True angular rate Ω (radians/s)	Output rate (counts/s)	Quant step $-Q$ (radians/count)
0.05	0.076	50.8	0.001501
0.1	0.152	102.5	0.001483
0.2	0.303	199	0.001522
0.3	0.455	294	0.001548

To generate the quantization step size Q, we use the relation:

$$Q \cdot (\text{Output Rate}) = \Omega$$
$$\frac{\text{radians}}{\text{count}} \cdot \frac{\text{counts}}{\text{s}} = \frac{\text{radians}}{\text{s}}. \tag{16}$$

The true angular rate Ω is computed with equation (5) with $\omega_y = 0$ (the OFS was not rotating):

$$\Omega_x = -\frac{V_x}{A}. \tag{17}$$

For an altitude of 0.66 m and the velocity setpoint values of column 1, negated since the rail system velocities were in the negative direction, one computes the quantization step size given in column 4, which differ slightly due to random error present in the test setup.

The mean Quantization Step Q for the PMW3901 computed by averaging these values is thus 0.0015 radians/count or 0.086 deg/count. This value has been used in the optical sensor model as well as in the computation of platform velocity from sensor output data. The jumpiness in this data is due in part to the pixel quantization; however, quantization error does not account for all of the "noise" that is observed. It is conjectured that this jumpiness represents real velocity perturbations caused by stepper motor velocity jitter and imperfections in the test setup. In addition, there may be additive measurement error contributed by the sensor itself. Our suspicion, however, is that the hashiness in the measured OFS output data is caused primarily by actual velocity perturbations. The OFS output can be used to measure translation displacement by summing the ΔX_k values. This was done for all 4 velocity test cases, with the results shown here in Fig. 6.

These summed outputs reach values of 526, 521, 506, and 504 counts for the four velocities of 0.5, 1.0, 2.0, and 3.0, respectively. Converting from counts to distance using the formula $QA\,(\sum \Delta X_k)$ produces a set of distance measurements ranging from 0.52 to 0.05 m, a difference of 4%.

B. Comparison to Simulation

The OFS Truth Model developed in Section II and defined in equations (9)–(11) is shown here in Fig. 7 as a Simulink model in block diagram form.

The results produced by this model are compared to the experimental results plotted in Fig. 5 first by driv-

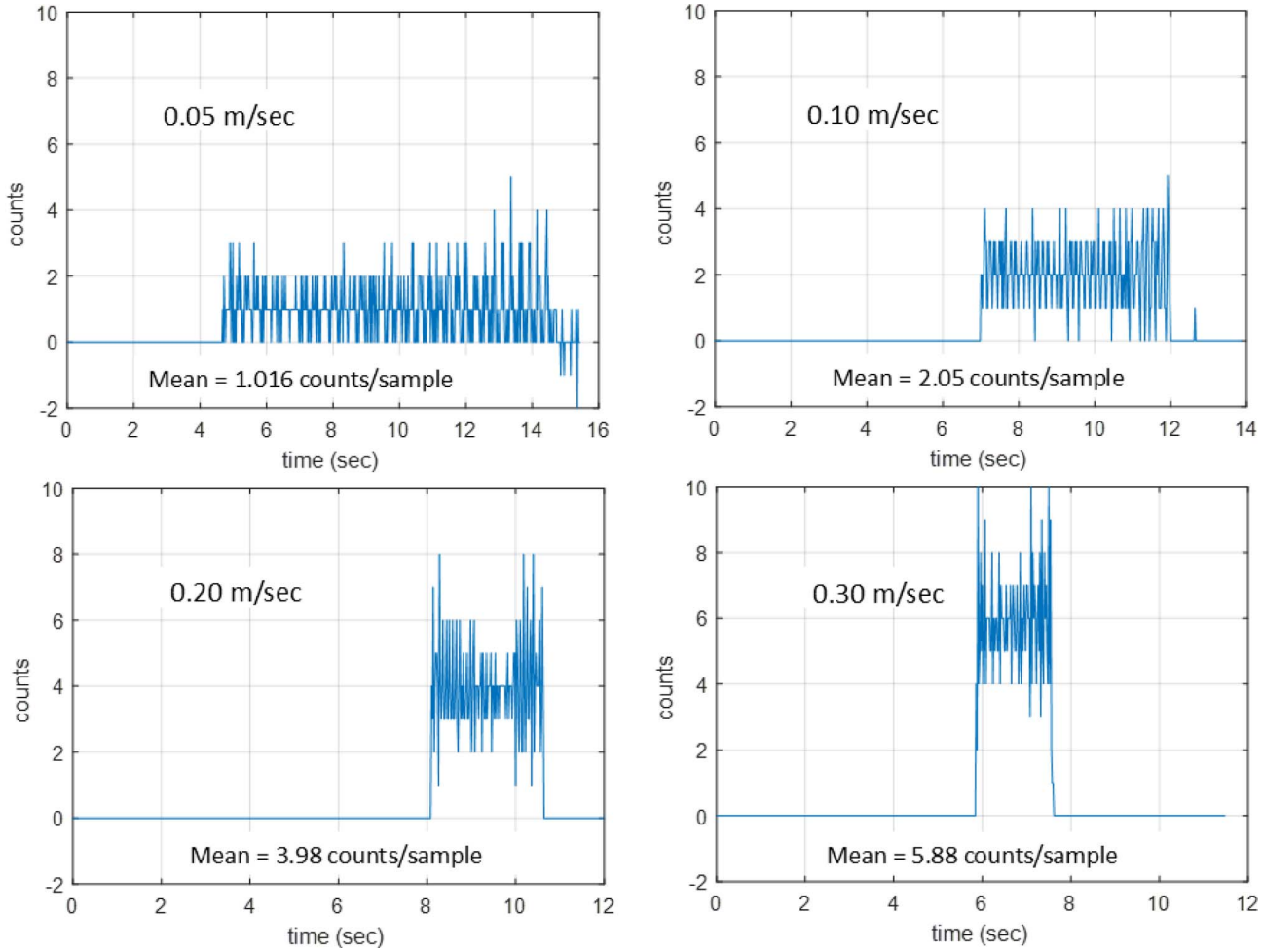


Figure 5. Optical flow sensor output on one axis during rail testing with approximately constant linear velocities.

ing the model with a purely constant velocity of 0.5 m/s, and second by driving the model with a velocity profile derived from the experimentally captured ΔX_k samples, creating a piecewise constant, continuous-time velocity signal as $\Delta X_k / \Delta T$. The results of the simulations are given in Fig. 8, with the constant velocity results on the left, and the experimentally measured velocity data on the right.

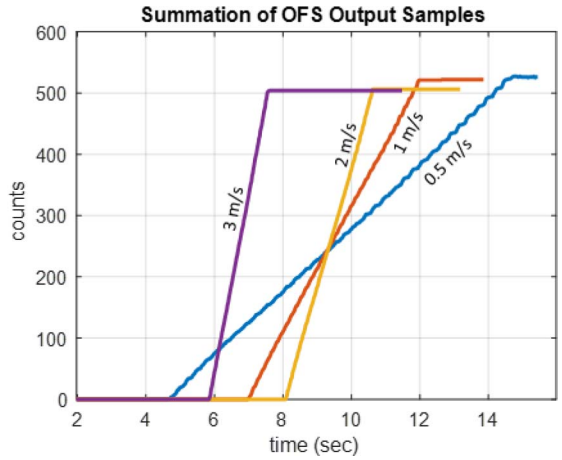


Figure 6. Running sum (discrete integration) of OFS output samples for four velocity setpoints.

For the constant velocity case, the OFS models' output is very regular and periodic, giving an output of 1 count most often, separated by an occasional output of 0. For the case on the right, in which the velocity is derived from the experimental data, the OFS model produces an output that precisely matches the output of the actual physical OFS device (see Fig. 5). This indicates that the model faithfully represents what is happening within the OFS in this scenario, i.e., constant velocity, low altitude, and zero angular velocity conditions. It should be noted that the new model produces results that match the output of the OFS when driving it with velocity time histories derived from the data of the other "constant" velocity cases. Since the simulated OFS outputs matched the experimental data of Fig. 5, they are not replotted here.

C. Test Results with Oscillatory Input Motion and Varying Altitude

Dependence of Sensor Output Flow on Altitude: Tests were performed to highlight the impact of altitude on the magnitude of flow vectors and the level of noise present

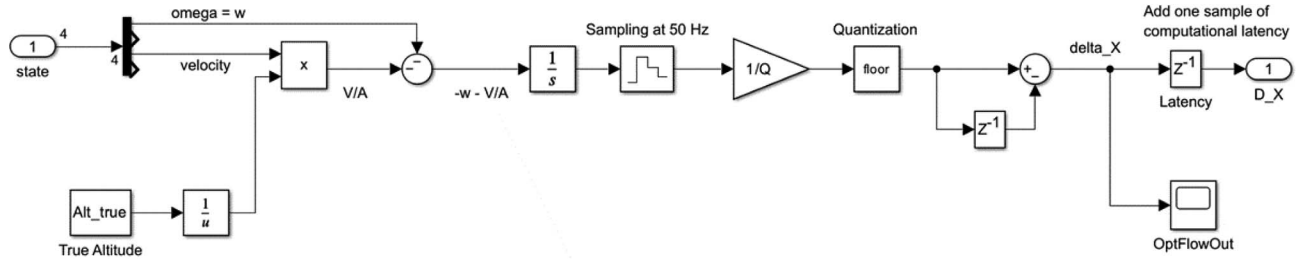


Figure 7. The OFS truth model.

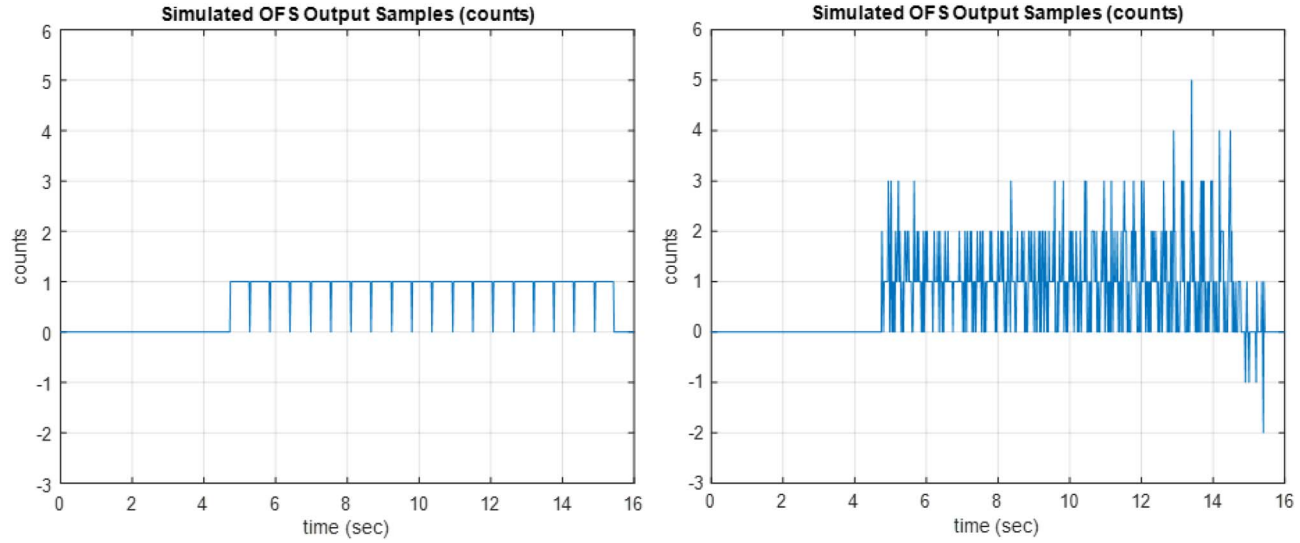
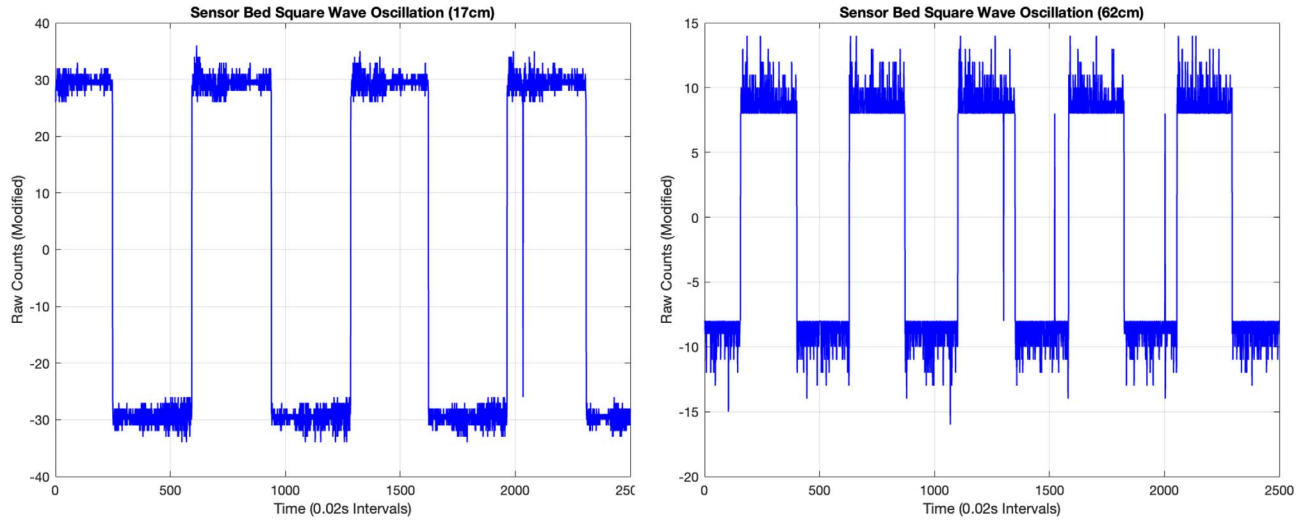


Figure 8. Comparison of OFS model outputs for constant and variable velocity, experimentally measured velocity data, velocity setpoint of 0.5 m/s.

in the data. These observations underscored the significance of altitude adjustments in accurately computing platform velocity. In this test, the rail system was moving in a triangular manner with directions of travel reversals separated by periods of constant velocity magnitudes of 0.02 m/s. The sample frequency was 25 Hz, so in a single sample, the slider moved 0.8 mm. At a distance of 17 cm above the ground, this equates to an angular change of 47 mrad ($8e^{-4}/17e^{-3}$). Given the quantization size $Q = 0.0015$ rad/count, this leads to an estimated ΔX_k optical flow count [see equation (11)] of $0.047/0.0015 = 31.4$ counts/sample. As noted in



(a) (b)

Figure 9. Sensor output data during periodic, constant velocity triangular wave motion at two altitudes, 17 and 62.5 cm.

Fig. 9a, the output is nominally equal to this expected output level. When elevated to 62.5 cm, the angular change becomes 12.8 mrad, and the output nominally 8.5 counts/sample. This is clearly reflected in the plot of Fig. 9(b).

D. Additional Characterization Tests

To better understand the impact of quantization and altitude, we expanded our experimental study to include other environmental variables to verify if consistent velocity readings are maintained at a fixed altitude.

1) Light Level Tests: Given the OFS's reliance on shadow changes for velocity determination, we examined how varying light conditions affect displacement accuracy. Two tests evaluated the PMW3901MB sensor's performance under different lighting: one at dusk and another in direct sunlight, both at 0.66 m altitude. The observed displacements were 0.59 m and 0.57 m, respectively, demonstrating the sensor's ability to produce comparable displacement readings across light levels.

2) Surface Quality Tests: The sensor's response to surface texture was also assessed, given the significance of landmarks and texture in displacement detection. Two scenarios were compared: one with a checkerboard-taped grid surface and another over a wooden floor. The displacement estimations were 0.61 m for the textured surface and 0.58 m for the nontextured, with the actual displacement set at 0.6 m. Interestingly, flow counts varied significantly between surfaces, with nontextured surfaces eliciting larger vector magnitudes due to fewer but more pronounced feature changes.

V. HORIZONTAL HOLD CONTROL DESIGN AND SIMULATION TESTING

A. Plant Dynamics

The OFS model developed above has been used in two ways. First, to create a Sensor Truth Model for use in simulation of the system involving the sensor. Secondly, to produce an algorithm for converting the OFS output signals to translational velocity measurements. These measurements in the x- and y-directions are used in generating the control signals that act to drive the drone velocity to zero, thereby holding the drone steady, or to some non-zero setpoint velocity provided by the pilot. A single-axis controller is developed and applied separately to each axis. One controls the translational motion of the drone in the Fwd-Rev direction, and the other controls the Left-Right translational motion.

To produce a dynamic model of the drone along a single axis, we assume that the platform has two degrees of freedom: translation along one axis (x-axis) and rotational about an axis that is perpendicular to the x-axis and horizontal (i.e. the pitch axis). A diagram of the drone is given below in Fig. 10. We assume the drone is

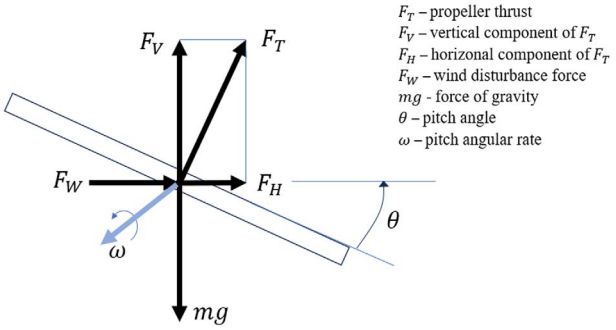


Figure 10. Free-body diagram of drone—single-axis horizontal motion.

not accelerating vertically, and therefore the weight of the drone is equal to the component of thrust in the vertical direction (F_V). The thrust of the props (F_T) has vertical component

$$F_V = F_T \cos(\theta) \tag{18}$$

and horizontal component

$$F_H = -F_T \sin(\theta). \tag{19}$$

The horizontal component of thrust causes the drone to accelerate. The vertical causes it to remain at a fixed altitude.

Since

$$F_V = mg, \tag{20}$$

we have:

$$F_T = \frac{mg}{\cos(\theta)}, \tag{21}$$

$$\begin{aligned} F_H &= -F_T \sin(\theta) \\ &= -\left(\frac{mg}{\cos(\theta)}\right)\sin(\theta), \\ &= -mg\tan(\theta) \end{aligned} \tag{22}$$

Using Newton's second law, the sum of the forces acting on a body along an axis equals the change in momentum along that axis

$$\sum F = m\ddot{x} \tag{23}$$

leads to:

$$m\ddot{x} = F_H + F_W \tag{24}$$

$$= -mg\tan(\theta) + F_W, \tag{25}$$

where F_W is the disturbance force due, for example, to wind gusts.

The dynamic angular response of the platform is assumed to be governed by a pitch control system having a second-order response with a rise time of 0.3 s. The corresponding second-order system is:

$$\frac{\theta}{u} = \frac{\Omega_p^2}{s^2 + 2\zeta\Omega_p s + \Omega_p^2} \tag{26}$$

with $\Omega_p = 5.4$ rad/s and $\zeta = 0.71$. In this model, the input u is the commanded pitch angle, or the pitch angle setpoint.

The following state vector is defined:

$$x = \begin{bmatrix} \omega & \theta & v & x \end{bmatrix}' = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}', \tag{27}$$

which leads to the following state space model:

$$\begin{aligned} \dot{x}_1 &= \Omega_p^2 (u - x_2) - 2\zeta \Omega_p \dot{x}_1 \\ \dot{x}_2 &= x_1 \\ \dot{x}_3 &= -g \tan(x_2) + F_W \\ \dot{x}_4 &= x_{3,} \end{aligned} \tag{28}$$

where F_W is the force of the wind and u is the control input.

B. Simulink Simulation Model and Control Design

A model of the single-axis translational motion of a drone platform, the OFS, IMU, and Proportional-Integral-Derivative (PID) controller was developed, with the top level block diagram shown in Fig. 11.

A velocity setpoint, shown at left, is the signal injected by the drone pilot. This command is compared to the velocity achieved to produce an error signal, which enters a PI controller producing a commanded pitch angle u same as in equation (28) into the plant.

The OFS block content is provided in Fig. 7. The IMU model is pictured in Fig. 14.

The coefficients and parameters required to simulate this model and reproduce the results presented in this paper are detailed in Table VI. A PI controller was designed with gains specifically chosen to achieve a 1 s step response rise time and sufficient integral action to mitigate the effects of constant wind disturbances. The PI controller coefficients, along with other essential parameters, including the UAV's rotational dynamic coefficients for an attitude-stabilized quadcopter, are provided in Table VI. These parameters were used in the simulations model of Figs. 7 and 11–15, and simulation results of Figs. 16 and 17.

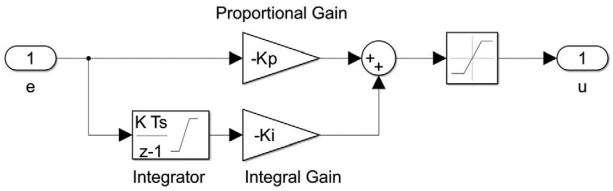


Figure 12. Controller with proportional and integral paths, and pitch angle command saturation.

C. Simulation Test Results

Simulation tests were performed to assess the effectiveness of a horizontal velocity control system for a drone, incorporating the OFS and an IMU. The results were generated using the single-axis dynamic model discussed earlier, a model that includes a continuous-time fourth-order representation of plant dynamics and discrete-time models for the OFS, IMU, and the digitally implemented PI controller. Both sensors operate at a frequency of 50 Hz, and each sensor model includes a computational latency of one sample period, 20 ms. There are no other impairments or noise sources added to the sensor outputs.

D. Step Response in Velocity

The step response test is conducted to ensure that the system responds to changes in velocity setpoints with an appropriate speed of response sufficient to meet a drone pilot's expectations for responsiveness yet avoiding jerky or erratic behavior. The results of this simulation test are presented in Fig. 16. At the 5 s mark, a step change command is initiated by the pilot, commanding a velocity change from 0 to 2 m/s. This is shown in Fig. 16(a) where we see that the velocity increases from 0 to 2 m/s with a rising time of less than 1 s and a setting time of about 2 s. The true velocity exceeds the setpoint with an acceptable level of overshoot, settling at 2 m/s as desired. To effect this change, the controller caused the platform's pitch angle to drop to -20° as shown in Fig. 16(b). The drone then pitches forward and accelerates along the positive x-axis. During this transient response, the OFS and IMU generated outputs are depicted in Fig. 16(c) and (d), respectively.

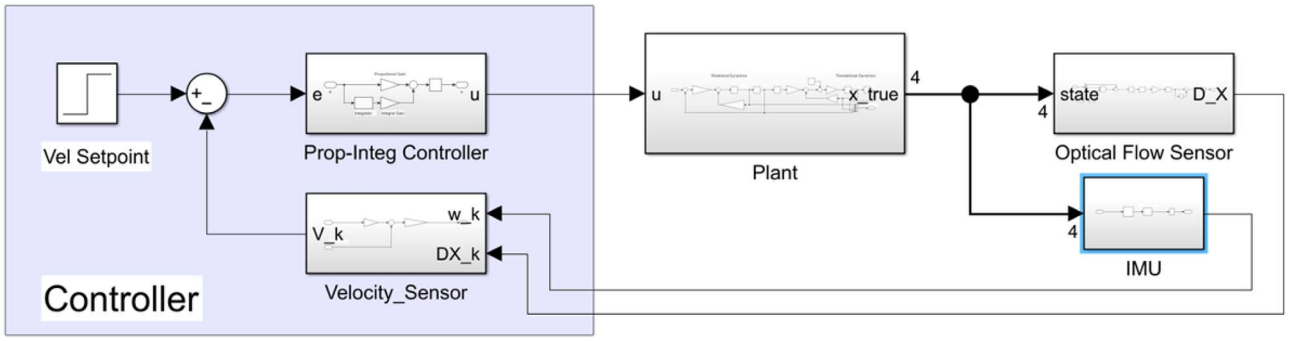


Figure 11. Control system model with single-axis drone (plant) and sensors.

Table VI
Simulation Model Parameters

```
% Translational dynamics
m = 2.3;                    % kg
g = 9.81;                   % m/sec^2
b = 0.2;                    % N/(m/sec)
% Rotational dynamics
OmegP = 5.4;                % natural frequency (rad/sec) with rise time ~= 0.3 sec
zet = 0.71;                 % damping factor
Fw = 1;                     % Wind Force (N)
% Controller Gains
Kp = 0.2;                   % Proportional Gain (radians/(m/sec))
Ki = 0.062;                 % Integral Gain (radians/m)
Pitch_max = 20/57.3;        % pitch command saturation limit (20 degrees)
% Sensor parameters
Fs = 50;                    % Sample Frequency (Hz)
Alt_true = 1;               % true altitude (m)
Alt_meas = 1;               % measured altitude from baro (m)
Q = 0.09 / 57.3;            % Opt Flow quantization (radians)
```

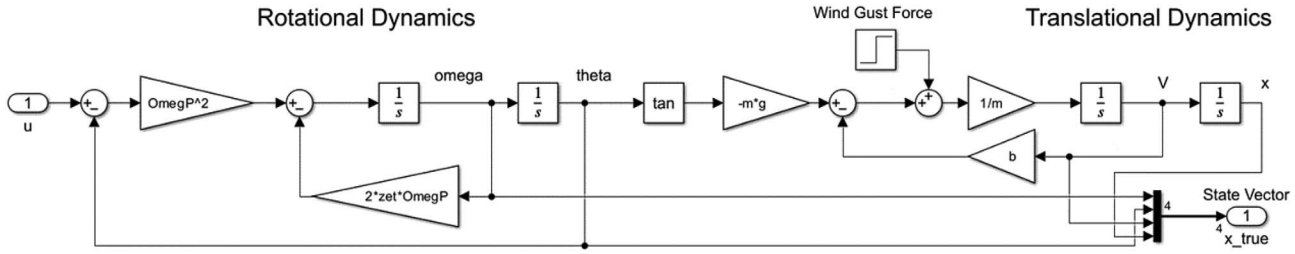


Figure 13. Plant model based on drone dynamics.

E. Response to a Wind Gust

The purpose of the wind gust test is to verify that the controlled system's response to a wind gust is acceptably small. The results of this simulation test are as follows: Commencing at 5 s, a headwind force of 4 Newtons is applied, representing a significant force relative to the drone's weight of 22 Newtons. The headwind drives the drone backward and the velocity sensed is negative as shown in Fig. 17(a). In response to this horizontal velocity disturbance, the controller initiates a 10° downward pitch, as illustrated in Fig. 17(b), to counteract the wind. The drone is driven backwards a distance of about 2.7 m at which point it comes to a halt with zero velocity. The integral part of the PI controller has developed a 10° pitch angle to hold the drone against the headwind. The pilot can command the drone to follow a positive velocity setpoint if they wish to move the drone back to its original location. The OFS output during this time is shown in Fig. 17(d).

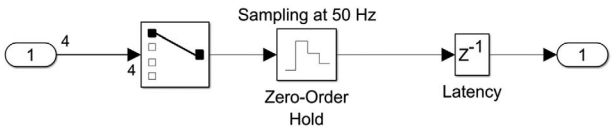


Figure 14. Inertial measurement unit Simulink block diagram.

F. Very Slow Movement

The very slow movement test is designated to verify that the highly quantized output from the OFS is adequate to enable the velocity controller to induce extremely low-velocity horizontal motion. The results of this simulation test are illustrated in Fig. 18. The velocity-commanded setpoint transitions from 0 to 1 cm/s at 2 s [see Fig. 18(a)]. At this speed, the drone moves at a barely perceptible pace. The resulting OFS output is a pulse stream with a very low duty cycle, as shown in Fig. 19, with pulses that alternate between 0 and −1 at a rate of approximately 7 Hz. Noting that we are sampling the sensor's output at a rate of 50 Hz, this stream is producing one count over a period of 7 or 8 samples. The results show that the controller achieves the commanded 1 cm/s velocity as noted in Fig. 18(a). Although the pulsing introduces some jumpiness in the control signal (pitch angle command), the plant's dynamics effectively filter out this "noise," resulting in a smooth achieved veloc-

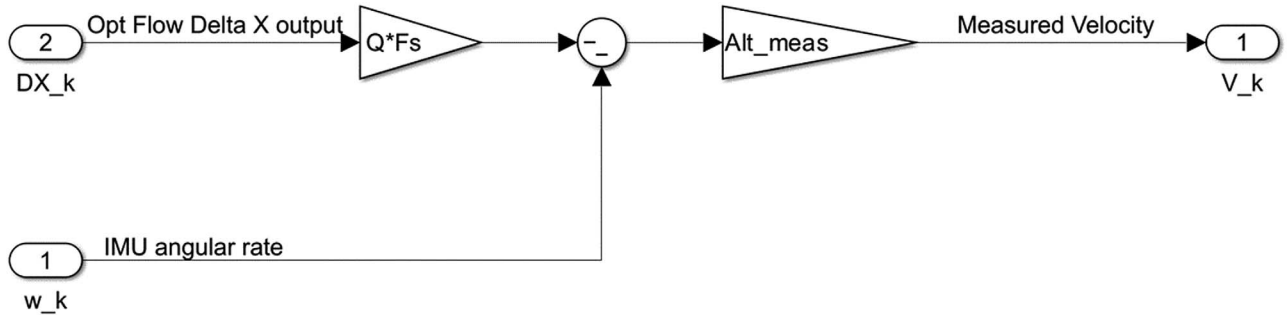


Figure 15. Velocity sensor Simulink block diagram.

ity. The IMU's Pitch Rate output during this period is presented in Fig. 18(d). The successful completion of this test indicates that the OFS's output is indeed adequate for the velocity controller to induce and regulate extremely low-velocity horizontal motion, thereby affirming its suitability for tasks that demand high levels of precision and control.

Zooming into the OFS output in Fig. 19, we see that it does not produce a non-zero output for 0.7 s after the commanded step change at 2 s. During this period, the sensor has not yet experienced sufficient motion to generate an output. Subsequently, once the output initiates, the sensor produces a pulsed signal stream with a low - duty cycle. The average output is effectively a fraction of one count.

VI. CONCLUSIONS AND FUTURE WORK

The ability to effectively process and integrate data from a diverse set of sensors is important in UAV navigation, localization, and control, and can improve system operational performance and safety. In this paper, a new OFS model was presented, a model that captures the impact of pixel quantization and uses platform angular

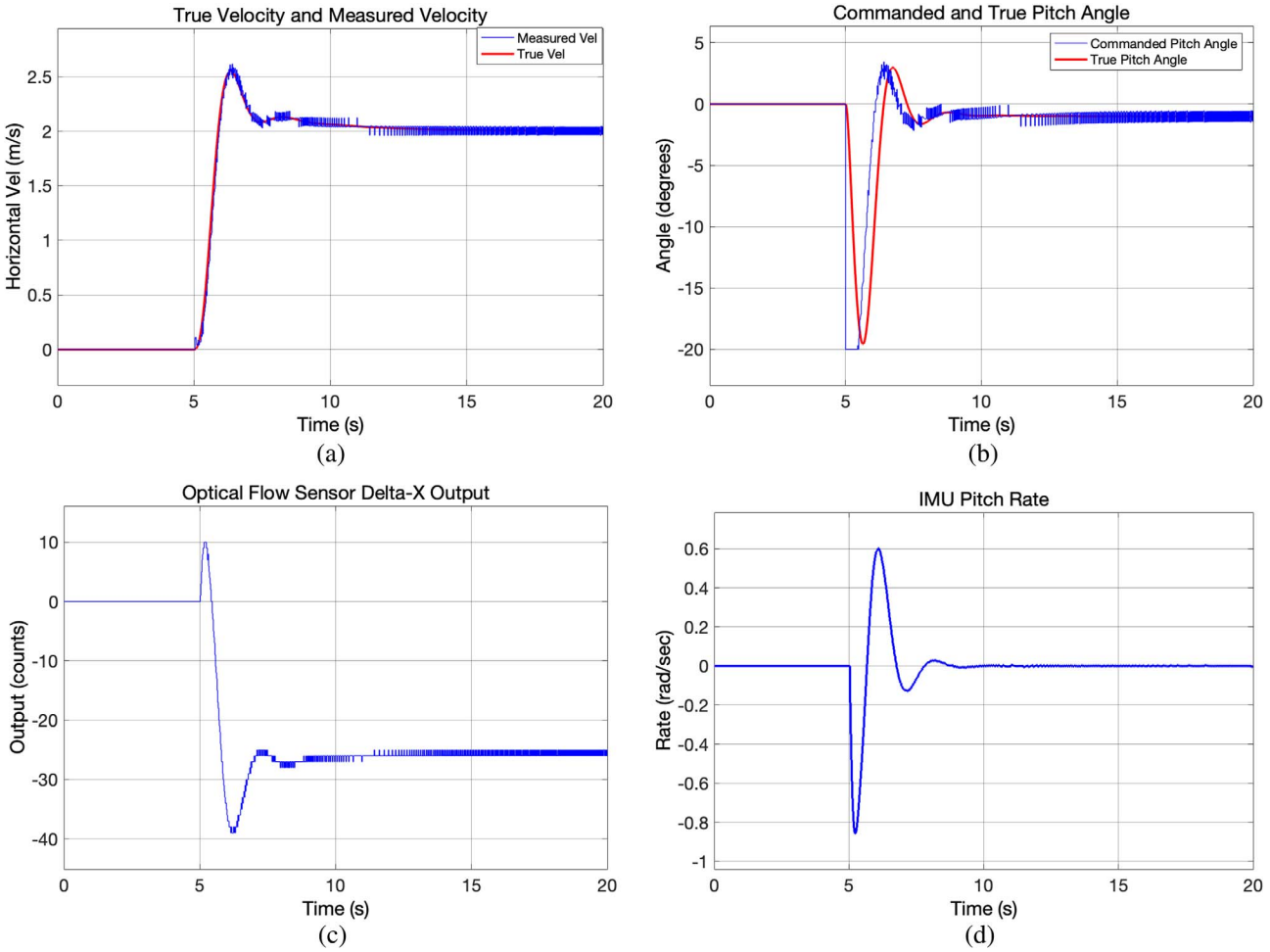


Figure 16. Step response in velocity.

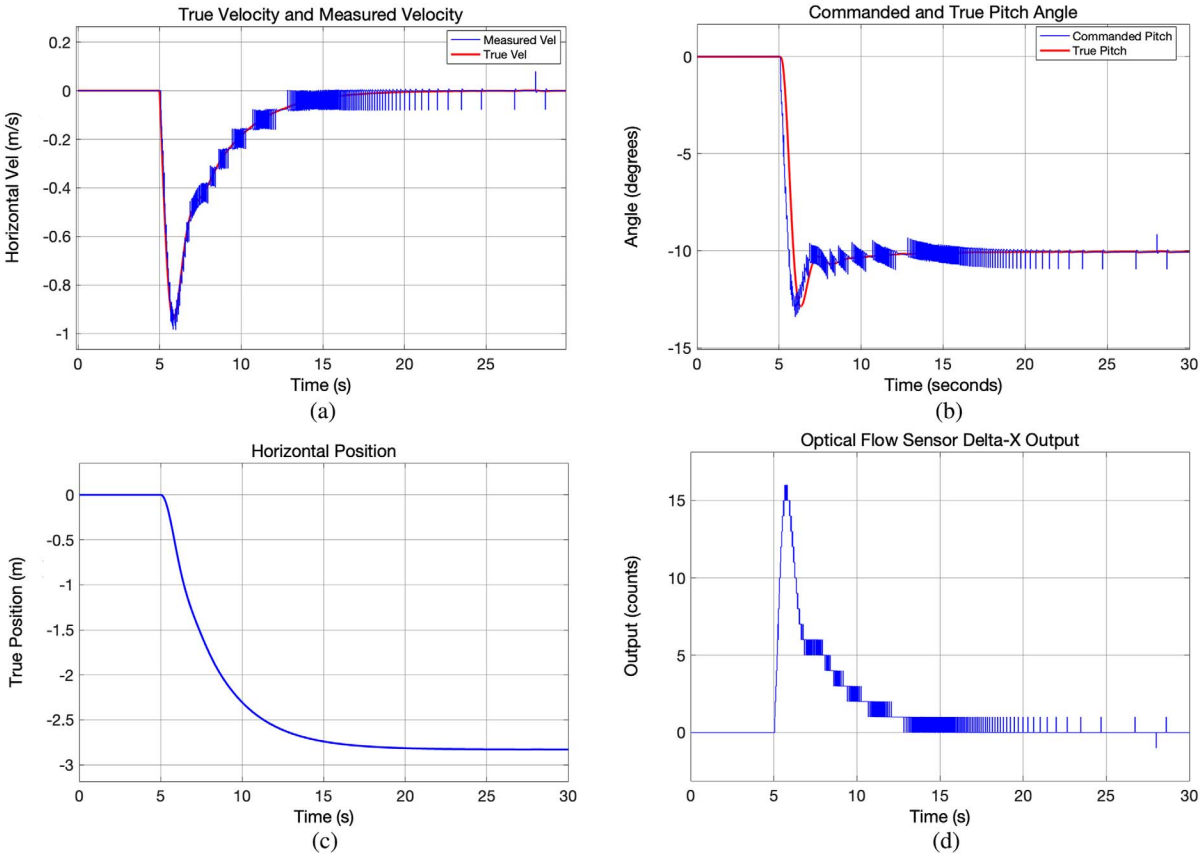


Figure 17. Response to wind gust.

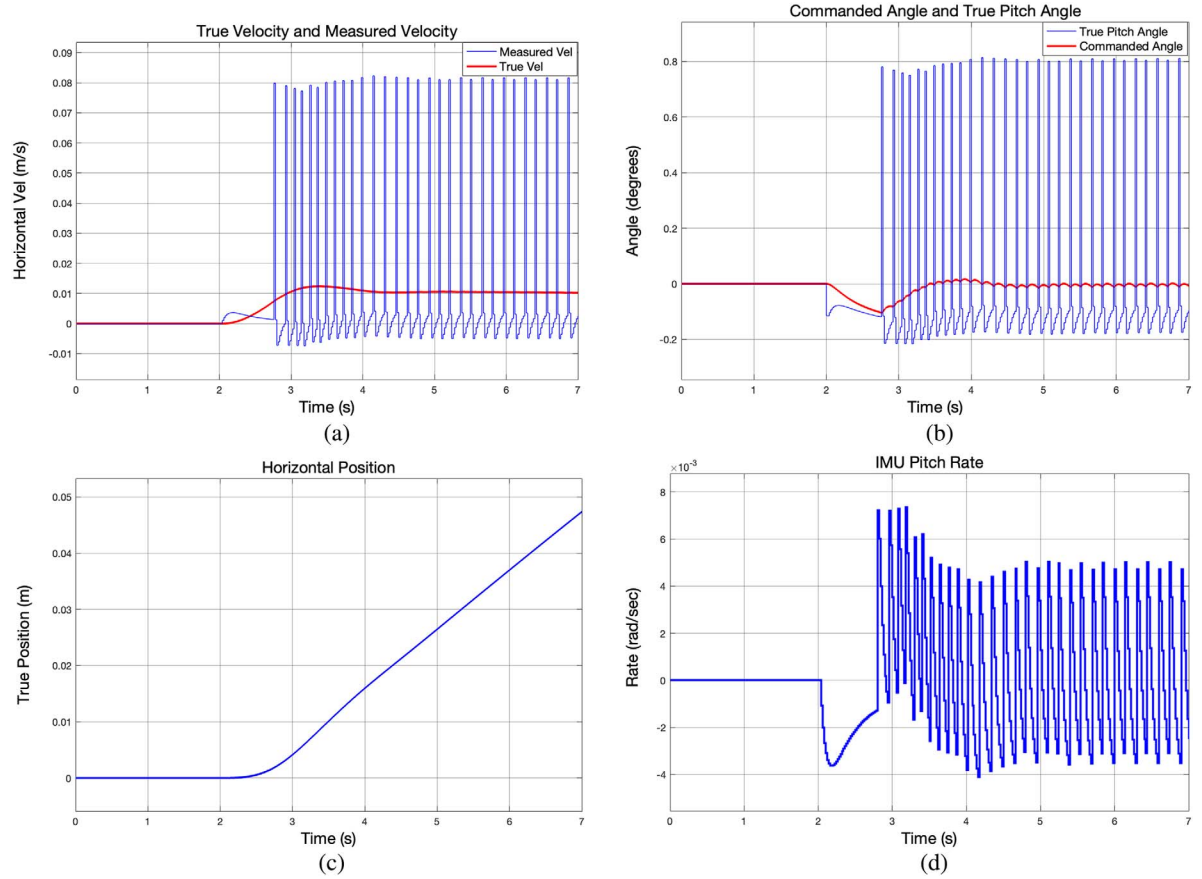


Figure 18. Very slow commanded velocity of 1 cm/s (0.01 m/s).

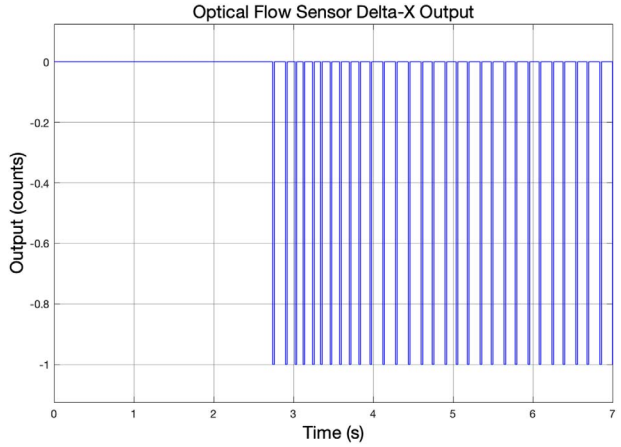


Figure 19. Optical flow sensor output during very slow commanded velocity of 10 cm/s.

rate and linear velocity to generate the OFS output. This model accurately reflects a key feature of the sensor—the fact that it is measuring angular displacement rather than angular velocity. How this impacts platform horizontal velocity control is discussed. From this new model, a velocity measurement equation was derived that fuses platform IMU and OFS data. The combination of data from these sensors has been shown to enable the generation of accurate velocity estimates in two dimensions, a feature particularly beneficial in GPS-compromised environments. Using this velocity measurement as feedback, a control system for horizontally stabilizing drone location has been proposed. Future work will include the assessment of IMU bias and random noise error on horizontal hold control system performance.

REFERENCES

[1] S. Huang and G. Dissanayake
"Robot localization: An introduction,"
in *Wiley Encyclopedia of Electrical and Electronics Engineering,* J. G. Webster, Ed. Hoboken, NJ, USA: Wiley, 2016, pp. 1–3.

[2] T. D. Barfoot
State Estimation for Robotics: Second Edition, Introduction. Cambridge, U.K.: Cambridge Univ. Press, 2024, pp. 1–4.

[3] Q. Yuan and I.-M Chen
"Localization and velocity tracking of human via 3 IMU sensors,"
Sensors Actuators A: Phys., vol. 212, pp. 25–33, 2014.

[4] S. M. Joshi
"Adaptive control in the presence of simultaneous sensor bias and actuator failures," National Aeronautics and Space Administration, Langley Research Center, Hampton, VA, NASA Tech. Rep. NASA/TM-2012-217231, 2012.

[5] B. Wang
"Reduced integration time improves accuracy in dead reckoning navigation systems,"
Analog Dialogue, vol. 47, no. 7, pp. 722–723, July 2013.

[6] M. L. Abell and J. P. Braselton
Differential Equations with Mathematica, 5th ed. San Diego, CA: Academic Press, 2022.

[7] H. B. Gilbert, O. Celik, and M. K. O'Malley
"Long-term double integration of acceleration for position sensing and frequency domain system identification," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2010, pp. 453–458 .

[8] Y. Nabavi, D. Asadi, and K. Ahmadi
"Image-based UAV position and velocity estimation using a monocular camera,"
Control Eng. Pract., vol. 134, no. 3, 2023.

[9] F. Santoso, M. A. Garratt, and S. G. Anavatti
"Visual–inertial navigation systems for aerial robotics: Sensor fusion and technology,"
IEEE Trans. Automat. Sci. Eng., vol. 14, no. 1, pp. 260–275, Jan. 2017.

[10] MathWorks
"What is SLAM (Simultaneous Localization and Mapping)—MATLAB," Accessed: April 7, 2024. [Online]. Available: https://www.mathworks.com/discovery/slam.html

[11] S. Lange, N. Sünderhauf, and P. Protzel
"Autonomous landing for a multirotor UAV using vision," in *Proc. SIMPAR 2008 Int. Conf. Simulation Simul., Model. Program. Auton. Robots*, 2008, pp. 482–491 .

[12] Z. Y. Chen, W. C. Chen, X. M. Liu, and C. Song
"Fault-tolerant optical flow sensor/SINS integrated navigation scheme for MAV in a GPS-denied environment,"
J. Sensors, vol. 2018, pp. 9678505, 2018.

[13] F. Kendoul, I. Fantoni, and K. Nonami
"Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles,"
Robot. Auton. Syst., vol. 57, no. 6–7, pp. 591–602, 2009.

[14] T. G. Kim, D. M. Kim, S. K. Kim, Y. S. Kim, and S. H. Han
"Improved optical sensor fusion in UAV navigation using feature point threshold filter,"
Int. J. Aeronautical Space Sci., vol. 23, no. 1, pp. 157–168, 2022.

[15] X. Li, C. X. He, and Y. M. Tang
"Improved modeling and fast in-field calibration of optical flow sensor for unmanned aerial vehicle position estimation,"
Measurement, vol. 225, 2023, Art. no. 114066.

[16] S. Lee and J.-B. Song
"Robust mobile robot localization using optical flow sensors and encoders," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004.

[17] H. Didari, F. Lotfi, H. Taghirad, and S. Germi
"Position estimation for drones based on visual SLAM and IMU in GPS-denied environment," in *Proc. 7th Int. Conf. Robot. Mechatron.*, 2019, pp. 120–124.

Djedjiga Belfadel received the Ph.D. degree in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2015. She is currently an Associate Professor and the ECE Graduate Program Director in the Department of Electrical and Biomedical Engineering Department, Fairfield University, Fairfield, CT, USA. Her research traverses a wide breadth of estimation theory, with a specific emphasis on practical applications such as drone navigation and target tracking. Her scholarly contributions extend to space-based infrared (IR)/electro-optical (EO) sensors, signal and image processing, machine learning, and big data. Alongside her research, she is committed to enhancing engineering education and boosting the representation of women and underrepresented groups within the engineering sector.

John Cain received the M.S. degree in Electrical and Computer Engineering in 2024 and the B.S. degree in Computer Science in 2022, both from Fairfield University, Fairfield, CT, USA, where he worked as a graduate research assistant. He is currently a firmware engineer at SmartSense. His research interests include sensor systems/communication and embedded electronics.

David Haessig received the B.S. and M.S. degrees in mechanical engineering from Lehigh University, Bethlehem, PA, USA and the Ph.D. degree in electrical engineering from NJIT, Newark, NJ, USA. With over 40 years of experience in communication, control, and navigation systems, he is currently the Chief Scientist at AuresTech Inc., Bridgewater, NJ USA. Previously, at BAE Systems, he was a Technical Director II, leading significant projects like the Joint Tactical Radio System's Anti-Jam Wideband Networking Waveform and developing UAV SATCOM control systems. He also worked on submarine control systems at General Dynamics and advanced wireless technologies at Lucent Technologies, where he earned a Bell Labs Gold Award. He is a registered Professional Engineer in New Jersey, an IEEE Life Senior member, and an Adjunct Professor at NJIT. He holds 13 patents and has 27 technical publications.

Cherif Chibane is the Founder and Chief Technology Officer at AuresTech Inc., Bridgewater, NJ, USA, where he leads R&D in configurable computing for wireless communications. He has also served as a Senior Research Scientist at MIT Lincoln Laboratory and as a Science and Technology PM at Draper Laboratory, focusing on advanced communication systems for NASA and DARPA. At BAE Systems, he led the development of System On Chip technology and UAV navigation solutions. Cherif holds an MSEE from Fairleigh Dickinson University. His expertise includes RF systems, control systems, and embedded systems development.

A Multiple Extended Object Tracker with the Gaussian Process Model Utilizing Negative Information

MARTIN BAERVELDT
MICHAEL ERNESTO LÓPEZ
EDMUND FØRLAND BREKKE

In multiple extended object tracking, the Poisson multi-Bernoulli mixture (PMBM) tracker is considered state-of-the-art. Originally, it was presented with the gamma Gaussian inverse Wishart (GGIW) target model, which is a random matrix model. When tracking larger objects using a light detection and ranging (LiDAR) sensor, measurements are generated by the contour rather than the whole target surface, and it is beneficial to model this with the target model. A target model that has this capability is the Gaussian process (GP) extent model. This paper presents a PMBM tracker using this target model. We also discuss considerations related to the use of the GP model in the PMBM framework. Secondly, we present improvements in the target model that increase the robustness of the model by dealing with the inherent non-linearities using the Gauss–Newton method. Furthermore, we incorporate an improvement to the tracker that utilizes the concept of negative information to generate virtual measurements that are then used in the Gauss–Newton optimization. In relation to this, we also present an occlusion model that utilizes the same negative information model to ensure that the state estimate is consistent in the presence of occluding targets. The presented methods are compared to the GGIW-PMBM tracker on simulated and real LiDAR data gathered from maritime vessels. The results show that the GP model outperforms the GGIW model by providing a better estimate of the extent and more accurate tracking, as measured by the GOSPA metric. Utilizing negative information for state estimation and occlusion modeling further improves the state estimate and tracking performance.

Manuscript received March 25, 2024; revised July 8, 2024; released for publication January 10, 2025

Refereeing of this contribution was handled by Florian Meyer.

The work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 955.768 (MSCA-ETN AUTOBarge). This publication reflects only the authors' view, exempting the European Union from any liability. Project website: http://etn-autobarge.eu/. This work was supported in part by the Research Council of Norway through Projects 295033 and 309230.

The authors are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7034 Trondheim, Norway (e-mail: martin.baerveldt@ntnu.no; ernesto.lopez@ntnu.no; edmund.brekke@ntnu.no).

1557-6418/2024/$17.00

I. INTRODUCTION

Target tracking, the issue of estimating the kinematic state of one or several objects, has long used the point approximation when parsing sensor data. With the advent of high-resolution sensors, it is now common that a measurement source gives rise to multiple measurements. This has given rise to extended object tracking models, which enable the modeling of a target's extent in addition to its kinematic properties by inferring information from the spatial distribution of these measurements [1]. Initial approaches assumed that the spatial distribution of the measurements could be modeled by a Gaussian distribution around the center of the target extent. This results in an estimated ellipsoidal extent [2]. This is known as the random matrix model. A version of this model, the gamma Gaussian inverse Wishart (GGIW) model, was used to demonstrate an extended object Poisson multi-Bernoulli mixture (PMBM) filter [3] based on the original PMBM filter [4]. This filter has also been used with a set of trajectory framework [5]. Work has also been conducted to investigate improvements with regard to data association [6] and reducing complexity by approximating the PMBM posterior as a PMB [7]. In [8], a factor graph representation of the PMBM posterior was used to present a PMB filter using the particle belief propagation method presented in [9]. However, the random matrix model is not the only target model for extended objects. Another method, the random hypersurface model, models the extent using star-convex shapes and represents the shape using a parametrization of the contour [10]. This enables the modeling of more complex shapes. It also allows an easier way to model measurements that originate from the contour, such as measurements generated by an LiDAR sensor. The most promising and investigated of these models uses Gaussian processes (GP) to estimate the extent [11]. This model has been further improved with different estimation methods [12], [13] and augmented with the use of virtual measurements that use negative information [14].

The GP model has also been used to implement multi-object tracking filters, such as the δ-GLMB filter [15] and the probability hypothesis density (PHD) filter [16]. In previous work comparing different filter structures, it has been shown that the PMBM has a more efficient structure and it can initialize a track faster with its Poisson birth model as compared to Bernoulli birth models [17].

A key challenge to consider for multiple extended object tracking is occlusion, since this will cause objects to not generate measurements. Previous methods have looked at the specific target model and calculated a non-constant probability of detection [18], [19] based on the state of other targets using the GGIW model; this has also been done in the context of the extended object PMBM filter [3]. Another approach was presented in [20], which calculates an occlusion likelihood based on random variables inferred from the current set of targets. This occlusion likelihood is represented by a Gaussian mixture, which can be used to update the state of undetected objects, infer the existence of objects, and inform data association.

In this paper, we aim to present an extended object PMBM tracker using the GP target model with the applicable prediction and update formulas. We also provide an example of a Poisson birth density for the GP target model. Furthermore, we introduce an improvement to dealing with the nonlinearity of the measurement model for the GP target model. In addition, we present a method to incorporate negative information into the estimate of object states for this model by the use of virtual measurements, similar to the method in [14]. To ensure that this can be utilized in a multi-object context, we also utilize these virtual measurements to model target occlusion. Finally, we show the application of the developed tracker on LiDAR data gathered by tracking smaller maritime vessels. This article is an extension of [21] with the additional inclusion of the method of using negative information for state estimation and occlusion handling, as well as a refined criterion for initialization for the Gauss–Newton (GN) optimization. It also includes a more complex simulation scenario and a more detailed exposition of the method. The article is organized as follows: In Section II, we introduce the relevant theory and previous work, and in Section III, we present the improvement to the GP target model, along with the incorporation of negative information into the state estimation. In Section IV, we present the applicable prediction and update formulas for a PMBM filter utilizing the GP target model, the specific approximations that are used in this paper, as well as the utilization of negative information to model occlusion. In Section V, we present the simulation study, and in Section VI, the results on the real maritime data are presented.

II. BACKGROUND

In this section, we present a summary of the method of extent estimation using GP presented in [11]. Then we outline the theory related to the extended object PMBM filter, which was presented in [3].

A. Notation

In the following, we present the most significant variables as well as the notation used in this work.

NOMENCLATURE

$(\cdot)^*$	Quantities related to the virtual measurements.
$(\cdot)^b$	Quantities related to the birth process.
$(\cdot)^f$	Quantities related to the radial function f.
$(\cdot)^u$	Quantities related to unknown targets.
$(\cdot)^u_n/(\cdot)_{u_n}$	Quantities related to nth component of the PPP intensity of unknown targets.
$(\cdot)^{j,i}$	Quantities related to the ith Bernoulli in the jth multi-Bernoulli in an MBM.
$(\cdot)_k$	Quantity at time step k.
α	Shape parameter of gamma distribution.
β	Inverse scale parameter of gamma distribution.
η_f	Forgetting factor for extent prediction.
η_v	Occlusion correction factor for gamma distribution.
η_γ	Forgetting factor for gamma prediction.
$\hat{(\cdot)}$	Estimation of a random variable.
$\kappa_{min/max}$	The minimum and maximum angle occupied by an extended object.
λ_c	Clutter rate.
λ_m	Measurement rate.
$\mathbb{I}$	Set of targets in a multi-Bernoulli.
$\mathbb{I}^{j,i}_O$	Set of occluding targets for target i in the jth MB.
$\mathbb{I}^{j,i}_{pO}$	Set of partially occluding targets for target i in the jth MB.
$\mathbb{J}$	Set of components in an MBM.
$\mathbf{F}$	Process model transition matrix.
$\mathbf{H}$	Measurement matrix.
$\mathbf{h}(\mathbf{x})$	Vector of predicted measurements for a target.
$\mathbf{P}$	State covariance matrix.
$\mathbf{Q}$	Process noise covariance matrix.
$\mathbf{R}$	Measurement noise covariance matrix.
$\mathbf{x}$	State space vector of a target.
$\mathbf{x}^c$	Position of target centroid.
$\mathbf{z}_k$	Vector of measurements for one scan at timestep k.
ϕ	Heading of a target.
$\sigma_{(\cdot)}$	Standard deviation of a quantity.
θ	Angle used in the radial function f.
D	A PPP intensity function.
f	Probability density function.
$f(\theta)$	Radial function which is estimated by a GP.
$k(\theta, \theta')$	Covariance function for a GP.
L^j_C	Likelihood of measurement cell assignment.
l_C	Predictive likelihood of a measurement set being assigned to an estimated target.
P_D	Probability of detection.
P_G	Gating probability.
P_O	Probability of occlusion.
P_S	Probability of survival.
P_{pO}	Probability of partial occlusion.
Q_D	Probability of missed detection.
r	Existence probability.
T	Sampling time.
v^i	Visibility ratio of object i.
w^j	Weight of the jth multi-Bernoulli in an MBM.
w_e	Window length of gamma prediction.
z^l	The lth measurement in a set.
$\theta^{(G)}$	θ defined in the global frame.
$\theta^{(L)}$	θ defined in the local target frame.

B. Gaussian Process

A GP can be considered a distribution over functions [22]. It is completely specified by its mean function $m(\gamma)$ and covariance function $k(\gamma, \gamma')$. Using GPs to estimate a radial function means that we can write

$$f(\theta) \sim \mathcal{GP}(m(\theta), k(\theta, \theta')), \tag{1}$$

where $f(\theta)$ defines the radius at angle θ. We want to estimate the values of this function using measurements of only some of its values. This is a method known as GP regression. We define a vector of N different points known as test points $\Theta^f = [\theta_1^f \quad ... \quad \theta_N^f]$. Further, we define a measurement model as

$$z_k = f(\theta_k) + \eta_k, \quad \eta_k \sim \mathcal{N}(0, \mathbf{R}), \tag{2}$$

where z_k is a measurement of the unknown function, θ_k is the training input, which is the point at which the measurement is taken, and η_k is the measurement noise. If we have m measurements of the function, we define $\mathbf{z} = [z^1 \quad ... \quad z^m]$ and their corresponding input values $\Theta = [\theta_1 \quad ... \quad \theta_m]$ to learn the function values for Θ^f. In the original paper [11], it is shown that the state $\mathbf{x}^f = \left[f(\theta_1^f) \quad ... \quad f(\theta_N^f)\right]^\mathsf{T}$, which defines the extent, can be recursively estimated using the following state space model:

$$\begin{aligned} \mathbf{x}_{k+1}^f &= \mathbf{F}^f \mathbf{x}_k^f + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}^f) \\ z_k &= \mathbf{H}^f(\theta_k)\mathbf{x}_k^f + \epsilon_k^f, \quad \epsilon_k^f \sim \mathcal{N}(0, \mathbf{R}^f). \end{aligned} \tag{3}$$

The measurement model is in turn given by the following matrices:

$$\begin{aligned} \mathbf{H}^f(\theta_k) &= \mathbf{K}(\theta_k, \Theta^f)\left[\mathbf{K}(\Theta^f, \Theta^f)\right]^{-1} \\ \mathbf{R}^f(\theta_k) &= k(\theta_k, \theta_k) + \mathbf{R} - \mathbf{H}^f(\theta_k)\mathbf{K}(\Theta^f, \theta_k), \end{aligned} \tag{4}$$

where $\mathbf{K}$ in turn is defined as a covariance matrix where the elements are made up of the elementwise evaluation of the covariance function $k(\theta, \theta')$. The process model is defined by

$$\mathbf{F}^f = e^{-\eta_f T}\mathbf{I}, \quad \mathbf{Q}^f = \left(1 - e^{-2\eta_f T}\right)\mathbf{K}(\Theta^f, \Theta^f), \tag{5}$$

where T is the sampling time. The parameter for this model is a forgetting factor η_f.

1) Covariance Functions: As can be inferred from the equations above, the covariance function is the component that defines the GP and any prior information about the shapes. Therefore, we want to encode the periodicity of $f(\theta)$ in the covariance function. Such a function was presented in the original article as

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{1}{2l^2}\left(\sin^2\frac{|\theta-\theta'|}{2}\right)} + \sigma_r^2. \tag{6}$$

This function gives a high correlation for two function values $f(\theta)$ and $f(\theta')$ when their respective angles are closer and a lower correlation when they are further apart. σ_f, the signal variance, defines the magnitude of this correlation, and the length-scale l defines the distance on which it acts. σ_r is a constant bias term, which can be used to formulate the GP as a zero mean GP by stating $m(\theta) \sim \mathcal{N}(0, \sigma_r^2)$, integrating out $m(\theta)$, and adding it as a contribution to the covariance function.

It is also desirable to design a covariance function that encodes axial symmetry since, in many cases, targets are symmetric about the longitudinal axis. Since the longitudinal axis is defined at $\theta = 0$, this is equivalent to an even function. One such function can be built using the smallest signed angle function

$$ssa(\theta) := \pi - [(\pi - \theta)(\mathrm{mod}2\pi)], \tag{7}$$

i.e., $ssa(\theta)$ is the only angle in $(-\pi, \pi]$ such that $ssa(\theta) \equiv \theta$. The absolute value of this function is both 2π-periodic and even, which is equivalent to the radial curve of $f(\theta)$ being closed and symmetric about the longitudinal axis, as desired. We define the symmetric covariance function as

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{1}{2l^2}(|ssa(\theta)| - |ssa(\theta')|)^2} + \sigma_r^2 + \sigma_n^2 \delta(\theta, \theta'), \tag{8}$$

where σ_n is a noise term that models individual features of a specific point, such as a sharp corner, by adding a small term to each diagonal part of the constructed covariance matrix. This has the added benefit of regularizing the covariance matrix [23].

C. State Space Model for Extended Targets

To perform joint estimation of the extent and state of the target, an augmented state space vector is defined

$$\mathbf{x}_k = \left[\mathbf{x}_k^{c\,\mathsf{T}} \quad \phi_k \quad \left(\mathbf{x}_k^{*\mathsf{T}}\right) \quad \mathbf{x}_k^{f\,\mathsf{T}}\right]^\mathsf{T}, \tag{9}$$

where $\mathbf{x}_k^c$ is the position of the centroid of the target from which the extent is defined, ϕ is the heading of the target, and $\mathbf{x}_k^*$ are any additional kinematic states of the target. In the original paper, these are the velocity in each direction in 2D $\dot{\mathbf{x}}_k^c$, and the angular velocity $\dot{\phi}_k$. We use the same state space vector in this paper.

For this augmented state space vector, we define the following state space description:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{w}, \quad \mathbf{w} \sim \mathcal{N}(0, \mathbf{Q}) \\ \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k) + \boldsymbol{\eta}_k, \quad \boldsymbol{\eta}_k \sim \mathcal{N}(0, \mathbf{R}_k), \end{aligned} \tag{10}$$

where $\mathbf{z}_k, \mathbf{h}_k(\mathbf{x}_k)$, and $\mathbf{R}_k$ are all augmentations given by measurements of one scan of the target

$$\begin{aligned} \mathbf{z}_k &= \left[\mathbf{z}_k^{1\,\mathsf{T}}, \quad ..., \quad \mathbf{z}_k^{m\,\mathsf{T}}\right]^\mathsf{T} \\ \mathbf{R}_k &= \mathrm{diag}\left[\mathbf{R}_k^1, \quad ..., \quad \mathbf{R}_k^m\right] \\ \mathbf{h}_k(\mathbf{x}_k) &= \left[\mathbf{h}_k^1(\mathbf{x}_k)^\mathsf{T}, \quad ..., \quad \mathbf{h}_k^m(\mathbf{x}_k)^\mathsf{T}\right]^\mathsf{T}. \end{aligned} \tag{11}$$

To define the measurement equation $\mathbf{h}_k^l(\mathbf{x}_k)$ for a single measurement $\mathbf{z}_k^l$, we utilize the measurement equation defined by the random hypersurface model for a target contour parametrized by a function f.

$$\begin{aligned} \mathbf{z}_k^l &= \mathbf{x}_k^c + \mathbf{p}(\theta_k^l) f(\theta_k^l) + \boldsymbol{\eta}_k^l \\ \mathbf{p}(\theta_k^l) &= \begin{bmatrix} \cos\theta_k^l \\ \sin\theta_k^l \end{bmatrix}, \end{aligned} \tag{12}$$

where $\mathbf{z}_k^l$ is the measurement l at time k and θ_k^l is the corresponding angle of the origin of the measurement of the target contour. θ_k^l can be expressed both in a global

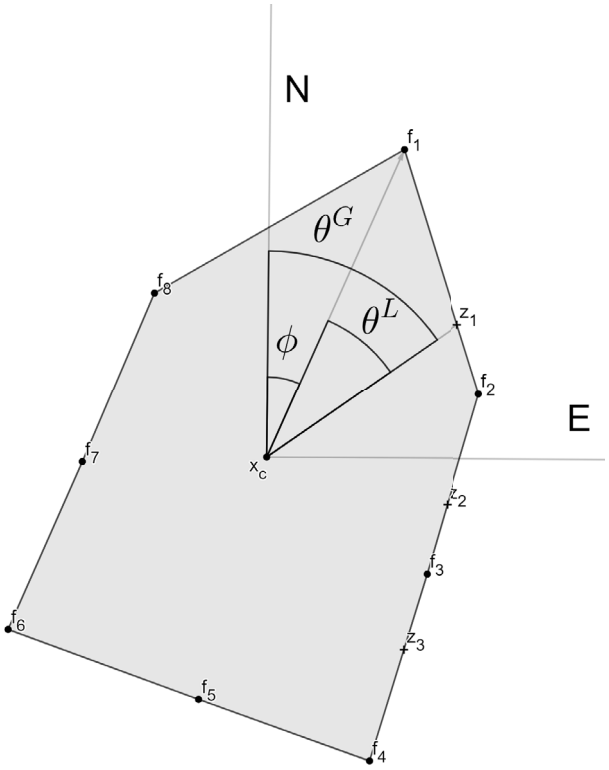


Figure 1. Visualization of the variables used in the measurement equation of the GP target model. The global frame is defined by N - North and E - East. The points marked with f_n correspond to the n-th element of $\mathbf{x}^f$, i.e., the test points, and the crosses marked with z_n represents measurements of the contour. The visualized θ^G and θ^L correspond to $\theta_k^{1\,(G)}(\mathbf{x}_k^c)$ and $\theta_k^{1\,(L)}(\mathbf{x}_k^c, \phi_k)$ respectively, i.e., they are defined for z_1.

frame $\theta_k^{l\,(G)}$ and the local target frame $\theta_k^{l\,(L)}$ as

$$\begin{aligned} \theta_k^{l\,(L)}(\mathbf{x}_k^c, \phi_k) &= \theta_k^{l\,(G)}(\mathbf{x}_k^c) - \phi_k \\ \theta_k^{l\,(G)}(\mathbf{x}_k^c) &= \angle\left(\mathbf{z}_{k,l} - \mathbf{x}_k^c\right). \end{aligned} \tag{13}$$

Inserting the expressions for GP regression (3) into the measurement equation (12), we attain

$$\begin{aligned} \mathbf{z}_k^l &= \mathbf{x}_k^c + \mathbf{p}_k^l(\theta_k^{l\,(G)}(\mathbf{x}_k^c))\mathbf{H}^f\left(\theta_k^{l\,(L)}(\mathbf{x}_k^c, \phi_k)\right)\mathbf{x}_k^f + \boldsymbol{\eta}_k^l \\ &= \mathbf{h}_k^l(\mathbf{x}_k) + \boldsymbol{\eta}_k^l, \qquad \boldsymbol{\eta}_k^l \sim \mathcal{N}(0, \mathbf{R}_k^l) \\ \mathbf{R}_k^l &= \mathbf{p}_k^l(\theta_k^{l\,(G)}(\mathbf{x}_k^c))\mathbf{R}^f\left(\theta_k^{l\,(L)}(\mathbf{x}_k^c, \phi_k)\right)\mathbf{p}_k^l(\theta_k^{l\,(G)}(\mathbf{x}_k^c))^\mathsf{T} + \mathbf{R}. \end{aligned} \tag{14}$$

This is a nonlinear measurement model and therefore needs to be estimated using a nonlinear filtering technique. It should be noted that this is an implicit equation due to the dependence of $\mathbf{z}_k^l$ contained in $\theta_k^{l\,(G)}(\mathbf{x}_k^c)$. See Fig. 1 for a graphical representation of the relationship between the variables used in the measurement equation.

For the motion model, the motion can be described with a linear state space model and this can be combined with the process model for the extent as

$$\mathbf{F} = \begin{bmatrix} \bar{\mathbf{F}} & 0 \\ 0 & \mathbf{F}^f \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} \bar{\mathbf{Q}} & 0 \\ 0 & \mathbf{Q}^f \end{bmatrix}, \tag{15}$$

where $\mathbf{F}^f$ and $\mathbf{Q}^f$ are given by (5), and $\bar{\mathbf{F}}$ and $\bar{\mathbf{Q}}$ are given by the motion model used. For this work, we use the constant velocity model combined with a constant angular velocity. The model matrices are defined as

$$\begin{aligned} \bar{\mathbf{F}} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_3, \\ \bar{\mathbf{Q}} &= \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \begin{bmatrix} \sigma_c^2 & 0 & 0 \\ 0 & \sigma_c^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix}, \end{aligned} \tag{16}$$

where σ_c is the standard deviation of the process noise for position and σ_ϕ is the standard deviation for the heading angle.

D. The PMBM Filter

To model the problem of tracking multiple targets, the PMBM filter utilizes random finite sets (RFS) to model both the unknown number of targets and the unknown number of measurements. The set of object states at time k is modeled as $X_k = \{x_k^1, \ldots, x_k^{n_k}\}$ and the measurements collected at time step k are defined as $Z_k = \{z_k^1, \ldots, z_k^{m_k}\}$ with z_k^l denoting a single measurement.

The PMBM conjugate prior is a combination of a Poisson point process (PPP) and a multi-Bernoulli mixture (MBM), where the PPP represents the targets that have not been detected $\mathbf{X}_k^u$ and the MBM represents the targets that have been detected $\mathbf{X}_k^d$. A PMBM density is fully parametrized by

$$D_k^u, \{w_k^j, \{r_k^{j,i}, (f_k^{j,i})\}_{i\in\mathbb{I}_{k|k'}}\}_{j\in\mathbb{J}_{k|k'}}, \tag{17}$$

where D_k^u is the intensity function of the PPP for the unknown targets. The Bernoulli modeling target i is represented by the probability density $f_k^{j,i}$, which represents both the kinematic state and the extent of the target, along with any additional information that can be inferred from it. A Bernoulli set also contains a parameter r that represents the existence probability of the target. The different components in the MBM are represented by an index $j \in \mathbb{J}$ and correspond to a data association hypothesis with the weight w^j representing the relative likelihood of each hypothesis. Additional assumptions are that new targets appear in the region according to a PPP with birth intensity function D_k^b, targets survive with probability P^S and evolve with a transition density $\mathbf{g}_{k|k-1}$. Clutter is modeled as a PPP with rate λ_c and a uniform spatial distribution. Each target is detected with a probability P^D and, if detected, generates measurements according to a PPP with rate $\lambda_m(\mathbf{x})$ and a spatial distribution $l(Z_C|\mathbf{x}_k)$, given by the chosen target model. Z_C is the subset of measurements assigned to a specific measurement cell C, and l_C is the likelihood of this assignment.

Recursions based on these assumptions are presented in the original paper on the PMBM filter for extended objects [3].

E. Estimating Measurement Rate

The Poisson rate $\lambda_m(\mathbf{x})$ models the cardinality of the measurement set, i.e., the expected number of measurements. The simplest assumption is a constant rate, but this is not in good agreement with the physical reality of many sensors, since the number of returns usually scales by distance. A more realistic approach was developed in [24] and has since been used as part of the GGIW target model in several works, such as [3], [25]. It has also been used in combination with the GP model [15]. It utilizes that a Poisson rate can be estimated using a gamma distribution because it is the conjugate prior to the Poisson distribution. A gamma distribution can be parametrized by parameters α and β, where α is the shape parameter and β is the inverse scale parameter, i.e., $\lambda_m \sim \mathcal{G}(\alpha, \beta)$. These can be updated using the following recursions:

$$\begin{aligned} \alpha_{k|k-1} &= \frac{\alpha_{k-1}}{\eta_\gamma}, \qquad \beta_{k|k-1} = \frac{\beta_{k-1}}{\eta_\gamma} \\ \alpha_k &= \alpha_{k|k-1} + |Z_C|, \qquad \beta_k = \beta_{k|k-1} + 1. \end{aligned} \tag{18}$$

The forgetting factor η_γ is defined as $\eta_\gamma = \frac{1}{1-\frac{1}{w_e}}$, which means that only information from the time steps within the window length w_e is trusted. By estimating these parameters for each target, we can determine a target-specific Poisson rate $\lambda_m(\mathbf{x})$ for each target.

III. IMPROVEMENTS TO THE GP TARGET MODEL

In this section, we present the suggested improvements to the GP target model by using the iterated extended Kalman filter to improve the linearization. We also present how negative information can be used in this framework.

A. Handling Nonlinearities in the Measurement Equation

Since the measurement equation (14) is nonlinear there arises a need to use nonlinear filtering to deal with this non-linearity and estimate $\mathbf{H}$. The original paper on the GP model applies the extended Kalman filter [11]. Subsequent work has been done to improve this method by dealing with the non-linearities differently or augmenting the approach [12]–[14]. In this work, we propose using the iterated extended Kalman filter (IEKF) to improve the linearization. It has been shown that applying the IEKF is equivalent to GN optimization of the maximum likelihood function defined as

$$q(\boldsymbol{\xi}) = \left(\begin{bmatrix} \mathbf{z}_k \\ \hat{\mathbf{x}} \end{bmatrix} - \begin{bmatrix} \mathbf{h}(\boldsymbol{\xi}) \\ \boldsymbol{\xi} \end{bmatrix} \right)^{\mathsf{T}} \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{P} \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{z}_k \\ \hat{\mathbf{x}} \end{bmatrix} - \begin{bmatrix} \mathbf{h}(\boldsymbol{\xi}) \\ \boldsymbol{\xi} \end{bmatrix} \right). \tag{19}$$

Here, $\mathbf{h}(\boldsymbol{\xi})$ and $\mathbf{z}_k$ are defined by (11) and (14). The IEKF is therefore a maximum a posteriori estimator of the state [26]. Equivalently, the IEKF will suffer from the same shortcomings as GN methods, in particular when there are several local optima or the initialization point is far away. In this specific case, $\mathbf{h}(\boldsymbol{\xi})$ is not globally convex and has several local optima.

To mitigate this, we suggest designing a set of heuristic constraints for the initial point of the optimization to ensure that it converges on the most relevant local optimum. In [14], the concept of negative information is used to augment the model. Inspired by this, we can define constraints for the centroid $\mathbf{x}^c$ for a given target. Consider a return from a laser- ranging sensor hitting an extended object. We can then state the following constraint for $\mathbf{x}^c := [x^p, y^p]$ given more than two measurements:

$$\begin{aligned} &\min(\angle \mathbf{z}_k) < \operatorname{atan2}(y^p, x^p) < \max(\angle \mathbf{z}_k) \\ &\min ||\mathbf{z}_k|| < \sqrt{(y^p)^2 + (x^p)^2}. \end{aligned} \tag{20}$$

Essentially, the center of the extended object's angle with regard to the sensor should be between the minimum and maximum angles, and it should be further away than the closest measurement return. We enforce this condition prior to optimization by first calculating the mean range and angle of the measurements generated by the contour

$$\begin{aligned} \theta^c &= \operatorname{mean}(\angle \mathbf{z}_k) \\ r^c &= \operatorname{mean}(||\mathbf{z}_k||) + \min \mathbf{x}^f. \end{aligned} \tag{21}$$

The mean of the angle is corrected to ensure that it is not affected by the discontinuity in the unit circle. Then we convert the point to cartesian coordinates according to

$$\mathbf{x}^c = r^c \begin{bmatrix} \cos(\theta^c) \\ \sin(\theta^c) \end{bmatrix}, \tag{22}$$

This means that if either constraint is violated, $\mathbf{x}^c$ will be initiated behind the wall of sensor measurements generated by the object contour. There is also a local optimum relating to the heading. In particular, $\phi + \pi$, i.e., the reverse heading, is a local optimum since it is also aligned with the symmetry axis defined by the covariance function. To avoid this local optimum, we can utilize the velocity vector to design a similar constraint for the heading as for the centroid, i.e., ϕ is initialized according to the following criteria:

$$\phi = \begin{cases} \phi & \text{if } (\phi - \operatorname{atan2}(\dot{y}^p, \dot{x}^p)) < \pi/3 \\ \operatorname{atan2}(\dot{y}^p, \dot{x}^p) & \text{if } (\phi - \operatorname{atan2}(\dot{y}^p, \dot{x}^p)) \geq \pi/3 \end{cases}. \tag{23}$$

This scheme will not be applicable in all cases, e.g., reversing targets or surface vessels that drift with a strong current, but for most cases where tracking targets is relevant, this condition will be applicable.

B. Negative Information

In [14], the concept of negative information is used to constrain the bounds of the estimated extent so that it doesn't expand beyond the received measurements. This is achieved by estimating the minimum and maximum angles of the extended object and comparing these to the minimum and maximum angles of the actual measurements associated with that object. The minimum and maximum angles of the measurement set are therefore used as an additional measurement; these measurements are termed virtual measurements. A generic measurement equation for these virtual measurements can be written as

$$\begin{aligned} \max \angle \mathbf{z}_k &= h_k^{max}(\mathbf{x}_k) + \eta_k^{max}, \qquad \eta_k^{max} \sim \mathcal{N}(0, \mathbf{R}_k^{max}) \\ \min \angle \mathbf{z}_k &= h_k^{min}(\mathbf{x}_k) + \eta_k^{min}, \qquad \eta_k^{min} \sim \mathcal{N}(0, \mathbf{R}_k^{min}). \end{aligned} \tag{24}$$

We use the same approach, but we perform the state estimation utilizing the IEKF framework presented above.

The key assumption that allows us to perform the state estimation in the IEKF framework is to assume that, for the GP target model, the minimum and maximum angles of an extended object are achieved by sampling points of the extent, i.e., one of the elements of Θ^f. Then, we can derive an expression for these points in global coordinates since we can calculate the global position of each test point from the following expression:

$$\mathbf{x}_G^f = \mathbf{x}_k^c + \mathbf{x}_k^f \begin{bmatrix} \cos\left(\Theta^f + \phi_k\right) \\ \sin\left(\Theta^f + \phi_k\right) \end{bmatrix}. \tag{25}$$

We can then calculate the minimum and maximum angles that the extended object occupies

$$\begin{aligned} \kappa_{min} &= \min \angle \mathbf{x}_G^f \\ \kappa_{max} &= \max \angle \mathbf{x}_G^f. \end{aligned} \tag{26}$$

This allows us to determine the angular sector that an extended object occupies. These relations are presented graphically in Fig. 2. To define the measurement equation for these angles, we define

$$\begin{aligned} \theta_{max}^{(L)} &= \theta_{max}^f \\ \theta_{max}^{(G)}(\phi_k) &= \angle\left(\mathbf{x}_G^f - \mathbf{x}_k^c\right) = \theta_{max}^f + \phi_k, \end{aligned} \tag{27}$$

where θ_{max}^f is the element of Θ^f which corresponds to κ_{max}, which can be defined both in the local target frame (L) and the global frame (G). The time index k has been omitted from θ for notational convenience. With these definitions, we can compute the points to the extent that correspond most closely to the measurements with the maximum and minimum angles using (14)

$$\mathbf{z}_k^{max} = \mathbf{x}_k^c + \mathbf{p}_k(\theta_{max}^{(G)}(\phi_k))\mathbf{H}^{\mathbf{f}}\left(\theta_{max}^f\right)\mathbf{x}_k^f. \tag{28}$$

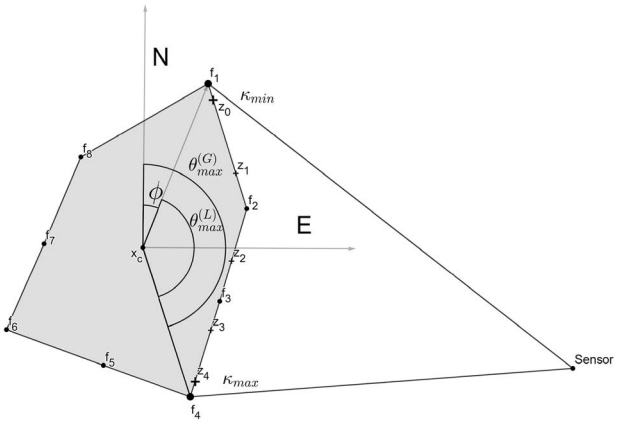


Figure 2. The extreme angles of the extended object κ_{min} and κ_{max} are given by the points f_1 and f_4, respectively, which are equivalent to $\mathbf{x}_G^f$ calculated for the 1st and 4th test point. The angles given by measurements z_0 and z_4 are the virtual measurements, and κ_{min} and κ_{max} are the corresponding predicted virtual measurements.

Utilizing this expression, we can calculate the global angle of these points

$$\begin{aligned} h_k^{max}(\mathbf{x}_k) &= \angle \mathbf{z}_k^{max} \\ &= \operatorname{atan2}\left(\frac{y_k^p + \sin(\theta_{max}^{(G)}(\phi_k))\mathbf{H}^{\mathbf{f}}\left(\theta_{max}^f\right)\mathbf{x}_k^f}{x_k^p + \cos(\theta_{max}^{(G)}(\phi_k))\mathbf{H}^{\mathbf{f}}\left(\theta_{max}^f\right)\mathbf{x}_k^f}\right). \end{aligned} \tag{29}$$

For the minimum angle, substitute θ_{max}^f for θ_{min}^f. The measurement Jacobian for this equation can be found by calculating the partial derivatives; see Appendix A for the applicable expressions. The noise component should also be transformed to extract the angular component. However, since we have chosen θ_{max}^f to be a test point, there is no uncertainty from the GP regression [22], $\mathbf{R}^f = 0$, and we are simply left with the noise component of the measurement, which we can define as

$$\mathbf{R}_k^{max} = \sigma_\kappa^2, \tag{30}$$

where σ_κ is the standard deviation of the angular component of the measurement noise. The actual measurements, which are used as virtual measurements, can be found by finding the measurements with the minimum and maximum angles. These virtual measurements are then used to augment the quantities in (11) such that

$$\begin{aligned} \mathbf{z}_k^* &= [\mathbf{z}_k, \quad \max \angle \mathbf{z}_k, \quad \min \angle \mathbf{z}_k]^\mathsf{T} \\ \mathbf{R}_k^* &= \operatorname{diag}\left[\mathbf{R}_k, \quad \mathbf{R}_k^{max}, \quad \mathbf{R}_k^{min}\right] \\ \mathbf{h}_k^*(\mathbf{x}_k) &= \left[\mathbf{h}_k(\mathbf{x}_k), \quad h_k^{max}(\mathbf{x}_k), \quad h_k^{min}(\mathbf{x}_k), \right]^\mathsf{T}, \end{aligned} \tag{31}$$

where $\mathbf{z}_k, \mathbf{R}_k$, and $\mathbf{h}_k(\mathbf{x})$ are defined by (11). Correspondingly, we can define a new cost function as

$$q(\boldsymbol{\xi}) = \left(\begin{bmatrix} \mathbf{z}_k^* \\ \hat{\mathbf{x}} \end{bmatrix} - \begin{bmatrix} \mathbf{h}^*(\boldsymbol{\xi}) \\ \boldsymbol{\xi} \end{bmatrix}\right)^\mathsf{T} \begin{bmatrix} \mathbf{R}^* & 0 \\ 0 & \mathbf{P} \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{z}_k^* \\ \hat{\mathbf{x}} \end{bmatrix} - \begin{bmatrix} \mathbf{h}^*(\boldsymbol{\xi}) \\ \boldsymbol{\xi} \end{bmatrix}\right). \tag{32}$$

This ensures that the virtual measurements are included in the GN optimization.

Table I
GP Update Algorithm

Input: Predicted state and covariance $\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}$, Associated measurements Z_C
Output: Updated state and covariance $\hat{\mathbf{x}}_k, \mathbf{P}_k$, Measurement likelihood l_C
if Equation (20) **and** $|Z_C| \geq 2$ **then**
  $\mathbf{x}_{iter} \leftarrow \mathbf{x}_{k|k-1}$ modified by Equations (22) and (23)
else
  $\mathbf{x}_{iter} \leftarrow \mathbf{x}_{k|k-1}$
end if
if $|Z_C| \geq 2$ **and not** partialOcclusion **and** $\mathbf{x}_{iter} \neq \mathbf{x}_{k|k-1}$ **then**
  negativeInformation = True
else
  negativeInformation = False
end if
$i \leftarrow 0$
while $i \leq GPMaxIterations$ **and** $eps > 10^{-4}$ **do**
  $\mathbf{x}_{upd} \leftarrow$ SingleGPUpdate$(\mathbf{x}_{k|k-1}, \mathbf{x}_{iter}, \mathbf{P}_{k|k-1}, Z_C)$
  $eps \leftarrow \mathbf{x}_{iter} - \mathbf{x}_{upd}$
  $\mathbf{x}_{iter} \leftarrow \mathbf{x}_{upd}$
  $i \leftarrow i + 1$
  if $i > GPMaxIterations$ **and** NegativeInformation **and** $(eps > 10^{-4}$
    or $\dot{\phi}_{upd} > \pi/4)$ **then**
    $i \leftarrow 0$
    $eps \leftarrow 10^3$
    $\mathbf{x}_{iter} \leftarrow \mathbf{x}_{k|k-1}$
    NegativeInformation = False
  end if
end while
$\mathbf{x}_k \leftarrow \mathbf{x}_{upd}$
Update covariance $\mathbf{P}_k$
Update α and β according to (18)
Calculate measurement set likelihood l_C via (41)

Table II
SingleGPUpdate

Input: Predicted state and covariance $\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}$,
  Current state $\mathbf{x}_{iter}$, Associated measurements Z_C
Output: Updated state $\mathbf{x}_{upd}$
for all $z \in Z_C$ **do**
Evaluate equation (14) with $\mathbf{x}_{iter}$
end for
Construct $\mathbf{H}_k, \mathbf{R}_k, \mathbf{h}(\mathbf{x})$ via equation (11)
if NegativeInformation **then**
$\kappa_{min/max} \leftarrow$ via equation (26)
Evaluate equation (29) and augment $\mathbf{H}_k, \mathbf{R}_k, \mathbf{h}(\mathbf{x}), \mathbf{z}_k$ with result
end if
$\mathbf{S}_k \leftarrow \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$
$\mathbf{W}_k \leftarrow \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$
$\mathbf{x}_{upd} \leftarrow \mathbf{x}_{k|k-1} + \mathbf{W}_k \left((\mathbf{z}_k - \mathbf{h}(\mathbf{x})) - \mathbf{H}_k (\mathbf{x}_{k|k-1} - \mathbf{x}_{iter}) \right)$

1) Practical Considerations: We only generate these virtual measurements if we have at least two measurements; otherwise, the virtual measurements will be identical, which implies a very small extent and will have the practical effect of shrinking extents. Incorporating these virtual measurements can be viewed as further constraining the optimization problem. In some cases, this can prevent convergence of the iterative optimization, and this is usually not desirable. Therefore, if the optimization does not converge while using virtual measurements, we repeat the optimization without including them in the cost function. Similarly, the use of virtual measurements has a tendency to estimate a rather large rotation of the extent in certain situations, which can be seen as another undesirable local optimum. We therefore also repeat the optimization without virtual measurements if the estimated rotational velocity is higher than a certain threshold. We also do not use virtual measurements when the initialization criteria in (20) and (23) are used, as convergence is harder to achieve in that case. The full method is summarized in algorithmic form in Tables I and II.

IV. THE GP-PMBM TRACKER

Given the state space model presented above, we now provide the specific closed-form expressions for the PMBM filter recursions for the GP model based on the general form given in [3]. We then discuss specific considerations for using the GP model in the PMBM framework and present the approximations used to make the tracker computationally feasible. Finally, we present the way occlusion is modeled using the information given by the virtual measurements in the previous section.

A. PMBM Filter Recursions with a GP Target Model

For the special case where the probability of survival P_S is constant and the following holds:

$$\begin{aligned} D_{k-1}^u(\mathbf{x}) &= \sum_{n=1}^{N_u} d_n^u \mathcal{N}(\mathbf{x}; \mathbf{x}_n^u, \mathbf{P}_n^u) \mathcal{G}(\alpha_n^u, \beta_n^u) \\ f_{k-1}^{j,i}(\mathbf{x}) &= \mathcal{N}(\mathbf{x}; \mathbf{x}_{k-1}^{j,i}, \mathbf{P}_{k-1}) \mathcal{G}(\alpha_{k-1}^{j,i}, \beta_{k-1}^{j,i}) \\ \mathbf{g}_{k|k-1}(\mathbf{x}|\mathbf{x}') &= \mathcal{N}(\mathbf{x}; \mathbf{F}\mathbf{x}', \mathbf{Q}), \end{aligned} \tag{33}$$

i.e., the probability distribution representing the target state in the Bernoulli components is a gamma-Gaussian distribution, and the PPP intensity is a linear combination of gamma-Gaussian distributions, i.e., a gamma-Gaussian mixture. The state transition density for the Gaussian and the gamma component is assumed to be independent, which enables separate prediction of the state and extent from the measurement rate. This assumption was used in [24]. The closed-form expression is then given by

$$\begin{aligned} D_{k|k-1}^u(\mathbf{x}) &= D^b(\mathbf{x}) \\ &\quad + P_S \sum_{n=1}^{N_u} d_n^u \mathcal{N}(\mathbf{x}; \mathbf{F}\mathbf{x}_n^u, \mathbf{F}\mathbf{P}_n^u\mathbf{F}^T + \mathbf{Q}) \mathcal{G}(\alpha_n^u, \beta_n^u) \\ w_{k|k-1}^j &= w_k^j \\ r_{k|k-1}^{j,i} &= r_k^{j,i} P_S \\ f_{k|k-1}^{j,i} &= \mathcal{N}(\mathbf{x}; \mathbf{F}\mathbf{x}_{k-1}^{j,i}, \mathbf{F}\mathbf{P}_{k-1}^{j,i}\mathbf{F}^T + \mathbf{Q}) \\ &\quad \times \mathcal{G}(\alpha_{k|k-1}^{j,i}, \beta_{k|k-1}^{j,i}). \end{aligned} \tag{34}$$

For the update step, we define the following predicted set densities

$$f^{j,i}_{k|k-1}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{x}^{j,i}_{k|k-1}, \mathbf{P}^{j,i}_{k|k-1})\mathcal{G}(\alpha^{j,i}_{k|k-1}, \beta^{j,i}_{k|k-1})$$
$$D^{u}_{k|k-1}(\mathbf{x}) = \sum_{n=1}^{N_u} d^u_n \mathcal{N}(\mathbf{x}; \mathbf{x}^u_n, \mathbf{P}^u_n)\mathcal{G}(\alpha^u_n, \beta^u_n). \quad (35)$$

For a nonempty set of measurements Z_C conditioned on a target $\mathbf{x}$, the extended object measurement set likelihood [3] is given by

$$l(Z_C|\mathbf{x}) = P_D e^{-\lambda_m} \lambda_m^{|Z_C|} \prod_{\mathbf{z}\in Z_C} l(\mathbf{z}|\mathbf{x}). \quad (36)$$

$l(\mathbf{z}|\mathbf{x})$ is the single measurement likelihood, which in our case is given by the GP target model, i.e., (14). Due to the non-linearity of the measurement model, we make the following assumption:

$$l(\mathbf{z}^l|\mathbf{x}) \approx \mathcal{N}(\mathbf{z}^l; \mathbf{h}^l(\mathbf{x}), \mathbf{R}^l) \approx \mathcal{N}(\mathbf{z}^l; \mathbf{H}^l\mathbf{x}, \mathbf{R}^l), \quad (37)$$

i.e., $\mathbf{h}(\mathbf{x})$ defined in (14) is approximated by linearization with the jacobian matrix $\mathbf{H}^l$. With this, the product of the single measurement likelihoods can be written as

$$\mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \mathbf{R}) = \prod_{\mathbf{z}^l\in Z_C} \mathcal{N}(\mathbf{z}^l; \mathbf{H}^l\mathbf{x}, \mathbf{R}^l). \quad (38)$$

Here, $\mathbf{z}$ and $\mathbf{R}$ correspond to the augmented vectors and matrices defined in (11) for all measurements in the set Z_C and $\mathbf{H}$ is the concatenation of all matrices $\mathbf{H}^l$ given by these measurements. This is the joint likelihood of all the measurements in the set Z_C. Furthermore, to incorporate virtual measurements, we define the following:

$$\mathcal{N}(\mathbf{z}^*; \mathbf{H}^*\mathbf{x}, \mathbf{R}^*) = \mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \mathbf{R})\times$$
$$\mathcal{N}(\max\angle\mathbf{z}; \mathbf{H}^{max}\mathbf{x}, \mathbf{R}^{max})\mathcal{N}(\min\angle\mathbf{z}; \mathbf{H}^{min}\mathbf{x}, \mathbf{R}^{min}), \quad (39)$$

where $\mathbf{z}^*$ and $\mathbf{R}^*$ are defined in (31) and $\mathbf{H}^*$ is the linearization of $\mathbf{h}^*_k(\mathbf{x}_k)$. This is the joint likelihood of all measurements in the set Z_C if virtual measurements are incorporated. Given this, we can write the extended object measurement set likelihood as

$$l(Z_C|\mathbf{x}) = P_D e^{-\lambda_m} \lambda_m^{|Z_C|} \mathcal{N}(\mathbf{z}^*; \mathbf{H}^*\mathbf{x}, \mathbf{R}^*). \quad (40)$$

Using this, we can define the likelihood of a measurement set belonging to a measurement cell C conditioned on a given target estimate as

$$l_C(\alpha, \beta, \hat{\mathbf{x}}, \mathbf{P}, Z_C) = P_D \frac{\Gamma(\alpha + |Z_C|)\beta^\alpha}{\Gamma(\alpha)(\beta+1)^{(\alpha+|Z_C|)}|Z_C|!} \times \mathcal{N}(\mathbf{z}^*; \mathbf{H}^*\hat{\mathbf{x}}, \mathbf{S}^*), \quad (41)$$

where $\hat{\mathbf{x}}$ is the estimated mean of the target and the innovation covariance $\mathbf{S}^*$ is given by a Kalman filter update step and is as such defined by the matrices $\mathbf{H}^*$, $\mathbf{R}^*$, and the predicted covariance $\mathbf{P}$ corresponding to the target estimate. It is again assumed that the measurement rate and the combined state and extent are independent. The predicted likelihood of the gamma component was derived in [24]. We assume that the probability of detection P_D can be approximated as

$$P_D(\mathbf{x})p(\mathbf{x}) \approx P_D(\hat{\mathbf{x}})p(\mathbf{x}), \quad (42)$$

where $p(\mathbf{x})$ is a generic gamma-Gaussian probability distribution and $\hat{\mathbf{x}}$ is the mean of that distribution. This assumption holds trivially when $P_D(\cdot)$ is constant and is expected to hold when it is a sufficiently smooth function within the uncertainty area of the estimate [3]. With these definitions, we can state the following closed-form expressions. The PPP component representing undetected targets is updated as

$$D^u_k(\mathbf{x}) = Q_D D^u_{k|k-1}(\mathbf{x}), \quad (43)$$

i.e., the weight of each undetected target in the mixture is updated with the effective probability of a missed detection, defined as

$$Q_D = 1 - P_D + P_D e^{-\lambda_m}, \quad (44)$$

where the exponential term is the Poisson probability of a target generating no detections, this is equivalent to the likelihood of an empty set of measurements. The MBM is updated based on the associations made of measurements to measurement cells. The weights for the association hypotheses are updated as

$$w^{j,A}_k = \frac{w^j_{k|k-1}\prod_{C\in A} L^j_C}{\sum_{j\in\mathbb{J}_{k|k-1}}\sum_{A\in\mathcal{A}_j} w^j_{k|k-1}\prod_{C\in A} L^j_C}, \quad (45)$$

i.e., the weight of an association hypothesis A is given by a product of the likelihoods L of all measurement cells, normalized over all association hypotheses.

The updated parameters for the Gaussian distributions are given by a Kalman filter update step, and the updated gamma parameters for a measurement cell are given by (18). This corresponds to applying the steps in Table I.

The form of the update step for measurement cell C depends on if the measurement cell is associated with a detected or undetected target. The current time index k is omitted for brevity. For detected targets, we have two cases to consider

$$L^j_C = \begin{cases} 1 - r^{j,i_C}_{k|k-1} + r^{j,i_C}_{k|k-1}Q_D & |Z_C| = 0 \\ r^{j,i_C}_{k|k-1} l_C(\alpha^{j,i_C}, \beta^{j,i_C}, \hat{\mathbf{x}}^{j,i_C}, \mathbf{P}^{j,i_C}, Z_C) & |Z_C| \neq 0 \end{cases}$$

$$r^{j,i_C}_k = \begin{cases} \frac{r^{j,i_C}_{k|k-1}Q_D}{1 - r^{j,i_C}_{k|k-1} + r^{j,i_C}_{k|k-1}Q_D} & |Z_C| = 0 \\ 1 & |Z_C| \neq 0 \end{cases}$$

$$f^{j,i_C}_k(\mathbf{x}) = \begin{cases} \mathcal{N}(\mathbf{x}; \mathbf{x}^{j,i_C}_{k|k-1}, \mathbf{P}^{j,i_C}_{k|k-1})\mathcal{G}(\alpha^{j,i_C}, \beta^{j,i_C}) & |Z_C| = 0 \\ \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^{j,i_C}, \mathbf{P}^{j,i_C})\mathcal{G}(\alpha^{j,i_C}, \beta^{j,i_C}) & |Z_C| \neq 0 \end{cases}. \quad (46)$$

If measurements are assigned to undetected targets, there are also two cases to consider, since it is assumed that a cell containing more than one measurement cannot be clutter-originated. Note that the result is a component of the MBM since the target has now been detected.

For the following, we define $l_C = l_C(\alpha_n^u, \beta_n^u, \hat{\mathbf{x}}_n^u, \mathbf{P}_n^u, Z_C)$:

$$L_C^j = \begin{cases} D^c + \sum_{n=1}^{N^u} d_n^u l_C & |Z_C| = 1 \\ \sum_{n=1}^{N^u} d_n^u l_C & |Z_C| > 1 \end{cases}$$

$$r_k^{j,i_C} = \begin{cases} \frac{\sum_{n=1}^{N^u} d_n^u l_C}{D^c + \sum_{n=1}^{N^u} d_n^u l_C} & |Z_C| = 1 \\ 1 & |Z_C| > 1 \end{cases} \tag{47}$$

$$f_k^{j,i_C}(\mathbf{x}) = \sum_{n=1}^{N^u} d_n^u \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_n^u, \hat{\mathbf{P}}_n^u) \mathcal{G}(\alpha_n^u, \beta_n^u).$$

See Appendix B for a derivation of these expressions.

B. Initialization and Birth Process

The GP model is generally not observable, especially with few measurements. Therefore, there is no unique solution, and the solution depends on the choice of prior used to initialize the estimate. The choice of prior is therefore of key importance. In particular, the general characteristics of the extent prior and the prior value of the heading need to be specified. In the PMBM framework, the prior estimates are encoded in the birth process intensity D^b. Using the mixture representation of the PPP intensity function, i.e.,

$$D^b(\mathbf{x}) = \sum_{n=1}^{N^b} w_n^b p_n^b(\mathbf{x}), \tag{48}$$

we can define several priors, and since the weight of the PPP components will be updated based on the likelihood of the measurements, the resulting estimate will be weighted. Selecting these priors is still not a trivial choice and, in most cases, are tuned to fit the particular problem. For instance, in [15] where a GP model was used to track vehicles, the prior for the extent ($\mathbf{x}_0^{\mathbf{f}}$) was chosen to correspond to the extent of a real vehicle.

In this work, we use a similar method to the one that was used to define a birth density along the edge of the surveillance area in [18]. We assume an expected maximum range of the sensor R^b, and use it to set the positional component of the birth process intensity. Given N^b components, an angle for each component is defined as $\psi_n^b = \frac{n2\pi}{N^b}$. The prior for the centroid is then given by

$$\mathbf{x}_n^c = R^b \begin{bmatrix} \cos\psi_n^b \\ \sin\psi_n^b \end{bmatrix}, \tag{49}$$

i.e., spread the mixture components uniformly on or slightly beyond the edge of the circle defined by the maximum range of the sensor. By placing it beyond, the centroid will be placed behind the first measurements. The heading can be defined by

$$\phi_n^b = \psi_n^b + \pi, \tag{50}$$

i.e., the direction toward the sensor. The direction of the velocity vector can be defined similarly along with a predefined magnitude v^b, i.e.,

$$\dot{\mathbf{x}}_n^c = v^b \begin{bmatrix} \cos\phi_n^b \\ \sin\phi_n^b \end{bmatrix}. \tag{51}$$

The angular velocity $\dot{\phi}_n^b$ can be assumed to be 0 rad/s. The covariance of all these kinematic states is inflated to ensure that the mixture components can represent a variety of states. With regards to the extent, it should be tailored to the targets that are expected to appear. In this case, because we are tracking ships, we define the extent prior as a ship-like shape with a pointed bow and a flat stern with symmetry along the vertical axis. If it is desirable to track targets with very different shapes, one can also include different shapes in the birth intensity function.

C. Mixture Reduction

Mixture reduction is also a necessary tool used in the birth process to reduce all the components in the PPP mixture into one Bernoulli. It can also be used to merge Bernoullis that are similar. The merging is done using standard Gaussian mixture reduction for the kinematic and extent states and by the method derived in [24] for the parameters of the gamma distribution.

D. Reducing Associations

To reduce the number of data associations, gating is performed as an initial step. Gating for the GP model was presented in the original paper presenting the model [11], and the same method is used in this work. This separates the targets and measurements into independent subgroups. Further reduction of association hypotheses is done using the stochastic optimization method presented in [6] to find the most likely associations. As an implementational detail, we calculate the predicted measurement and the measurement matrix for each measurement-object pair once during the gating process and store them for use during the stochastic optimization method to avoid redundant computation of (14), which involves one GP regression per measurement, a relatively expensive computation.

E. Occlusion

To utilize negative information for state estimation in a multiobject framework, we have to account for the occluding effect of other targets. A natural choice to model occlusion would be to modify the probability of detection P_D. For multiple extended object tracking, this has been done in [18] and further modified in [19]. They used the GIW model and calculated a probability of detection based on the angles occupied by each target and their distance to the sensor. Partial occlusion was handled by discretizing each point on the extent of a target, calculating the probability of detection for each point,

and taking the maximum as the probability of detection for that target. This method was reused in the PMBM framework in [3]. Another approach was presented in [20], which calculates an occlusion likelihood based on the current targets. This occlusion likelihood is represented by a Gaussian mixture that can be used to update the state of undetected objects, infer the existence of objects, and inform the association of data.

The zone being occluded by a specific object is given by the angles occupied and the distance of the object from the sensor which we define as $(\kappa^i_{min}, \kappa^i_{max}, \rho^i)$. In [20], this is expressed as a combination of the following conditions. Object i is fully occluded by object i_O if the following conditions are true

$$\begin{aligned} B &= (\kappa^i_{min} \geq \kappa^{i_O}_{min}) \cap (\kappa^{i_O}_{max} \geq \kappa^i_{max}) \\ R &= (\rho^i \geq \rho^{i_O}). \end{aligned} \tag{52}$$

Condition B can be referred to as the bearing condition and condition R can be referred to as the radial condition. An object is partially occluded if either of the following bearing conditions are true in combination with the radial condition R

$$\begin{aligned} B_{min} &= (\kappa^{i_O}_{max} \geq \kappa^i_{max}) \cap (\kappa^i_{max} \geq \kappa^{i_O}_{min}) \\ B_{max} &= (\kappa^{i_O}_{max} \geq \kappa^i_{min}) \cap (\kappa^i_{min} \geq \kappa^{i_O}_{min}). \end{aligned} \tag{53}$$

Figure 3 shows an example where these cases occur. With this, we can define the probability of a target i being occluded by another target i_O as

$$P^{i,i_O}_O = p(B, R, E) = p(B|E)p(R|E)p(E), \tag{54}$$

where $p(B|E)$ and $P(R|E)$ is the probability of the conditions of (52) being true conditional on the existence of target i_O and $p(E) = r^{i_O}$, i.e., the probability of existence of target i_O. For partial occlusion, we can similarly state

$$P^{i,i_O}_{pO} = p(B_{max/min}, R, E) = p(B_{max/min}|E)p(R|E)p(E). \tag{55}$$

In [20], $(\kappa^i_{min}, \kappa^i_{max}, \rho^i)$ were all assumed Gaussian distributed. We can then calculate the resulting probability of the occlusion conditions by using the cumulative distribution function

$$p(B|E) = \Phi\left(\frac{\kappa^{i_O}_{max} - \kappa^i_{max}}{\sqrt{\sigma^2_{\kappa^{i_O}_{max}} + \sigma^2_{\kappa^i_{max}}}}\right) \Phi\left(\frac{\kappa^i_{min} - \kappa^{i_O}_{min}}{\sqrt{\sigma^2_{\kappa^{i_O}_{min}} + \sigma^2_{\kappa^i_{min}}}}\right), \tag{56}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a Gaussian distribution with zero mean and unit variance; for more details, see [20]. Using the virtual measurements generated in the previous section, we can find a Gaussian distribution for κ^i_{min} and κ^i_{max} from (28) and calculate the resulting innovation variance for each virtual measurement according to

$$\begin{aligned} \sigma^2_{\kappa^i_{min}} &= \mathbf{H}^i_{min}\mathbf{P}^i{\mathbf{H}^i_{min}}^{\mathsf{T}} + \mathbf{R}^{min} \\ \sigma^2_{\kappa^i_{max}} &= \mathbf{H}^i_{max}\mathbf{P}^i{\mathbf{H}^i_{max}}^{\mathsf{T}} + \mathbf{R}^{max}. \end{aligned} \tag{57}$$

The probability of partial occlusion can be calculated in the same manner using the same terms, which becomes

$$\begin{aligned} p(B_{min}|E) &= \Phi\left(\frac{\kappa^{i_O}_{max} - \kappa^i_{min}}{\sqrt{\sigma^2_{\kappa^{i_O}_{max}} + \sigma^2_{\kappa^i_{min}}}}\right) \Phi\left(\frac{\kappa^i_{max} - \kappa^{i_O}_{max}}{\sqrt{\sigma^2_{\kappa^{i_O}_{max}} + \sigma^2_{\kappa^i_{max}}}}\right) \\ p(B_{max}|E) &= \Phi\left(\frac{\kappa^{i_O}_{min} - \kappa^i_{min}}{\sqrt{\sigma^2_{\kappa^{i_O}_{min}} + \sigma^2_{\kappa^i_{min}}}}\right) \Phi\left(\frac{\kappa^i_{max} - \kappa^{i_O}_{min}}{\sqrt{\sigma^2_{\kappa^{i_O}_{min}} + \sigma^2_{\kappa^i_{max}}}}\right). \end{aligned} \tag{58}$$

For ρ^i, we do not have an equivalent way to calculate a Gaussian distribution, so we instead define

$$p(R|E) = \begin{cases} 1 & \text{if} \quad ||\mathbf{x}^i|| > ||\mathbf{x}^{i_O}|| \\ 0 & \text{otherwise} \end{cases}. \tag{59}$$

With this, we can calculate the probability of occlusion for each target pair. Furthermore, with a PMBM framework, both the detected and undetected targets can be occluded, and therefore, we should calculate the probability of occlusion for both these sets. However, only detected targets will have an occluding effect.

Occlusion or partial occlusion have different effects on other model parameters. If we determine that a target

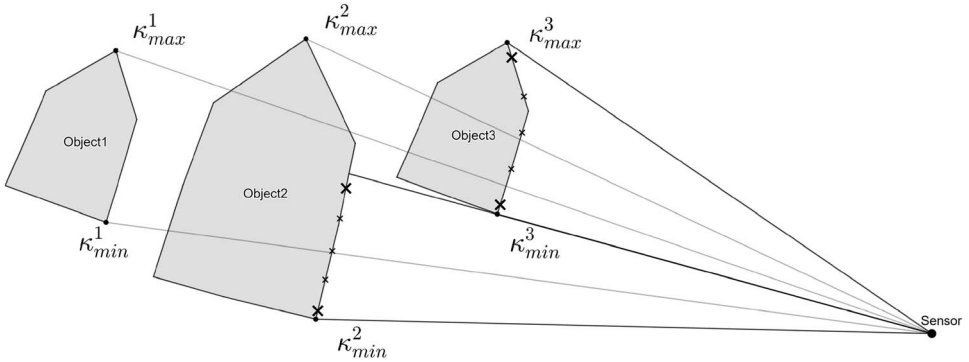


Figure 3. Three targets, where object 1 is occluded by object 2, which in turn is partially occluded by object 3. The angles κ are shown for each object. The measurements generated by the sensor are represented by crosses, and the measurements used for the virtual measurements are shown by the larger crosses.

is wholly occluded by another, the probability of detection will be lowered for that target. Taking the probability of occlusion into account therefore gives a target-specific probability of detection

$$P_D^i = P_D \prod_{i_O \in \mathbb{I}_O} (1 - P_O^{i,i_O}), \tag{60}$$

where $\mathbb{I}_O$ is the total set of occluding targets for target i. If the occlusion probability of any target is 1, then we will have $P_D = 0$; this will cause issues with likelihood calculations, so we enforce a minimum probability of detection. For a component of the PPP mixture u_n, the expression becomes

$$P_D^{u_n} = P_D \sum_{j \in \mathbb{J}} w_j \prod_{i_O \in \mathbb{I}_O^j} (1 - P_O^{u_n,i_O}), \tag{61}$$

i.e., the probability of detection of a component is a weighted average of the probability of detection for every hypothesis in the MBM.

F. Partial Occlusion

Partial occlusion will not affect the probability of detecting the target according to the model assumptions since the target will still generate measurements. However, it will affect the model parameters in other ways. The first is the way negative information is utilized for state estimation since partial occlusion will not match with the assumptions underlying the generation of the virtual measurements. Specifically, if the target is partially occluded, the expected angle κ will no longer correspond to either the minimum or maximum angles of the measurement set, depending on the part of the target that is occluded. In this work, we therefore do not utilize negative information for the state estimation when a target is partially occluded, but we note that it is possible to do so by calculating where κ would intersect on the target contour and use that in place of $\theta^f_{max/min}$ in the measurement equation for negative information. In addition, partial occlusion will affect the number of measurements that are generated by the objects, which will have an impact on the association likelihood in (41) since the gamma distribution will not be able to account for this. So this effect needs to be managed as well, and in this work, we do this by defining a heuristic visibility ratio

$$v^i = \frac{\min(\kappa^{i^{pO}}_{max}) - \max(\kappa^{i^{pO}}_{min})}{\kappa^i_{max} - \kappa^i_{min}}, \tag{62}$$

where $i^{pO} \in \mathbb{I}^{pO}$ are the partially occluding objects. Here, we simply consider the mean of κ, ignoring the uncertainty of each estimate. With this ratio, we also need to consider the probability of occlusion. We define a correction factor as

$$\eta_v^i = 1 - P^i_{pO}(1 - v^i). \tag{63}$$

The probability of occlusion becomes more complicated to calculate for multiple occluding objects since they will all occlude different sectors of the object; each part of an object's extent will therefore, in theory, have its own probability of occlusion. For the sake of simplicity, we utilize the maximum probability of occlusion of the objects partially occluding object i

$$P^i_{pO} = \max P^{i,i_O \in \mathbb{I}^{pO}}_{pO}. \tag{64}$$

Again, for the PPP component, the correction factor should also consider the weights of the MBM, giving the resulting expression

$$\eta_v^{u_n} = \sum_{j \in \mathbb{J}} w_j \eta_v^j, \tag{65}$$

where η_v^j is defined as the correction factor calculated for component u_n using the Bernoullis in the jth MBM. This correction factor is then utilized to modify the parameters of the gamma distribution. We assume that the Poisson rate of a partially occluded target is $\eta_v \lambda_m$, and recall that λ_m is gamma distributed with shape parameter α and inverse scale parameter β. Given this, the Poisson rate of a partially occluded target is gamma distributed with the following parameters:

$$\eta_v \lambda_m \sim \mathcal{G}\left(\alpha, \frac{\beta}{\eta_v}\right). \tag{66}$$

This is equivalent to scaling the gamma distribution since we have divided the inverse scaling parameter with our scaling factor η_v. We can express this in the existing gamma recursions as

$$\begin{aligned} &\alpha_{k|k-1} = \frac{\alpha_{k-1}}{\eta_\gamma}, \qquad \beta_{k|k-1} = \frac{\beta_{k-1}}{\eta_v \eta_\gamma} \\ &\alpha_k = \alpha_{k|k-1} + |Z_C|, \qquad \beta_k = \eta_v(\beta_{k|k-1} + 1). \end{aligned} \tag{67}$$

In this way, the parameters of the gamma distribution are in line with the number of measurements we expect to receive; in other words, the predictive likelihood of the expected number of measurements is preserved

$$\frac{\Gamma(\alpha_{k|k-1} + |Z_C|)\beta_{k|k-1}^{\alpha_{k|k-1}}}{\Gamma(\alpha_{k|k-1})(\beta_{k|k-1} + 1)^{(\alpha_{k|k-1}+|Z_C|)}|Z_C|!}. \tag{68}$$

1) Practical Considerations: The visibility ratio defined in (62) is only well defined between 0 and 1. However, it could take on values larger than 1 if the expressions are applied naively as stated. Therefore, we enforce the conditions in (53) explicitly. In the presence of full occlusion, the visibility ratio will become a negative value. In this case, we set the correction factor to 1. This is for practical reasons since a very low correction factor while not receiving any measurements for an extended period could cause the target estimate to be overly attracted to clutter.

The whole procedure for correcting for occlusion is presented in Table III. Finally, the complete update step for the GP-PMBM tracker is presented in Table IV.

Table III
Occlusion Correction

Input: A predicted PMBM
Output: A probability of detection P_D and a correction factor η_v for every component in the predicted PMBM
for $j \in \mathbb{J}$ **do**
for $i \in \mathbb{I}$ **do**
$(\kappa^i_{min}, \kappa^i_{max}) \leftarrow$ via Equation (26)
end for
for $i \in \mathbb{I} \cup D^u$ **do**
for $i_O \in \mathbb{I}_{\notin i}$ **do**
condmin $\leftarrow \kappa^{i_O}_{min} \leq \kappa^i_{min}$ **and** $\kappa^{i_O}_{max} \geq \kappa^i_{min}$
condmax $\leftarrow \kappa^{i_O}_{min} \leq \kappa^i_{max}$ **and** $\kappa^{i_O}_{max} \geq \kappa^i_{max}$
conddistance $\leftarrow ||\mathbf{x}^{\mathbf{c}i}|| > ||\mathbf{x}^{\mathbf{c}i_O}||$
if conddistance **then**
${P_O}^{i,i_O} \leftarrow$ via Equation (54)
end if
if condmin **and** condmax **and** conddistance **then**
$\eta_v \leftarrow 1$
else if not condmin **and** condmax **and** conddistance **then**
Save $\kappa^{i_O}_{max}$
Assign $i_O \in \mathbb{I}^{pO}$
${P_{pO}}^{i,i_O} \leftarrow$ via Equation (55)
else if condmin **and not** condmax **and** conddistance **then**
Save $\kappa^{i_O}_{min}$
Assign $i_O \in \mathbb{I}^{pO}$
${P_{pO}}^{i,i_O} \leftarrow$ via Equation (55)
end if
end for
Determine $\min(\kappa^{i^{pO}}_{max})$ and $\max(\kappa^{i^{pO}}_{min})$
$v^i \leftarrow$ via Equation (62)
$\eta^i_v \leftarrow$ via Equation (63)
$P^i_D \leftarrow$ via Equation (60)
end for
end for
$P^{u_n}_D \leftarrow$ via Equation (61)
$\eta^{u_n}_v \leftarrow$ via Equation (65)

Table IV
Full GP-PMBM Update

Input: A predicted Poisson Multi-Bernoulli Mixture and a measurement set Z
Output: An updated Poisson Multi-Bernoulli Mixture
Correct for occlusion via Table III
Perform gating for each component of the PMBM
for $j \in \mathbb{J}$ **do**
Compute most likely subset of associations $\mathcal{A}_j$
for $A \in \mathcal{A}_j$ **do**
for $C \in A$ **do**
if New Target **then**
for Each component of $D^u \in C$ **do**
Update component(s) via Table I
end for
$f^{j,i_C} \leftarrow$ via mixture reduction of component(s)
$L^j_C, r^{j,i_C} \leftarrow$ via Equation (47)
else if Existing Target **then**
$f^{j,i_C} \leftarrow$ via Table I
$L^j_C, r^{j,i_C} \leftarrow$ via Equation (46)
end if
end for
for Targets without Detection **do**
$f^{j,i_C}, L^j_C, r^{j,i_C} \leftarrow$ via Equation (46)
end for
end for
end for
Update weights via Equation (45)
Update D^u via Equation (43)

V. SIMULATION STUDY

In this section, we present the result from a Monte Carlo simulation study where the performance of the PMBM-tracker using the presented GP model is compared with the implementation using the GGIW model as presented in [5][1]. For the GP model, we present results without the negative information and occlusion handling as GP, and the results including those features are presented as GP-NI.

A. Simulation Scenario

The scenario consists of 8 ships that are born from timestep 1 to 50. The simulation area is 200×200 m in total with a sensor placed at the center, which has a measurement range of 100 m. Two vessels spawn from each side of the area and approach the center, where they turn. The scenario was particularly handcrafted to simulate occlusion, so that some vessels are born behind another vessel. When the vessels reach the center, they appear close together and are frequently wholly or partially occluded during this time, with vessels traveling alongside one another. See Fig. 4 for a view of the scenario. The scenario lasts for 240 timesteps, and each vessel persists for 190 timesteps. The extent is modeled by a ship that is 6 m long, 3 m wide, and has a pointed bow where the full width is achieved 2 m behind it. The measurements are generated by simulating an LiDAR with a simulated maximum range of 100 m, angular resolution 0.5°, and a modeled radial accuracy of 0.1 m. Measurements are only generated if they hit a simulated hull, and only one measurement is generated per angle, which simulates occlusion. In addition, clutter is generated using a PPP with $\lambda_c = 20$ and a uniform spatial distribution, but the clutter measurements are also corrected for occlusion. The results are averaged over 100 Monte Carlo simulation runs.

B. Parameters

The PMBM parameters are chosen as follows: probability of detection $P_D = 0.90$ for the standard model and $P_D = 0.99$ when occlusion is modeled, probability of survival $P_S = 0.95$, and clutter rate $\lambda_c = 20$. The gating probability is set at $P_G = 0.99$; the pruning parameters are 0.01 for the existence probability, 0.005 for PPP mixture components, and 0.005 for MBM components. Both target models use $\sigma_c = 0.2$ m as the noise parameter for the CV model and $\sigma_r = 0.1$ m for the measure-

[1]The implementation for the GGIW model was taken from github.com/yuhsuansia/Extended-target-PMBM-tracker, and this implementation was modified for use with the GP model

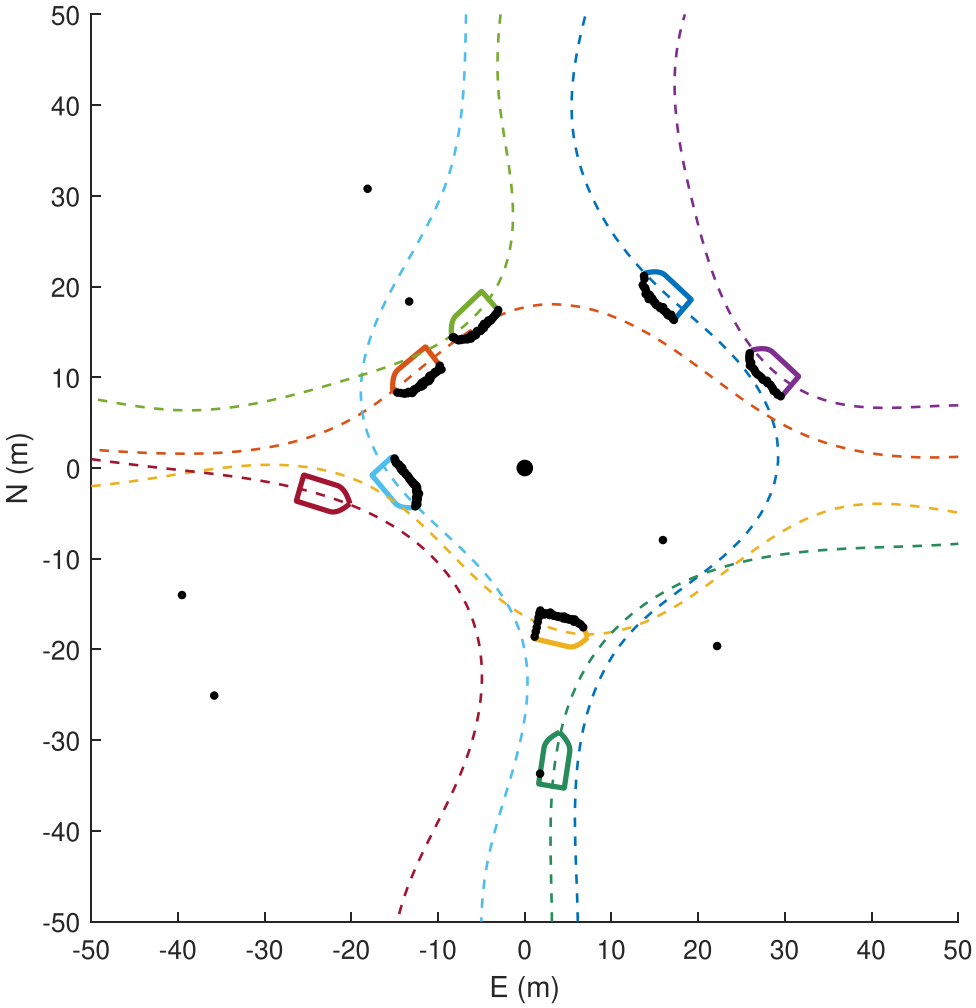


Figure 4. Visualization of the simulation scenario with the trajectories of all vessels, along with the ship extents and measurements visualized for timestep 105. Note that this does not show the whole simulation area but only the 100×100 m area in the center. The simulated LiDAR sensor is placed at the origin.

ment noise; the GP model uses $\sigma_\phi = 0.1$ rad as noise for the constant angular velocity model. For the virtual measurements, the measurement noise strength is $\sigma_\kappa = 0.5^\circ$, the same as the simulated sensor resolution. Both target models use $w_e = 20$ as the length of the gamma prediction window. For the GP target model, we use nine test angles to parametrize the extent, and the hyperparameters are $\sigma_f = 1.0$ m, $\sigma_r = 0.5$ m, $\sigma_n = 0.01$ m, $l = \pi/4$ rad and the forgetting factor $\eta_f = 0.001$. The maximum amount of IEKF iterations is 50. For the GGIW target model, we use 200 for the extent prediction window. The birth intensity function is defined according to the method defined above with $N^b = 36$ components and a range $R^b = 105$ m and a velocity magnitude of $v^b = 1$ m/s. The extent prior is roughly equivalent to the true extent for the GP model, and for the GGIW model, it is an ellipse with the same length and width defining the semi-axes; this is combined with the prior heading to calculate a prior value for the shape matrix X. The prior value of the gamma distribution is $\alpha_0 = 900$ and $\beta_0 = 100$. The covariance of the Gaussians is inflated to ensure coverage of the whole circle, the standard deviation of the positional component is 20 m, 3 m/s for the velocity component, and for the GP model the heading component is $\pi/4$ rad and the angular velocity is $\pi/4$ rad. In the case of the extent, for the GP model, the prior covariance is given by the covariance function.

C. Performance Evaluation

To compare the performance of the trackers, the generalized optimal sub-pattern assignment (GOSPA) metric [27] is used to provide a single metric for the performance of a multiobject tracking algorithm by incorporating localization error, missed targets, and false targets into a single metric. The parameters for the GOSPA metric were cut off $c = 10$ and power $p = 2$. To compare the extent estimates of the target models, we use the process of associating estimates to targets to generate additional measures that are comparable between them. One such measure is the Intersection-Over-Union (IOU) metric, which has been used in previous work to compare methods for extent estimation [11], [12]. To calculate the IOU metric for the GGIW model, the shape

Table V
Mean Value of Metrics for the Simulated Scenario

Model	GP	GP-NI	GGIW
GOSPA	9.52	**8.26**	11.40
Loc. Err.	5.47	**4.53**	11.07
Missed	1.55	**1.03**	1.91
False	0.33	0.48	**0.31**
IOU	0.59	**0.64**	0.33
Heading (rad)	0.40	**0.30**	1.75
Time (s)	**115.90**	158.96	118.19

The best values are highlighted in bold.

matrix X is decomposed to retrieve the length of the semi-axes, corresponding to the 2-σ ellipsoid, and the ellipse orientation. The heading error is calculated using the same method. Finally, the computation time for each run is also presented; however, it should be noted that the GP methods were made more computationally efficient by precomputing the measurement matrices during gating, which makes it hard to compare their computational times with that of the GGIW method, which does not have this implementation efficiency.

D. Results

The metrics are presented in Table V. Note the disparity of the IOU metric. This is primarily due to the inability of the GGIW model to model contour-generated measurements, since the GGIW model assumes a uniform distribution, and thus centers the ellipse on the contour instead, which causes large localization errors and large errors in extent estimation. Note also the larger heading error, showing an inability to estimate the heading as a separate state. We can also note the improvement in utilizing negative information and modeling occlusion as compared to the base GP method with the improvement in localization error and the IOU metric, as well as the reduction of the number of missed targets. However, it increases the number of false targets. The evolution of the metrics during the simulation run is shown in Fig. 5. It shows that the GGIW model has worse IOU and a worse localization error across the whole run, although the localization error is smaller while the ships are close to the sensor, due to the fact that the side of the vessel is measured rather than the front or rear. Comparing the regular method with the negative information method, we can note the disparity in the IOU metric starting around timestep 100, which is when occlusion occurs in the simulation, as well as the subsequent disparity in the localization error. This shows that the use of negative information results in an improved state estimate, particularly when targets are close together. The handling of occlusion also results in a notable improvement in the number of missed targets, due to the method being better at maintaining a track when occlusions occur, but the track is still lost for some targets due to occlusion. This is paired with an increase in false targets, particularly during the middle of the run, when a large part of the surveillance area is occluded by the targets. The reason for this could be that the way we model partial occlusion for undetected targets results in more erroneous detections due to clutter. With these results, we can state that the combination of modeling occlusion and negative information constitutes an improvement over the base GP method. However, it comes with an increase in computational time and false targets.

VI. TEST DATA

In this section, we present the result from real LiDAR data gathered from tests in Trondheim utilizing the two platforms milliAmpere and milliAmpere2 in the Trondheim canal [28].

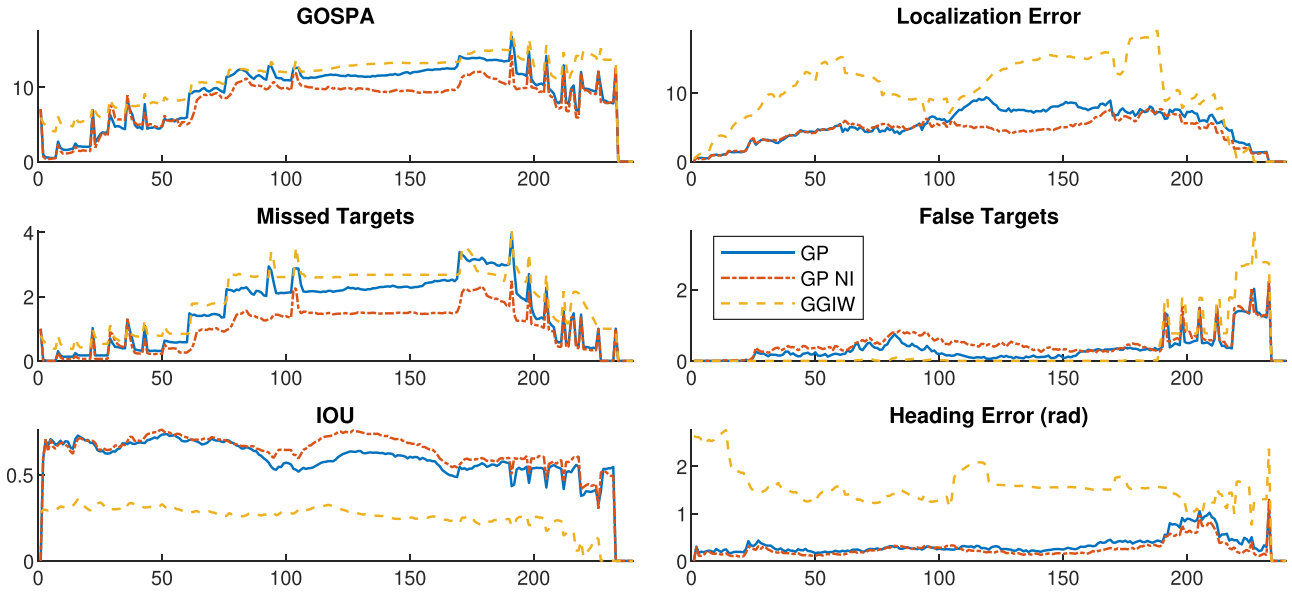


Figure 5. The evolution over the course of the simulation run for selected metrics as an average over all MC simulations.

A. Test Scenario

We present two separate scenarios, one with a single vessel performing maneuvers in front of the sensor in the canal (see Fig. 6a) and one with two vessels traveling in separate directions in the canal and passing each other (see Fig. 6b). The data from the first scenario was gathered using milliAmpere2, which is equipped with two Ouster OS1 32 LiDARs. These two point clouds were combined, and the returns from land and static obstacles along the canal were filtered out using manual land masking, and the point cloud was transformed to 2D by only retaining the point closest to the sensor in each angular resolution sector. The second scenario was published in [29] (as scenario 13) and is reused in this work.

B. Parameters

Most of the parameters used are similar to the simulation study. The range used to define the birth intensity R^b is reduced to 40 and 65 m, respectively, due to the observed range at which the LiDARs were able to detect the target vessels. For the second scenario, α_0 was set to 500 to account for the lower sensor resolution. The extent priors were set such that the length and width of the prior were roughly equivalent to the target vessels, but the same prior was used to represent both ships in the second scenario. In addition, some tweaks were made to attempt to mitigate some observed effects that are not modeled. To account for wake clutter, the clutter rate was increased to $\lambda_c = 60$ and $\lambda_c = 100$ for the first and second scenario, respectively. Finally, to account for errors related to sway affecting the pitch of the LiDAR sensor, the measurement noise σ_r was set to 0.5 m.

C. Performance Evaluation

We use the same metrics that were used in the simulation study, with the ground truth data gathered used to calculate the metrics. For the first scenario, ground truth was measured by using a dual antenna inertial navigation system (INS), and the extent of the vessel was measured to be able to compare the estimated extent with the ground truth. For the second scenario, the ground truth data gathered was only positional global navigation satellite system (GNSS) data without heading; in addition, the exact position of the GNSS receiver was unknown, which is a significant source of error for the IOU calculation. The heading was inferred from the velocity vector, which is also a source of error for the calculation of the IOU and heading error metrics.

Table VI
Mean Value of Metrics for the Real LiDAR Data

	Test 1			Test 2		
Model	GP	GP-NI	GGIW	GP	GP-NI	GGIW
GOSPA	1.51	**1.31**	2.45	4.76	**3.95**	4.06
Loc. Err.	0.85	**0.66**	1.80	4.81	3.68	**3.49**
Missed	0.08	**0.07**	**0.07**	**0.08**	0.08	0.10
False	0.00	**0.00**	0.01	0.01	0.00	**0.00**
IOU	0.36	**0.50**	0.12	0.20	0.26	**0.31**
Heading (rad)	0.37	**0.19**	1.30	0.46	**0.34**	1.86
Time (s)	**96.35**	111.51	161.31	256.53	257.85	**253.26**

The best values are highlighted in bold.

D. Results

The first scenario is quite simple from a target tracking perspective, it is simply a test of target birth and the ability of the target models to track the ship while it is performing complex maneuvers. The relevant metrics are presented in Table VI, and the plots are shown in Fig. 7. The GP model is able to track the target over the whole scenario. However, as the target gets closer to the sensor, the IOU value degrades; this is due to wake clutter being detected by the LiDAR around timestep 400. These wake measurements are associated with the

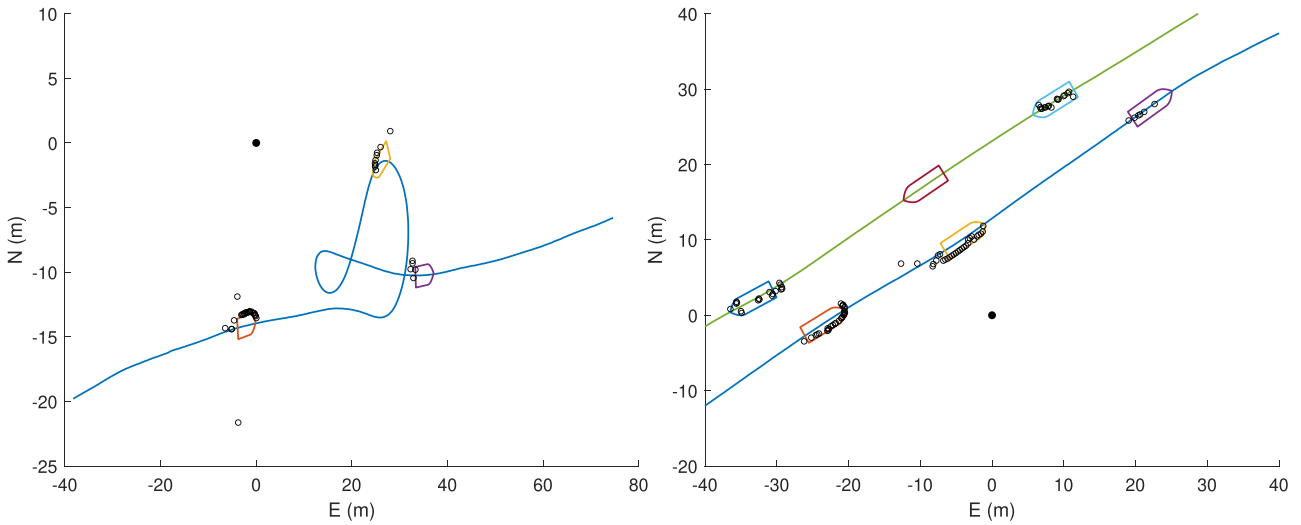


Figure 6. Visualization of the test scenarios, along with the extent and measurements visualized for three different timesteps, with the sensor platform placed at the origin. Note the measurements generated by the wake, as well as the occlusion in the second scenario.

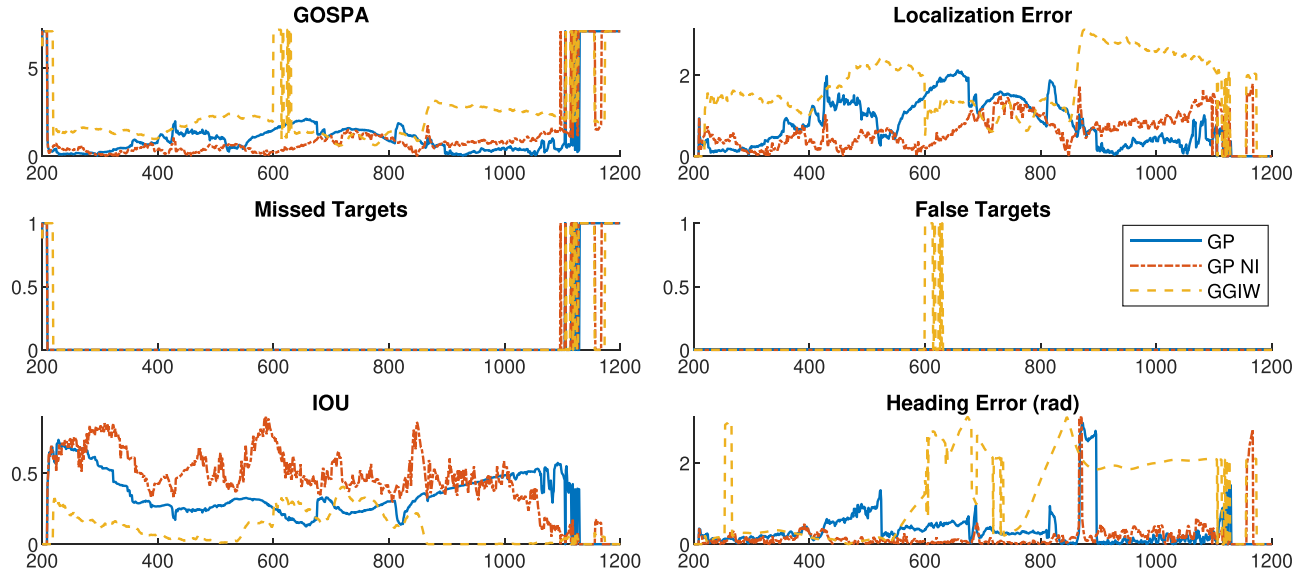


Figure 7. The evolution throughout the single target test run for selected metrics.

target because it does not adhere to the uniform clutter model. This affects the extent estimate and consequently the state estimate. The estimate is able to recover when the vessel completes a turn. In this case, utilizing negative information confers an improvement in the IOU and correspondingly in the localization error. This is due to the ability to more quickly adjust the extent estimate after being affected by the wake clutter, which we are able to do immediately due to the use of virtual measurements in the update step, whereas the base GP method cannot adjust the extent estimate if it is estimated as too large. The GGIW model initializes another track during one of the turning maneuvers to continue tracking the target, resulting in a false target for a few timesteps.

The second scenario is more complex, as it entails two targets, with one target being occluded by the other. The metrics are given in Table VI, and the evolution over time is shown in Fig. 8. Here, the base GP model performs worse compared to the GGIW model, with a higher localization error and lower IOU metric. This is due to the disruptive effect of wake clutter on the GP model, which causes the extent estimates to get significantly worse when the wake is detectable, between timestep 1200 and 1300. This coincides with one vessel occluding the other, and the combined effect of these two phenomena has a major negative effect on the extent estimate. This also affects the centroid estimate, resulting in a larger localization error. Utilizing negative information mitigates this effect since the occlusion is better managed and the effect of the wake clutter is somewhat mitigated by the extent estimate being corrected faster due to the use of negative information in

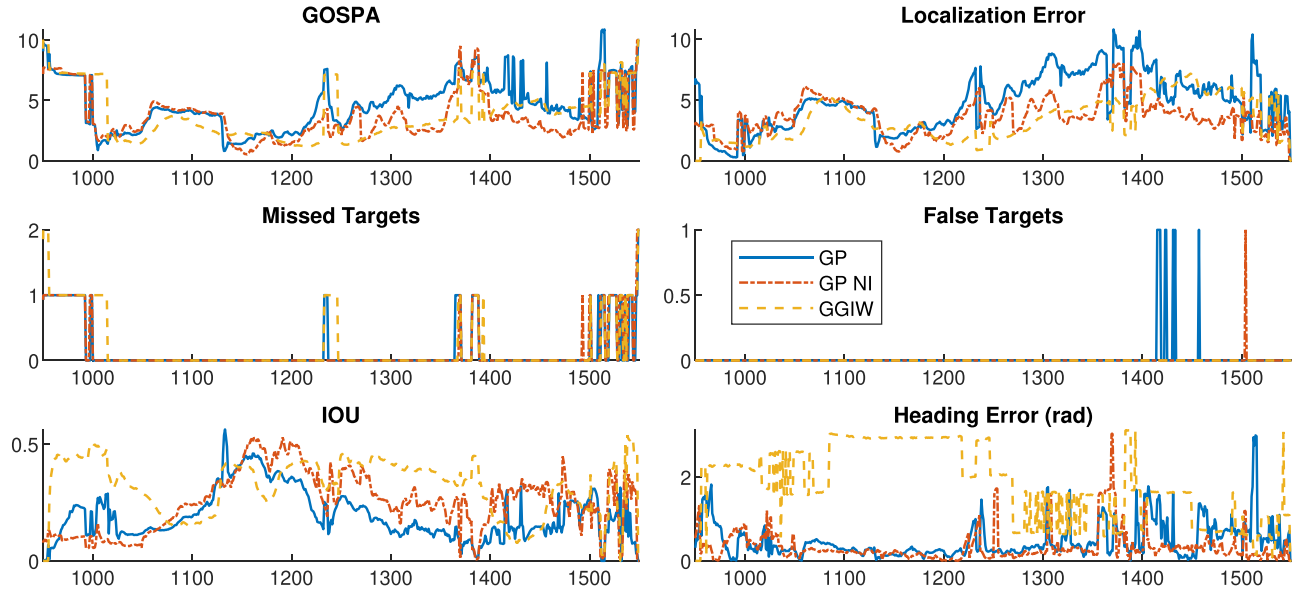


Figure 8. The evolution over the course of the multi-target test run for selected metrics.

the state estimate, but it is still worse than the GGIW model, albeit it is quicker to initialize an estimate, resulting in a lower share of missing targets and a lower GOSPA score. However, even with the use of negative information, the wake clutter still has an effect on the state estimate, as observed by the peak in localization error and the dip in IOU around 1230. The GGIW model is less affected by this shortcoming because of the assumption that measurements are uniformly distributed. Another thing of note is the relatively poor extent estimate of the GP methods at the beginning and the end of the scenario; this is due to scans from the LiDAR hitting at different heights of the target vessel, where the extent may be radically different. Since this method only utilizes information in 2D, this is a source of model inconsistency, and therefore the extent of the vessel is not estimated correctly. This is also a contributing factor to the GGIW model being able to estimate the extent better, which can be seen at the start of the scenario.

VII. CONCLUSION

This paper has presented the use of the GP model as a target model in the extended object PMBM-tracker, presented an improvement of the GP target model by using GN optimization, suggested a heuristic method to mitigate the fact that the measurement model is nonconvex, and we have highlighted the need for a well-designed birth density and provided an example. We have also presented a method to utilize negative information both to improve the state estimation and to handle occlusion. Furthermore, we have demonstrated the resulting tracker on both simulated and real data and compared the performance against the standard GGIW-PMBM tracker. It shows that the GP model can generally track targets more accurately, as measured by GOSPA, and provide a better extent estimate when only a part of the target is detected by the sensor, as measured by the IOU metric. It also enables a correct heading estimate since the heading is explicitly modeled as a part of the state. In addition, applying the method to real maritime data shows that wake clutter is an issue that needs to be addressed. The use of negative information further increases tracking performance and improves the state estimation. Furthermore, it specifically mitigates issues with occlusion and partially mitigates the effect of wake clutter, providing more robust performance on real maritime data.

A. Future Work

It would be of interest to look further into the issue of data association for this target model. Particularly given the computational cost of the method, of which the main part of the computational time is taken up by the data association step. Recent developments in this area for extended object PMBM filters have shown that it is possible to achieve a drastic reduction in computational time by reducing the PMBM to a PMB representation [7]. Directly estimating a PMB either by the use of belief propagation [8], [9] or blocked Gibbs sampling [30] has also shown a drastic reduction in computational time and better performance compared to estimating a PMBM. It would therefore be interesting to explore if the GP target model can be integrated into these methods. Another venue of future work would be to extend the method to also include 3D information, for which there already exist several target models [31], [32]. This would resolve the issue of model consistency when applying it to real data, which could prevent the issues where different LiDAR beams hit the target boat. It could also aid in reducing the effect of wake clutter on the extent estimate. It would also be of interest to directly incorporate a model that accounts for wake clutter into the filter. Recent work has explored how to model arbitrary sources of clutter in the PMBM framework [33], and wake clutter could be modeled in this framework. Clutter models for wake clutter in the context of target tracking already exist [34], [35], and they could be applied by adapting them to extended targets.

ACKNOWLEDGMENT

The authors would like to acknowledge the help of Øystein Kaarstad Helgesen in providing data sets from milliAmpere, Erik Wilthil at Zeabuz, and Egil Eide for assistance in collecting the data from milliAmpere2, and Simen Eldevik at DNV for providing ideas that contributed to the improvement of the Gauss–Newton optimization method used in the paper.

APPENDIX A
PARTIAL DERIVATIVES FOR NEGATIVE INFORMATION

Following the approach in [11], we divide the derivative into the following components:

$$\frac{dh^{max}(\mathbf{x})}{d\mathbf{x}} = \begin{bmatrix} \frac{dh^{max}(\mathbf{x})}{d\mathbf{x}^c} & \frac{dh^{max}(\mathbf{x})}{d\phi} & \frac{dh^{max}(\mathbf{x})}{d\mathbf{x}^f} \end{bmatrix}. \tag{69}$$

The time index has been omitted for brevity. The derivatives can be found by applying the chain rule and the quotient rule. We start by making the substitution

$$u = \frac{u_y}{u_x} = \frac{y^p + \sin\left(\theta^f_{max} + \phi\right)\mathbf{H}^{\mathbf{f}}\left(\theta^f_{max}\right)\mathbf{x}^f}{x^p + \cos(\theta^f_{max} + \phi)\mathbf{H}^{\mathbf{f}}\left(\theta^f_{max}\right)\mathbf{x}^f}. \tag{70}$$

We also introduce the following shorthands:

$$\begin{aligned} \mathbf{H}^f\left(\theta^f_{max}\right) &= \mathbf{H}^f \\ \theta^f_{max} + \phi &= \theta^{(G)}_{max}. \end{aligned} \tag{71}$$

The derivative of $\arctan(u)$ is $\frac{1}{1+u^2}$; therefore, according to the chain rule, we get

$$\frac{dh^{max}(\mathbf{x})}{d\mathbf{x}} = \frac{1}{1+u^2}\begin{bmatrix} \frac{du}{d\mathbf{x}^c} & \frac{du}{d\phi} & \frac{du}{d\mathbf{x}^f} \end{bmatrix}. \tag{72}$$

To calculate the partial derivatives, we apply the quotient rule. Doing this results in

$$\begin{aligned}
\frac{du}{dx^p} &= -\frac{y^p + \sin\left(\theta_{max}^{(G)}\right)\mathbf{H}^f\mathbf{x}^f}{{u_x}^2} \\
\frac{du}{dy^p} &= \frac{1}{u_x} \\
\frac{du}{d\phi} &= \frac{\mathbf{H}^f\mathbf{x}^f\left(\mathbf{H}^f\mathbf{x}^f + x^p\cos\left(\theta_{max}^{(G)}\right) + y^p\sin\left(\theta_{max}^{(G)}\right)\right)}{{u_x}^2} \\
\frac{du}{d\mathbf{x}^f} &= \frac{\mathbf{H}^f\left(x^p\sin\left(\theta_{max}^{(G)}\right) - y^p\cos\left(\theta_{max}^{(G)}\right)\right)}{{u_x}^2}.
\end{aligned} \tag{73}$$

By expanding

$$\frac{1}{1+u^2} = \frac{{u_x}^2}{{u_x}^2 + {u_y}^2}, \tag{74}$$

then multiplying this expression with the ones above we end up with the final partial derivatives

$$\begin{aligned}
\frac{du}{dx} &= -\frac{y + \sin\left(\theta_{max}^{(G)}\right)\mathbf{H}^f\mathbf{x}^f}{{u_x}^2 + {u_y}^2} \\
\frac{du}{dy} &= \frac{x^p + \cos\left(\theta_{max}^{(G)}\right)\mathbf{H}^f\mathbf{x}^f}{{u_x}^2 + {u_y}^2} \\
\frac{du}{d\phi} &= \frac{\mathbf{H}^f\mathbf{x}^f\left(\mathbf{H}^f\mathbf{x}^f + x\cos\left(\theta_{max}^{(G)}\right) + y\sin\left(\theta_{max}^{(G)}\right)\right)}{{u_x}^2 + {u_y}^2} \\
\frac{du}{d\mathbf{x}^f} &= \frac{\mathbf{H}^f\left(x\sin\left(\theta_{max}^{(G)}\right) - y\cos\left(\theta_{max}^{(G)}\right)\right)}{{u_x}^2 + {u_y}^2}.
\end{aligned} \tag{75}$$

It follows that the partial derivatives for the minimum angle can be found by substituting θ_{max}^f with θ_{min}^f.

APPENDIX B
PMBM FLTER RECURSIONS FOR A GP TARGET MODEL

B.1. Prediction

As stated in the main text, we utilize the expressions derived for an extended object PMBM filter in [3] and use those to derive closed-form expressions.

The posterior distribution of a PMBM prior is a PMBM with parameters

$$D_{k-1}^u, \{w_{k-1}^j, \{r_{k-1}^{j,i}, (f_{k-1}^{j,i})\}_{i\in\mathbb{I}_{k|k'}}\}_{j\in\mathbb{J}_{k|k'}}. \tag{76}$$

Then, the predicted distribution is a PMBM with parameters

$$D_{k|k-1}^u, \{w_{k|k-1}^j, \{r_{k|k-1}^{j,i}, (f_{k|k-1}^{j,i})\}_{i\in\mathbb{I}_{k|k'}}\}_{j\in\mathbb{J}_{k|k'}}, \tag{77}$$

where the parameters are given by

$$\begin{aligned}
D_{k|k-1}^u &= D^b(\mathbf{x}) + \langle D_{k-1}^u; P_S\mathbf{g}_{k|k-1}(\mathbf{x})\rangle \\
w_{k|k-1}^j &= w_{k-1}^j \\
r_{k|k-1}^{j,i} &= \langle f_{k-1}^{j,i}; P_S\rangle r_{k-1}^{j,i} \\
f_{k|k-1}^{j,i} &= \frac{\langle f_{k-1}^{j,i}; P_S\mathbf{g}_{k|k-1}(\mathbf{x})\rangle}{\langle f_{k-1}^{j,i}; P_S\rangle}.
\end{aligned} \tag{78}$$

For the Gaussians, we can derive the expressions using the product rule and integrating the resulting expression

$$\begin{aligned}
&\left\langle f_{k-1}^{j,i}; P_S\mathbf{g}_{k|k-1}(\mathbf{x})\right\rangle \\
&= P_S\int \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{x}_{k-1}^{j,i}, \mathbf{P}_{k-1})\mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}', \mathbf{Q})d\mathbf{x}_{k-1} \\
&= P_S\mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}_{k-1}^{j,i}, \mathbf{F}\mathbf{P}_{k|k-1}^{j,1}\mathbf{F}^{\mathrm{T}} + \mathbf{Q})\int \mathcal{N}(\mathbf{x}_{k-1}; \cdot, \cdot)d\mathbf{x}_{k-1} \\
&= P_S\mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}_{k-1}^{j,i}, \mathbf{F}\mathbf{P}_{k|k-1}^{j,1}\mathbf{F}^{\mathrm{T}} + \mathbf{Q}) \\
&\langle D_{k-1}; P_S\mathbf{g}_{k|k-1}(\mathbf{x})\rangle \\
&= P_S\int \sum_{n=1}^{N^u} d_n^u\mathcal{N}(\mathbf{x}_{k-1}, \mathbf{x}_n^u, \mathbf{P}_n^u)\mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}', \mathbf{Q})d\mathbf{x}_{k-1} \\
&= P_S\sum_{n=1}^{N^u} d_n^u\mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}_n^u, \mathbf{F}\mathbf{P}_n^u\mathbf{F}^{\mathrm{T}} + \mathbf{Q})\int \mathcal{N}(\mathbf{x}_{k-1}; \cdot, \cdot)d\mathbf{x}_{k-1} \\
&= P_S\sum_{n=1}^{N^u} d_n^u\mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}_n^u, \mathbf{F}\mathbf{P}_n^u\mathbf{F}^{\mathrm{T}} + \mathbf{Q}) \\
&\left\langle f_k^{j,i}; P_S\right\rangle = P_S\int f_{k-1}^{j,i}(\mathbf{x})d\mathbf{x} = P_S.
\end{aligned} \tag{79}$$

The gamma component does not have a similar expression, and we instead use the heuristic defined in [24] and presented in (18). Inserting these expressions into (78) and simplifying results in (34).

B.2. Update

Starting with the predicted PMBM and given the set of measurements Z_k. The posterior distribution is then also a PMBM with parameters

$$D_k^u, \{w_k^{j,A}, \{r_k^{j,C}, (f_k^{j,C})\}_{C\in A}\}_{j\in\mathbb{J}_{k|k-1}, A\in\mathcal{A}^j}, \tag{80}$$

where $\mathcal{A}^j$ is the set of all possible data associations for the association hypothesis with index j. If measurement cell C belongs to a detected target, the following

expressions apply:

$$
\begin{aligned}
L_C^j &= \begin{cases} 1 - r_{k|k-1}^{j,i_C} + r_{k|k-1}^{j,i_C} \langle f_{k|k-1}^{j,i_C}; Q_D \rangle & |Z_C| = 0 \\ r_{k|k-1}^{j,i_C} \langle f_{k|k-1}^{j,i_C}; l(Z_C|\mathbf{x}) \rangle & |Z_C| \neq 0 \end{cases} \\
r_k^{j,i_C} &= \begin{cases} \frac{r_{k|k-1}^{j,i_C} \langle f_{k|k-1}^{j,i_C}; Q_D \rangle}{1 - r_{k|k-1}^{j,i_C} + r_{k|k-1}^{j,i_C} \langle f_{k|k-1}^{j,i_C}; Q_D \rangle} & |Z_C| = 0 \\ 1 & |Z_C| \neq 0 \end{cases} \\
f_k^{j,i_C}(\mathbf{x}) &= \begin{cases} \frac{Q_D f_{k|k-1}^{j,i_C}}{\langle Q_D; f_{k|k-1}^{j,i_C} \rangle} & |Z_C| = 0 \\ \frac{l(Z_C|\mathbf{x}) f_{k|k-1}^{j,i_C}}{\langle l(Z_C|\mathbf{x}); f_{k|k-1}^{j,i_C} \rangle} & |Z_C| \neq 0 \end{cases}.
\end{aligned} \tag{81}
$$

If measurements are assigned to an undetected target, we instead have the following expressions:

$$
\begin{aligned}
L_C^j &= \begin{cases} D^c + \langle D_{k|k-1}^u; l(Z_C|\mathbf{x}) \rangle & |Z_C| = 1 \\ \langle D_{k|k-1}^u; l(Z_C|\mathbf{x}) \rangle & |Z_C| > 1 \end{cases} \\
r_k^{j,i_C} &= \begin{cases} \frac{\langle D_{k|k-1}^u; l(Z_C|\mathbf{x}) \rangle}{D^c + \langle D_{k|k-1}^u; l(Z_C|\mathbf{x}) \rangle} & |Z_C| = 1 \\ 1 & |Z_C| > 1 \end{cases} \\
f_k^{j,i_C}(\mathbf{x}) &= \frac{l(Z_C|\mathbf{x}) D_{k|k-1}^u}{\langle l(Z_C|\mathbf{x}); D_{k|k-1}^u \rangle}.
\end{aligned} \tag{82}
$$

Q_D is assumed scalar, so we get

$$
\begin{aligned}
\langle Q_D; f_{k|k-1}^{j,i_C} \rangle &= Q_D \int f_{k|k-1}^{j,i_C}(\mathbf{x}) d\mathbf{x} = Q_D \\
\langle Q_D; D_{k|k-1}^u \rangle &= Q_D \int D_{k|k-1}^u(\mathbf{x}) d\mathbf{x} = Q_D \\
Q_D f_{k|k-1}^{j,i_C} &= Q_D \mathcal{N}(\mathbf{x}, \mathbf{x}_{k|k-1}^{j,i}, \mathbf{P}_{k|k-1}),
\end{aligned} \tag{83}
$$

where the integrals all evaluate to 1 since the integrands are probability distribution functions. The remaining expressions are a collection of products and inner products of distributions. We can write a generic product

$$
\begin{aligned}
&p(\mathbf{x}) l(Z_C|\mathbf{x}) = \\
&P_D \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \mathcal{G}(\lambda_m; \alpha_{k|k-1}, \beta_{k|k-1}) \\
&\quad \times P_D e^{-\lambda_m} \lambda_m^{|Z_C|} \mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \mathbf{R}).
\end{aligned} \tag{84}
$$

Given that we have assumed independence between the measurement rate and the combined state and extent, we can treat the Gaussian components and the gamma and Poisson components separately. First, we evaluate the products of the Gaussian distributions, which is done using the product rule for Gaussians, which states that

$$
\begin{aligned}
&\mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \mathbf{R}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \\
&= \mathcal{N}(\mathbf{z}; \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_k, \mathbf{P}_k),
\end{aligned} \tag{85}
$$

where the first term is recognized as the posterior distribution and the second term is the predictive likelihood. The inner product can consequently be written as

$$
\begin{aligned}
&\int \mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \mathbf{R}) \mathcal{N}(\mathbf{x}; \mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) d\mathbf{x} \\
&= \mathcal{N}(\mathbf{z}; \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}) \int \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_k, \mathbf{P}_k) d\mathbf{x} \\
&= \mathcal{N}(\mathbf{z}; \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}),
\end{aligned} \tag{86}
$$

which corresponds to a marginalization of $\mathbf{x}$. For the gamma component, the key element is that the gamma distribution is the conjugate prior of the Poisson distribution, and the number of measurements $|Z_C|$ is Poisson distributed $\mathcal{PS}(|Z_C|; \lambda_m)$. In [24], it was shown that the product evaluates to

$$
\begin{aligned}
&\mathcal{G}(\lambda_m; \alpha_{k|k-1}, \beta_{k|k-1}) \mathcal{PS}(|Z_C|; \lambda_m) \\
&= \mathcal{G}(\lambda_m; \alpha_{k|k-1} + |Z_C|, \beta_{k|k-1} + 1) \\
&\quad \times \frac{\Gamma(\alpha_{k|k-1} + |Z_C|) \beta_{k|k-1}^{\alpha_{k|k-1}}}{\Gamma(\alpha_{k|k-1}) (\beta_{k|k-1} + 1)^{(\alpha_{k|k-1} + |Z_C|)} |Z_C|!},
\end{aligned} \tag{87}
$$

where the first term is the posterior of the gamma distribution and the second term is the predictive likelihood. From this, we can combine the two expressions and write a combined predictive likelihood as

$$
\begin{aligned}
l_C(\alpha, \beta, \mathbf{x}, \mathbf{P}, Z_C) = P_D &\frac{\Gamma(\alpha + |Z_C|) \beta^\alpha}{\Gamma(\alpha) (\beta + 1)^{(\alpha + |Z_C|)} |Z_C|!} \\
&\times \mathcal{N}(\mathbf{z}; \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}).
\end{aligned} \tag{88}
$$

The posterior can be found from Bayes rule

$$
\frac{p(\mathbf{x}) l(Z_C|\mathbf{x})}{\langle p(\mathbf{x}); l(Z_C|\mathbf{x}) \rangle} = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_k, \mathbf{P}_k) \mathcal{G}(\lambda_m; \alpha_k, \beta_k). \tag{89}
$$

Replacing the generic probability density p with f^{j,i_C} or the components of the mixture D^u results in the following predictive likelihoods:

$$
\begin{aligned}
\langle l(Z_C|\mathbf{x}); f_{k|k-1}^{j,i_C} \rangle &= l_C(\alpha^{j,i_C}, \beta^{j,i_C}, \mathbf{x}_{k|k-1}^{j,i_C}, \mathbf{P}_{k|k-1}^{j,i_C}, Z_C) \\
\langle l(Z_C|\mathbf{x}); D_{k|k-1}^u \rangle &= \sum_{n=1}^{N^u} d_n^u l_C(\alpha_n^u, \beta_n^u, \mathbf{x}_n^u, \mathbf{P}_n^u, Z_C)
\end{aligned} \tag{90}
$$

and the following posterior distributions:

$$
\begin{aligned}
\frac{l(Z_C|\mathbf{x}) D_{k|k-1}^u}{\langle l(Z_C|\mathbf{x}); D_{k|k-1}^u \rangle} &= \sum_{n=1}^{N^u} d_n^u \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_n^u, \hat{\mathbf{P}}_n^u) \mathcal{G}(\alpha_n^u, \beta_n^u) \\
\frac{l(Z_C|\mathbf{x}) f_{k|k-1}^{j,i_C}}{\langle l(Z_C|\mathbf{x}); f_{k|k-1}^{j,i_C} \rangle} &= \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_k^{j,i_C}, \hat{\mathbf{P}}_k^{j,i_C}) \mathcal{G}(\alpha_k^{j,i_C}, \beta_k^{j,i_C}).
\end{aligned} \tag{91}
$$

Inserting these expressions into (81) and (82) results in (46) and (47), respectively.

REFERENCES

[1] K. Granström, M. Baum, and S. Reuter
"Extended object tracking: Introduction, overview and applications,"
J. Adv. Inf. Fusion, vol. 12, no. 2, pp. 139–174, Dec. 2017.

[2] J. W. Koch
"Bayesian approach to extended object and cluster tracking using random matrices,"
IEEE Trans. Aerosp. Electron. Syst., vol. 44, no. 3, pp. 1042–1059, Jul. 2008.

[3] K. Granström, M. Fatemi, and L. Svensson
"Poisson multi-Bernoulli mixture conjugate prior for multiple extended target filtering,"
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 1, pp. 208–225, Feb. 2020.

[4] J. L. Williams
"Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based member,"
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 3, pp. 1664–1687, Jul. 2015.

[5] Y. Xia, K. Granström, L. Svensson, A. F. García-Fernández, and J. L. Williams
"Extended target Poisson multi-Bernoulli mixture trackers based on sets of trajectories,"
in *Proc. 22th Int. Conf. Inf. Fusion*, Jul. 2019, pp. 1–8.

[6] K. Granström, L. Svensson, S. Reuter, Y. Xia, and M. Fatemi
"Likelihood-based data association for extended object tracking using sampling methods,"
IEEE Trans. Intell. Veh., vol. 3, no. 1, pp. 30–45, Mar. 2018.

[7] Y. Xia, K. Granström, L. Svensson, M. Fatemi, A. F. García-Fernández, and J. L. Williams
"Poisson multi-Bernoulli approximations for multiple extended object filtering,"
IEEE Trans. Aerosp. Electron. Syst., vol. 58, no. 2, pp. 890–906, Apr. 2022.

[8] Y. Xia, A. F. García-Fernández, F. Meyer, J. L. Williams, K. Granström, and L. Svensson
"Trajectory PMB filters for extended object tracking using belief propagation,"
IEEE Trans. Aerosp. Electron. Syst., vol. 59, no. 6, pp. 9312–9331, 2023.

[9] F. Meyer and J. L. Williams
"Scalable detection and tracking of geometric extended objects,"
IEEE Trans. Signal Process., vol. 69, pp. 6283–6298, 2021.

[10] M. Baum and U. D. Hanebeck
"Shape tracking of extended objects and group targets with star-convex RHMs,"
in *Proc. 14th Int. Conf. Inf. Fusion*, 2011, pp. 1–8.

[11] N. Wahlstrom and E. Ozkan
"Extended target tracking using Gaussian processes,"
IEEE Trans. Signal Process., vol. 63, no. 16, pp. 4165–4178, Aug. 2015.

[12] M. Kumru, H. Köksal, and E. Özkan
"Variational measurement update for extended object tracking using Gaussian processes,"
IEEE Signal Process. Lett., vol. 28, pp. 538–542, 2021.

[13] M. Michaelis, P. Berthold, T. Luettel, D. Meissner, and H.-J Wuensche
"Extended object tracking with an improved measurement-to-contour association,"
in *Proc. IEEE 23rd Int. Conf. Inf. Fusion*, Jul. 2020, pp. 1–6.

[14] S. Lee and J. McBride
"Extended object tracking via positive and negative information fusion,"
IEEE Trans. Signal Process., vol. 67, no. 7, pp. 1812–1823, Apr. 2019.

[15] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer
"Multiple extended object tracking using Gaussian processes,"
in *Proc. 19th Int. Conf. Inf. Fusion*, 2016, pp. 868–875.

[16] M. Michaelis, P. Berthold, D. Meissner, and H.-J Wuensche
"Heterogeneous multi-sensor fusion for extended objects in automotive scenarios using Gaussian processes and a GMPHD-filter,"
in *Proc. Sensor Data Fusion: Trends, Solutions, Appl. (SDF)*, 2017, pp. 1–6.

[17] Y. Xia, K. Granström, L. Svensson, and A. F. García-Fernández
"Performance evaluation of multi-Bernoulli conjugate priors for multi-target filtering,"
in *Proc. 20th Int. Conf. Inf. Fusion*, 2017, pp. 1–8.

[18] K. Granström, C. Lundquist, and O. Orguner
"Extended target tracking using a Gaussian-mixture PHD filter,"
IEEE Trans. Aerosp. Electron. Syst., vol. 48, no. 4, pp. 3268–3286, Oct. 2012.

[19] K. Granström and U. Orguner
"A PHD filter for tracking multiple extended targets using random matrices,"
IEEE Trans. Signal Process., vol. 60, no. 11, pp. 5657–5671, Nov. 2012.

[20] K. Wyffels and M. Campbell
"Negative information for occlusion reasoning in dynamic extended multiobject tracking,"
IEEE Trans. Robot., vol. 31, no. 2, pp. 425–442, Apr. 2015.

[21] M. Baerveldt, M. E. Lopez, and E. F. Brekke
"Extended target PMBM tracker with a Gaussian process target model on LiDAR data,"
in *Proc. 26th Int. Conf. Inf. Fusion*, 2023, pp. 8.

[22] C. E. Rasmussen and C. K. I. Williams
Gaussian Processes for Machine Learning, in Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 2006.

[23] M. E. Lopez
"Poisson multi-Bernoulli mixture filter for multiple extended object tracking of maritime vessels using LiDAR and Gaussian processes," 2020. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2781005

[24] K. Granström and U. Orguner
"Estimation and maintenance of measurement rates for multiple extended target tracking,"
in *Proc. 15th Int. Conf. Inf. Fusion*, 2012, pp. 2170–2176.

[25] C. Lundquist, K. Granström, and U. Orguner
"An extended target CPHD filter and a gamma Gaussian inverse Wishart implementation,"
IEEE J. Sel. Topics Signal Process., vol. 7, no. 3, pp. 472–483, Jun. 2013.

[26] B. Bell and F. Cathey
"The iterated Kalman filter update as a Gauss–Newton method,"
IEEE Trans. Autom. Control, vol. 38, no. 2, pp. 294–297, Feb. 1993.

[27] A. S. Rahmathullah, A. F. García-Fernández, and L. Svensson
"Generalized optimal sub-pattern assignment metric,"
in *Proc. 20th Int. Conf. Inf. Fusion*, 2017, pp. 1–8.

[28] E. F. Brekke, E. Eide, B.-O. H. Eriksen, E. F. Wilthil, M. Breivik, E. Skjellaug, Ø. K. Helgesen, A. M. Lekkas, A. B. Martinsen, E. H. Thyri, T. Torben, E. Veitch, O. A. Alsos, and T. A. Johansen
"milliAmpere: An Autonomous Ferry Prototype," J. Phys.: Conf. Ser., vol. 2311, no. 1, pp. 012029, Jul. 2022, publisher: IOP Publishing.

[29] Ø. K. Helgesen, K. Vasstein, E. F. Brekke, and A. Stahl
"Heterogeneous multi-sensor tracking for an autonomous surface vehicle in a littoral environment,"
Ocean Eng., vol. 252, May 2022.

[30] Y. Xia, A. F. García-Fernández, and L. Svensson
"An efficient implementation of the extended object trajectory PMB filter using blocked Gibbs sampling,"
in *Proc. 26th Int. Conf. Inf. Fusion*, 2023, pp. 1–8.

[31] M. Kumru and E. Özkan
"Three-dimensional extended object tracking and shape learning using Gaussian processes,"

IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 5, pp. 2795–2814, Oct. 2021.

[32] T. Baur, J. Reuter, A. Zea, and U. D. Hanebeck "Extent estimation of sailing boats applying elliptic cones to 3D LiDAR data," in *Proc. 2022 25th Int. Conf. Inf. Fusion*, 2022, pp. 1–8.

[33] A. F. García-Fernández, Y. Xia, and L. Svensson "Poisson multi-Bernoulli mixture filter with general target-generated measurements and arbitrary clutter," *IEEE Trans. Signal Process.*, vol. 71, pp. 1895–1906, 2023.

[34] E. Brekke, O. Hallingstad, and J. Glattetre "Improved target tracking in the presence of wakes," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 2, pp. 1005–1017, 2012.

[35] A. G. Hem, H.-G. Alvheim, and E. F. Brekke "Wakeipda: Target tracking with existence modeling in the presence of wakes," in *Proc. 26th Int. Conf. Inf. Fusion*, 2023, pp. 1–7.

Martin Baerveldt received the M.Sc. degree in complex adaptive systems from Chalmers University of Technology, Gothenburg, Sweden, in 2019. He is currently pursuing Ph.D. at the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He is a part of the European training and research network on Autonomous Barges for Smart Inland Shipping (AUTOBarge). His research focuses on automated situational awareness with a view to autonomous surface vessels, with a particular focus on extended object tracking.

Michael Ernesto López received the M.Sc. degree in mathematics from the Complutense University of Madrid, Spain, in 2012, and the M.Sc. degree in engineering cybernetics from Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2021. Since 2021, he has been pursuing Ph.D. at the Department of Engineering Cybernetics, NTNU, with a focus on detection and pose estimation of maritime vessels.

Edmund Førland Brekke received the M.Sc. degree in industrial mathematics in 2005 and the Ph.D. degree in engineering cybernetics in 2010, both from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. From 2010 to 2014, he worked with the Acoustic Research Laboratory (ARL), NUS in Singapore as a Postdoctoral Research Fellow. In 2014, he rejoined NTNU and the Department of Engineering Cybernetics, where he is currently a Professor of sensor fusion. Brekke's research interests are in the area of sensor fusion and situational awareness, with a particular focus on multitarget tracking and maritime surface autonomy. He is an Associate Editor of the IEEE Journal of Oceanic Engineering.

SalFAU-Net: Saliency Fusion Attention U-Net for Salient Object Detection

KASSAW ABRAHAM MULAT
ZHENGYONG FENG
TEGEGNE SOLOMON ESHETIE
AHMED ENDRIS HASEN

Salient object detection (SOD) remains an important task in computer vision, with applications ranging from image segmentation to autonomous driving. Fully convolutional network-based methods have made remarkable progress in visual saliency detection over the last few decades. However, these methods have limitations in accurately detecting salient objects, particularly in challenging scenes with multiple objects, small objects, or objects with low resolutions. To address this issue, we proposed a Saliency Fusion Attention U-Net (SalFAU-Net) model, which incorporates an saliency fusion module into each decoder block of the attention U-net model to generate saliency probability maps from each decoder block. SalFAU-Net employs an attention mechanism to selectively focus on the most informative regions of an image and suppress nonsalient regions. We train SalFAU-Net on the DUTS dataset using a binary cross-entropy loss function. We conducted experiments on six popular SOD evaluation datasets to evaluate the effectiveness of the proposed method. The experimental results demonstrate that our method, SalFAU-Net, achieves competitive performance compared to other methods in terms of mean absolute error, *F*-measure, *S*-measure, and *E*-measure.

Manuscript received February 5, 2024; revised October 22, 2024; released for publication January 10, 2025

Refereeing of this contribution was handled by Ruixin Niu.

K. A. Mulat and T. S. Eshetie are with the School of Computer Science, China West Normal University, Nanchong, 637002 Sichuan, China (e-mail: abraham88@stu.cwnu.edu.cn; solomoneshetie8@gmail.com).

Z. Feng is with the School of Electronic and Information Engineering, China West Normal University, Nanchong, 637002 Sichuan, China (e-mail: zhyfeng@cwnu.edu.cn).

A. E. Hasen is with the Paul C. Lauterbur Research Center, Shenzhen Institute of Advanced Technology, Chinese Academy of Science, Shenzhen, Guangdong 518055, China (e-mail: ahmeduestc53@outlook.com).

(Corresponding author. Kassaw Abraham Mulat.)

I. INTRODUCTION

Salient object detection (SOD), also referred to as visual saliency detection, is detecting the most noticeable, unique, and visually distinct objects or regions in a scene that attract the human eye [3]. The human visual perception system has an exceptional ability to rapidly recognize and focus its attention toward visually unique and prominent objects or regions within scenes [37]. This innate capability has captivated the interest of many researchers in the field of computer vision, where the aim is to simulate this process based on the psychological and biological properties of the human visual attention system. The goal is to identify prominent objects in images and videos that hold significant importance and valuable information.

Given the diverse applications of SOD in various domains of computer vision, it plays a crucial role as a preprocessing step in tasks like image segmentation [1], [7], [12], [31], object detection [4], [35], [33], image captioning [43], autonomous driving [29], and augmented reality [8]. Numerous visual saliency detection methods have been proposed. These methods aim to distinguish the most unique foreground images from less significant backgrounds. While traditional saliency detection approaches rely on low-level heuristic visual features, these methods often fail to detect salient objects in challenging scenes. Recently, deep learning methods, particularly convolutional neural networks (CNNs), have exhibited exceptional efficacy across diverse computer vision tasks, including saliency detection. In contrast to traditional methods, CNN-based methods have made remarkable advancements by harnessing advanced semantic features [16].

Due to the significant impact of representative features on algorithms performance, it is beneficial to investigate models that leverage multilevel features and contextual information to enhance saliency detection. Furthermore, despite the introduction of end-to-end models based on fully convolutional networks (FCNs), there remains significance in incorporating and advancing conventional FCN models like U-Net [34] and its variate for the task of saliency detection. One of the variants of U-Net that is well known for its efficacy in medical image segmentation is the Attention U-Net network [28], which selectively focuses on relevant regions of the input image by integrating attention mechanisms into its architecture, which improves the model's ability to capture intricate patterns and important features. The attention mechanism facilitates improved performance in tasks such as image segmentation. Drawing on its success in medical image segmentation, this study explores the application of Attention U-Net for saliency detection tasks. We added a saliency fusion module (SFM) to each decoder block of the network. This module allows us to generate saliency maps effectively, which we then concatenate with each decoder's side output saliency map to get the final saliency map. The

attention gate (AG) module in the proposed method helps the model learn to focus on salient features with varying sizes and shapes. In this way, SalFAU-Net has the ability to suppress irrelevant regions from an input image while emphasizing the features that are most important for saliency detection. To summarize, the main contributions of this paper are as follows:

(1) We proposed a Saliency Fusion Attention U-Net (SalFAU-Net) for the task of visual saliency detection.

(2) SFM is added to each decoder block of the network to generate a saliency map from each decoder, and these saliency maps are concatenated together to obtain the ultimate visual representation that highlights the most important areas or objects in an image.

(3) We conducted experiments on six publicly available challenging SOD datasets, and the results demonstrate the effectiveness of SalFAU-Net for the task of visual saliency detection.

II. RELATED WORKS

Generally, saliency detection methods can be classified into two categories. These are traditional methods and deep learning-based methods. Traditional methods are based on low-level heuristic visual features such as contrast, location, and texture. Most of these methods are unsupervised or semi-supervised. Examples of traditional saliency detection methods include those based on local contrast [22], global contrast [44], backgroundness prior [45], center prior [42], objectness prior [20], and others. These methods achieved good results in uncomplicated images or scenarios featuring solitary objects. Nonetheless, these methods failed to detect salient objects that are in complex scenes, low resolutions, or scenes with multiple salient objects. This limitation arises from their reliance on low-level features, which prove inadequate for addressing the complexities introduced by such challenging visual contexts.

Recently, deep learning-based methods, particularly CNNs, have demonstrated remarkable performance across diverse computer vision tasks, including image classification [17], semantic image segmentation [24], and object detection [41]. CNNs have the capability to learn rich and hierarchical representations of input data by extracting high-level semantic features. However, in SOD, both low-level and high-level features are important for developing good visual saliency detection models. The introduction of FCNs [25] has revolutionized the approach to end-to-end pixel-level saliency detection. Initially designed for semantic segmentation, FCN seamlessly combines the tasks of feature extraction and pixel label prediction in a single network structure composed of down-sampling and upsampling paths. Subsequently, numerous FCN-based visual saliency detection models have been proposed, including deep contrast learning (DCL) [19], aggregating multilevel convolutional feature frameworks (Amulet) [46], recurrent fully convolutional networks (RFCN) [39], and deep uncertain convolutional features (UCF) [47]. These advancements have notably enhanced the effectiveness of algorithms designed for visual saliency detection. Nonetheless, exploring effective FCN-based models designed for different purposes is still beneficial. U-Net is one of the most widely used networks in medical image segmentation [34]. Following the success of U-Net, numerous network variations have been introduced for different tasks. One exemplary variant of U-Net is the Attention U-Net model, which is designed for pancreas image segmentation, has shown impressive results in other tissue and organ segmentation, benefiting from the AG module to focus on relevant and variable size regions in an image. Most FCN-based saliency models are based on plain U-Net and have achieved remarkable performance for saliency detection. In [32], Qin *et al.* proposed a two-level nested U-structure by using a residual U-block (RSU) as a backbone for visual saliency detection. Compared to many other networks that use pretrained networks as backbones, U-2-Net's RSU block increases architecture depth without significantly increasing computational costs while achieving competitive performance. In [14], Han *et al.* proposed a modified U-Net network for saliency detection, utilizing an edged convolution constraint. This variant effectively integrates features from multiple layers, reducing information loss and enabling pixel-wise saliency map prediction rather than patch-level prediction, which is common in CNN-based models.

Although these methods based on plain U-Net achieved remarkable performance for saliency detection, their performance can be boosted by incorporating different techniques into the encoder and decoder blocks of their architecture. Recently, attention mechanisms have shown remarkable results across various computer vision applications, encompassing saliency detection. In [21], Li *et al.* proposed a U-shape network with stacked layers incorporating channel-wise attention to extract the most important channel features and effectively utilize these features by integrating a parallel dilated convolution (PDC) module and a multilevel attention cascaded feedback (MACF) module.

In order to recurrently translate and aggregate the context features separately with various attenuation factors, Hu *et al.* [15] proposed a spatial attenuation context module. After that, the module carefully learned the weights to adaptively incorporate the collective contextual features. In [48], Zhang *et al.* proposed a novel approach to visual saliency detection that leverages attention mechanisms for refining saliency maps, incorporating bi-directional refinement for enhanced accuracy. The introduction of bi-directional refinement highlights the focus on comprehensive feature extraction and optimization. In [49], Zhao and Wu applied spatial attention (SA) and channel-wise attention (CA) to distinct aspects of the model. Specifically, SA was employed for low-level feature maps, while CA was incorporated into context-

aware pyramid feature maps. This strategic approach aims to direct the network's focus toward the most relevant features for the given sample. In [13], Gong*et al.* proposed an enhanced U-Net model incorporating pyramid feature attention, channel attention, and a pyramid feature extraction module to improve the performance of the U-Net backbone network.

In this research, we attempt to explore the applications of attention U-Net architecture in the realm of visual saliency detection. We added an SFM to each decoder of the network and concatenated their output to obtain the final saliency map. The AG module in the proposed method helps the model learn to focus on salient features with varying sizes and shapes. Thus, SalFAU-Net adeptly learns the capability to suppress irrelevant or undesirable regions within an input image while highlighting the most crucial and salient features essential for the task of saliency detection.

III. METHODOLOGY

In this section, we provide a detailed description of the architecture of our proposed method. This is followed by the network supervision, the datasets and evaluation metrics used, and the implementation details.

A. Architecture of SalFAU-Net

The proposed SalFAU-Net for visual saliency detection in this paper mainly consists of four parts: (1) a five-level encoder block, (2) a four-level decoder block, (3) an AG module, and (4) an SFM. Figure 1 shows the architecture of the proposed SalFAU-Net model. Compared with the Attention U-Net model proposed for pancreas image segmentation [28], we add an SFM to each decoder of the architecture and finally concatenate them together to obtain the final saliency map.

1) Encoder Block: Each encoder block consists of two convolutional layers, each followed by a batch normalization layer and ReLu activation function, which increases the number of feature maps from 3 to 1024. Max pooling with a stride of 2×2 is applied at the end of every block except the last block for downsampling, reducing the image size from 288×288 to 18×18. The encoder block progressively reduces the spatial resolution of feature maps while increasing the number of channels, capturing features at different scales.

2) Decoder Block: The decoder block is responsible for upsampling and generating salient maps. It consists of an up-sampling layer followed by two convolutional layers, batch normalization, and ReLU activation function. The decoder block is connected to the AG block through skip connections. Each decoder block reduces the number of feature maps by two while increasing the size of the spatial resolutions from 18×18 to 288×288. The goal is to recover spatial details lost during the downsampling in the encoder, facilitating precise localization and detection of salient objects.

3) AG Module: AGs have demonstrated remarkable effectiveness in capturing crucial regions, diminishing feature responses in irrelevant background areas, and eliminating the need to crop a region of interest (ROI) within an image. This is particularly important for the task of visual saliency detection. The integration of AGs into the conventional U-Net architecture enhances the model's ability to emphasize salient features transmitted through skip connections. Given a skip connection feature $F_s \in R^{C\times H\times W}$, where C is the number of channels, and H and W are the height and width of F, we first apply a convolution layer, batch norm, and ReLu activation function to obtain a key feature K, and let Q be the input from the previous layer or the gating signal obtained by applying a convolution layer followed by a batch

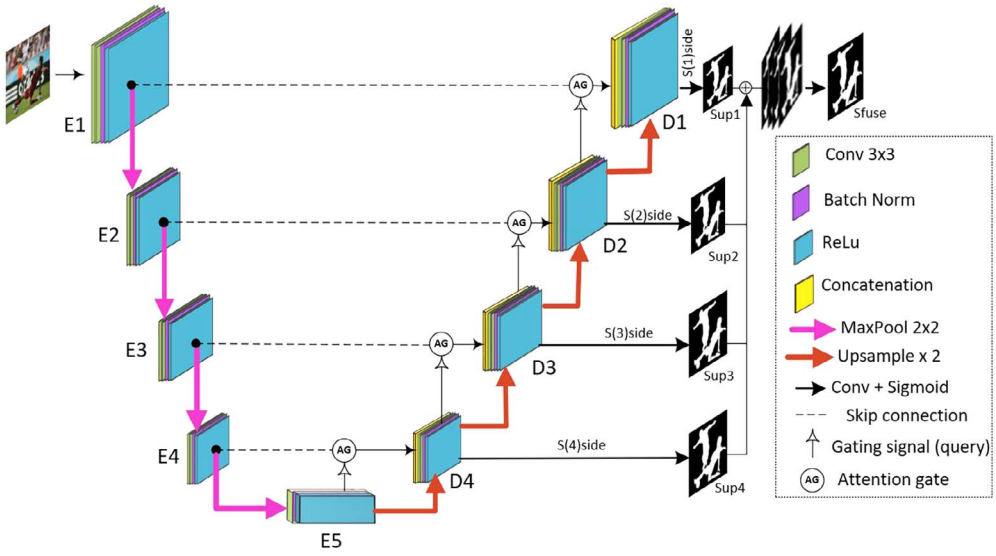


Figure 1. Architecture of our proposed Saliency Fusion Attention U-Net (SalFAU-Net model).

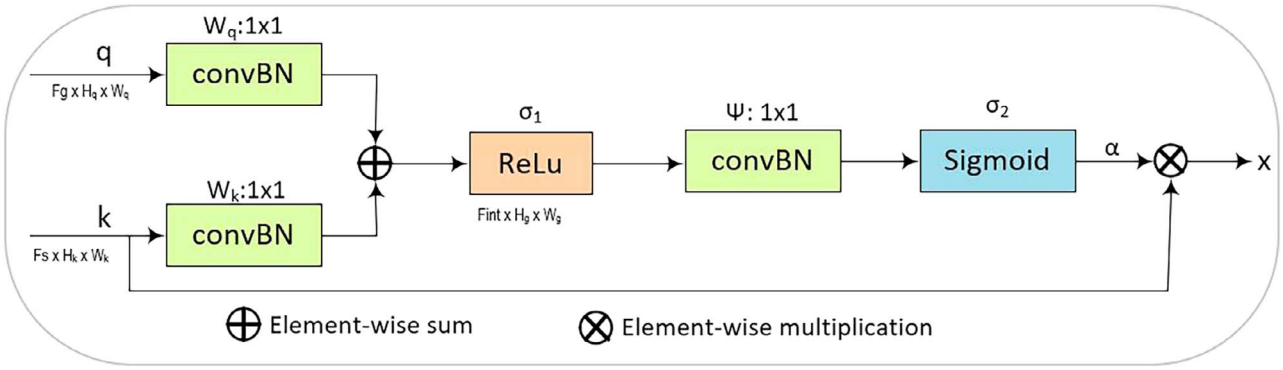


Figure 2. Diagram of the additive attention gate (AG) module.

normalization and relu activation to the input gate feature F_g. The attention coefficient α is obtained by applying a relu function to the element-wise sum of Q and K. The final attention coefficient value V is obtained by feeding a convolution layer, batch normalization, and sigmoid activation function to the attention coefficient α.

Finally, the attention coefficient value V and the skip connection feature map are multiplied element-wise to produce the final AG output $\hat{x}^l_{i,c}$, which is calculated as equation (3)

$$q^l_{att} = \Psi\left(\sigma_1\left(W_q^T \times q_i^l + W_k^T \times K_i + b_k\right)\right) + b_\Psi, \quad (1)$$

$$\alpha_i^l = \sigma_2\left(q^l_{att}\left(q_i^l, K_i; \Theta_{att}\right)\right), \quad (2)$$

$$\hat{x}^l_{i,c} = \alpha_i^l \cdot K_{i,c}, \quad (3)$$

where $\sigma_2(x_i, c) = \frac{1}{1+exp^{(-x_i,c)}}$ represents the sigmoid activation function. Thus, AG is defined by a set of parameter set θ_{att}, which includes linear transformations $W_k \in R^{F_l \times F_{\text{int}}}, W_q \in R^{F_g \times F_{\text{int}}}, \Psi \in R^{F_{\text{int}} \times 1}$ and bias terms $b_\Psi \in R, b_k \in R^{F_{\text{int}}}$. The linear transformation can be computed using channel-wise 1×1 convolutions for the input tensors.

4) Saliency Fusion Module: The saliency map fusion module serves as a pivotal component in generating saliency probability maps. Similar to the methodology in [32], our model undertakes a multistage approach. Initially, it generates four-side output saliency probability maps, denoted as $S(1)_{\text{side}}, S(2)_{\text{side}}, S(3)_{\text{side}}$, and $S(4)_{\text{side}}$, originating from the respective stages decoder1, decoder2, decoder3, and decoder4. This generation is facilitated by a 3×3 convolution layer, followed by a sigmoid activation function. Subsequently, the convolution outputs prior to sigmoid functions of these side-output saliency maps are upsampled to have the same size as the input image. The integration of these saliency maps is accomplished through a concatenation operation, followed by a 1×1 convolution layer and a sigmoid function. The result of this fusion process is the final saliency map S_{fuse} (depicted in the bottom right of Fig. 1).

Mathematically, the saliency probability maps at each stage are generated as follows:

$$S(i)_{\text{side}} = \sigma(\text{Conv}(i)(X)), \quad (4)$$

where i represents the stage (1, 2, 3, or 4), σ denotes the sigmoid function, Conv(i) is the convolution operation at stage i, and X is the decoder's output feature map. The side outputs are then upsampled and concatenated to generate the final saliency map S_{fuse}:

$$S_{\text{fuse}} = \sigma(\text{Conv}_{\text{fuse}}(\text{Concat}(S(i)_{\text{side}}))), \quad (5)$$

where Concat represents the concatenation operation, $\text{Conv}_{\text{fuse}}$ is the 1×1 convolution layer specific to the fusion process, and σ represents the sigmoid function.

B. Network Supervision

Loss functions play a significantly important role in optimizing a saliency detection model. One of the most widely employed loss functions for binary classification problems is the binary cross-entropy (BCE) loss [5]. For visual saliency detection, it measures the dissimilarity between the predicted saliency map and the ground truth in a binary classification setting.

We use a deep supervision approach similar to that in [32], which has demonstrated efficacy. Our training loss is formulated as follows:

$$\mathcal{L} = \sum_{m=1}^{M} w^m_{\text{side}} l^m_{\text{side}} + w_{\text{fuse}} l_{\text{fuse}} \quad (6)$$

The total loss comprises two components. The first component is the loss associated with the side-output saliency maps, denoted as l^m_{side}, where m represents the four supervision stages (Sup1, Sup2, Sup3, and Sup4) shown in Fig. 1. The second component is the loss of the final fusion-output saliency map, represented by l_{fuse}. The weights assigned to these loss terms are w^m_{side} and w_{fuse}, respectively.

We compute the loss for each term l using the conventional BCE to calculate the pixel-level comparison between the predicted saliency map and the ground

truth.

$$l = -\sum_{(x,y)}^{(H,W)} [G_{(x,y)} \log P_{(x,y)} + (1 - P_{(x,y)}) \log(1 - P_{(x,y)})], \tag{7}$$

where (H, W) is the height and width of the image, and (x, y) is the coordinate of a pixel. The ground truth and predicted saliency probability map's pixel values are represented by the symbols $G(x, y)$ and $P(x, y)$, respectively. The goal of the training procedure is to reduce the total loss $\mathcal{L}$ of (6). We select the fusion output l_{fuse} as our final saliency map during the testing process.

IV. EXPERIMENTAL RESULTS

A. Datasets

Training dataset: We train our model using the DUTS-TR dataset, which is a subset of the DUTS dataset [38]. DUTS-TR is curated from the training and validation sets of ImageNet DET [6], and comprises a total of 10 553 images, each with its corresponding ground truth. DUTS is the largest and most widely used dataset for saliency detection. We performed horizontally flipping data augmentation technique, resulting in 21,106 images for training.

Evaluation dataset: We use the following six widely used saliency detection datasets in order to evaluate the detection performances of our model.

ECSSD [36]: The ECSSD (Extended Complex Scene Saliency Dataset) contains semantically significant yet structurally complex and challenging images. This dataset contains 1000 natural images with carefully annotated ground truth saliency masks.

PASCAL-S [23]: This dataset was collected on eight subjects with a 3-second viewing time and the utilization of the Eyelink eye tracker collected from the PASCAL VOC (Visual Object Classes 2010) [9] validation dataset. This dataset contains 850 images featuring multiple salient objects within their scenes, providing a rich and diverse visual context.

HKU-IS [18]: The HKU-IS dataset is a more challenging benchmark for visual saliency detection, aimed at advancing the research and evaluating the performance of visual saliency models. This dataset comprises 4447 challenging images, featuring high-quality pixel-wise annotations with characteristics of either low contrast or presence of multiple salient objects.

DUT-OMRON [45]: DUT-OMRON comprises 5168 high-quality nature images meticulously chosen from more than 140 000 images. These images possess dimensions of either $400\times x$ or $x\times 400$ pixel dimensions, where x is less than 400. Notably, each image features one or more salient objects set against a relatively complex background.

DUTS-TE: DUTS-TE is the test set of the DUTS dataset, which comprises 5019 test images sourced from the ImageNet DET test set and the SUN dataset [40]. This dataset contains highly challenging scenarios for the evaluation of saliency detection models.

SOD [27]: SOD comprises salient object boundaries derived from the Berkeley Segmentation Dataset (BSD) [26]. It consists of 300 particularly challenging images, initially intended for image segmentation.

B. Evaluation Metrics

The probability maps that are produced by deep salient object algorithms often have the same dimension as the input images. In predicted saliency maps, each pixel has a value between 0 and 1 (or [0, 255]). The ground truths are often binary masks, where each pixel is either 0 or 1 (or 0 and 255), with 1 denoting the pixels of the foreground salient object and 0 denoting the background.

To comprehensively evaluate the performance of our model and the quality of the predicted saliency maps against the actual saliency masks, we used the following four evaluating measures: (1) mean absolute error (MAE) [30], (2) maximal F-measure ($maxF\beta$) [2], (3) structure measure (Sm) [10], and (4) enhanced alignment measure (Em) [11]. Using multiple evaluation metrics is crucial when evaluating an SOD model because it provides a more comprehensive assessment of the model's performance across different aspects. Each metric measures a specific quality of the prediction, and no single metric can fully capture all aspects of model generalization. The detailed descriptions of these measures are presented below.

1) F-measure*:* F-measure comprehensively evaluates both precision and recall as:

$$F_\beta = \frac{(1 + \beta_2)\text{Precision} \times \text{Recall}}{\beta_2 \text{Precision} + \text{Recall}}. \tag{8}$$

Since the rate of recall is not as important as precision, β_2 is empirically set to 0.3 to emphasize precision more.

2) Mean Absolute Error: MAE, or mean absolute error, represents the average difference per pixel between a predicted saliency map and its corresponding ground truth mask. It is used as a metric to accurately assess false negative pixels.

$$MAE = \frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{W} |P(x, y) - G(x, y)|, \tag{9}$$

where P and G are the probability map of saliency detection and the corresponding ground truth, respectively, and (H, W) and (x, y) are the (height, width) and the pixel coordinates. A lower MAE value signifies a high degree of similarity between the ground truth and the predicted saliency map.

3) *Structure Measure:* S-measure (Sm) assesses the structural similarity between the predicted saliency map

Table I
Comparison of the Proposed Method and Four Other Methods on DUT-OMRON, DUTS-TE, and ECSSD Datasets, Using $MAE(\downarrow)$, F-Measure $F_\beta(\uparrow)$, Structure Measure $S_m(\uparrow)$, and E-Measure $E_m(\uparrow)$ as Evaluation Metrics

Dataset	DUT-OMRON				DUTS-TE				ECSSD			
Metrics	MAE	F_β	S_m	E_m	MAE	F_β	S_m	E_m	MAE	F_β	S_m	E_m
DCL	0.132	0.734	0.758	0.763	0.174	0.771	0.776	0.764	0.078	0.910	0.883	0.888
RFCN	0.110	0.742	0.764	0.778	0.09	0.784	0.794	0.839	0.107	0.890	0.852	0.876
UCF	0.132	0.734	0.758	0.763	0.117	0.771	0.776	0.764	0.078	0.910	0.883	0.888
Amulet	0.098	**0.743**	**0.780**	0.784	0.085	0.778	0.802	0.797	**0.059**	**0.912**	**0.893**	**0.911**
SalFAU-Net	**0.080**	0.722	0.755	**0.811**	**0.061**	**0.794**	**0.809**	**0.856**	0.063	0.895	0.862	0.892

The best values are highlighted in bold.

and the binary ground truth. It quantifies how well the predicted salient regions align with the actual salient regions in terms of structure, which are essential for real-world applications. It is defined as the weighted sum of region-aware *Sr* and object-aware *So* structural similarity:

$$S = (1-\alpha)Sr + \alpha So. \tag{10}$$

Typically, α is set to 0.5.

4) Enhanced Alignment Measure: Enhanced alignment measure (Em) incorporates both local pixel matching information and image-level statics by combining local pixel values and the image-level mean or global average value in a single term, which allows for a more comprehensive evaluation of detection algorithms, ensuring that the accuracy and quality of the salient regions are effectively measured.

$$Q_{FM} = \frac{1}{h \times w} \sum_{x=1}^{h} \sum_{y=1}^{w} \phi_{FM}(x, y), \tag{11}$$

where h and w are the height and width of the saliency map, respectively. ϕ_{FM} is enhanced alignment matrix, reflecting the correlation between P and G after subtracting their global means, respectively.

C. Implementation Details

The proposed network is implemented using the PyTorch framework, and training and testing are performed on an NVIDIA GeForce RTX 4070Ti GPU with 12 GB of video memory. The training dataset consists of 10,553 images from the DUTS-TR subset of DUTS [38]. To augment the dataset, each image is horizontally flipped, resulting in a doubled training set with 21,106 images. Prior to feeding the images into the network, they are resized to 320 × 320 and then cropped to 288 x 288 during training. Model optimization employs the Adam optimizer with default hyperparameter values ($lr = 1e-3, betas = (0.9, 0.999), eps = 1e-8, weight\_decay = 0$). The network is trained for approximately 500 000 iterations with a batch size of 12 to ensure convergence of the loss. While testing, the input images are first resized to 320 × 320 before being inputted into the trained network. The resulting predicted saliency map is then restored to its original dimensions through bilinear interpolation.

D. Comparison With Other Methods

In this section, we evaluate the effectiveness of the proposed model through both qualitative and quantitative analysis. We perform experiments to compare its performance with that of other models, utilizing four evaluation metrics, namely, MAE, *F*-measure, *S*-measure, and *E*-measure. We compare the results of the proposed method with some FCN-based methods, including Amulet [46], DCL [19], RFCN [39], and UCF [47].

1) Quantitative Comparison: The quantitative results on the six evaluation datasets using the four evaluation metrics are reported in Tables I and II. Based

Table II
Comparison of the Proposed Method and Four Other Methods on HKU-IS, PASCAL-S, and SOD Datasets, Using $MAE(\downarrow)$, F-Measure $F_\beta(\uparrow)$, Structure Measure $S_m(\uparrow)$, and E-measure $E_m(\uparrow)$ as Evaluation Metrics

Dataset	HKU-IS				PASCAL-S				SOD			
Metrics	MAE	F_β	S_m	E_m	MAE	F_β	S_m	E_m	MAE	F_β	S_m	E_m
DCL	0.074	0.886	0.866	0.891	0.126	0.824	0.803	0.785	0.164	0.798	0.754	0.755
RFCN	0.089	0.893	0.859	0.906	0.132	0.824	0.798	0.807	0.169	0.797	0.732	0.778
UCF	0.074	0.886	0.866	0.891	0.126	0.824	0.803	0.785	0.164	0.798	0.754	0.755
Amulet	0.052	0.895	0.882	0.912	0.098	**0.833**	**0.819**	0.827	0.0141	**0.802**	**0.759**	**0.791**
SalFAU-Net	**0.044**	**0.896**	**0.885**	**0.927**	**0.091**	0.814	0.801	**0.834**	**0.137**	0.798	0.718	0.759

The best values are highlighted in bold.

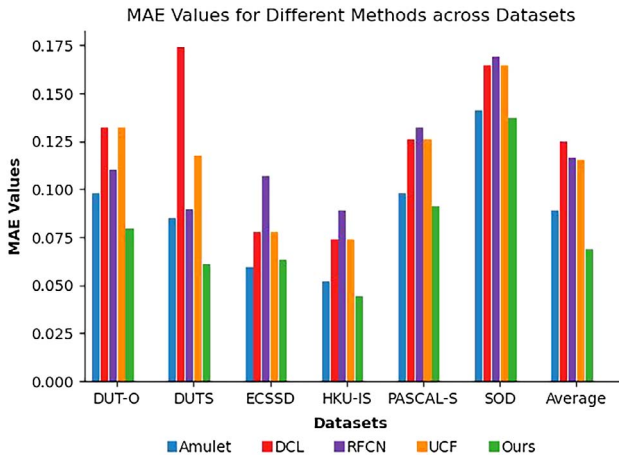


Figure 3. MAE values of Amulet, DCL, RFCN, UCF, and the proposed SalFAU-Net model across six evaluation datasets and the average MAE values of each method across all datasets. A lower MAE value signifies superior performance.

on the results, it is evident that our proposed method outperforms the benchmark methods on DUTS-TE and HKU-IS datasets in all evaluation metrics. On the DUT-OMRON dataset, our model achieves impressive results, outperforming other methods with the best MAE values. Furthermore, on the ECSSD dataset, our model demonstrates competitive performance with the second lowest MAE value of 0.063, surpassed only by Amulet with a slightly lower MAE of 0.059. Furthermore, we calculate the average MAE values for each method across the six datasets. Impressively, our proposed method achieves the lowest average MAE value of 0.068, indicating superior performance compared to other methods. Figure 3 presents the results of the average MAE values of each method, which clearly demonstrate that our model outperforms the comparison algorithms based on the average MAE value.

2) Qualitative Comparison: In addition to quantitative evaluations, we present predicted saliency maps generated by the proposed method and the comparison methods in Fig. 4. The images in the first and second columns of Fig. 4 represent the original input images and their corresponding ground truth saliency maps, respectively. The third column showcases the predicted saliency maps of our proposed method, while the fourth, fifth, sixth, and seventh columns exhibit the results of the comparison methods. The first two rows depict scenarios with multiple salient objects; the third row showcases a single large salient object; the fourth row contains small objects; and the fifth and sixth rows depict images with both small and large salient objects. The last row features relatively low-contrast salient objects. As we can see from Fig. 4, the results demonstrate that SalFAU-Net generates saliency maps more accurately for different challenging scenes, while the comparison methods generate incomplete or noisy saliency maps.

From the qualitative and quantitative results presented above, it is evident that our proposed method yields competitive results in tackling the challenge of visual saliency detection. These findings also highlight the crucial role of attention mechanisms in enhancing the effectiveness of the visual saliency models, as the proposed model places significant emphasis on extracting highly representative features while effectively eliminating unwanted or noisy features. This emphasis on attention mechanisms not only contributes to the competitiveness of our approach but also enhances the overall performance by prioritizing the extraction of relevant or salient visual information.

E. Failure Cases

The proposed method demonstrated effective SOD in most cases. However, there are some cases where it exhibits limitations. Figure 5 shows some failure cases for the proposed method. In the first column of Fig. 5, the presence of the shadow of the person is erroneously detected as a salient object. This is because the presence of shadows in salient objects may cause a decrease in the visibility or distinguishability of salient objects, making them harder to detect accurately. In the second column, the reflection of the duck is identified as a salient object, which is caused by reflections, which can create distracting image regions, potentially diverting attention away from the true salient objects. The third and fourth columns depict situations where the salient objects have low contrast, causing difficulty for our model in accurate saliency detection. In the last column, although most part of the airplane is detected, the model fails to capture its entirety.

In general, these images are very challenging for most deep learning models to detect accurately. These challenges arise due to the sensitivity of deep learning models to factors such as shadows, reflections, and low contrasts in salient objects. In the future, we will carry out further research aiming to address these problems and develop more accurate saliency detection models.

V. CONCLUSION

In this paper, we proposed SalFAU-Net as an approach for visual saliency detection tasks. Our method integrates an SFM into each decoder block of the Attention U-Net model, which enables efficient generation of saliency maps. The use of an AG module in our method facilitates selective focus on informative regions and suppression of nonsalient regions within an image. The comprehensive evaluation across six diverse SOD datasets, both quantitatively and qualitatively, underscores the effectiveness of our proposed method compared to the benchmark methods. SalFAU-Net not only showcases competitive performance but also highlights the potential of attention-based models in advancing the capabilities of saliency detection models.

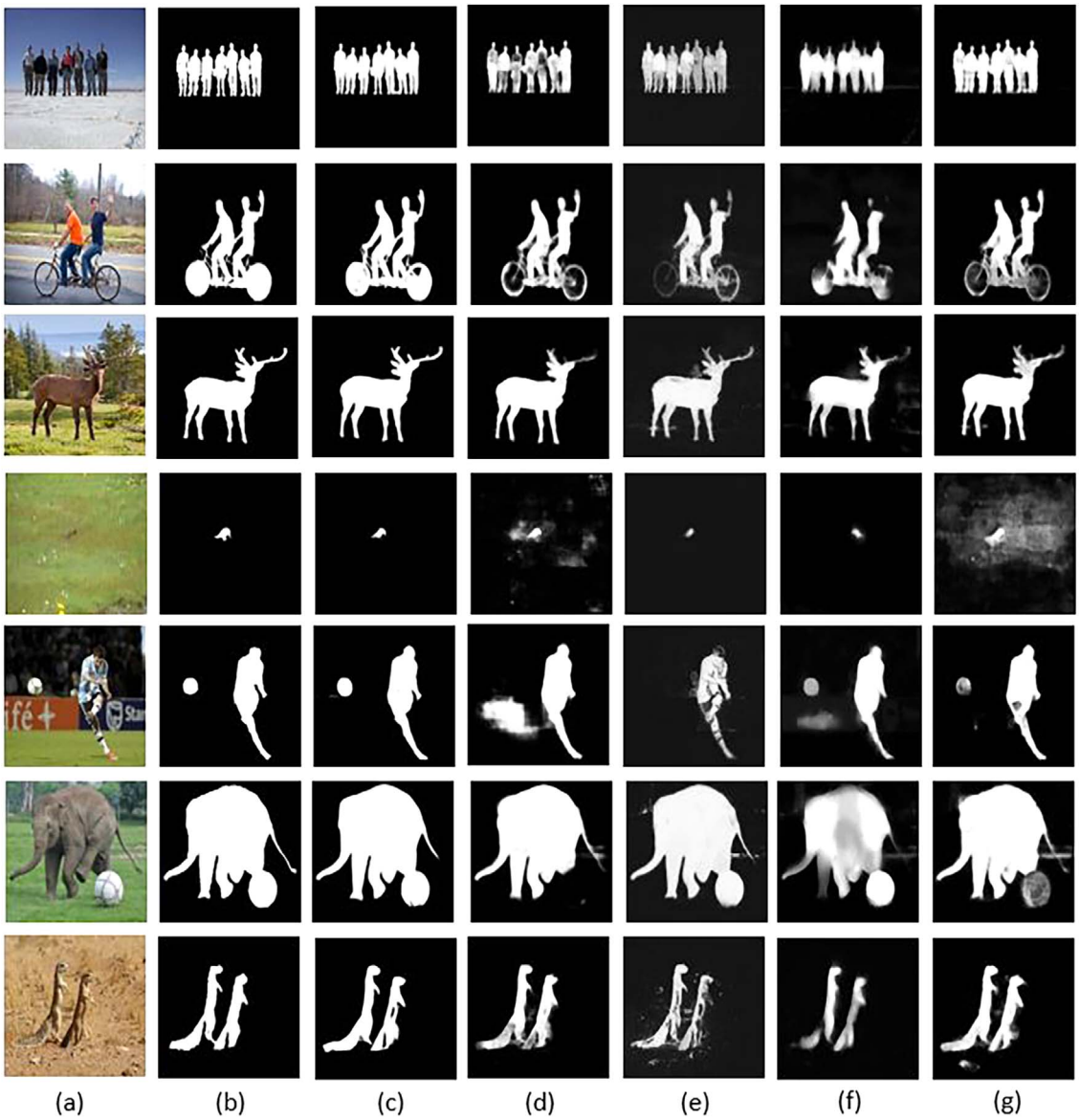


Figure 4. Qualitative comparison of the proposed method with four other SOTA methods: (a) original image, (b) GT, (c) Ours, (d) Amulet, (e) DCL, (f) RFCN, and (g) UCF.

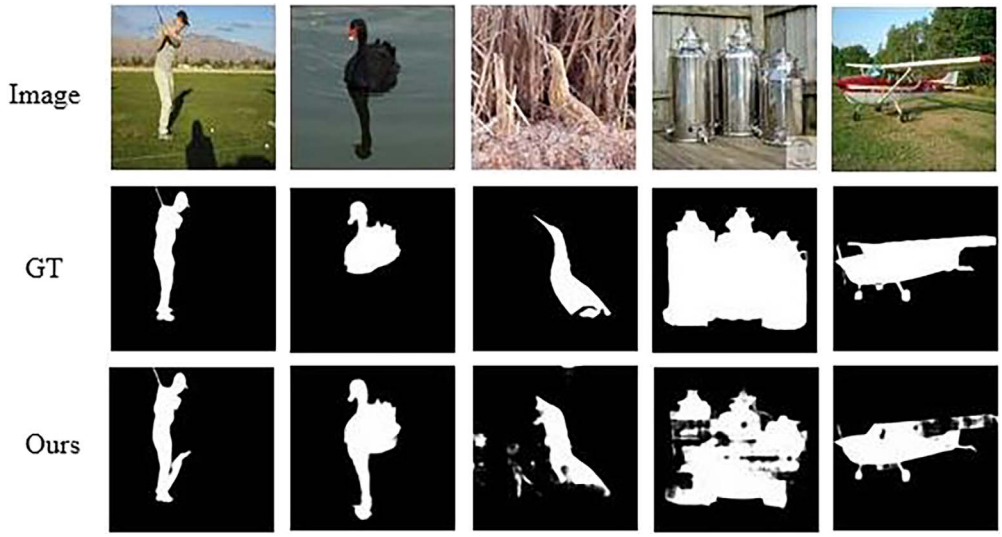


Figure 5. Failure cases of the proposed method.

REFERENCES

[1] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk
"Salient region detection and segmentation,"
in *Proc. 6th Int. Conf. Comput. Vis. Syst.*, Santorini, Greece, 2008, pp. 66–75.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk
"Frequency-tuned salient region detection,"
in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1597–1604.

[3] A. Borji
"What is a salient object? a dataset and a baseline model for salient object detection,"
IEEE Trans. Image Process., vol. 24, no. 2, pp. 742–756, Feb. 2015.

[4] A. Borji and L. Itti
"Scene classification with a sparse set of salient regions,"
in *Proc. IEEE Int. Conf. Roboti. Automat.*, 2011, pp. 1902–1908. IEEE.

[5] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein
"A tutorial on the cross-entropy method,"
Ann. Operations Res., vol. 134, pp. 19–67, 2005.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and L. Fei-Fei
"Imagenet: A. large-scale hierarchical image database,"
in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2009, pp. 248–255.

[7] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof
"Saliency driven total variation segmentation,"
in *Proc. IEEE 12th Int. Conf. Comput. Vis.* 2009, pp. 817–824.

[8] H. Duan, W. Shen, X. Min, D. Tu, J. Li, and G. Zhai
"Saliency in augmented reality,"
in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 6549–6558.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman
"The pascal visual object classes (VOC) challenge,"
Int. J. Comput. Vis., vol. 88, pp. 303–338, 2010.

[10] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji
"Structure-measure: A. new way to evaluate foreground maps,"
in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4548–4557.

[11] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji
"Enhanced-alignment measure for binary foreground map evaluation," 2018, arXiv:1805.10421.

[12] S. H. Fu, D. Xu, and D. Lin
"Object-based multiple foreground segmentation in rgbd video,"
IEEE Trans. Image Process., vol. 26, no. 3, pp. 1418–1427, Mar. 2017.

[13] X. Gong, L. Qingge, Q. Liu, and P. Yang
"Improved U-net-like network for visual saliency detection based on pyramid feature attention,"
Wireless Communi. Mobile Comput., vol. 2022, no. 1, p. 1108462, 2022.

[14] L. Han, X. Li, and Y. Dong
"Convolutional edge constraint-based U-net for salient object detection,"
IEEE Access, vol. 7, pp. 48890–48900, 2019.

[15] X. Hu, C.-W. Fu, L. Zhu, T. Wang, and P.-A. Heng
"Sac-net: Spatial attenuation context for salient object detection,"
IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 3, pp. 1079–1090, 2020.

[16] Y. Ji, H. Zhang, Z. Zhang, and M. Liu
"Cnn-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances,"
Inf. Sci., vol. 546, pp. 835–857, 2021.

[17] X. Lei, H. Pan, and X. Huang
"A dilated cnn model for image classification,"
IEEE Access, vol. 7, pp. 124087–124095, 2019.

[18] G. Li and Y. Yu
"Visual saliency based on multiscale deep features,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2015, 2015, pp. 5455–5463 2015.

[19] G. Li and Y. Yu
Deep contrast learning for salient object detection.
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 478–487, 2016.

[20] H. Li, H. Lu, Zhe Lin, Xiaohui Shen, and Brian Price
"Inner and inter label propagation: Salient object detection in the wild,"
IEEE Trans. Image Process., vol. 24, no. 10, pp. 3176–3186, Oct. 2015.

[21] J. Li, Z. Pan, Q. Liu, and Z. Wang
"Stacked U-shape network with channel-wise attention for salient object detection"
IEEE Trans. Multimedia, vol. 23, pp. 1397–1409, 2020.

[22] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang
Saliency detection via dense and sparse reconstruction.
in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2976–2983.

[23] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille
"The secrets of salient object segmentation,"
in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2014, pp. 280–287.

[24] X. Liu, L. Song, S. Liu, and Y. Zhang
Sustainability, vol. 13 no. 3, 2021, Art. no. 1224.

[25] J. Long, E. Shelhamer, and T. Darrell
"Fully convolutional networks for semantic segmentation,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik
"A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,"
in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, 2001, pp. 416–423.

[27] V. Movahedi and J. H. Elder
"Design and perceptual validation of performance measures for salient object segmentation,"
in *Proc. IEEE Comput. Soci. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 49–56.

[28] O. Oktay, J. Schlemper, L.L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz et al.
"Attention U-net: Learning where to look for the pancreas,"
2018, arXiv:1804.03999.

[29] A. Pal, S. Mondal, and H. I. Christensen
"Looking at the right stuff"-guided semantic-gaze for autonomous driving.
in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11883–11892.

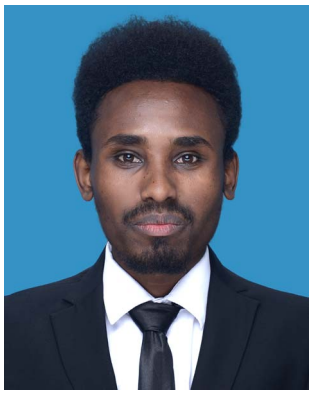
[30] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung
"Saliency filters: Contrast based filtering for salient region detection,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 733–740.

[31] C. Qin, G. Zhang, Y. Zhou, W. Tao, and Zhiguo Cao
"Integration of the saliency-based seed extraction and random walks for image segmentation,"
Neurocomputing, vol. 129, pp. 378–391, May 2014.

[32] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand
"U2-net: Going deeper with nested U-structure for salient object detection,"
Pattern Recognit., vol. 106, 2020, Art. no. 107404.

[33] Z. Ren, S. Gao, L.-T. Chia, and I.W.-H. Tsang
"Region-based saliency detection and its application in object recognition,"
IEEE Trans. Circ. Syst. Video Tech, vol. 24, no. 5, pp. 769–779,

May 2014.
[34] O. Ronneberger, P. Fischer, and T. Brox
U-net: Convolutional networks for biomedical image segmentation.
in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, Munich, Germany, 2015, pp. 234–241.
[35] U. Rutishauser, D. Walther, C. Koch, and P. Perona
"Is bottom-up attention useful for object recognition?,"
in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. II–II.
[36] J. Shi, Q. Yan, L. Xu, and J. Jia
"Hierarchical image saliency detection on extended CSSD,"
IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 4, pp. 717–729, Apr. 2016.
[37] I. Ullah, M. Jian, S. Hussain, J. Guo, H. Yu, X. Wang, and Y. Yin
"A brief survey of visual saliency detection,"
Multimed. Tools Appl., vol. 79, pp. 34605–34645, 2020.
[38] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan
"Learning to detect salient objects with image-level supervision,"
in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2017, pp. 3796–3805.
[39] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan
"Saliency detection with recurrent fully convolutional networks,"
in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 825–841. Springer, 2016.
[40] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba
"Sun database: Large-scale scene recognition from abbey to zoo,"
in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.
[41] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han
"Oriented r-cnn for object detection,"
in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3520–3529.
[42] Y. Xie, H. Lu, and M.-Hsuan Yang
"Bayesian saliency via low and mid level cues,"
IEEE Trans. Image Process., vol. 22, no. 5, pp. 1689–1698, May 2013.
[43] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio
"Show, attend and tell: Neural image caption generation with visual attention,"
in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015 pp. 2048–2057.
[44] Q. Yan, L. Xu, J. Shi, and J. Jia
"Hierarchical saliency detection,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1155–1162.
[45] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang
"Saliency detection via graph-based manifold ranking,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3166–3173.
[46] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan
"Amulet: Aggregating multi-level convolutional features for salient object detection,"
in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 202–211.
[47] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin
"Learning uncertain convolutional features for accurate saliency detection",
in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 212–221.
[48] Q. Zhang, Y. Shi, and X. Zhang
"Attention and boundary guided salient object detection,"
Pattern Recognit., vol. 107, pp. 107484, 2020.
[49] T. Zhao and X. Wu
"Pyramid feature attention network for saliency detection,"
in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3085–3094.



Kassaw Abraham Mulat received his bachelor's degree in software engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2021. He is currently pursuing his master's degree at the School of Computer Science, China West Normal University, Nanchong, China. His research interests include computer vision, pattern recognition, salient object detection, and medical image processing.

Zhengyong Feng received the B.S. degree in Physics from China West Normal University, Nanchong, China, in 2001, the M.S. degree in Communication and Information Systems from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2004, and the Ph.D. degree in Communication and Information Systems from University of Electronic Science and Technology of China, Chengdu, China, in 2013. From 2010 to 2011, he was a visiting PhD student at the School of Computer Science and Engineering, University at Buffalo, the State University of New York, USA. From 2004 to 2007, he was a Research Assistant at the School of Physics and Electronic Information, China West Normal University. Since 2017, he has been a Professor at the School of Electronic Information Engineering, China West Normal University. He is the author of more than 80 articles. His research interests include machine vision with deep learning, optimal control with MPC (Model Predictive Control), and RL (Reinforcement Learning).

Tegegne Solomon Eshetie received his bachelor's degree in Electronic Information Engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2021. He is currently pursuing his master's degree at the School of Computer Science, China West Normal University, Nanchong, China. His research interests include computer vision, medical image processing machine learning, and deep learning.

Ahmed Endris Hasen received his Bachelor's and Master's degree in software engineering from the University of Electronic Science and Technology of China (UESTC) in 2021 and 2023, respectively. Currently, he is a PhD Student at Paul C. Lauterbur Research Center, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. His research interests include Deep Learning, Computer vision, Image processing, and medical imaging.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: https://www.isif.org

2024 BOARD OF DIRECTORS*

2022–2024	2023–2025	2024–2026
Felix Govaers	Gustaf Hendeby	Paolo Costa
Lyudmila Mihaylova	Wolfgang Koch	Manon Kok
Paul Thomas	Claire Laudy	Terry Ogle

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Uwe Hanebeck, 2023	Darin Dunham, 2014	W. Dale Blair, 2005
Simon Maskell, 2022	Wolfgang Koch, 2013	Chee Chong, 2004
Simon Maskell, 2021	Roy Streit, 2012	Xiao-Rong Li, 2003
Paulo Costa, 2020	Joachim Biermann, 2011	Yaakov Bar-Shalom, 2002
Paulo Costa, 2019	Stefano Coraluppi, 2010	Pramod Varshney, 2001
Lyudmila Mihaylova, 2018	Elisa Shahbazian, 2009	Yaakov Bar-Shalom, 2000
Lyudmila Mihaylova, 2017	Darko Musicki, 2008	Jim Llinas, 1999
Jean Dezert, 2016	Erik Blasch, 2007	
Darin Dunham, 2015	Pierre Valin, 2006	

SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

SOCIETY MISSION

Advocate

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes– look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors are encouraged to submit their manuscripts for peer review https://isif.org/journal.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at https://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 19 Number 2 December 2024 ISSN 1557-6418