



DEPARTMENT OF ENGINEERING CYBERNETICS

PROJECT REPORT

Towards accurate multitarget tracking

Author:

Erling HJERMSTAD

Supervisor:

Edmund F. BREKKE

Co-supervisor:

Lars-Christan N. TOKLE

December 2023

Abstract

Incorporating prior information of an area can aid in the performance of target tracking. This project derives two different formulations of the VIMMJIPDA tracker of [1] supporting spatially varying birth and clutter intensities. The first model makes approximations to avoid the need for numerical integrations. The second does not. To make the second model efficient in simulations a precomputation scheme is developed and presented.

Based on data from a radar surveying the Ravnkloa canal, the intensities are estimated. Using the estimates, the models are evaluated over a selected subset of the entire data set from the canal. The approximation model is shown to achieve comparable results to the second model, without the computational overhead. Both models outperform the baseline VIMMJIPDA dramatically on a number of criteria, and achieve comparable scores on the other.

Table of Contents

List of Figures	iv
List of Tables	vi
1 Introduction	1
2 Notation	2
3 Target tracking theory	3
3.1 Bayesian estimation	3
3.2 Kalman Filter	4
3.3 Target Tracking	5
3.4 Finite set statistics	5
3.5 Poisson Point Process and Bernoulli process	6
3.6 Probability generating functionals	8
3.7 Poisson Multi-Bernoulli Mixture Filter	10
3.8 Limitations	13
3.9 Interacting Multiple Models	15
3.10 VIMMJIPDA	17
4 Method	21
4.1 Data	21
4.2 Preprocessing	21
4.3 Filter setup	24
4.4 Implementation	25
4.5 Estimating clutter and birth intensity	31
4.6 Custom intensities	31
4.7 Discretization of precompute grid	34
5 Model analysis	36
5.1 Results	37
5.2 Case 1 - Effect of amended marginal probability	38
5.3 Case 2 - Initial position adjustment	38
5.4 Case 3 - Kinematic models	40
5.5 Case 4 - Initial existence probability of the Integral and Approximate models	41
5.6 Case 5 - Early detection	43

5.7	Discussion	44
6	Conclusion	46
	Appendix	47
A	All scenarios used for evaluation	47
	Bibliography	52

List of Figures

1	An example of development of hypotheses after 2 scans of 2 measurements.	12
2	The position of the radar at Fosenkaia.	21
3	All measurements in the dataset. The circle has a radius of 150 m. The Fosenkaia frame of the aFerry ROS implementation is shown. A green dot is placed at the radar position in the Fosenkaia frame; $(0, -8)$	22
4	Approximate area monitored by the radar. The needle is placed at the radar position. The area measures approximately 0.02km^2	22
5	Illustration of a radar spoke. This spoke has 3 cells for which the echo intensity is measured. Since the cell size increases with range, the radar is less sensitive at distance.	23
6	A single scan from the radar shown in RViz. The radar is at the blue dot. Due to a bug in RViz, the background map is not shown correctly in the lower right corner.	23
7	Distinct targets can be fused together to a single measurement in the preprocessing. This degrades the performance of the tracker.	23
8	Extracting points from a radar scan.	24
9	An example of measurements extracted from a radar recording. The measurements in the fjord north of the canal are bug points since there is a railway blocking the radar view. These are ignored in analysis.	25
10	The transform described by Equation 60. The samples are rotated by 45° with respect to the principal axes for better numerical properties.[8] Due to symmetry this is unproblematic.	28
11	Adjusting the initial position distribution of a target towards a high birth-intensity area.	29
12	The effect of approximating the new target state density as Gaussian. Gray is a high-intensity area, and white is a low-intensity area.	30
13	Computing the weighted average of the 4 closest precomputed values, with inverse distance weights. X marks the measurement location.	30
14	Bar graphs of the birth and clutter intensities extracted from the data.	32
15	The high- and low-intensity areas for clutter and birth found by thresholding Figure 14. Red squares show high-intensity grid cells. The range of the radar is shown with a circle.	33
16	The clutter and birth zones used going forward in this project.	33
17	A constructed scenario where the overlap of the calculation points of the precompute grid and the birth intensity grid cause problems for the averaging approximation of the exact solution. Intensity values are exaggerated in this example.	34
18	Comparison between performance of the VIMMJIPDA tracker without (left) and with (right) spatially varying intensities describing the area.	36
19	The two main sources of error tracks in the spatial clutter and birth aware models.	37
20	The clutter areas at $(20, -100)$ and $(-70, -90)$ affect the clutter aware model (left) and the normal VIMMJIPDA (right) differently due to how the marginal association probabilities are calculated.	38

21	The effect of the calculation of the initial state distribution is prone to errors when U is not exact. Here a target spawns in the low-intensity birth zone and is erroneously corrected towards the high-intensity area. The intensity difference of the high- and low-intensity zones are exaggerated in this example.	39
22	The Integral model correctly adjusts the track beginning towards a high birth zone in this example. The difference of the track in the Approximate model is nearly undetectable.	39
23	An example of a track where all models of the IMM are used.	40
24	The IMM makes a difference in some scenarios.	41
25	When the tracks are less dense, the extra existence probability can be decisive. . .	41
26	Scatter plot of the birth intensity integral values, and the smoothed birth intensity. As expected, the points on the border between high- and low-intensity differ from the birth intensity. Points where the integral and approximate equal the low intensity are not shown.	42
27	Comparisons of when the trackers detect targets.	43
28	The tracking of the approximate model is good. The dark green track in the lower left corner is a false track caused by very noisy measurements in this area. The turquoise track was initialized late and should have been continued instead of a new blue track being created.	44
29	All scenarios used for evaluation of the trackers. Newer measurements are darker..	51

List of Tables

1	Overview of notation	2
2	The standard model for multi-target tracking.	5
3	Assumptions in VIMMJIPDA. All non-contradictive assumptions of Table 2 are assumed.	17
4	The amendments done to the VIMMJIPDA assumptions in this project.	18
5	Parameter estimates found in [7].	25
6	An overview of the parameters used in the tracker.	26
7	The values used for the intensities in the analysis of the tracker.	34
8	The Frobenius norm of the difference between the exact and averaged covariance matrix with different measures to combat discrepancies. A lower value is better. . .	35
9	Evaluation criteria	36
10	Evaluation results of the different trackers over the evaluation set.	37
11	Euclidean norm of the error in the smoothing approach, compared to the numerical solution of the integral $\mathbb{I}(\mathbf{z})$	42

1 Introduction

It is reasonable to believe that driving in an unknown environment increases the risk of accidents compared to driving in a known environment. Locals know where to expect traffic, poor conditions, pedestrian crossings, etc. Some research supports this, but the area has not received much attention.[2], [3] The same logic can be extended to maritime traffic, although it is arguably to a lesser degree due to less traffic. Urban environments, such as harbors and canals, offer more possible dangers. Autonomous Surface Vehicles (ASVs) use tracking algorithms to survey their working area to detect and monitor boats, canoes, or other movable or stationary obstacles. Using this information, collision avoidance algorithms can calculate optimal actions. One of the main challenges in target tracking is establishing new targets to track in the presence of false measurements. By learning the usual behavior of targets and clutter in an environment, ASVs can possibly gain some of the intuition of "locals" and use this to make more informed decisions.

This project investigates the impact of monitoring an area for some time to learn the usual traffic patterns and clutter areas for robustness in tracking. The effect of prior information on track initiation has been investigated in [4] and was found to improve performance slightly. This is investigated again in this project, together with the effect of prior information on the remainder of the tracking process.

First, notation and relevant theory are presented. This is followed by a review of the method used. Then results and conclusion follow.

2 Notation

The notation in this project largely follows that of [1]. Vectors are in bold. Bold and capitalized letters indicate matrices or sets.

$\mathcal{N}(x; \mu, \sigma^2)$	x is normal distributed with mean μ and variance σ^2 .
$(\cdot)_{k k-1}$	Prediction at timestep k , given measurements up to timestep $k-1$.
$(\cdot)_k$	Quantity at timestep k , given measurements up to time step k .
$(\cdot)_k^j$	Quantity at timestep k dependent on measurement j .
$(\cdot)_k^s$	Quantity at timestep k dependent on model s .
m_k	The number of measurements received at timestep k .
\mathbf{z}_k^j	Measurement j received at timestep k .
$\mathbf{Z}_k = \{\mathbf{z}_k^1, \dots, \mathbf{z}_k^{m_k}\}$	The set of measurements received at timestep k .
$\mathbf{Z}_{1:k} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_k\}$	The set of sets of measurements received at time steps 1 through k .
\mathbf{x}_k^t	State of target t at timestep k .
$\mathbf{X}_k = \{\mathbf{x}_k^1, \dots, \mathbf{x}_k^n\}$	The (possibly empty) set of targets.
\mathbf{F}	Kinematic transition matrix.
\mathbf{Q}	Kinematic transition uncertainty covariance.
\mathbf{H}	Measurement matrix.
\mathbf{H}^*	Inverse measurement matrix, extracts all states \mathbf{H} does not.
\mathbf{R}	Measurement uncertainty covariance.
\mathbf{P}	State uncertainty covariance.

Table 1: Overview of notation

In this report, the notation for random variables will sometimes be confused with their realizations. E.g $\mathbf{x} \sim f(\mathbf{x})$ might be written for the random variable X with probability distribution $f(\mathbf{x})$. This is for notational simplicity, as the other approach would quickly become overwhelmingly verbose, and to avoid confusion between the set-valued random variables \mathbf{X} and regular random variables \mathbf{x} .

To describe functionals the notation $G[h]$ will be used frequently. Here, G is a mapping from function space to the reals. This notation is also used for inner products on function-space; $f[h] = \int f(\mathbf{x})h(\mathbf{x})d\mathbf{x}$. In the case of the ambiguity cased by multivariate functions, the variable to marginalize over will be denoted by (\cdot) , e.g. $f_{k|k-1}[f_{\mathbf{z}}(\mathbf{z}_k^j|\cdot)P_D] = \int f_{k|k-1}(\mathbf{x})f_{\mathbf{z}}(\mathbf{z}_k^j|\mathbf{x})P_D(\mathbf{x})d\mathbf{x}$.

3 Target tracking theory

In this section, a review of the underlying theory in target tracking is presented. Some of the underlying assumptions are motivated, and the equations of the filter used in this work are derived.

3.1 Bayesian estimation

Bayes' rule gives us a way of updating our beliefs when new information is discovered. It mixes our previous beliefs, the *prior*, with the likelihood of the new information, to give the correct new understanding, the *posterior*. The rule is often written on the proportional form without the denominator, since it is easier to work with, \mathbf{z} is known, and the correct probabilities can be retrieved by normalization.

$$\underbrace{p(\mathbf{x}|\mathbf{z})}_{\text{posterior}} = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \propto \underbrace{p(\mathbf{z}|\mathbf{x})}_{\text{likelihood}} \underbrace{p(\mathbf{x})}_{\text{prior}} \quad (1)$$

In the Bayesian filter framework, this approach is taken whenever new information is received, i.e. measurements. This gives an iterative recursive approach to the filtering problem. A few assumptions underlie this way of thinking, called the Markov assumptions.

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{k-1}, \mathbf{z}_1, \dots, \mathbf{z}_{k-1}) &= p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ p(\mathbf{z}_k|\mathbf{x}_1, \dots, \mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{z}_1, \dots, \mathbf{z}_{k-1}) &= p(\mathbf{z}_k|\mathbf{x}_k) \end{aligned} \quad (2)$$

The first equation in Equation 2 expresses the assumption that all information from all past iterations is captured by the previous state estimate. That means that \mathbf{x}_k is independent of all previous measurements and states when \mathbf{x}_{k-1} is given. The second assumption says that the current state estimate sums up all information about the likelihood of the current measurements. Therefore, \mathbf{z}_k is independent of all previous measurements and states given \mathbf{x}_k .

Using Bayes' rule and the Markov assumptions, the Bayesian filter can be derived.

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \quad (3)$$

Equation 3 is the update step of the filter. To complete the iterative process, the form of $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ must be generated from the posterior in the previous timestep. Here, the Markov assumptions come in, with the help of the total probability theorem.

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) &= \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \end{aligned} \quad (4)$$

The first equality follows from the total probability theorem, the second from the definition of conditional probability, and the third from the first Markov assumption. Equation 4 is known as the prediction step of the filter, or the Chapman-Kolmogorov equation. Intuitively the equation makes sense. $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ encodes our understanding of how the system evolves from one state to another between timesteps. The probability of getting to state \mathbf{x}_k will then be the probability of being in state \mathbf{x}_{k-1} and it evolving to \mathbf{x}_k for all states \mathbf{x}_{k-1} .

Together, the prediction and update steps give the Bayes filter. Given the posterior at timestep $k - 1$, we can predict the state at timestep k . Using this prediction as our prior, the update equation fuses it with new measurements to give the posterior at timestep k . The posterior is then predicted on in the next timestep. To start the process, an initial estimate $p(\mathbf{x}_0)$ is required.

In the event that no measurements arrive, the prior is the best guess for the posterior and is fed through to the prediction step at the next timestep. It can be argued that no measurements also

carry information about the target. However, assuming a constant probability of detection in the surveillance region, we get $p(\mathbf{x}_k | \mathbf{z}_{1:k}) \propto (1 - P_D)p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) \propto p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$, where $1 - P_D$ is the constant probability of misdetection. The case of nonuniformity is investigated later, as is the case of multiple measurements in each timestep and multiple targets. In a multisensor version of the Bayes filter, the update step can be repeated for each sensor.

The forms $p(\mathbf{z}_k | \mathbf{x}_k)$ and $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ deserve extra attention. The first is the likelihood of a measurement. This is called the measurement model and is a consequence of the sensors and software chosen to use for measuring. The second is the kinematic prediction. This follows from the system dynamics. If both of these are known, this leaves $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ to be found. However, generally there does not exist a closed-form solution.

3.2 Kalman Filter

Under Gaussian linear assumptions the Bayes filter has a closed form solution. The resulting filter is known as the Kalman filter. In the Kalman filter, the kinematic model, sensor model and initial conditions are all Gaussian.

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{x}_{k-1}) &= \mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}_{k-1}, \mathbf{Q}) \\ p(\mathbf{z}_k | \mathbf{x}_k) &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}\mathbf{x}_k, \mathbf{R}) \\ p(\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_0, \mathbf{P}_0) \end{aligned} \quad (5)$$

\mathbf{Q} , \mathbf{R} , \mathbf{P}_0 are the covariance of the noise in the kinematic model, measurement model and initial conditions respectively. Equation 5 can equivalently be written on state space form.

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k, & \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{Q}) \\ \mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{w}_k, & \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{R}) \\ \mathbf{x}_0 &\sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0) \end{aligned} \quad (6)$$

Under these assumptions, the Kalman filter can be derived. The proof relies on the product identity of Gaussian variables.

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) &= \int p(\mathbf{x}_k | \mathbf{x}_{k-1})p(\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \\ &= \int \mathcal{N}(\mathbf{x}_k; \mathbf{F}\mathbf{x}_{k-1}, \mathbf{Q})\mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})d\mathbf{x}_{k-1} \\ &= \mathcal{N}(\mathbf{x}_k; \mathbf{F}\hat{\mathbf{x}}_{k-1}, \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^\top + \mathbf{Q}) \\ &\quad \times \int \mathcal{N}(\mathbf{x}_{k-1}; \text{some vector, some matrix})d\mathbf{x}_{k-1} \\ &= \mathcal{N}(\mathbf{x}_k; \mathbf{F}\hat{\mathbf{x}}_{k-1}, \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^\top + \mathbf{Q}) \end{aligned} \quad (7)$$

From Equation 7 the prediction step equations can be extracted. They are summarized in Equation 8.

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}\hat{\mathbf{x}}_{k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^\top + \mathbf{Q} \end{aligned} \quad (8)$$

The proof of the update step is another exercise in using the product identity.

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_{1:k}) &\propto p(\mathbf{z}_k | \mathbf{x}_k)p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) \\ &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}\mathbf{x}_k, \mathbf{R})\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k) \\ &\propto \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k) \end{aligned} \quad (9)$$

where the values of $\hat{\mathbf{x}}_k$ and \mathbf{P}_k are listed in Equation 10.

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_{1:k}) &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{W}_k \mathbf{H}) \mathbf{P}_{k|k-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{W}_k \boldsymbol{\nu}_k \end{aligned} \quad (10)$$

where

$$\begin{aligned}\boldsymbol{\nu}_k &= \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R} \\ \mathbf{W}_k &= \mathbf{P}_{k|k-1}\mathbf{H}^\top\mathbf{S}_k^{-1}\end{aligned}$$

\mathbf{W}_k is known as the Kalman gain. It is the optimal gain that yields the minimum mean square error estimate of the state $\hat{\mathbf{x}}$.

3.3 Target Tracking

Target tracking is the task of supervising a set of targets and estimating their states. There are a few complications when comparing target tracking to ownship state estimation. For one, the kinematic models of the targets are unknown. This includes control signals and dynamics. Moreover, there are usually multiple measurements that must be filtered and assigned to the present targets. This is a difficult task in the presence of false measurements, missed detections, unknown targets, clutter, track coalescence, and target extent. First when a measurement and a model are assigned to a target, can the Kalman filter equations be applied. When dealing with multitarget tracking, some simplifying assumptions are usually made. The set of assumptions is known as the standard model for multitarget tracking. [5]

- M1 New targets are born according to a Poisson process with intensity $b(\mathbf{x})$.
- M2 Existing targets survive from timestep $k-1$ to k with probability $P_S(\mathbf{x}_{k-1})$.
- M3 The motion of a surviving target is given by $f_{\mathbf{x}}(\mathbf{x}_k|\mathbf{x}_{k-1})$.
- M4 A target with state \mathbf{x}_k generates a single measurement \mathbf{z}_k with probability $P_D(\mathbf{x}_k)$.
- M5 Clutter measurements occur according to a Poisson process with intensity $\lambda(\mathbf{z})$.
- M6 The measurement of a detected target is related to its state according to $f_{\mathbf{z}}(\mathbf{z}_k|\mathbf{x}_k)$.
- M7 The Markov assumptions of Equation 2 apply.
- M8 A target generates at most one measurement.
- M9 A measurement comes from at most one target.

Table 2: The standard model for multi-target tracking.

Often, these assumptions are amended to a small degree in a tracking algorithm. In particular, both P_S and P_D are often assumed to be constant in the surveillance region. $f_{\mathbf{x}}(\mathbf{x}_k|\mathbf{x}_{k-1})$ in assumption M3 corresponds to $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and is the notation that will be used going forward in this report. This also applies to $f_{\mathbf{z}}(\mathbf{z}_k|\mathbf{x}_k)$ from M6 and $p(\mathbf{z}_k|\mathbf{x}_k)$.

3.4 Finite set statistics

From M1 and M2 it is clear that the number of targets present at any timestep varies. When both the cardinality and the state of random variables must be modeled, Random Finite Sets (RFS) is a convenient mathematical tool of Finite Set Statistics (FISST). An RFS is a set-valued random variable; this means that both the size of the set and the state of the members of the set are random. A sample of an RFS consists of a set of realizations of random variables. The applications for processes where the number of entities varies, like multi-target tracking, are obvious. Using FISST we can define equations like the Bayesian filter of Equation 4 and Equation 3 for the case where both the states and the measurements are sets of unknown cardinality.

In the RFS-framework, there exist multitarget densities analogous to probability densities in probability theory. They can be constructed from a cardinality conditioned joint distribution and a cardinality distribution. As sets are unordered by nature, care needs to be taken. To eliminate the effect of the order of the state variables, the joint-probability distribution must be summed over all $n!$ possible orderings of the \mathbf{x}_i 's. Let Π be the set of the $n!$ lists of unique index orders $\Pi = \{\pi_1, \dots, \pi_{n!}\}$. Then the RFS density of the random variable Ξ with density f_Ξ , cardinality

distribution $p(n)$ and cardinality conditioned state distribution f_n is given by

$$\begin{aligned}\mathbf{X} &= \{\mathbf{x}_1, \dots, \mathbf{x}_n\}, \quad n \sim p(n), \quad \mathbf{x}_1, \dots, \mathbf{x}_n \sim f_n(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ f_{\Xi}(\mathbf{X}) &= p(n) \sum_{\pi \in \Pi} f_n(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(n)}).\end{aligned}\tag{11}$$

RFS comes with its own calculus. The set density integral is defined as in Equation 12. This is a natural consequence of the definition in Equation 11, where the integral accumulates the density of the n \mathbf{x} 's in D^n for all n in N .

$$\int_{N,D} f_{\Xi}(\mathbf{X}) \delta \mathbf{X} = \sum_{n \in N} \frac{1}{n!} \int_{D^n} f_{\Xi}(\mathbf{X}) d\mathbf{x}_1 \dots d\mathbf{x}_n\tag{12}$$

where the special case of the empty set $f_{\Xi}(\emptyset) = p(0)$ is implied when $n = 0$. It is easy to show that the RFS density integrates to 1 over the entire space of n and \mathbf{x} .

$$\begin{aligned}\int_{N,D} f_{\Xi}(\mathbf{X}) \delta \mathbf{X} &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{D^n} f_{\Xi}(\mathbf{X}) d\mathbf{x}_1 \dots d\mathbf{x}_n \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{D^n} p(n) \sum_{\pi \in \Pi} f_n(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(n)}) d\mathbf{x}_1 \dots d\mathbf{x}_n \\ &= \sum_{n=0}^{\infty} \frac{p(n)}{n!} \sum_{\pi \in \Pi} \int_{D^n} f_n(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(n)}) d\mathbf{x}_1 \dots d\mathbf{x}_n \\ &= \sum_{n=0}^{\infty} \frac{p(n)}{n!} \sum_{\pi \in \Pi} 1 \\ &= \sum_{n=0}^{\infty} p(n) = 1\end{aligned}\tag{13}$$

Having a notion of integral in FISST, we can formulate the Bayesian estimation problem of Equation 4 and Equation 3 in this context.

$$\begin{aligned}f_{k|k-1}(\mathbf{X}_k) &= \int f_X(\mathbf{X}_k | \mathbf{X}_{k-1}) f_{k-1}(\mathbf{X}_{k-1}) \delta \mathbf{X}_{k-1} \\ f_k(\mathbf{X}_k) &\propto f_Z(\mathbf{Z}_k | \mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k)\end{aligned}\tag{14}$$

where the notation $f_{k|k-1}$ means the density is implicitly conditioned on all measurements up to and including $k - 1$, and f_k is including timestep k . E.g. $f_{k|k-1}(\mathbf{X}_k) = f(\mathbf{X}_k | \mathbf{Z}_{1:k-1})$. This is a multi-target formulation of Equation 3 and Equation 4, where both the number of targets and measurements can vary. However, the distributions of the measurement and kinematic models are now a bit more mysterious. Given the assumptions of Table 2, there is only a need for two RFS-densities to fill the gaps. The Poisson Point Process (ppp) spawns targets and clutter, while the Bernoulli process monitors the existence and state of known targets.

3.5 Poisson Point Process and Bernoulli process

The filter will be shown to consist of a mix of Bernoulli and Poisson processes. A Bernoulli trial models an event with only two possible outcomes. In the filter, it will be used to model the existence of targets. For this, a Bernoulli process seems fitting. The Poisson process warrants some more reflection. Besides its many useful properties, there are multiple ways to motivate its pertinence. Two will be presented here.

A Poisson can be viewed as the limit of n Bernoulli trials with rate r such that $\lim_{n \rightarrow \infty} nr = \lambda$. The binomial distribution models the number of successes of n Bernoulli trials. Let M be the binomial distributed random variable that models the number of successes. Using the probability

generating function of M , the binomial theorem, a definition of the exponential function, and letting $\lim_{n \rightarrow \infty} nr = \lambda$ we get the following.

$$\begin{aligned}
\lim_{n \rightarrow \infty} G_M(t) &= \lim_{n \rightarrow \infty} \mathbb{E}[t^M] \\
&= \lim_{n \rightarrow \infty} \sum_{m=0}^n \binom{n}{m} (tr)^m (1-r)^{n-m} \\
&= \lim_{n \rightarrow \infty} (1-r+rt)^n \\
&= \lim_{n \rightarrow \infty} (1 + \underbrace{nr}_{\lambda} (t-1)/n)^n \\
&= e^{\lambda(t-1)}
\end{aligned} \tag{15}$$

This is the probability generating function of the Poisson distribution. Therefore, a Poisson process can be seen as a large number of Bernoulli trials with a tiny chance of success. In the context of clutter measurements, we have a great number of resolution cells on our sensor and a small probability of false alarms. Therefore, the number of false measurements should be approximately Poisson distributed.

The next argument pertains to the birth process being Poisson. For convenience, it is normal to assume that the targets arrive independently of each other and without regard for when the previous target arrived. Usually, this is a very good approximation. There is only one continuous distribution which has this property of memorylessness; the exponential distribution. Let T be the exponentially distributed random variable that models the time between the arrival of two targets.

$$\begin{aligned}
p_T(t) &= \lambda e^{-\lambda t}, & t \geq 0 \\
p_{T|T \geq t_0}(t|t_0) &= \frac{Pr(T \geq t_0|T=t)p_T(t)}{Pr(T \geq t_0)} \\
&= \begin{cases} \frac{1 \times p_T(t)}{1 - P_T(t_0)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t_0}} = \lambda e^{-\lambda(t-t_0)} & t \geq t_0 \\ \frac{0 \times p_T(t)}{1 - P_T(t_0)} = 0 & t \leq t_0 \end{cases} \\
&= p_T(t - t_0)
\end{aligned}$$

This proves the memorylessness. Given the exponential distribution for inter-arrival time, the distribution for the number of targets to arrive up to time t : $p_N(n) = Pr(T_{n+1} > t \cap T_n \leq t)$ is found to be Poisson with intensity λt .

$$\begin{aligned}
p_N(n) &= Pr(T_{n+1} > t \cap T_n \leq t) \\
&= \int_t^\infty \int_0^t p_{T_{n+1}, T_n}(t_{n+1}, t_n) dt_n dt_{n+1} \\
&= \int_0^t \int_t^\infty p_{T_{n+1}|T_{n+1} > t_n}(t_{n+1}|t_n) dt_{n+1} p_{T_n}(t_n) dt_n \\
&= \int_0^t \int_t^\infty p_{T_{n+1}}(t_{n+1} - t_n) dt_{n+1} p_{\sum_i \Delta T_i}(t_n) dt_n \\
&= \int_0^t e^{-\lambda(t-t_n)} \int_0^{t_n} \cdots \int_0^{t_2} \prod_{i=1}^n \lambda e^{-\lambda(t_i-t_{i-1})} dt_1 \cdots dt_n \\
&= \int_0^t e^{-\lambda(t-t_n)} \lambda^n e^{-\lambda t_n} \frac{(t_n)^{n-1}}{(n-1)!} dt_n \\
&= e^{-\lambda t} \frac{(\lambda t)^n}{n!}
\end{aligned}$$

Therefore, the number of targets born between two timesteps should be distributed according to a Poisson distribution. These results can be transferred to the analogous Poisson and Bernoulli distributions in the FISST framework.

In a ppp the state distributions are independent and identically distributed, thus the cardinality conditioned joint distribution can be factorized as the product of the individual distributions $\sum_{\pi \in \Pi} f_n(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(n)}) = \sum_{\pi \in \Pi} \prod_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) = n! \prod_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$. With this in mind, a ppp is easily shown to have the following RFS density;

$$\begin{aligned} f_{\Xi}(\mathbf{X}) &= p(n) \sum_{\pi \in \Pi} f_n(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(n)}) \\ &= e^{-\Lambda} \frac{\Lambda^n}{n!} \sum_{\pi \in \Pi} \prod_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \\ &= e^{-\Lambda} \Lambda^n \prod_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}). \end{aligned} \quad (18)$$

A ppp can be written as a function of the intensity $\lambda(\mathbf{x})$. The intensity is the expected number of objects per area, such that the expected value in an area S is given by $\int_S \lambda(\mathbf{x}) d\mathbf{x}$.

$$f_{\Xi}(\mathbf{X}) = e^{-\int_S \lambda(\mathbf{x}) d\mathbf{x}} \prod_{\mathbf{x} \in \mathbf{X}} \lambda(\mathbf{x}) \quad (19)$$

The formulations of Equation 18 and Equation 19 are equivalent for a homogeneous ppp where $\lambda(\mathbf{x}) = \Lambda f(\mathbf{x})$, since $\int \lambda(\mathbf{x}) d\mathbf{x} = \Lambda$. For the non-homogeneous case, the expected value per area varies over the surveillance area and therefore there exists no equivalent formulation without the integral.

The cardinality distribution can be retrieved from the set density by marginalizing over the state variables. Again, some care must be taken due to the unordered nature of sets. We eliminate this effect by noting the independence between states and, therefore, dividing by the number of permutations $n!$.

$$\begin{aligned} p(n) &= \frac{1}{n!} \int_{D^n} f_{\Xi}(\mathbf{X}) d\mathbf{x}_1 \dots d\mathbf{x}_n \\ &= \frac{1}{n!} \int_{D^n} e^{-\Lambda} \Lambda^n \prod_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{n!} e^{-\Lambda} \Lambda^n \prod_{\mathbf{x} \in \mathbf{X}} \int_D f(\mathbf{x}) d\mathbf{x} \\ &= \frac{\Lambda^n}{n!} e^{-\Lambda} \end{aligned} \quad (20)$$

where D is the domain of \mathbf{x} . The last equality comes from the fact that a probability density integrated over its entire domain is 1. We recognize Equation 20 as the Poisson distribution.

The Bernoulli process is the other building block of multi-target tracking in FISST. A Bernoulli process models existence.

$$f_{\Xi}(\mathbf{X}) = \begin{cases} 1 - r, & \mathbf{X} = \emptyset \\ r \cdot f(\mathbf{x}), & \mathbf{X} = \{\mathbf{x}\} \\ 0, & \text{else} \end{cases} \quad (21)$$

In the Bernoulli process given by Equation 21, r is the probability of existence, and f is the existence-conditioned probability density function. The Bernoulli process handles a single target. Combining many independent Bernoulli processes, we get the multi-Bernoulli process. The RFS-density of the set $\mathbf{Y} = \bigcup_{i=1}^N \mathbf{X}_i$ is complicated, however, Probability Generating Functionals (pgfl's) gives us a handy way of representing it.

3.6 Probability generating functionals

Like probability generating functions are a practical tool in statistics, probability generating functionals are a practical tool in FISST. A functional takes a function as input and returns a number. Their application is convenience in the same way that Laplace transforms are convenient. The Laplace transform is an alternative representation of a signal. Notably, the Laplace transform

turns convolutions into products. This is also the case for the pgfl. The probability density function of a sum of random variables is the convolution of the individual probability density functions. The union of random finite sets is analogous to the sum of random variables. The pgfl of a union of random finite sets is a product of the individual probability generating functionals. Computing the product is much easier than computing the convolution. The definition of the pgfl is given in Equation 22.

$$G_{\Xi}[h] = \mathbb{E}_{\Xi}[h^{\mathbf{X}}] = \int h^{\mathbf{X}} f_{\Xi}(\mathbf{X}) \delta \mathbf{X} \quad (22)$$

where $h^{\mathbf{X}} = \prod_{\mathbf{x} \in \mathbf{X}} h(\mathbf{x})$ with the convention $h^{\emptyset} = 1$, and \mathbb{E} is the usual expectation operator with the obvious changes needed to fit with FISST. h is known as the test-function. The proof that the pgfl of a union of independent sets is the product of the pgfl of each set is straightforward.

Let Υ be the random finite set variable describing the set $\mathbf{Y} = \bigcup_{i=1}^N \mathbf{X}_i$, and Ξ_i describe the individual subsets.

$$G_{\Upsilon}[h] = \mathbb{E}_{\Upsilon}[h^{\mathbf{Y}}] = \mathbb{E}_{\Upsilon}[h^{\bigcup_{i=1}^N \mathbf{X}_i}] = \mathbb{E}_{\Upsilon}\left[\prod_{i=1}^N h^{\mathbf{X}_i}\right] = \prod_{i=1}^N \mathbb{E}_{\Xi_i}[h^{\mathbf{X}_i}] = \prod_{i=1}^N G_{\Xi_i}[h] \quad (23)$$

Using the definition from Equation 22 the pgfl of a Poisson and Bernoulli process can be found. For ppp, pgfl is given by

$$\begin{aligned} G^{\text{PPP}}[h] &= \int h^{\mathbf{X}} e^{-\int \lambda(\mathbf{x}) d\mathbf{x}} \prod_{\mathbf{x} \in \mathbf{X}} \lambda(\mathbf{x}) \delta \mathbf{X} \\ &= e^{-\int \lambda(\mathbf{x}) d\mathbf{x}} \int \prod_{\mathbf{x} \in \mathbf{X}} h(\mathbf{x}) \lambda(\mathbf{x}) \delta \mathbf{X} \\ &= e^{-\int \lambda(\mathbf{x}) d\mathbf{x}} \sum_{n=0}^{\infty} \frac{1}{n!} \prod_{\mathbf{x} \in \mathbf{X}} \int h(\mathbf{x}) \lambda(\mathbf{x}) d\mathbf{x} \\ &= e^{-\int \lambda(\mathbf{x}) d\mathbf{x}} \sum_{n=0}^{\infty} \frac{1}{n!} \left(\int h(\mathbf{x}) \lambda(\mathbf{x}) d\mathbf{x} \right)^n \\ &= e^{-\int \lambda(\mathbf{x}) d\mathbf{x}} e^{\int \lambda(\mathbf{x}) h(\mathbf{x}) d\mathbf{x}} \\ &= e^{\int \lambda(\mathbf{x})(h(\mathbf{x}) - 1) d\mathbf{x}} \\ &= e^{\lambda[h-1]}. \end{aligned} \quad (24)$$

Here, the Taylor expansion of the exponential function is used along with the i.i.d. properties of the \mathbf{x} s. With this notation, the resemblance to the probability generating function of the Poisson distribution in Equation 15 is striking. The pgfl of the Bernoulli process is more straightforward.

$$\begin{aligned} G^b[h] &= \int h^{\mathbf{X}} f_{\Xi}(\mathbf{X}) \delta \mathbf{X} \\ &= f_{\Xi}(\emptyset) + \int h(\mathbf{x}) r f(\mathbf{x}) d\mathbf{x} \\ &= 1 - r + r f[h] \end{aligned} \quad (25)$$

In the derivation the RFS-density from Equation 21 is used together with the definition of the set integral defined in Equation 12. Using Equation 23 and Equation 25 we get the pgfl of a multi-Bernoulli consisting of N independent Bernoulli sets.

$$G^{\text{mb}}[h] = \prod_{i=1}^N (1 - r^i + r^i f^i[h]) \quad (26)$$

In the presence of multiple targets and multiple measurements, we can assign the measurements to different targets in an amount of ways. One such hypothesis will lead to one Multi-Bernoulli. A Multi-Bernoulli mixture (mbm) is a linear combination of the hypotheses.

$$G^{\text{mbm}}[h] = \sum_{j=1}^M \prod_{i=1}^N w^{ij} (1 - r^{ij} + r^{ij} f^{ij}[h]) \quad (27)$$

The weights of the mixture w^{ij} are used to reflect the belief we have in each hypothesis. We will refer to r^{ij} as the hypothesis conditioned probability of existence, as it will be used to describe the probability of existence of target i under hypothesis j . $f^{ij}[h]$ is the pgfl of the existence conditioned probability distribution of the state of target i under hypothesis j .

3.7 Poisson Multi-Bernoulli Mixture Filter

Having defined the pgfl, the Poisson Multi-Bernoulli Mixture filter (PMBM) can be derived. The proof is based on taking the pgfl of the prediction and the update step from Equation 14. The pgfl of the prediction step are as follows.

$$\begin{aligned} G_{k|k-1}[h] &= \int h^{\mathbf{X}} f_{k|k-1}(\mathbf{X}) \delta \mathbf{X} \\ &= \int h^{\mathbf{X}} \int f_X(\mathbf{X}|\mathbf{X}') f_{k-1}(\mathbf{X}') \delta \mathbf{X}' \delta \mathbf{X} \\ &= \iint h^{\mathbf{X}} f_X(\mathbf{X}|\mathbf{X}') \delta \mathbf{X} f_{k-1}(\mathbf{X}') \delta \mathbf{X}' \\ &= \int G_X[h|\mathbf{X}'] f_{k-1}(\mathbf{X}') \delta \mathbf{X}' \end{aligned} \quad (28)$$

G_X is the pgfl of the kinematic set prediction function. It is described by assumptions 1 to 3 of Table 2. The resulting set is a union of a new set resulting from a ppp and multiple Bernoulli components. The ppp has an intensity given by $b(\mathbf{x})$, as described by the first assumption. The second assumption discusses the probability that a target survives from one timestep to the next. This is a Bernoulli process with an existence probability of $P_S(\mathbf{x})$. Finally, the third assumption says how the state develops from one timestep to the next. This is the existence-conditioned probability density of the Bernoulli component and is given by the kinematic prediction function $f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}')$.

$$\begin{aligned} G_X[h|\mathbf{X}'] &= e^{b[h-1]} \prod_{\mathbf{x}' \in \mathbf{X}'} (1 - P_S(\mathbf{x}') + P_S(\mathbf{x}') f_{\mathbf{x}}[h](\mathbf{x}')) \\ &= e^{b[h-1]} (1 - P_S + P_S f_{\mathbf{x}}[h])^{\mathbf{X}'} \end{aligned} \quad (29)$$

where $f_{\mathbf{x}}[h](\mathbf{x}') = \int h(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') d\mathbf{x}$ is the probability generating functional of the kinematic prediction function as a function of the state. For ease of notation, $f_{\mathbf{x}}$ is used both for the functional and for the actual function, but they should be discernible from context. The set exponent notation defined previously is also used. Inserting Equation 29 into Equation 28, we obtain the pgfl of the prediction step.

$$\begin{aligned} G_{k|k-1}[h] &= \int G_X[h|\mathbf{X}] f_{k-1}(\mathbf{X}) \delta \mathbf{X} \\ &= e^{b[h-1]} \int (1 - P_S + P_S f_{\mathbf{x}})^{\mathbf{X}} f_{k-1}(\mathbf{X}) \delta \mathbf{X} \\ &= e^{b[h-1]} G_{k-1}[1 - P_S + P_S f_{\mathbf{x}}] \end{aligned} \quad (30)$$

Assuming the posterior of the previous timestep to be a union of Bernoulli and Poisson components $G_{k-1}[h] = G_{k-1}^{\text{PPP}}[h] G_{k-1}^{\text{mbbm}}[h]$, the pgfl of the prediction is shown to keep this property.

$$\begin{aligned} G_{k|k-1}[h] &= e^{b[h-1]} G_{k-1}[1 - P_S + P_S f_{\mathbf{x}}] \\ &= \underbrace{e^{b[h-1]} G_{k-1}^{\text{PPP}}[1 - P_S + P_S f_{\mathbf{x}}]}_{G_{k|k-1}^{\text{PPP}}[h]} \underbrace{G_{k-1}^{\text{mbbm}}[1 - P_S + P_S f_{\mathbf{x}}]}_{G_{k|k-1}^{\text{mbbm}}[h]} \end{aligned} \quad (31)$$

Calling the intensity of the previous posterior's ppp-component for the unknown target intensity

u_{k-1} , the predicted ppp-component can be found as follows.

$$\begin{aligned}
G_{k|k-1}^{\text{PPP}}[h] &= \exp(b[h-1]) G_{k-1}^{\text{PPP}}[1 - P_S + P_S f_{\mathbf{x}}[h]] \\
&= \exp(b[h-1]) \exp(u_{k-1}[1 - P_S + P_S f_{\mathbf{x}}[h] - 1]) \\
&= \exp(b[h] + u_{k-1}[P_S f_{\mathbf{x}}[h]] - b[1] - u_{k-1}[P_S]) \\
&= \exp(b[h] + u_{k-1}[P_S f_{\mathbf{x}}[h]] - (b[1] + u_{k-1}[P_S f_{\mathbf{x}}[1]])) \\
&= \exp(\underbrace{\int h(\mathbf{x})(b(\mathbf{x}) + \int f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') P_S(\mathbf{x}') u_{k-1}(\mathbf{x}') d\mathbf{x}') d\mathbf{x}}_{u_{k|k-1}(\mathbf{x})} \\
&\quad - \underbrace{\int 1(\mathbf{x})(b(\mathbf{x}) + \int f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') P_S(\mathbf{x}') u_{k-1}(\mathbf{x}') d\mathbf{x}') d\mathbf{x}}_{u_{k|k-1}(\mathbf{x})}) \\
&= \exp(u_{k|k-1}[h-1])
\end{aligned} \tag{32}$$

The fourth equality follows from the fact that $f_{\mathbf{x}}[1] = \int 1(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') d\mathbf{x}' = 1$ regardless of \mathbf{x}' , where $1(\mathbf{x})$ is the function that is 1 everywhere. As can be seen in Equation 32, $G_{k|k-1}^{\text{PPP}}$ is still Poisson. The same can be proved for the mbm-component by breaking it down into the individual Bernoullis. The following proves that they are still Bernoullis after the prediction step.

$$\begin{aligned}
G_{k|k-1}^{\text{mbm}}[h] &= G_{k-1}^{\text{mbm}}[1 - P_S + P_S f_{\mathbf{x}}] \\
&= \sum_j \prod_t w^{t,j} G_{k-1}^{t,j}[1 - P_S + P_S f_{\mathbf{x}}] \\
&= \sum_j \prod_t w^{t,j} G_{k|k-1}^{t,j}[h] \\
G_{k|k-1}^{t,j}[h] &= G_{k-1}^{t,j}[1 - P_S + P_S f_{\mathbf{x}}] \\
&= 1 - r_{k-1}^{t,j} + r_{k-1}^{t,j} f_{k-1}^{t,j}[1 - P_S + P_S f_{\mathbf{x}}] \\
&= 1 - r_{k-1}^{t,j} + r_{k-1}^{t,j} \int f_{k-1}^{t,j}(\mathbf{x}) (1 - P_S(\mathbf{x}) + P_S(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x})) d\mathbf{x} \\
&= 1 - r_{k-1}^{t,j} \int P_S(\mathbf{x}) f_{k-1}^{t,j}(\mathbf{x}) d\mathbf{x} \\
&\quad + r_{k-1}^{t,j} \int f_{k-1}^{t,j}(\mathbf{x}) P_S(\mathbf{x}) \int h(\mathbf{x}') f_{\mathbf{x}}(\mathbf{x}'|\mathbf{x}) d\mathbf{x}' d\mathbf{x} \\
&= 1 - \underbrace{r_{k-1}^{t,j} f_{k-1}^{t,j}[P_S]}_{r_{k|k-1}^{t,j}} + \underbrace{r_{k-1}^{t,j} f_{k-1}^{t,j}[P_S]}_{\int h(\mathbf{x}') \frac{\int f_{k-1}^{t,j}(\mathbf{x}) P_S(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'}{f_{k-1}^{t,j}[P_S]} d\mathbf{x}'} \\
&= 1 - r_{k|k-1}^{t,j} + r_{k|k-1}^{t,j} \int h(\mathbf{x}') f_{k|k-1}^{t,j}(\mathbf{x}') d\mathbf{x}' \\
&= 1 - r_{k|k-1}^{t,j} + r_{k|k-1}^{t,j} f_{k|k-1}^{t,j}[h]
\end{aligned} \tag{33}$$

We recognize Equation 33 as a Bernoulli, so $G_{k|k-1}^{\text{mbm}}$ preserves the multi-Bernoulli mixture form. Therefore, the combined prediction step preserves the ppp-mbm form of the previous posterior. The results found above are summarized in Equation 34.

$$u_{k|k-1}(\mathbf{x}) = b(\mathbf{x}) + \int f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') P_S(\mathbf{x}') u_{k-1}(\mathbf{x}') d\mathbf{x}' \tag{34a}$$

$$r_{k|k-1}^{t,j} = r_{k-1}^{t,j} f_{k-1}^{t,j}[P_S] \tag{34b}$$

$$f_{k|k-1}^{t,j}(\mathbf{x}) = \frac{\int f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') f_{k-1}^{t,j}(\mathbf{x}') P_S(\mathbf{x}') d\mathbf{x}'}{f_{k-1}^{t,j}[P_S]} \tag{34c}$$

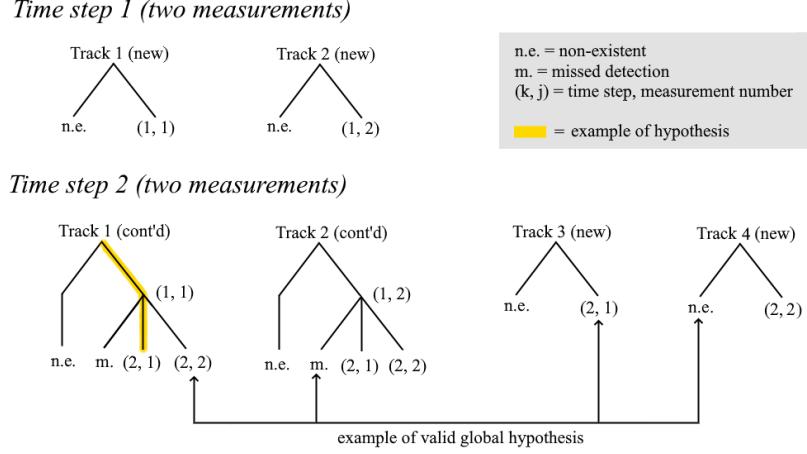


Figure 1: An example of development of hypotheses after 2 scans of 2 measurements.

The other quantities, like the number of Bernoulli components and the hypothesis weights remain the same. The rules stated in Equation 34 make intuitive sense. The predicted unknown target intensity at a point x is the sum of the birth intensity at this point and the probability that unknown targets survive and progress to this point. The new existence probability is the previous multiplied by the probability of survival of the kinematic distribution. The predicted kinematic distribution is the probability of a target surviving from all previous states to a new state x .

There are two meaningful ways that a measurement can occur. A measurement can be created by one of the targets known by the filter, or a measurement can come from an unknown target or clutter. Also, a target might not be detected. For each measurement, all of these hypotheses are considered in the filter update step. This motivates the need for three different sets of update equations. When a new set of measurements arrives, a new track is created for each measurement in the set, since it is possible that the measurement originates from an unknown target. A track is a collection of all possible hypotheses originating from the same measurement. In Figure 1 a track is one tree. Hypotheses are a collection of measurements and misdetections describing a possible path of a target. In the figure, one hypothesis is a path to a leaf node in the track trees. A global hypothesis is a collection of one hypothesis from each track in such a way that no two hypotheses share a measurement. When a new track is made it has two hypotheses, one from the measurement and one for the case that the track does not actually exist. The non-existence is a special hypothesis for each track, which does not require any attention as its weight $w^{t,n.e.} = 1$ and hypothesis conditioned existence probability $r^{t,n.e.} = 0$. This means that it will never have any mathematical influence, except for allowing the track to be excluded in a valid global hypothesis. Figure 1 illustrates the growth of hypotheses and tracks in the update step. The figure explains well how each hypothesis branches into $1+m_k$ at each iteration when m_k is the number of measurements registered. The 1 corresponds to the missed detection case where none of the measurements are hypothesized to come from the target.

As in the prediction step, the equations for the update step can be derived from the pgfl of the update step in Equation 14, and the demand that it preserves the form of the prior. The proof for the update step is longer and more involved, and is skipped here. It can be found in the original paper [6]. Since the update step is shown to preserve the ppp-mbm form, the PMBM is proved by induction. The update equations take different forms depending on the type of hypothesis on which they are conditioned. The posterior unknown target intensity is the same for all.

$$u_k(\mathbf{x}) = (1 - P_D(\mathbf{x}))u_{k|k-1}(\mathbf{x}) \quad (35)$$

Let a_k^t be the current hypothesis number and a_{k-1}^t the hypothesis number for the node above in the tree. The next hypothesis number is given by $a_k^t + 1$, i.e. it increments per hypothesis. In the

case of the missed detection hypothesis, the Bernoulli is updated according to the following.

$$w_k^{t,a_k^t} = w_{k-1}^{t,a_{k-1}^t} (1 - r_{k|k-1}^{t,a_{k-1}^t} + r_{k|k-1}^{t,a_{k-1}^t} f_{k|k-1}^{t,a_{k-1}^t} [1 - P_D]) \quad (36a)$$

$$r_k^{t,a_k^t} = \frac{r_{k|k-1}^{t,a_{k-1}^t} f_{k|k-1}^{t,a_{k-1}^t} [1 - P_D]}{1 - r_{k|k-1}^{t,a_{k-1}^t} + r_{k|k-1}^{t,a_{k-1}^t} f_{k|k-1}^{t,a_{k-1}^t} [1 - P_D]} \quad (36b)$$

$$f_k^{t,a_k^t}(\mathbf{x}) = \frac{(1 - P_D(\mathbf{x})) f_{k|k-1}^{t,a_{k-1}^t}(\mathbf{x})}{f_{k|k-1}^{t,a_{k-1}^t} [1 - P_D]} \quad (36c)$$

For each measurement, a new track is initiated. The track indexes for the new tracks will be $n_{k-1} + j$, where n_{k-1} is the number of tracks in the previous timestep. Since it is the first hypothesis of the track, the hypothesis number is 1. The initial values are as follows.

$$w_k^{n_{k-1}+j,1} = \lambda(\mathbf{z}_k^j) + u_{k|k-1}[f_{\mathbf{z}}(\mathbf{z}_k^j|\cdot)P_D] \quad (37a)$$

$$r_k^{n_{k-1}+j,1} = \frac{u_{k|k-1}[f_{\mathbf{z}}(\mathbf{z}_k^j|\cdot)P_D]}{\lambda(\mathbf{z}_k^j) + u_{k|k-1}[f_{\mathbf{z}}(\mathbf{z}_k^j|\cdot)P_D]} \quad (37b)$$

$$f_k^{n_{k-1}+j,1}(\mathbf{x}) = \frac{u_{k|k-1}(\mathbf{x}) f_{\mathbf{z}}(\mathbf{z}_k^j|\mathbf{x}) P_D(\mathbf{x})}{u_{k|k-1}[f_{\mathbf{z}}(\mathbf{z}_k^j|\cdot)P_D]} \quad (37c)$$

For each track-hypothesis-measurement combination, the track is updated with that new hypothesis. Let t be the track number, j be the index of the measurement of the current set to update based on, and a_{k-1}^t the index of the hypothesis to update. Then the new hypotheses are given by the following equations for all combinations t, a_{k-1}^t, j .

$$w_k^{t,a_k^t} = w_{k-1}^{t,a_{k-1}^t} r_{k|k-1}^{t,a_{k-1}^t} f_{k|k-1}^{t,a_{k-1}^t} [f_{\mathbf{z}}(\mathbf{z}_k^j|\cdot)P_D] \quad (38a)$$

$$r_k^{t,a_k^t} = 1 \quad (38b)$$

$$f_k^{t,a_k^t}(\mathbf{x}) = \frac{f_{\mathbf{z}}(\mathbf{z}_k^j|\mathbf{x}) P_D(\mathbf{x}) f_{k|k-1}^{t,a_{k-1}^t}(\mathbf{x})}{f_{k|k-1}^{t,a_{k-1}^t} [f_{\mathbf{z}}(\mathbf{z}_k^j|\cdot)P_D]} \quad (38c)$$

3.8 Limitations

For each iteration of the filter, the number of hypotheses increases according to $h_k = h_{k-1}(1 + m_k)$ and the number of tracks is $n_k = n_{k-1} + m_k$ where m_k is the number of measurements received at timestep k. Due to the number of hypotheses growing exponentially and the number of tracks growing linearly, the PMBM is computationally infeasible. Many techniques to mitigate the computation problem exist.

The perhaps most effective way to reduce computational cost is to end each iteration of the filter with approximating the multi-Bernoulli mixture by a single multi-Bernoulli. This means creating one hypothesis for each track and discarding the rest. Assuming this is done after each iteration, there will only be $(1 + m_k)n_k$ hypotheses before the mixture reduction to n_k , excluding the nonexistence hypotheses. We approximate the joint hypotheses probability as the product of the marginal probabilities. Assuming that this is done after each iteration, a_k^t will take the values from 0 to m_k , corresponding to the measurement that caused this hypothesis, and 0 being a miss-detection. The marginal probability of target t claiming measurement j , denoted $a_k^t = j$, is then the sum of the probabilities of all hypotheses $a_k \in \mathcal{A}_k$ such that $a_k^t = j$. Defining W_k to be the total weight of all hypotheses, $W_k = \sum_{a_k \in \mathcal{A}_k} \prod_{t=1}^{n_k} w_k^{t,a_k^t}$, and assuming spatially constant P_D , the

probability of a single hypotheses is given by Equation 39.

$$\begin{aligned}
\Pr(a_k) &= \frac{1}{W_k} \prod_{t=1}^{n_k} w_k^{t,a_k^t} \\
&= \frac{1}{W_k} \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t = 0} (1 - r_{k|k-1}^{t,0} P_D) \times \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t > 0} P_D r_{k|k-1}^{t,a_k^t} f_{k|k-1}^{t,a_k^t} [f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} \cdot)] \\
&\quad \times \prod_{j \in \{1, \dots, m_k\} | a^{n_{k-1}+j} = 1} (\lambda(\mathbf{z}_k^j) + P_D u_{k|k-1} [f_{\mathbf{z}}(\mathbf{z}_k^j \cdot)]) \\
&\propto \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t = 0} (1 - r_{k|k-1}^{t,0} P_D) \times \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t > 0} \frac{P_D r_{k|k-1}^{t,a_k^t} f_{k|k-1}^{t,a_k^t} [f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} \cdot)]}{\lambda(\mathbf{z}_k^{a_k^t}) + P_D u_{k|k-1} [f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} \cdot)]} \tag{39}
\end{aligned}$$

Equation 39 says the probability of a global hypothesis is proportional to the product of the weights of the individual hypotheses compromising the global hypothesis. Since all measurements must be taken into account in a valid hypothesis, we can divide by the constant $\prod_{j=1}^{m_k} \lambda(\mathbf{z}_k^j) + P_D u_{k|k-1} [f_{\mathbf{z}}(\mathbf{z}_k^j \cdot)]$. This is what is done in the last step of Equation 39, noting that the measurements are either claimed by the new track weights or the detection weights. Equation 39 describes a proportional relationship, but true probabilities can be retrieved by normalization. Using this, the mixture reduction can be found.

$$\begin{aligned}
G_k^{\text{mbm}}[h] &= \sum_{a_k \in \mathcal{A}_k} \frac{1}{W_k} \prod_{t=1}^{n_k} w_k^{t,a_k^t} (1 - r_k^{t,a_k^t} + r_k^{t,a_k^t} f_k^{t,a_k^t}[h]) \\
&\approx \prod_{t=1}^{n_k} \frac{1}{W_k} \sum_{j=0}^{m_k} \sum_{\substack{a_k \in \mathcal{A}_k \\ \text{s.t. } a_k^t = j}} \left(\prod_{t'=1}^{n_k} w_k^{t'a_k^{t'}} \right) (1 - r_k^{tj} + r_k^{tj} f_k^{tj}[h]) \\
&= \prod_{t=1}^{n_k} \sum_{j=0}^{m_k} \sum_{\substack{a_k \in \mathcal{A}_k \\ \text{s.t. } a_k^t = j}} \Pr(a_k) (1 - r_k^{tj} + r_k^{tj} f_k^{tj}[h]) \\
&= \prod_{t=1}^{n_k} \sum_{j=0}^{m_k} (1 - r_k^{tj} + r_k^{tj} f_k^{tj}[h]) \sum_{\substack{a_k \in \mathcal{A}_k \\ \text{s.t. } a_k^t = j}} \Pr(a_k) \tag{40}
\end{aligned}$$

Defining $p_k^{tj} = \sum_{\substack{a_k \in \mathcal{A}_k \\ \text{s.t. } a_k^t = j}} \Pr(a_k)$ and inserting into Equation 40 give the equations needed. Recall that $r_k^{tj} = 1 : \forall j > 0$.

$$\begin{aligned}
G_k^{\text{mbm}}[h] &\approx \prod_{t=1}^{n_k} \sum_{j=0}^{m_k} (1 - r_k^{tj} + r_k^{tj} f_k^{tj}[h]) p_k^{tj} \\
&= \prod_{t=1}^{n_k} \sum_{j=0}^{m_k} (p_k^{tj} - p_k^{tj} r_k^{tj} + p_k^{tj} r_k^{tj} f_k^{tj}[h]) \\
&= \prod_{t=1}^{n_k} \left(1 - \underbrace{1 - p_k^{t0} (1 - r_k^{t0})}_{r_k^t} + \sum_{j=0}^{m_k} p_k^{tj} r_k^{tj} f_k^{tj}[h] \right) \\
&= \prod_{t=1}^{n_k} \left(1 - r_k^t + r_k^t \underbrace{\sum_{j=0}^{m_k} \frac{1}{r_k^t} p_k^{tj} r_k^{tj} f_k^{tj}[h]}_{f_k^t[h]} \right) \\
&= \prod_{t=1}^{n_k} (1 - r_k^t + r_k^t f_k^t[h]) \tag{41}
\end{aligned}$$

The mixture reduction equations found above are summarized in Equation 42.

$$\Pr(a_k) \propto \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t = 0} (1 - r_{k|k-1}^{t,0} P_D) \times \prod_{\substack{t \in \{1, \dots, n_{k-1}\} \\ |a_k^t| > 0}} \frac{P_D r_{k|k-1}^{t,a_k^t} f_{k|k-1}^{t,a_k^t}[f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} | \cdot)]}{\lambda(\mathbf{z}_k^{a_k^t}) + P_D u_{k|k-1}[f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} | \cdot)]} \quad (42a)$$

$$p_k^{tj} = \sum_{\substack{a_k \in \mathcal{A}_k \\ \text{s.t. } a_k^t = j}} \Pr(a_k) \quad (42b)$$

$$r_k^t = 1 - p_k^{t0}(1 - r_k^{t0}) \quad (42c)$$

$$f_k^t = \sum_{j=0}^{m_k} \frac{1}{r_k^t} p_k^{tj} r_k^{tj} f_k^{tj} \quad (42d)$$

Assuming each f_k^{tj} to be Gaussian, f_k^t can be approximated as Gaussian using standard moment-matching and the weights $\beta_k^{tj} = \frac{1}{r_k^t} p_k^{tj} r_k^{tj}$. These are the same reduction equations used in the JIPDA tracker, although with a slight modification to $\Pr(a_k)$. In the original JIPDA, the unknown target influence is neglected in the calculation of the association probabilities.[1], [5], [6]

Mixture reduction reduces the number of hypotheses dramatically, but does nothing to combat the growing number of tracks. For this purpose, pruning is used. Pruning is the removal of hypotheses with an existence probability below some threshold, called the termination threshold. After mixture reduction, a single hypothesis constitutes each track. Pruning these means pruning entire tracks. Many tracks will never gather any significant existence probability mass before being pruned. In the context of making decisions based on the target tracking, we would like to ignore these tracks. For this reason, tracks are placed in different categories based on their existence probability. New tracks are put in the preliminary category. Tracks are confirmed and moved to the confirmed category when they gather an existence probability above a certain threshold. The confirmed tracks are the tracks that we care about. Confirmed tracks stay in the confirmed category until their existence probability reaches the termination threshold, at which point they are pruned. In many scenarios, most tracks never escape the preliminary category. The termination threshold and the confirmation threshold are tuning variables.

In combination with this, gating is often employed. A gate is an area around a prediction where measurements are considered. Measurements outside the gate are not included in the hypotheses for that prediction. When the prediction is normal distributed, the gate is usually chosen to be 3.5 standard deviations. This captures 99.95% of the probability mass of the measurement likelihood function and is therefore likely to only remove wrong hypotheses. Only tracks sharing measurements in their gates are processed together, effectively splitting the tracker into multiple trackers only knowing about their own cluster of tracks and measurements. This is called single-linkage clustering and is another way to speed up computations.

3.9 Interacting Multiple Models

The derivations of this subchapter follow [5] closely.

After all the previous derivations, there still is one problem that needs to be solved. The kinematic model of the targets is still unknown, and therefore the kinematic prediction function $f_{\mathbf{x}}(\mathbf{x} | \mathbf{x}')$ is unspecified. In tracking theory, there are two main models for this density. The targets are assumed to either have constant velocity (CV) or constant turn rate (CT) in combination with a high enough process noise for the filter to smooth over the models' inconsistencies. The models are simple in the continuous case, but are somewhat more intricate in the discrete case due to the propagation of the states and uncertainty between timesteps. The kinematic equations of a

discrete CV model are given by

$$\begin{aligned}\mathbf{x} &= [x, y, v_x, v_y]^\top \\ \mathbf{x}_k &= \mathbf{F}_{\text{cv}} \mathbf{x}_{k-1} + \mathbf{v}_k, \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\text{cv}}) \\ \mathbf{F}_{\text{cv}} &= \begin{bmatrix} \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \\ \mathbf{Q}_{\text{cv}} &= \begin{bmatrix} T^3/3\mathbf{I}_2 & T^2/2\mathbf{I}_2 \\ T^2/2\mathbf{I}_2 & T\mathbf{I}_2 \end{bmatrix} \sigma_a^2.\end{aligned}\tag{43}$$

This is equivalent to $f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; \mathbf{F}_{\text{cv}}\mathbf{x}', \mathbf{Q}_{\text{cv}})$. σ_a^2 is the one tuning variable capturing the uncertainty of the acceleration, enabling the model to allow some change in velocity and direction. A CT model is more complicated due to the non-linearities that stem from the variable turn rate.

$$\begin{aligned}\mathbf{x} &= [x, y, v_x, v_y, \omega]^\top \\ \mathbf{x}_k &= \mathbf{F}_{\text{ct}}(\mathbf{x}_{k-1}) \mathbf{x}_{k-1} + \mathbf{v}_k, \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\text{ct}}) \\ \mathbf{F}_{\text{ct}}(\mathbf{x}) &= \begin{bmatrix} 1 & 0 & \frac{\sin T\omega}{\omega} & \frac{-1+\cos T\omega}{\omega} & 0 \\ 0 & 1 & \frac{1-\cos T\omega}{\omega} & \frac{\sin T\omega}{\omega} & 0 \\ 0 & 0 & \cos T\omega & -\sin T\omega & 0 \\ 0 & 0 & \sin T\omega & \cos T\omega & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{Q}_{\text{ct}} &= \begin{bmatrix} \mathbf{Q}_{\text{cv}} & \mathbf{0} \\ \mathbf{0} & T\sigma_a^2 \end{bmatrix}\end{aligned}\tag{44}$$

The kinematic prediction function then takes the form $f_{\mathbf{x}}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; \mathbf{F}_{\text{ct}}(\mathbf{x}')\mathbf{x}', \mathbf{Q}_{\text{ct}})$, which must be linearized before it can be used in a Kalman filter. In a CT model, both σ_a^2 and σ_α^2 are tuning variables.

Interacting Multiple Models (IMM) is the idea of letting the filter have multiple kinematic models, and let the filter estimate which is most accurate for each target at any timestep. To do this, the filter must estimate the discrete random variable s , taking the value of any of the M possible implemented models. The new state vector to estimate is then the previous state vector \mathbf{x} appended this variable.

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix}$$

This complicates the equations of the filter, as a new discrete state is introduced. The integral with respect to y is now to be understood as a combination of a sum over the discrete parts and an integral over the continuous, i.e. $\int f(\mathbf{y})d\mathbf{y} = \sum_s \int f(\mathbf{x}|s)d\mathbf{x}$. The new prediction and update equations can be factored into a continuous and discrete part. In practice, these are predicted and updated separately, meaning that the full density $f(\mathbf{y})$ is not evaluated.

$$\begin{aligned}f_{k|k-1}(\mathbf{y}_k) &= f_{k|k-1}(\mathbf{x}_k|s_k) \Pr(s_k|\mathbf{z}_{1:k-1}) = f_{k|k-1}^{s_k}(\mathbf{x}_k) \mu_{k|k-1}^{s_k} \\ f_k(\mathbf{y}_k) &= f_k(\mathbf{x}_k|s_k) \Pr(s_k|\mathbf{z}_{1:k}) = f_k^{s_k}(\mathbf{x}_k) \mu_k^{s_k}\end{aligned}\tag{45}$$

The notation f^s means the distribution f under model s , i.e. $f^s(\mathbf{x}) = f(\mathbf{x}|s)$. μ_k^s is the probability of model s at timestep k . Through some work, the equations of Equation 45 can be shown to take the following form.

$$f_{k|k-1}(\mathbf{y}_k) = \mu_{k|k-1}^{s_k} \overbrace{\int f_{\mathbf{x}}^{s_k}(\mathbf{x}_k|\mathbf{x}_{k-1}) \underbrace{\frac{\sum_{s_{k-1}} \pi^{s_{k-1}s_k} \mu_{k-1}^{s_{k-1}} f_{k-1}^{s_{k-1}}(\mathbf{x}_{k-1})}{\mu_{k|k-1}^{s_k}}} d\mathbf{x}_{k-1}}^{f_{k-1}^{s_k}(\mathbf{x}_{k-1})}\tag{46}$$

where $\mu_{k|k-1}^{s_k} = \sum_{s_{k-1}} \pi^{s_{k-1}s_k} \mu_{k-1}^{s_{k-1}}$. Here, $\pi^{s's}$ is the known transition probability from model s' to s . All transition probabilities are tuning parameters in the same way that the choice of models

are tuning parameters.

$$f_k(\mathbf{y}_k) = \underbrace{\frac{f_z^{s_k}(\mathbf{z}_k|\mathbf{x}_k)f_{k|k-1}^{s_k}(\mathbf{x}_k)}{l_k^{s_k}}}_{f_k^{s_k}(\mathbf{x}_k)} \underbrace{\frac{l_k^{s_k} \mathbf{p}_{k|k-1}^{s_k}}{\sum_{s_k} l_k^{s_k} \mathbf{p}_{k|k-1}^{s_k}}}_{\mathbf{p}_k^{s_k}} \quad (47)$$

Where the model dependent measurement likelihood is defined as $l_k^{s_k} = \int f_z^{s_k}(\mathbf{z}_k|\mathbf{x}_k)f_{k|k-1}^{s_k}(\mathbf{x}_k)d\mathbf{x}_k$. Note that under Gaussian linear assumptions this can be reduced to $l_k^{s_k} = \mathcal{N}(\mathbf{z}_k; \mathbf{H}^{s_k}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k)$. This can clearly be seen from the first three lines of the proof of the Kalman update step found in Equation 9, when we notice the independence of \mathbf{x}_k in the Gaussian above.

Some care must be taken when mixing CV and CT models, as they have a different number of states. In practice, this can be done by adding another dimension to the CV model and letting it always be zero.

In the IMM framework $f_{k|k-1}^{s_k}(\mathbf{x}_k)$ is approximated as Gaussian before the prediction step runs. This is done by means of moment matching. In a PMBM derived filter with IMM, moment matching is then used twice each iteration to preserve Gaussianity. Once in the update step, when reducing the number of hypotheses, and once in the prediction step when merging model dependent priors. One such filter is the VIMMJIPDA.

3.10 VIMMJIPDA

VIMMJIPDA is the tracker employed in this project. It is a computationally efficient tracker which can be derived from the PMBM filter, with interacting multiple models and an added visibility state. The visibility state is not used in this project and will not be discussed further. The tracker uses pruning, gating, and approximates the posterior as a single multi-Bernoulli, along with a few other tricks to keep the implementation efficient. For a complete overview, see the original paper.[1] Compared to the multi target tracking assumptions in Table 2, VIMMJIPDA is more restrictive. In particular, the birth intensity is assumed to be stationary, and the clutter intensity is spatially constant as well. Furthermore, detection and survival probabilities are assumed constant. The amendments to the standard model used in VIMMJIPDA are summarized in Table 3.

- V1 The birth intensity is stationary $b(\mathbf{y}) = BV_\Omega \mathbf{p}_0^s f_b(\mathbf{x})$, where $f_b(\mathbf{x}) = \frac{1_\Omega(\mathbf{H}^s \mathbf{x})}{V_\Omega} \mathcal{N}(\mathbf{H}^{*,s} \mathbf{x}; \mathbf{0}, \mathbf{P}_v^s)$.
- V2 The probability of survival is constant $P_S(\mathbf{x}) = P_S$.
- V3 The kinematic prediction function follows $f_y(\mathbf{y}_k|\mathbf{y}_{k-1}) = \mathcal{N}(\mathbf{x}; f^{s_k}(\mathbf{x}_{k-1}), \mathbf{Q}^s) \pi^{s_{k-1}s_k}$.
- V4 The probability of detection is constant $P_D(\mathbf{x}) = P_D$.
- V5 Clutter is uniformly distributed with intensity $\lambda(\mathbf{H}^s \mathbf{x}) = \lambda V_\Omega \frac{1_\Omega(\mathbf{H}^s \mathbf{x})}{V_\Omega}$.
- V6 The measurement likelihood function is Gaussian linear, $f_z(\mathbf{z}|\mathbf{y}) = \mathcal{N}(\mathbf{z}; \mathbf{H}^s \mathbf{x}, \mathbf{R}^s)$.

Table 3: Assumptions in VIMMJIPDA. All non-contradictive assumptions of Table 2 are assumed.

The most important result of the VIMMJIPDA article is that the unknown intensity converges when the birth intensity is stationary. Given the form of the birth intensity in V1, the form of the unknown intensity is similar.

$$\begin{aligned} u(\mathbf{y}) &= \lim_{k \rightarrow \infty} u_{k|k-1}(\mathbf{y}) = UV_\Omega \mathbf{p}_u^s f_u(\mathbf{x}) \\ f_u(\mathbf{x}) &\approx f_b(\mathbf{x}) \end{aligned} \quad (48)$$

Note that both the birth and unknown intensity are distributed over state-space, while clutter is distributed over measurement-space. In Equation 48 and V1, V_Ω is the area of the terrain monitored by the tracker. \mathbf{H}^s is the measurement matrix under model s . In this project, it will always extract the positional components of the state $\mathbf{H}^s \mathbf{x} = [x, y]^\top$. $\mathbf{H}^{*,s}$ is the matrix that extracts all the other states, $\mathbf{H}^{*,s} \mathbf{x} = [v_x, v_y, \omega]^\top$. $f_u(\mathbf{x}) \approx f_b(\mathbf{x})$ is the assumed state distribution of the unknown targets. $1_\Omega([x, y]^\top)$ is the function that takes a position and returns 1 if it is in the monitored area

and 0 otherwise. This makes $1_\Omega([x, y]^\top)/V_\Omega$ a uniform position distribution over the monitored area Ω . The normal distribution describes the assumed velocities of the unknown targets. They are assumed to be distributed around zero, and therefore \mathbf{P}_v^s must be great enough to capture all potential initial velocities with reasonable probability mass. If the covariance is too small, a target might be able to get so far from its prediction that the correct measurement falls outside of the gate in the gating process. μ_u^s is the assumed initial model probabilities of an unknown target. U is the parameter that captures the expected number of targets per unit area that appear. Similarly, λ quantifies the expected number of clutter measurements per unit area. U is related to B by the following equation, which is easily found by setting $u_{k|k-1}(\mathbf{y}) = u_{k-1|k-2}(\mathbf{y}) = u(\mathbf{y})$ and using Equation 34a, Equation 35 and V1-V4.

$$U = B + UP_S(1 - P_D) = \frac{B}{(1 - P_S(1 - P_D))} \quad (49)$$

Since the unknown target intensity converges to a deterministic value proportional to the birth intensity, it can be used as a tuning variable directly, rather than the birth intensity. This is done in the VIMMJIPDA implementation. For this reason, the birth intensity and the unknown intensity are used interchangeably for the rest of this report. This project aims to quantify the importance of the tuning variables U and λ . To do this, two changes to the assumptions are made. The amended assumptions are summarized in Table 4.

- V1* The birth intensity follows $b(\mathbf{y}) = \mu_0^s B(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{H}^{*,s} \mathbf{x}; \mathbf{0}, \mathbf{P}_v^s)$, where $B(\mathbf{H}^s \mathbf{x})$ is proportional to some distribution in $\mathbf{H}^s \mathbf{x}$ over Ω .
- V5* Clutter is not uniformly distributed, but has intensity $\lambda(\mathbf{H}^s \mathbf{x})$, where $\lambda(\mathbf{H}^s \mathbf{x})$ is proportional to some distribution in $\mathbf{H}^s \mathbf{x}$ over Ω .

Table 4: The amendments done to the VIMMJIPDA assumptions in this project.

Given V1*, we find the new form of the converged unknown intensity in the same way as before. The spatial dependence $B = B([x, y]^\top)$ does not change the convergence, and therefore Equation 49 is still valid. The amendments seem simple, but have some consequences. In the update step in the new target case given by Equation 37, we find that the following term dominates.

$$\begin{aligned} & u_{k|k-1}(\mathbf{x}_k) f_{\mathbf{z}}(\mathbf{z}_k^j | \mathbf{x}_k) P_D(\mathbf{x}) \\ &= \mu_u^s U(\mathbf{H}^s \mathbf{x}_k) \mathcal{N}(\mathbf{H}^{*,s} \mathbf{x}_k; \mathbf{0}, \mathbf{P}_v^s) \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}^s \mathbf{x}_k, \mathbf{R}^s) P_D \\ &= P_D \mu_u^s U(\mathbf{H}^s \mathbf{x}_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s) \end{aligned} \quad (50)$$

where $\hat{\mathbf{x}}_{k0}^{sj} = \mathbf{H}^{s\top} \mathbf{z}_k^j$, and $\mathbf{P}_0^s = \mathbf{H}^{s\top} \mathbf{R}^s \mathbf{H}^s + \mathbf{H}^{*,s\top} \mathbf{P}_v^s \mathbf{H}^{*,s}$. If $U(\mathbf{H}^s \mathbf{x}) = U$ as in the original assumption, an integral of this quantity with respect to \mathbf{y} would remove the normal distribution and model state probability, however, this is no longer the case.

Using assumptions V2 and V4 of Table 3, the prediction equations of Equation 34 can be simplified. Keeping in mind that the unknown intensity no longer needs to be updated and that the mixture reduction reduces the number of indexes needed, further simplifications can be made. Finally, incorporating the IMM equations (Equation 46), we arrive at the prediction step of VIMMJIPDA.

$$r_{k|k-1}^t = r_{k-1}^t P_S \quad (51a)$$

$$\mu_{k|k-1}^{ts} = \sum_{\tilde{s}} \pi^{\tilde{s}s} \mu_{k-1}^{t\tilde{s}} \quad (51b)$$

$$f_{k|k-1}^{ts}(\mathbf{x}) = \int f_{\mathbf{x}}^s(\mathbf{x} | \mathbf{x}') f_{k-1}^{ts,0}(\mathbf{x}') d\mathbf{x}' \quad (51c)$$

$$f_{k-1}^{ts,0}(\mathbf{x}) \approx \frac{\sum_{\tilde{s}} \pi^{\tilde{s}s} \mu_{k-1}^{t\tilde{s}} f_{k-1}^{t\tilde{s}}(\mathbf{x}_{k-1})}{\mu_{k|k-1}^{ts}}$$

The approximation signifies the mixture reduction that is used to keep $f_{k-1}^{ts,0}(\mathbf{x})$ Gaussian. The same procedure as above can also be used for the update step. Using the VIMMJIPDA assumptions,

the equations for the update step in the missed detection case are found from Equation 36 and the IMM Equation 47.

$$w_k^{t0} = 1 - r_{k|k-1}^t P_D \quad (52a)$$

$$r_k^{t0} = \frac{r_{k|k-1}^t (1 - P_D)}{1 - r_{k|k-1}^t P_D} \quad (52b)$$

$$f_k^{ts0}(\mathbf{x}) = f_{k|k-1}^{ts}(\mathbf{x}) \quad (52c)$$

$$\mathbf{p}_k^{ts0} = \mathbf{p}_{k|k-1}^{ts} \quad (52d)$$

Repeating the process for the equations of a new track from Equation 37 the VIMMJIPDA new track equations are found. The amendments made in Table 4 must be remembered this time. Here, Equation 50 comes in. Taking into account that $\hat{\mathbf{x}}_{k0}^{sj} = \mathbf{H}^{s\top} \mathbf{z}_k^j$ and $\mathbf{P}_0^s = \mathbf{H}^{s\top} \mathbf{R}^s \mathbf{H}^s + \mathbf{H}^{*,s\top} \mathbf{P}_v^s \mathbf{H}^{*,s}$, we arrive at the following.

$$w_k^{tj} = \lambda(\mathbf{z}_k^j) + P_D \sum_s \mathbf{p}_u^s \int U(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s) d\mathbf{x} \quad (53a)$$

$$r_k^{tj} = \frac{P_D \sum_s \mathbf{p}_u^s \int U(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s) d\mathbf{x}}{\lambda(\mathbf{z}_k^j) + P_D \sum_s \mathbf{p}_u^s \int U(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s) d\mathbf{x}} \quad (53b)$$

$$f_k^{tsj}(\mathbf{x}) \approx \frac{U(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s)}{\sum_{\tilde{s}} \mathbf{p}_{\tilde{s}}^s \int U(\mathbf{H}^{\tilde{s}} \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{\tilde{s}}, \mathbf{P}_0^{\tilde{s}}) d\mathbf{x}} \quad (53c)$$

$$\mathbf{p}_k^{tsj} = \mathbf{p}_u^s \quad (53d)$$

The equation for the state distribution must be made Gaussian to fit with the rest of the filter framework, again signified with the approximation symbol. Different approaches are discussed later. Compared to the standard VIMMJIPDA equations of [1], the expressions are more complicated. The measurement-dependent clutter intensity and the position-dependent unknown intensity are responsible for this, but especially the latter.

Finally, the case of continuing an existing track is described by Equation 38. The simplified equations, due to the assumptions of Table 3, take the same form as in [1].

$$w_k^{tj} = P_D r_{k|k-1}^t \sum_s \mathbf{p}_{k|k-1}^{ts} l_k^{tsj} \quad (54a)$$

$$r_k^{tj} = 1 \quad (54b)$$

$$f_k^{tsj}(\mathbf{x}) = f_{\mathbf{z}}^s(\mathbf{z}_k^j | \mathbf{x}) f_{k|k-1}^{ts}(\mathbf{x}) / l_k^{tsj} \quad (54c)$$

$$\mathbf{p}_k^{tsj} = \frac{\mathbf{p}_{k|k-1}^{ts} l_k^{tsj}}{\sum_{\tilde{s}} \mathbf{p}_{k|k-1}^{t\tilde{s}} l_{\tilde{s}}^{tsj}} \quad (54d)$$

$$l_k^{tsj} = \int f_{\mathbf{z}}^s(\mathbf{z}_k^j | \mathbf{x}) f_{k|k-1}^{ts}(\mathbf{x}) d\mathbf{x}$$

Due to the Gaussianity of priors and posteriors enforced by moment matching, the state prediction of Equation 51c and the state update of Equation 54c can be seen to be exactly those of the Kalman filter.

The final change that must be made to the VIMMJIPDA implementation is in the calculation of the marginal association probabilities. Using Equation 39 and the expressions for the weights listed above, the equation takes the following form.

$$\begin{aligned} \Pr(a_k) &\propto \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t = 0} (1 - r_{k|k-1}^{t,0} P_D) \times \prod_{\substack{t \in \{1, \dots, n_{k-1}\} \\ |a_k^t| > 0}} \frac{P_D r_{k|k-1}^{t,a_k^t} f_{k|k-1}^{t,a_k^t} [f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} | \cdot)]}{\lambda(\mathbf{z}_k^{a_k^t}) + P_D u_{k|k-1} [f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} | \cdot)]} \\ &= \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t = 0} (1 - r_{k|k-1}^t P_D) \times \prod_{\substack{t \in \{1, \dots, n_{k-1}\} \\ |a_k^t| > 0}} \frac{P_D r_{k|k-1}^t \sum_s \mathbf{p}_{k|k-1}^{ts} l_k^{tsa_k^t}}{\lambda(\mathbf{z}_k^{a_k^t}) + P_D \sum_s \mathbf{p}_u^s \int U(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{sa_k^t}, \mathbf{P}_0^s) d\mathbf{x}} \end{aligned} \quad (55)$$

which, again, is far more complicated than the standard JIPDA marginal association probabilities calculated as below.

$$\Pr(a_k) \propto \prod_{\substack{t \in \{1, \dots, n_{k-1}\} \\ |a_k^t| = 0}} (1 - r_{k|k-1}^t P_D) \times \prod_{\substack{t \in \{1, \dots, n_{k-1}\} \\ |a_k^t| > 0}} \frac{P_D r_{k|k-1}^t \sum_s l_{k|k-1}^{ts} l^{tsa_k^t}}{\lambda}$$

Note that the influence of unknown targets is entirely neglected.

4 Method

In this section, the method used to investigate the effect of U and λ is introduced. The relevant setup and thought processes are also presented.

4.1 Data

For the analysis in this project, data captured by a radar monitoring the Ravnkloa canal is used. The Ravnkloa canal runs from the north-east to the south-west and has a ship tunnel in the north that leads to the Trondheimsfjord. It has many docks for recreational boats and is a popular area for kayaking and rowing. The data were captured in the late summer and fall of 2023. The data set consists of 644 GB of raw radar data recorded over approximately 22 hours split over 847 recordings and 18 days, or 63 337 radar scans. The radar was only run during the day and recorded only when detecting movement in the canal. This biases the data towards traffic, which will be discussed later. The radar is located at the Fosenkaia dock north of the canal. Figure 2 shows imagery of the radar.



(a) The radar's shadow can be seen left of the tree in the middle of the image. Image from [norgebilder.no](#).

(b) The radar seen from the north. Image from [Google streetview](#).

Figure 2: The position of the radar at Fosenkaia.

The radar is placed next to a tree, and there are multiple boats in front of it. As can be seen in Figure 3, these obstruct the view to the east. Based on radar measurements, it has a range of 150 meters. The sweep rate is 48 rotations per minute. This gives a new set of measurements every 1.25 seconds. Based on Figure 3 the surveillance area can be approximated. The size of the area is useful for estimating bounds on the intensities. A rough estimate of the area gives an area of 0.02 km^2 . The estimated area is shown in Figure 4. Assuming that we should receive at least one clutter measurement per scan, a lower bound on the clutter intensity is given by $1/(0.02 \text{ km}^2) = 1/(2 \times 10^4 \text{ m}^2) = 5 \times 10^{-5}/\text{m}^2$. This is not an unreasonable assumption, due to the amount of clutter caused by the docks in the area. In fact, the lower bound could arguably be higher. Bounds on the unknown intensity are more difficult to reason about. The frequency with which targets arrive is not easy to reason about without knowing the traffic patterns of the area.

4.2 Preprocessing

The data from the radar is not readily available for use in the VIMMJIPDA. The output from the radar consists of echo intensities over the discretized surveillance area. The radar has an angular resolution of 0.088° and a radial resolution of 0.15m. A radar spoke is a measurement of the echo intensities in each of the cells defined by the radial resolution, for one angular cell. Figure 5

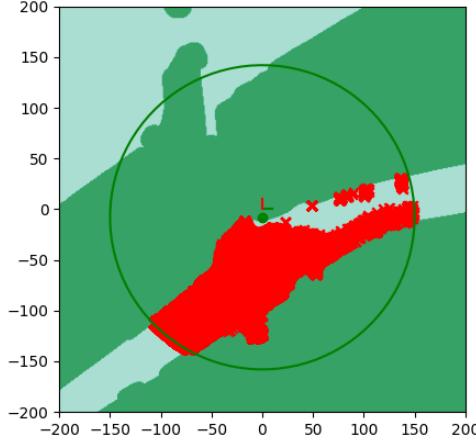


Figure 3: All measurements in the dataset. The circle has a radius of 150 m. The Fosenkaia frame of the aFerry ROS implementation is shown. A green dot is placed at the radar position in the Fosenkaia frame; $(0, -8)$.

visualizes a spoke for a 30° radial resolution. By collecting all the spokes for a single revolution of the radar, a full radar scan can be assembled. An existing implementation from the aFerry project at NTNU is used for processing the spokes. The aFerry project uses a ROS¹ pipeline to process the data step by step. The preprocessing pipeline is as follows. First, the spokes are assembled to a scan and converted to a pointcloud in cartesian coordinates. The program RViz can be used to visualize the output of each step. Figure 6 shows the pointcloud created in the first step. Next, all points on land are removed based on geographical information of the area. The remaining points are clustered using convex-hull clustering. Figure 8 shows this process. Convex-hull clustering groups together all points within some minimal distance from each other to a convex polygon. Finally, the centroids of the polygons are calculated. These centroids are considered the detection point of each cluster. Using ROS, the radar data is preprocessed and the detection points, together with their timestamps, are saved. After preprocessing, the size of the data set is dramatically reduced. The files containing the measurements ready for use in VIMMJIPDA measure only a total of 32

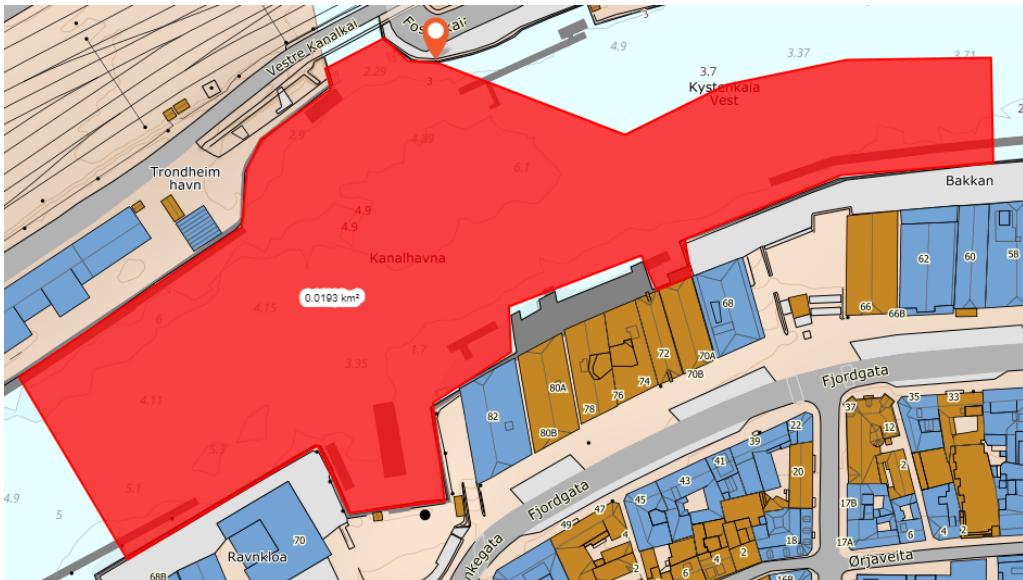


Figure 4: Approximate area monitored by the radar. The needle is placed at the radar position. The area measures approximately 0.02km^2 .

¹Robotic Operating System (ROS) is a framework for building robotic applications.

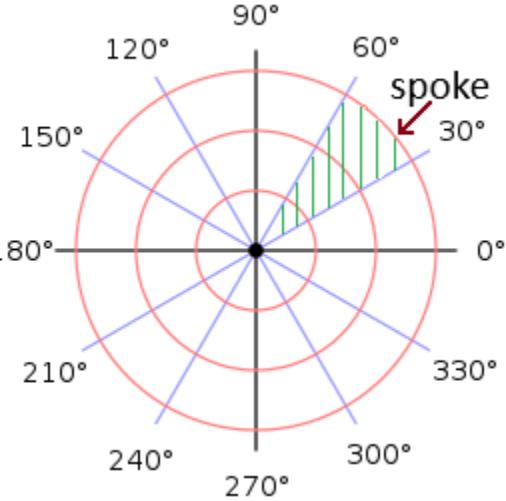


Figure 5: Illustration of a radar spoke. This spoke has 3 cells for which the echo intensity is measured. Since the cell size increases with range, the radar is less sensitive at distance.

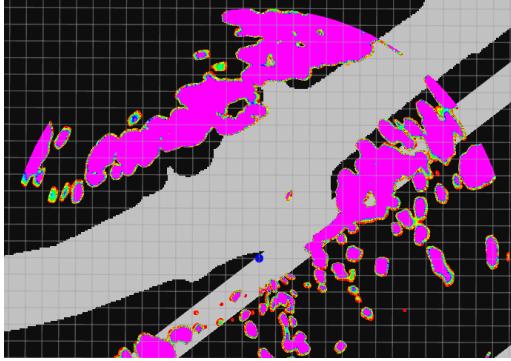


Figure 6: A single scan from the radar shown in RViz. The radar is at the blue dot. Due to a bug in RViz, the background map is not shown correctly in the lower right corner.

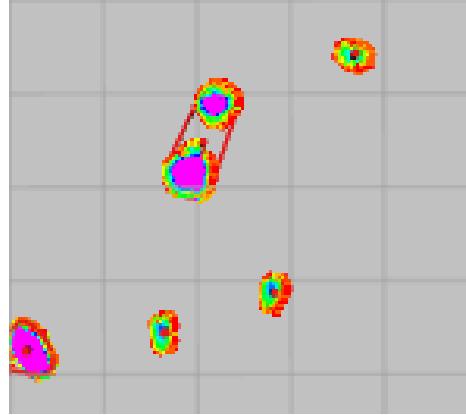
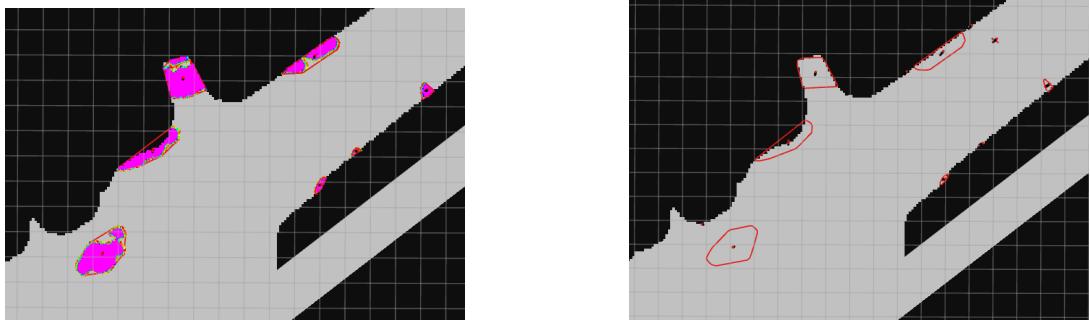


Figure 7: Distinct targets can be fused together to a single measurement in the pre-processing. This degrades the performance of the tracker.

MB, approximately 0.005% of the original size.

Using a radar makes the measurements robust to lighting and weather conditions, but there are some drawbacks. Radars suffer from smearing. Smearing is when a measurement leaks into nearby sensor cells. This can make objects appear larger than they are. Visually, this makes radar scans diffuse and blob-like. This smearing is obvious in Figure 6. Returns from land smear into the canal and cause false measurements. This causes there to be many false measurements close to land, even after filtering away land returns. Smearing worsens another drawback of radars. It can be difficult to separate two neighboring targets from one larger. When targets get too close to each other, the clustering extracts a single measurement. Figure 7 shows an example of this. This violates assumption M9 of Table 2, as a single measurement then corresponds to two targets. This degrades performance. Since spokes are aggregated every full revolution of the radar, there will be some mismatch in the age of the data processed together. The radar uses 1.25 seconds per revolution, meaning that the data will lag by up to this amount. This is not a big concern because of the low speed of the targets in the area. Similarly, radars suffer from stretching and squashing of targets moving normal to the scan direction. This is due to the sweeping motion of the radar scan, similar to how cameras suffer from motion blur. Again, this is not of any concern for the same reason as above.



(a) The radar scan after removing all returns from land.

(b) Clustered radar scan. The centroids of the clusters are the measurements used in the VIMMJIPDA.

Figure 8: Extracting points from a radar scan.

4.3 Filter setup

The filter runs with three different kinematic models. Two of the models are CV models. One of these is a high-covariance model, while the other is a low-covariance model. In addition, a CT model is included. The CT model uses the acceleration covariance of the low-covariance CV model. The respective covariance deviation tuning parameters are denoted $\sigma_{a,\text{low}}^2$, $\sigma_{a,\text{high}}^2$ and σ_α^2 .

Since the radar measures position independent of model, the measurement prediction function has the same form for all models.

$$f_z^s(\mathbf{z}|\mathbf{x}) = f_z(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \mathbf{R})$$

Where the measurement matrix is given by $\mathbf{H} = [\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{smallmatrix}]$. \mathbf{R} is the measurement covariance. Since a radar works in polar coordinates, so does its covariance. However, the measurements, state and rest of the framework are treated in a Cartesian frame. For this reason, the measurement covariance is converted from polar to cartesian coordinates, through linearization and uncertainty propagation. In addition, a cartesian uncertainty is added.

$$\mathbf{R} = \mathbf{J} \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \mathbf{J}^\top + \sigma_c^2 \mathbf{I}_2 \quad (56)$$

where \mathbf{J} is the jacobian of the polar to cartesian mapping evaluated at $\mathbf{H}\mathbf{x}$. The necessity of the polar component is evident when viewing the captured data. In Figure 9 the measurements in the trajectory can be seen to be more spread out farther away from the radar. Since the cells of Figure 5 are greater further away from the radar, the radar is less precise at longer ranges. This is modeled by σ_θ^2 of the polar covariance.

Revisiting Equation 50 we can simplify the normal distribution $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s)$ in light of the above arguments, making it entirely independent of the model s . We have $\hat{\mathbf{x}}_{k0}^{sj} = \mathbf{H}^\top \mathbf{z}_k^j = [\mathbf{z}_k^j, \mathbf{0}]^\top$, which is independent of s . $\mathbf{P}_0^s = \mathbf{H}^\top \mathbf{R} \mathbf{H} + \mathbf{H}^{*\top} \mathbf{P}_v^s \mathbf{H}^* = \text{diag}(\mathbf{R}, \mathbf{P}_v^s)$ is also independent of s if the initial velocity covariance is independent of s . This we can do by design, as this is a tuning parameter. We choose $\mathbf{P}_v^s = \mathbf{P}_v = \text{diag}(\sigma_v^2, \sigma_v^2, \sigma_\omega^2)$, leaving σ_v^2 and σ_ω^2 as the two parameters that quantify the variance in the starting linear and angular velocities of the newborn targets. These parameters must be high enough to capture all possible starting velocities with reasonable probability. The assumed state distribution of newborn targets is now $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k0}^j, \mathbf{P}_0)$. It should be noted that $\mathbf{P}_0 = \text{diag}(\mathbf{R}, \mathbf{P}_v)$, now is a function of the measurement \mathbf{z}_k^j because the jacobians in Equation 56 are being evaluated at $\hat{\mathbf{x}}_{k0}^j = [\mathbf{z}_k^j, \mathbf{0}]^\top$. To make this relationship explicit, we write $\mathbf{R}(\mathbf{z}_k^j) = \mathbf{R}_k^j$ to note the dependence of the covariance on the measurement \mathbf{z}_k^j . By extension, this makes $\mathbf{P}_{0k}^j = \mathbf{P}_0$ at \mathbf{z}_k^j .

The radar at Fosenkaia has been used in previous projects. In [7] parameters related to the radar were estimated. The estimates are listed in Table 5. These estimates were used as a starting point in the tuning process of the tracker, together with the values used in [1]. In [7] only cartesian

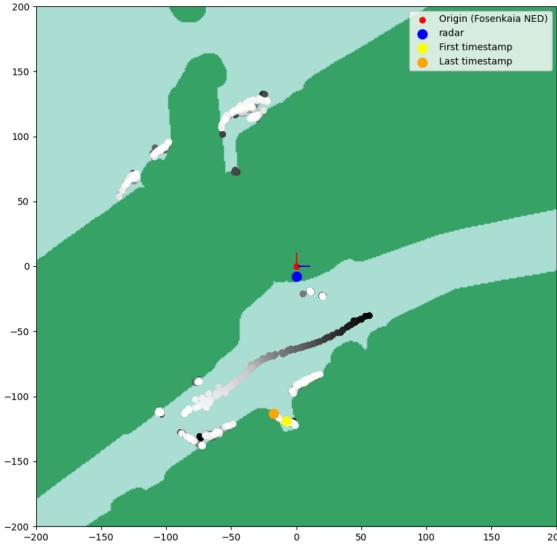


Figure 9: An example of measurements extracted from a radar recording. The measurements in the fjord north of the canal are bug points since there is a railway blocking the radar view. These are ignored in analysis.

Quantity	Symbol	Unit	Value
Detection probability	P_D	[%]	70
Survival probability	P_S	[%]	100
CV process noise std	σ_a	[m/s ²]	0.07
Cartesian noise std	σ_c	[m]	4.5
Clutter rate	λ	[1/m ²]	10^{-4}

Table 5: Parameter estimates found in [7].

variance was used in the measurement covariance. The tuning process was largely based on trial and error. Table 6 summarizes the parameters used in the filter in this project. The detection probability is inherited from [7]. So is the process noise, however, since this project uses multiple models, a slightly lower covariance and a slightly higher covariance CV model are used. Gate size, confirmation threshold, CT process noise, model probabilities, and transition probabilities are the same as those used in [1]. The survival probability is slightly reduced compared to [1], [7] in an attempt to terminate tracks more quickly. For this reason, the termination threshold is also raised substantially. The initial velocity standard deviations are based on reasonable assumptions of the velocities in the canal. The measurement standard deviations are chosen to match the standard deviations of [7] at the most distant parts of the surveillance area, whilst still having a reasonable uncertainty closer to the radar.

4.4 Implementation

In the theory section, the necessary changes to VIMMJIPDA were derived. Equation 53 and Equation 55 are the new equations for the update step when creating new tracks and calculating the marginal association probabilities, respectively.

The main issue in the implementation is the integral in $P_D \sum_s p_u^s \int U(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s) d\mathbf{x}$. Evaluating the integral for every received measurement is laborious. All other integrals in the filter equations have been solved analytically with the help of Gaussian approximations and the Kalman filter.

One approach to simplify the calculations is to use the approximation $U(\mathbf{H}\mathbf{x}) \approx U(\mathbf{H}\hat{\mathbf{x}}_{k0}^j) = U(\mathbf{H}\mathbf{H}^\top \mathbf{z}_k^j) = U(\mathbf{z}_k^j)$. That is, assume that the unknown intensity is uniform and equal to the

Quantity	Symbol	Unit	Value
Detection probability	P_D	[%]	70
Survival probability	P_S	[%]	98
Gate size	g	[–]	3.5
CV low process noise std	$\sigma_{a,\text{low}}$	[m/s ²]	0.05
CV high process noise std	$\sigma_{a,\text{high}}$	[m/s ²]	0.1
CT process noise std	σ_α	[°/s ²]	2
Cartesian noise std	σ_c	[m]	2
Polar range noise std	σ_r	[m]	3
Polar bearing noise std	σ_θ	[°]	$\sqrt{2}$
Initial linear velocity std	σ_v	[m/s]	2
Initial angular velocity std	σ_ω	[°/s]	10
Confirmation threshold	[–]	[%]	99.9
Termination threshold	[–]	[%]	20
Initial model probabilities	\mathbf{p}_u	[%]	[80 10 10]
IMM transition probabilities	$\boldsymbol{\pi}$	[%]	$\begin{bmatrix} 99 & 0.5 & 0.5 \\ 0.5 & 99 & 0.5 \\ 0.5 & 0.5 & 99 \end{bmatrix}$

Table 6: An overview of the parameters used in the tracker.

value at the measurement. Since the integral is independent of model, we can simplify further.

$$\begin{aligned}
& P_D \sum_s \mathbf{p}_u^s \int U(\mathbf{H}^s \mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^{sj}, \mathbf{P}_0^s) d\mathbf{x} \\
&= P_D \int U(\mathbf{H}\mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^j, \mathbf{P}_0) d\mathbf{x} \sum_s \mathbf{p}_u^s \\
&\approx P_D \int U(\mathbf{z}_k^j) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^j, \mathbf{P}_0) d\mathbf{x} \underbrace{\sum_s \mathbf{p}_u^s}_{=1} \\
&= P_D U(\mathbf{z}_k^j)
\end{aligned} \tag{57}$$

Using this approach, the equations can be dramatically simplified.

$$\begin{aligned}
w_k^{tj} &\approx \lambda(\mathbf{z}_k^j) + P_D U(\mathbf{z}_k^j) \\
r_k^{tj} &\approx \frac{P_D U(\mathbf{z}_k^j)}{\lambda(\mathbf{z}_k^j) + P_D U(\mathbf{z}_k^j)} \\
f_k^{tsj}(\mathbf{x}) &\approx \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k0}^j, \mathbf{P}_0) \\
\mathbf{p}_k^{tsj} &= \mathbf{p}_u^s
\end{aligned} \tag{58}$$

Ignoring the spatially varying unknown intensity and clutter intensity, this is identical to the update step for new tracks of the standard VIMMJIPDA filter. The association probabilities take the following form.

$$\Pr(a_k) \propto \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t = 0} (1 - r_{k|k-1}^t P_D) \times \prod_{\substack{t \in \{1, \dots, n_{k-1}\} \\ |a_k^t| > 0}} \frac{P_D r_{k|k-1}^t \sum_s \mathbf{p}_k^{ts} l^{tsa_k^t}}{\lambda(\mathbf{z}_k^{a_k^t}) + P_D U(\mathbf{z}_k^{a_k^t})}$$

Clearly, this does simplify things, but does not correctly reflect the influence of the unknown intensity. The simplification deflates the hypothesis weight and existence probability of new tracks in low-intensity areas neighboring on high-intensity areas. Similarly, it inflates the values of new tracks in high-intensity areas that neighbor low-intensity areas. The unapproximated formulation corrects for this.

Another option is to precompute the integral at a number of points throughout the working area. First, we can make some more simplifications. Continuing from the first equality of Equation 57 and recalling the block diagonal form of \mathbf{P}_0 , we can simplify the integral to only consider positional

states. Let us define $\rho = [x, y]^\top$ and $v = [v_x, v_y, \omega]^\top$ for notational brevity.

$$\begin{aligned}
& P_D \int U(\mathbf{H}x) \mathcal{N}(x; \hat{x}_{k0}^j, \mathbf{P}_{0k}^j) dx \\
&= P_D \iint U(\rho) \mathcal{N}(\rho; z_k^j, \mathbf{R}_k^j) \mathcal{N}(v; \mathbf{0}, \mathbf{P}_v) d\rho dv \\
&= P_D \int U(\rho) \mathcal{N}(\rho; z_k^j, \mathbf{R}_k^j) \int \mathcal{N}(v; \mathbf{0}, \mathbf{P}_v) dv d\rho \\
&= P_D \int U(\rho) \mathcal{N}(\rho; z_k^j, \mathbf{R}_k^j) d\rho = P_D \mathbb{I}(z_k^j)
\end{aligned} \tag{59}$$

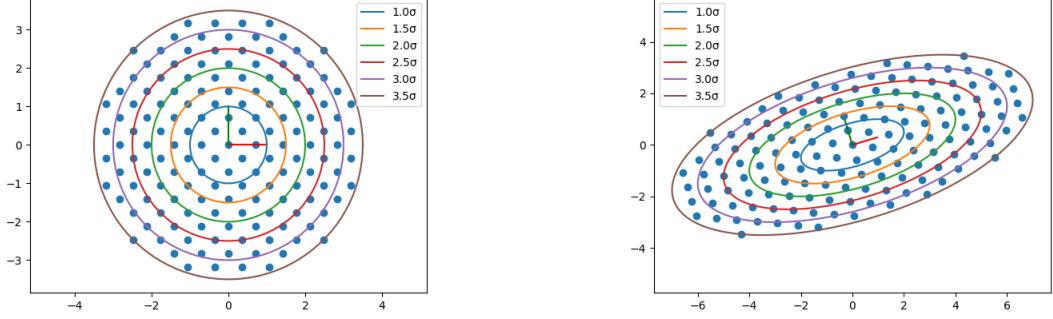
Note that Equation 59 is a function of z_k^j in both the mean and the covariance of the normal distribution. The dependence is made explicit through the notation $\mathbb{I}(z)$ for the integral at z . The discretization of the working area to precompute the integral above is worth some reflection, as is the way to compute the integral efficiently. However, before this, the form of $U(\rho)$ should be discussed.

From V1* and Equation 49, we find that U is proportional to some distribution in ρ . Physically, targets cannot appear anywhere in the surveillance area. They either enter the area from either side of the canal or through the tunnel from the fjord. Targets can also cast off from the docks along land. Ideally, there are no unknown targets in the middle of the canal, as they are detected earlier upon entering. Clearly, the shape of U will be dependent on the geography of the area, will not be an analytic function, and maybe even discontinuous. This makes approximation of the integral a difficult task, as small variations in the samples of U can take very different values. Nevertheless, let us give it a try.

To simplify the integral in Equation 59 for numerical applications, some tricks of linear algebra are used. To avoid cluttering the following derivation, sub- and superscripts are omitted. Since a covariance matrix is symmetric, it has a spectral decomposition. According to the spectral theorem $\mathbf{R} = \mathbf{V}\Lambda\mathbf{V}^\top$, where $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$. We have $\mathbf{R}^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{V}^\top = \mathbf{V}\sqrt{\Lambda^{-1}}\sqrt{\Lambda^{-1}}\mathbf{V}^\top$.

$$\begin{aligned}
\mathbb{I}(z) &= \int U(\rho) \mathcal{N}(\rho; z, \mathbf{R}) d\rho \\
&= \int U(\rho) \frac{1}{2\pi\sqrt{|\mathbf{R}|}} \exp\left\{-\frac{1}{2}(\rho - z)^\top \mathbf{R}^{-1}(\rho - z)\right\} d\rho \\
&= \int U(\rho) \frac{1}{2\pi\sqrt{|\mathbf{R}|}} \exp\left\{-\frac{1}{2} \underbrace{(\rho - z)^\top (\sqrt{\Lambda^{-1}}\mathbf{V}^\top)^\top}_{=\zeta^\top} \underbrace{\sqrt{\Lambda^{-1}}\mathbf{V}^\top(\rho - z)}_{=\zeta}\right\} d\rho \\
&= \int U(\rho) \frac{1}{2\pi\sqrt{|\mathbf{R}|}} \exp\left\{-\frac{1}{2}\zeta^\top \zeta\right\} d\rho \\
&= \int U(\rho(\zeta)) \frac{1}{2\pi\sqrt{|\mathbf{R}|}} \exp\left\{-\frac{1}{2}\zeta^\top \zeta\right\} |\mathbf{V}\sqrt{\Lambda}| d\zeta \\
&= \int U(\rho(\zeta)) \frac{1}{2\pi\sqrt{|\mathbf{R}|}} \exp\left\{-\frac{1}{2}\zeta^\top \zeta\right\} \sqrt{|\mathbf{R}|} d\zeta \\
&= \int U(\rho(\zeta)) \frac{1}{2\pi} \exp\left\{-\frac{1}{2}\zeta^\top \zeta\right\} d\zeta \\
&= \iint U(\rho(\zeta_1, \zeta_2)) \mathcal{N}(\zeta_1; 0, 1) \mathcal{N}(\zeta_2; 0, 1) d\zeta_1 d\zeta_2 \\
&\approx \int_{-3.5}^{3.5} \int_{-3.5}^{3.5} U(\rho(\zeta_1, \zeta_2)) \mathcal{N}(\zeta_1; 0, 1) \mathcal{N}(\zeta_2; 0, 1) d\zeta_1 d\zeta_2
\end{aligned} \tag{60}$$

where the change of coordinates is given by $\rho(\zeta) = \mathbf{V}\sqrt{\Lambda}\zeta + z$. The last approximation follows from the fact that 99.95% of the mass of the normal distribution is within 3.5 standard deviations. This allows us to sample from a standard normal distribution. The most straightforward approach



(a) Samples are made evenly in uncorrelated, normalized coordinates.

(b) The samples are transformed to the correct coordinates to evaluate $U(\rho)$.

Figure 10: The transform described by Equation 60. The samples are rotated by 45° with respect to the principal axes for better numerical properties.[8] Due to symmetry this is unproblematic.

is to use the definition of the Riemann integral to discretize the integrals.

$$\begin{aligned} & \iint U(\rho(\zeta_1, \zeta_2)) \mathcal{N}(\zeta_1; 0, 1) \mathcal{N}(\zeta_2; 0, 1) d\zeta_1 d\zeta_2 \\ & \approx \sum_i \sum_j w_i w_j U(\rho(\zeta_i, \zeta_j)) \mathcal{N}(\zeta_i; 0, 1) \mathcal{N}(\zeta_j; 0, 1) \end{aligned} \quad (61)$$

Now we need to decide how to choose the sampling points ζ and their weights w . In the Riemann approach, the weights are the distances between the sampling points. A simple approach is to have an equal spacing between samples, and only sample within 3.5 standard deviations. This makes the weights constant $w = \Delta\zeta$. The sample distance should be small enough to capture the changes in both the normal distribution and U . As discussed above, U is the critical factor here. Figure 10 shows this brute-force method.

A different approach is to use the Gaussian-Hermitian quadrature.[9] This is a computationally more efficient approach as it requires fewer sampling points. Another small change of coordinates transforms the problem to the correct form. Let $\vartheta_i = \frac{\zeta_i}{\sqrt{2}}$, making $d\zeta_i = \sqrt{2}d\vartheta_i$.

$$\begin{aligned} & \int U(\rho(\zeta_1, \zeta_2)) \mathcal{N}(\zeta_1; 0, 1) d\zeta_1 \\ & = \int U(\rho(\zeta_1, \zeta_2)) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\zeta_1^2} d\zeta_1 \\ & = \frac{1}{\sqrt{\pi}} \int U(\rho(\sqrt{2}\vartheta_1, \zeta_2)) e^{-\nu_1^2} d\vartheta_1 \\ & \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^n w_i U(\rho(\sqrt{2}\vartheta_i, \zeta_2)) \end{aligned}$$

The weights w_i and sampling points ϑ_i are given by tables of the quadrature. Their values depend on the order n . This gives the following approximation of the full problem.

$$\begin{aligned} & \iint U(\rho(\zeta_1, \zeta_2)) \mathcal{N}(\zeta_1; 0, 1) \mathcal{N}(\zeta_2; 0, 1) d\zeta_1 d\zeta_2 \\ & \approx \frac{1}{\pi} \sum_{i=1}^n \sum_{j=1}^n w_i w_j U(\rho(\sqrt{2}\vartheta_i, \sqrt{2}\vartheta_j)) \end{aligned}$$

The order n should be chosen so that a polynomial of degree $2n - 1$ approximates U well. As discussed above, the shape of U might be difficult to approximate. U is also not an expensive function to evaluate. The integral is computed ahead of time, so it is not very time-sensitive. For these reasons, the brute-force approach is used. However, if the measurement uncertainty is very

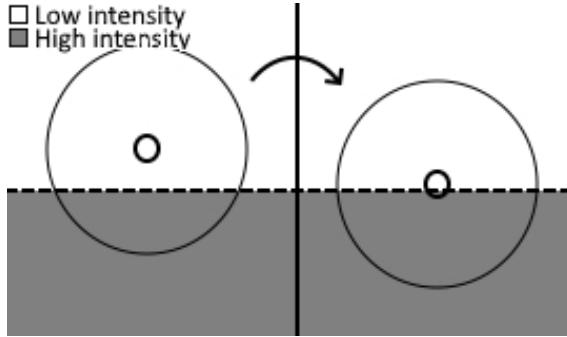


Figure 11: Adjusting the initial position distribution of a target towards a high birth-intensity area.

large, or the computation is needed online, a more refined approach might be convenient. For this purpose, the brute-force approach is good enough. Computing the integral for 9000 possible measurements, with roughly 275 points for each integral, takes about 12 seconds on the mid-range machine of the author.

The remaining mathematical challenge is the expression for the new target state density of Equation 53. The state must be approximated as Gaussian to work with the rest of the VIMMJIPDA. Since we know how to compute the integral of the denominator, our issue is the numerator. $U(\mathbf{H}\mathbf{x})\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_0, \mathbf{P}_0) = U(\boldsymbol{\rho})\mathcal{N}(\boldsymbol{\rho}; \mathbf{z}, \mathbf{R})\mathcal{N}(\mathbf{v}; \mathbf{0}, \mathbf{P}_v)$ must be approximated as Gaussian. Of this expression, the velocity component can be safely ignored and just tacked back on after the approximation because of the independence between the distributions. The simple approach is to again ignore U and assume that the normal distribution itself best approximates the product $U(\boldsymbol{\rho})\mathcal{N}(\boldsymbol{\rho})$. The effect of this is that the initial position distribution of the targets is not attracted to areas with high birth intensity. Consider the situation in Figure 11. A measurement is received close to a high-intensity area, and the hypothesis of it coming from a new target is evaluated. Considering that targets are more likely to originate in high-intensity areas, it is reasonable to believe that the target actually is closer to the high-intensity area and the error being due to measurement noise. The best fitting Gaussian matches the expectation and covariance of $U(\boldsymbol{\rho})\mathcal{N}(\boldsymbol{\rho})$. This is only proportional to a probability distribution, but the denominator scales it correctly. To find these for arbitrary U more integrals similar to the one we just discussed must be solved.

$$\mathbb{E}(\mathbf{z}) = \mathbb{E}[\boldsymbol{\rho}|\mathbf{z}] \propto \int \boldsymbol{\rho} U(\boldsymbol{\rho})\mathcal{N}(\boldsymbol{\rho}; \mathbf{z}, \mathbf{R})d\boldsymbol{\rho} \quad (62)$$

$$\mathbb{V}(\mathbf{z}) = \mathbb{V}[\boldsymbol{\rho}|\mathbf{z}] \propto \int (\boldsymbol{\rho} - \mathbb{E}[\boldsymbol{\rho}])(\boldsymbol{\rho} - \mathbb{E}[\boldsymbol{\rho}])^\top U(\boldsymbol{\rho})\mathcal{N}(\boldsymbol{\rho}; \mathbf{z}, \mathbf{R})d\boldsymbol{\rho} \quad (63)$$

where the dependence on \mathbf{z} is made explicit. The proportional symbol reflects the need for scaling. The integrals can be computed using the same approach as Equation 61. Figure 12 shows the output of the implemented approximation discussed above. The intuition of Figure 11 is clearly correct.

The expectation, the variance, and the integral of Equation 61 are all functions of \mathbf{z} . If they are calculated online, this is not an issue. However, this slows the tracker by a factor of approximately 10. For this reason, precomputing is attractive. For simplicity, a grid of points to precompute for is used. The grid has the same radius as the radar, 150m, and the cell size is a tuning parameter. Due to the discretization, the values will most likely not have been computed for the actual measurements. To combat this, a weighted average is used. Upon receiving a measurement, the 4 closest precomputed values are found. They correspond to the corners of the grid cell in which the measurement falls.² The values are then estimated as the weighted average with the weights being the inverse of the distance from the measurement. Figure 13 shows this scheme. To avoid zero-divison and numerical issues, if a measurement is sufficiently close to a precomputed value, the precomputed value is used without averaging. The reason for using the weighted average instead of simply using the closest precomputed value, is because this would cause all new tracks

²To be precise, for some measurement positions only 3 of the grid cell corners are in the 4 closest values.

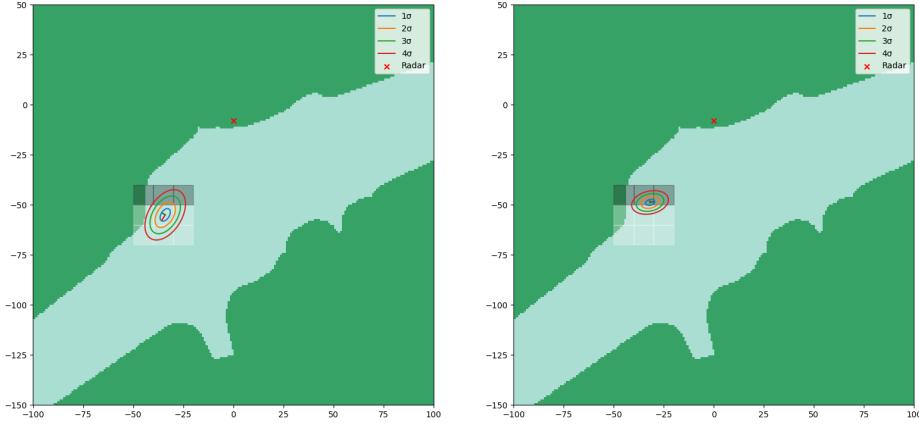


Figure 12: The effect of approximating the new target state density as Gaussian. Gray is a high-intensity area, and white is a low-intensity area.

to be moved close to grid corners. The expected value of Equation 62 is likely close to z , so using the precomputed expectation at the corner means moving z close to the corner, which is up to $\sqrt{2}/2 \times$ cell size away from the actual measurement.

Strictly speaking, the moment-matched covariance of a sum of Gaussians is the average covariance plus a term that stems from the spread of the means. This term inflates the computed covariance as a function of the distance to the means. When approximating multiple Gaussians as one, this is necessary to capture the spread of the Gaussians. This is not the intention here; the Gaussians are spread out, but we do not want to approximate their sum. We are approximating the covariance at the measurement. For this reason, a normal average is used. This is equivalent to approximating the sum of the Gaussians as if they all had a common mean. Simulations confirm this approach to be closer to the actual values.

Continuing with the notation $\mathbb{I}(z)$, $\mathbb{E}(z)$, $\mathbb{V}(z)$, for the integral, the expectation, and the variance,

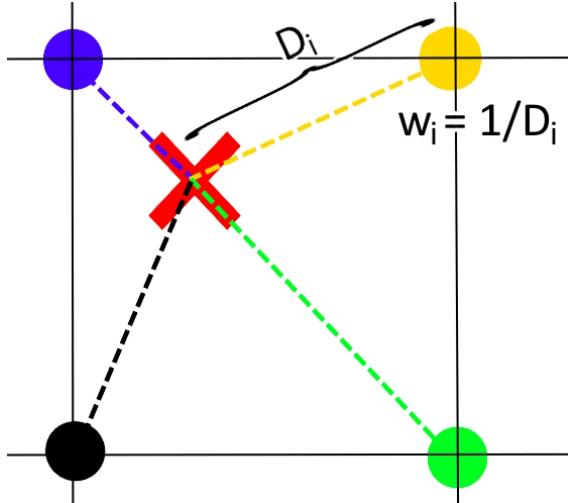


Figure 13: Computing the weighted average of the 4 closest precomputed values, with inverse distance weights. X marks the measurement location.

the new target update equations become

$$\begin{aligned}
w_k^{tj} &= \lambda(\mathbf{z}_k^j) + P_D \mathbb{I}(\mathbf{z}_k^j) \\
r_k^{tj} &= \frac{P_D \mathbb{I}(\mathbf{z}_k^j)}{\lambda(\mathbf{z}_k^j) + P_D \mathbb{I}(\mathbf{z}_k^j)} \\
f_k^{tsj}(\mathbf{x}) &\approx \mathcal{N}(\boldsymbol{\rho}; \mathbb{E}(\mathbf{z}_k^j), \mathbb{V}(\mathbf{z}_k^j)) \mathcal{N}(\mathbf{v}; \mathbf{0}, \mathbf{P}_v) \\
\mathbf{l}_k^{tsj} &= \mathbf{l}_u^s.
\end{aligned} \tag{64}$$

The corresponding association probabilities are

$$\Pr(a_k) \propto \prod_{t \in \{1, \dots, n_{k-1}\} | a_k^t = 0} (1 - r_k^t P_D) \times \prod_{\substack{t \in \{1, \dots, n_{k-1}\} \\ |a_k^t| > 0}} \frac{P_D r_k^t \sum_s \mathbf{l}_k^{ts} l^{tsa_k^t}}{\lambda(\mathbf{z}_k^{a_k^t}) + P_D \mathbb{I}(\mathbf{z}_k^{a_k^t})}.$$

Equation 58 and Equation 64 are two different ways of incorporating spatially variable clutter and birth intensity. They will be referred to as the Approximate model and the Integral model, respectively. To quantify the effect of the two different sets of equations on our data set, we first need to estimate the intensities and define the discretization resolution.

4.5 Estimating clutter and birth intensity

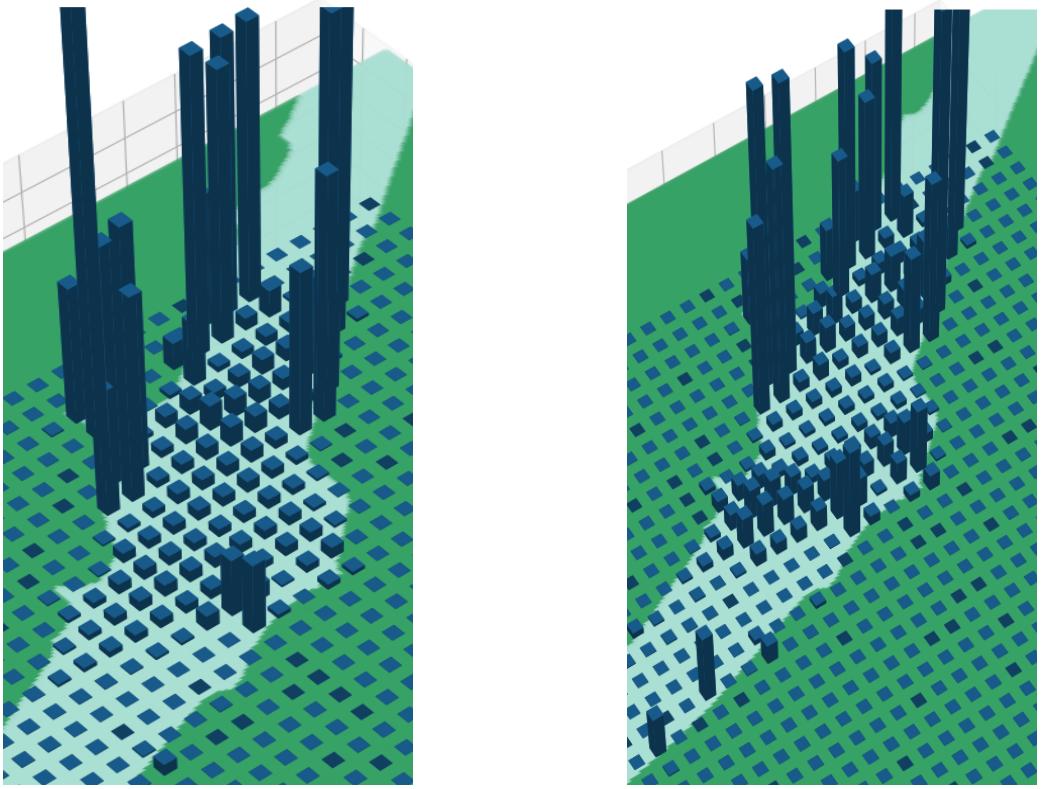
Assuming we have far more clutter measurements than target measurements, a decent approximation of the clutter intensity is the measurement intensity. This is a truth with modifications, as in some regions we might actually have a significant portion of target detections relative to clutter measurements. Nevertheless, due to the lack of ground truth, we proceed with this approximation. Discretizing the working area and counting the number of measurements in each cell, we get something proportional to the intensity. Dividing by cell area and number of scans, we get the number of expected measurements per square unit area per scan. This is the intensity. A 3D bar chart of the estimated clutter intensity in the canal is shown in Figure 14a. As expected, the clutter intensity is huge near land, due to obstacles by the docks and land-returns smearing into the corresponding sensor cells.

Estimating the birth intensity is less straight-forward. Without ground truth, the actual birth intensities are unknown. Using the normal VIMMJIPDA tracker without spatially varying birth and clutter gives far too many false tracks to be useful directly. The birth locations of all tracks over the entire data set are shown in Figure 14b. The same normalization as for the clutter is done. The birth intensity is almost identical to the clutter intensity, except some orders of magnitude smaller. A dominating amount of targets spawn in the clutter areas, most of these are false positives. Ignoring births in areas near land, some useful insight can be extracted. There are more births at the entry points to the surveillance area and few in the middle of the canal. This makes sense, targets must enter the canal from some place; there are no u-boats appearing in the middle of the canal. Ideally, all targets are detected at the time of entry. The intensities at the entry points seem to be roughly the same order of magnitude as well. This insight motivates the next choice.

4.6 Custom intensities

Based on the knowledge of the area, it seems reasonable to divide the birth intensity into high- and low-intensity zones. In reality, a smoother divide might be more fitting, but for analyzing the effect of spatially varying intensity, a binary divide is sufficient. Since the clutter intensity seems to either be huge or almost nonexistent, the same is done for this intensity. This choice simplifies the analysis.

The grid used to divide the area into zones has the same resolution as the grid used to count measurements and births. The grid cells are 10m × 10m. This choice was motivated by the nature



(a) A 3D bar graph of the clutter intensity.

(b) A 3D bar graph of the birth intensity extracted from the data set.

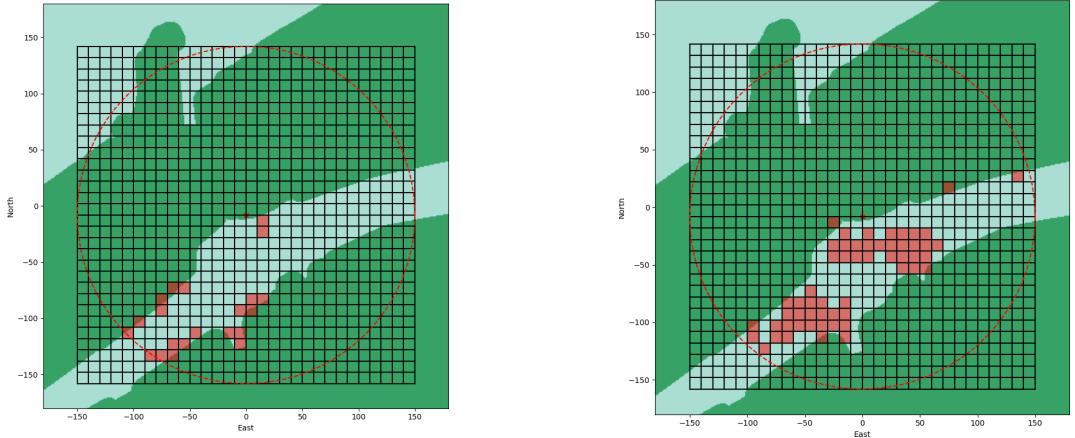
Figure 14: Bar graphs of the birth and clutter intensities extracted from the data.

of the clutter and traffic in the area, the size of the area monitored, and measurement uncertainties. Cells of $10m \times 10m$ were found to have a resolution decent enough to capture most of the intensity changes that occur in the area. For different areas, a different grid resolution is natural. For example, larger cell sizes naturally lend themselves to less busy and larger surveillance areas. This discretization does not mean that the discretization of the area for precomputing the integrals needs to be the same.

The divide into clutter areas was based on Figure 14. By thresholding the clutter intensities, all cells with an intensity above the threshold are extracted. Figure 15a shows the result of this for a threshold of $1 \times 10^{-3}/m^2$. About the top 20 clutter cells surpass this value and summarize the high-clutter locations well, based on experience from the data set. This intensity is equivalent to expecting a clutter measurement every 10 high-intensity cells per scan. If the entire area of Figure 4 were high-intensity, this would give an expectation of 20 false measurements per scan. When inspecting the scan in Figure 8, we see that we receive about six false measurements in a single scan. Also, there are often some false measurements in the eastern part of the canal as well. This is a single scan, but it represents the general trend in the data set well. This suggests that the intensity of high-clutter areas can be even greater than $1 \times 10^{-3}/m^2$.

Taking into account the areas extracted in Figure 15a combined with some experience from the data set and the geography of the area, the final division of the clutter intensity used for analysis was made. It is shown in Figure 16a.

A similar approach can be taken for the birth intensity zones, but perhaps deserves some more reflection, since the intensity is directly dependent on the model we are trying to improve. If we filter the bins of Figure 14b corresponding to the highest clutter intensity, we remove a substantial amount of false births occurring in high-clutter areas for the normal VIMMJIPDA tracker. Ignoring the top 30 bins and averaging the next 20 give an intensity of approximately $1 \times 10^{-5}/m^2$. Figure 15b shows the zone divide with a lower threshold of $5 \times 10^{-6}/m^2$, and upper threshold of



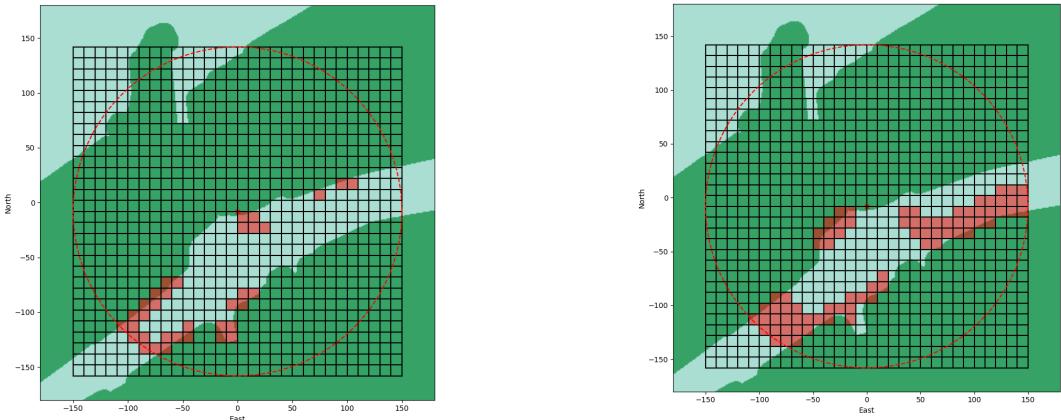
(a) Thresholding the estimated clutter intensity gives the following high- and low-intensity split of the area.

(b) Thresholding the estimated birth intensity gives the following high- and low-intensity split of the area.

Figure 15: The high- and low-intensity areas for clutter and birth found by thresholding Figure 14. Red squares show high-intensity grid cells. The range of the radar is shown with a circle.

$2 \times 10^{-5}/\text{m}^2$. These thresholds seem to remove most of the most clutter exposed bins near land, but also cover nearly all of the canal. As stated earlier, we want to detect targets before they reach too far into the canal. The variable birth intensity is a tool that may be useful for this purpose. Therefore, we define our birth intensity zones closer to the entrances of the surveillance area. Figure 16b shows the birth intensity zones used. The area to the east is especially large due to the poor visibility of the radar in this direction. This allows targets to get close before they are detected. Compared to Figure 15b, the low-intensity area in the middle of the canal is larger.

With the intensity zones defined, the natural next step is to define the values of the high- and low-intensity areas of both clutter and birth. So far, the following has been presented on the topic. A reasonable lower bound on the clutter intensity is $5 \times 10^{-5}/\text{m}^2$. This gives an expectation of one clutter measurement per scan over the entire area. However, this is a lower bound. The actual value used for the low intensity is probably higher, as the clutter map is guaranteed to cover all clutter areas. Especially in the western part of the canal, measurements seem to be more clutter prone. There are 35 high-intensity cells. The average intensity of the top 35 bins of Figure 14a is $5 \times 10^{-3}/\text{m}^2$. We can therefore expect a reasonable high-intensity value for clutter to be around



(a) The clutter zones used resemble the zones of Figure 15a.

(b) The used birth intensity zones. The high-intensity zones cover all entrances to the surveillance area to aid earlier detection of targets.

Figure 16: The clutter and birth zones used going forward in this project.

this order of magnitude. Simulations on the data set confirm that $1 \times 10^{-4}/\text{m}^2$ and $5 \times 10^{-3}/\text{m}^2$ work well.

The value of the high-intensity birth zones should be in the order of magnitude of $1 \times 10^{-5}/\text{m}^2$. In the case of targets escaping the high-intensity zones without being detected, the low-intensity zone should still have enough intensity to capture these targets. The average value of the 100 cells with the highest intensity excluded from Figure 15b is roughly $2 \times 10^{-6}/\text{m}^2$. A low-intensity in this range is therefore fitting. Simulations show $1 \times 10^{-6}/\text{m}^2$ and $1 \times 10^{-5}/\text{m}^2$ to work well.

	Low-intensity value	High-intensity value
Clutter intensity	$1 \times 10^{-4}/\text{m}^2$	$5 \times 10^{-3}/\text{m}^2$
Birth intensity	$1 \times 10^{-6}/\text{m}^2$	$1 \times 10^{-5}/\text{m}^2$

Table 7: The values used for the intensities in the analysis of the tracker.

4.7 Discretization of precompute grid

As written earlier, the precompute grid cell size is a hyperparameter that must be chosen. The grid size for the estimation of clutter and birth intensity is $10\text{m} \times 10\text{m}$, but this does not mean that the precompute grid must be the same. While the intensity grids should be able to capture the nature of the area, the precompute grid is more measurement oriented. The goal of the precompute grid is to approximate the values of $\mathbb{I}(z)$, $\mathbb{E}(z)$, and $\mathbb{V}(z)$ with reasonable accuracy using the method explained previously. The accuracy of the approximation is dependent on the size of the measurement covariance $R(z)$ and the birth intensity in the vicinity of the measurement.

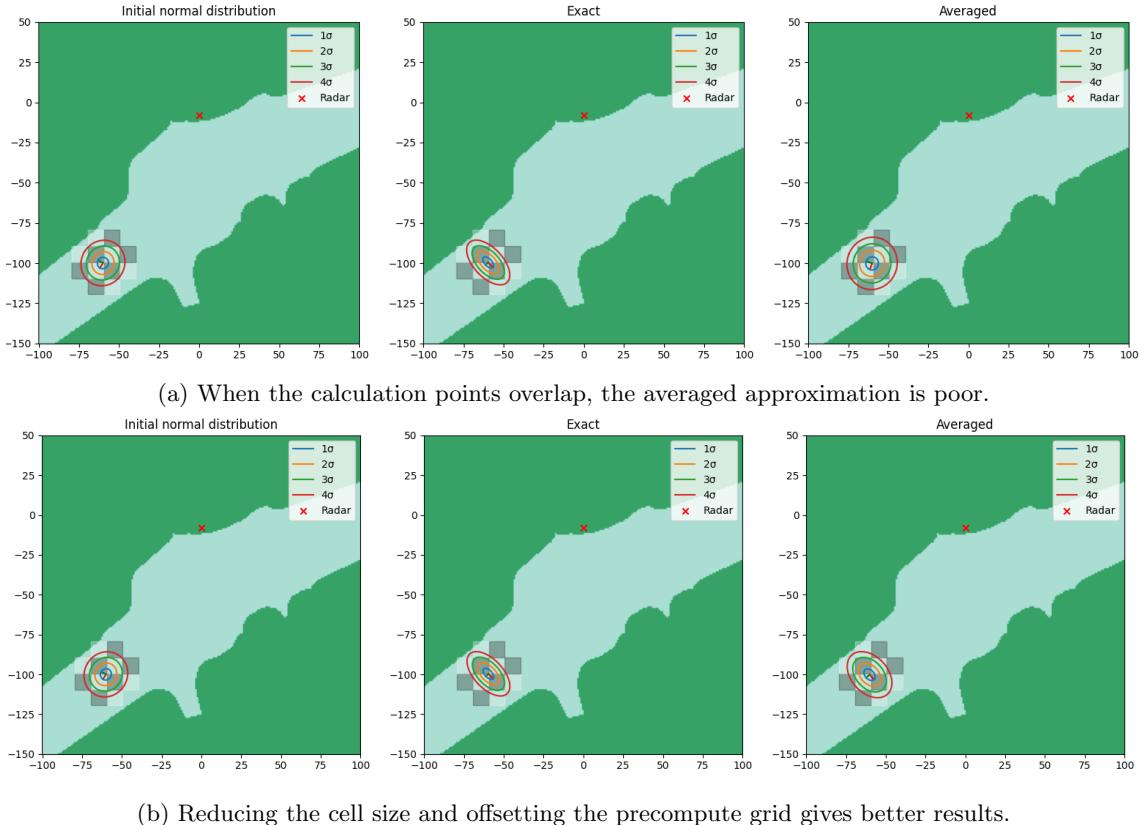


Figure 17: A constructed scenario where the overlap of the calculation points of the precompute grid and the birth intensity grid cause problems for the averaging approximation of the exact solution. Intensity values are exaggerated in this example.

To visualize the importance of how the precompute grid is arranged, consider the following example. Assume that the birth intensity and the precompute grid are identical. In certain scenarios, the precompute does not approximate the exact solution to the variance $\mathbb{V}(\mathbf{z})$ in the initial position distribution well. In Figure 17a the precompute points are in the same location as the birth intensity grid points; the center of each cell. In this constructed scenario, the birth intensity has a checkerboard pattern. Clearly, the averaged approximation is a poor representation of the exact solution. A simple solution to combat this issue is to reduce the size of the precomputed grid cells. Offsetting the grids from each other also helps avoid this issue. Of course, decreasing the cell size means increasing the number of points that need to be calculated. This makes precomputation take longer before the tracker can run, so there is a trade-off to be made.

To quantify the effect of the measures, the Frobenius norm of the difference between the exact and averaged covariance matrix is calculated. The Frobenius norm is the root of the sum of squared entries in the matrix.[10] The results are summarized in Table 8. Based on this, a cell size of $5\text{m} \times 5\text{m}$ is used for precomputing, giving a total of 2106 precomputed measurement positions. The precompute grid is also slightly offset from the birth intensity grid.

	No measures	Offset grid	Halved cell size	Halved and offset
Norm [m ²]	11.7	6.87	4.62	3.17

Table 8: The Frobenius norm of the difference between the exact and averaged covariance matrix with different measures to combat discrepancies. A lower value is better.

5 Model analysis

With the lack of ground truth, the performance of the two proposed variations of the VIMMJIPDA tracker must be evaluated manually. The evaluation criteria are chosen to shed light on the intended areas of improvement of the tracker. The intention of a spatially varying λ is to avoid the typical clutter tracks that the normal VIMMJIPDA is prone to in urban environments. At the same time, this should not be the same as simply filtering out the clutter areas in the preprocessing. Typical clutter areas include docks, an area where targets are likely to originate. A spatially varying U should enable earlier detection of targets, also in clutter areas. The evaluation criteria are summarized in Table 9.

- E1 Fewer clutter tracks
- E2 Detection of tracks originating in clutter areas
- E3 More robust tracking, less clutter disturbance
- E4 Earlier detection of tracks

Table 9: Evaluation criteria

Clutter tracks in criteria E1 means tracks spending their entire lifespan in a high-clutter area, or tracks obviously not tracking target-originated measurements. In Figure 18a there are a number of clutter tracks, and only one correct track.

To evaluate the performance of the trackers over the entire data set is far too much work, so a subset has been chosen. The subset consists of 66 radar recordings, roughly 8 % of the entire data set. The selected evaluation set consists of tracks in high-clutter areas and difficult maneuvers. Targets originating in high-clutter areas are especially overrepresented in the set. Additionally, some targets that spawn in low-intensity birth zones are included to validate that the trackers still capture them. The entire evaluation set can be found in Appendix Section A. Obtaining the performance of Figure 18 is not difficult, but is a proof that the clutter tracks are ignored. This result is achieved by both of the methods presented. However, the same result could be achieved by eliminating all measurements in high-clutter areas. For this reason, most recordings of scenarios like this are not included in the evaluation set.

In addition to evaluating performance over the evaluation set of the two trackers and a baseline normal VIMMJIPDA, some cases will be presented and discussed.

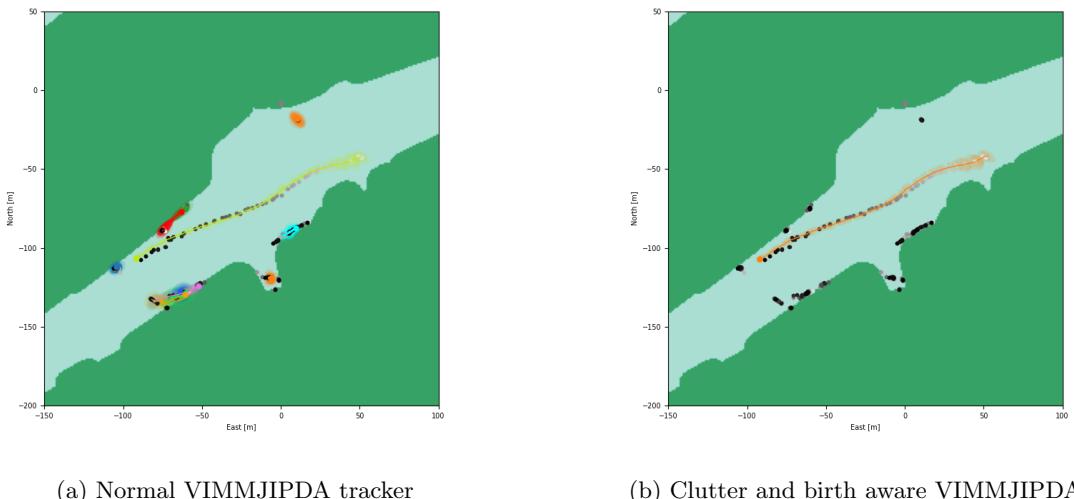


Figure 18: Comparison between performance of the VIMMJIPDA tracker without (left) and with (right) spatially varying intensities describing the area.

5.1 Results

Based on the criteria in Table 9, the three trackers are tested. E3 and E4 are not easy to quantify numerically, but are discussed based on observations of the tracker outputs. Statistics for criteria E1 and E2 are presented in Table 10, where the ground truth is defined manually.

All trackers share hyperparameters, except for the intensities in the baseline VIMMJIPDA. The baseline has constant intensities. It uses a birth intensity of $5 \times 10^{-6}/\text{m}^2$, approximately the average of the high- and low-intensity of the other trackers. Since the high-intensity clutter is 50 times larger than the low-intensity, this approach is not valid for the clutter intensity. Having a constant intensity of 25 times the low-intensity, would lead to all measurements in the middle of the canal being regarded as clutter, as there is virtually no clutter there. For this reason, a constant clutter intensity of $2 \times 10^{-4}/\text{m}^2$ is used. This is twice the clutter intensity of the low-intensity areas in the other trackers.

	Baseline	Approximate	Integral	Ground truth
False Tracks	625	56	55	0
Correct Tracks	135	136	140	146

Table 10: Evaluation results of the different trackers over the evaluation set.

The huge number of false tracks in the baseline comes largely from the high clutter areas, as can be seen in Figure 18a. The variable clutter density in the other two trackers clearly addresses this problem effectively with a reduction of more than 90%. The performance of the Approximate and Integral models are comparable in the number of false tracks. All three models perform about equally well in detecting actual targets. The difference in the number of correctly detected tracks between the Approximate and the Integral models are explored in Section 5.5.

There are mainly two types of false tracks appearing in both of the amended VIMMJIPDAs. When there are targets maneuvering in the middle of the canal, the radar view to the south-west corner of the canal is blocked for a period of time. This can cause false or noisy measurements in this part of the canal. This is caused by radar waves reflecting off other objects on their path from and to the radar. Figure 19a shows this phenomenon. There are two false tracks generated in the false measurements. The other typical source of false tracks is a violation of assumption M8 in Table 2. Sometimes, a single target generates more than one measurement. This can cause the tracker to identify two targets, although there is only one. Figure 19b shows an example of this. Here, four canoes pass through the canal and generate four correct tracks and three short false tracks. There is also some boat traffic in this example.

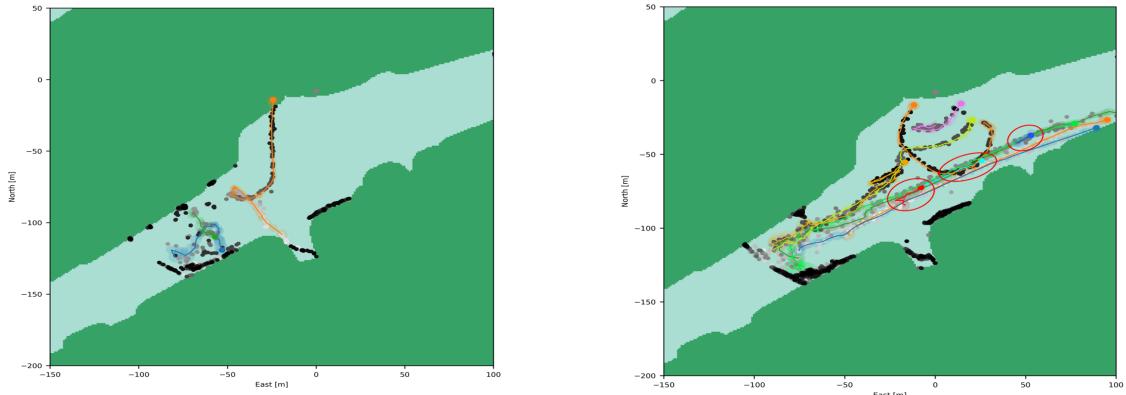


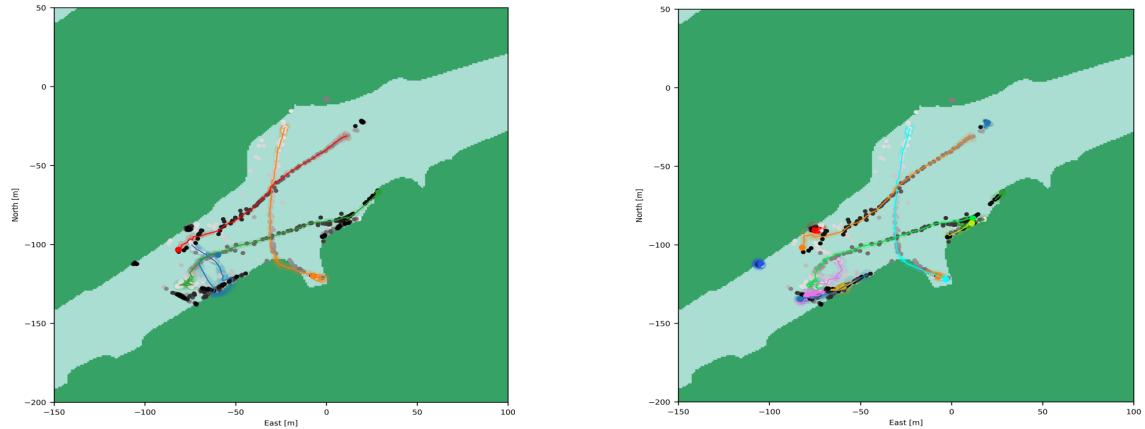
Figure 19: The two main sources of error tracks in the spatial clutter and birth aware models.

5.2 Case 1 - Effect of amended marginal probability

The marginal probability of a global hypothesis is one of the equations that change between models. This calculation reduces the effect of likely clutter measurements on existing tracks. Figure 20 shows an example of this. Clearly, the spatially aware VIMMJIPDAs are more robust against clutter disturbance, scoring better on evaluation criteria E3. The addition of the unknown target intensity integral in the denominator in Equation 39 allows the unknown targets to also influence the probabilities of the hypotheses. If a target is in a high-intensity birth zone, the probability that there exists an undetected target that causes measurements is greater than in a low-intensity area. Therefore, the clutter and birth aware VIMMJIPDA should be able to account for clutter and unknown targets not only when initializing tracks. This enables less noisy tracking.

Since even the high birth intensity is a factor of 10 smaller than the low clutter intensity, the birth intensity does not have a great effect on the marginal probability. However, clutter does. An interesting consequence of the influence of unknown targets on the marginal probability is that too high values for the birth intensity will not lead to more target births. Instead, all measurements will be claimed by unknown targets, making no targets able to claim more than their initial measurement, and therefore never amassing enough existence probability to pass the track confirmation threshold. Essentially, this is the same as what happens if the clutter intensity is set to high, except then all measurements are viewed as clutter.

Ignoring the unknown influence yields the marginal probability equations of a JIPDA tracker with spatially varying clutter. Although the unknown influence is less significant in the marginal probability calculation in our scenario, this is no reason to not include it. Mathematically, it belongs, as shown in the derivation of the trackers. In other scenarios, where clutter is less prominent, the unknown intensity can be more influential. Using the Approximate model, it is not costly to evaluate even online.



(a) The dark green track continues out of the clutter area, finishing the track correctly. The red track is not affected by the clutter towards the end of the track.

(b) The lime track is affected by the clutter area, and ends there. The orange track is affected by the clutter towards the end of the track.

Figure 20: The clutter areas at $(20, -100)$ and $(-70, -90)$ affect the clutter aware model (left) and the normal VIMMJIPDA (right) differently due to how the marginal association probabilities are calculated.

5.3 Case 2 - Initial position adjustment

The integral calculations defined by $\mathbb{I}(z)$, $\mathbb{V}(z)$, and $\mathbb{E}(z)$ come at a substantial numerical cost. Precomputing alleviates this, but is not transferable to natural extensions like measurements from a radar on a moving ownership. This is because the measurement covariance is dependent on the distance from the radar in the range of the polar coordinates. To account for this in precomputing,

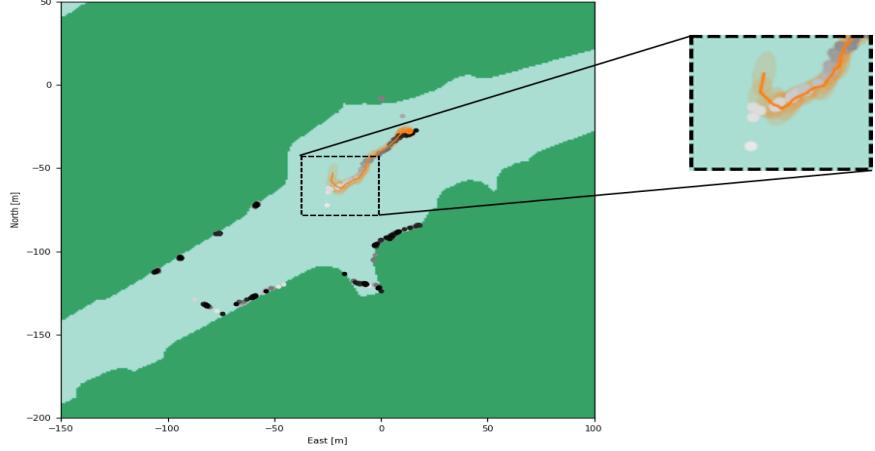


Figure 21: The effect of the calculation of the initial state distribution is prone to errors when U is not exact. Here a target spawns in the low-intensity birth zone and is erroneously corrected towards the high-intensity area. The intensity difference of the high- and low-intensity zones are exaggerated in this example.

it must be done for a discretized set of all combinations of ownership and measurement position, meaning that the precompute grid dimension has doubled. Transferred to the environment of this project, this means calculating for $2 \cdot 10^6 = 4\,435\,236$ position combinations. This obviously becomes impractical quickly. Also, if clutter and birth intensities are to be estimated online, pre-computing cannot be done. Therefore, the calculations must be done online in these scenarios. As previously mentioned, this slows down the tracker by a factor of approximately 10. The implementation of the author still runs faster than real-time, but not by a lot. There are ways to speed up computations, making the algorithm feasible in real-time applications, but the effort must be motivated.

The computation has two applications; the integral is used in the hypothesis weight and existence probability, and the expectation, variance, and integral are used in the initial state distribution. In this case, the effect of the initial state distribution is investigated. Section 5.5 discuss the weights and the existence probabilities.

The immediate argument against adjusting the initial state is the fact that it affects only the very first steps. The transient influence of the adjustment disappears after a few updates. Another argument is the dependence on the birth zones being correctly defined. Figure 21 shows how the birth intensity zones can affect the initial state of a target. Here, the target is more likely to have

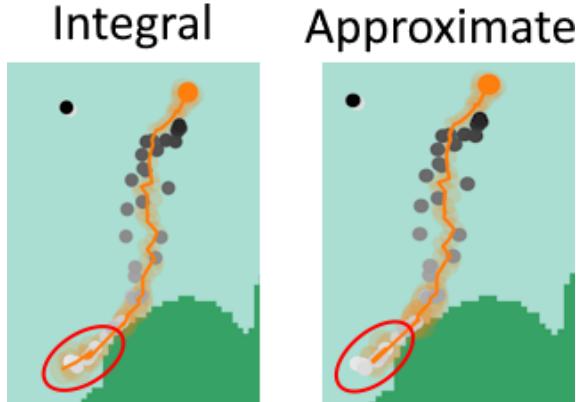


Figure 22: The Integral model correctly adjusts the track beginning towards a high birth zone in this example. The difference of the track in the Approximate model is nearly undetectable.

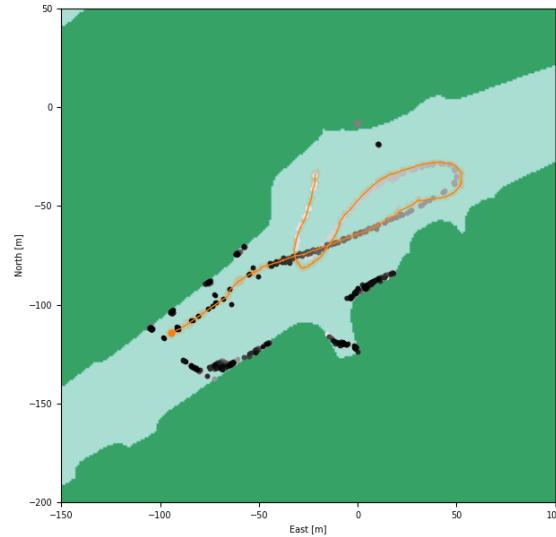
approached from the south, but is detected closer to the birth zone in the north and, therefore, pulled towards this area. The intensity difference is exaggerated in this example.

With the values used in this project, the effect is mostly undetectable or negligible. Figure 22 shows an example where the beginning of the track is correctly adjusted in the Integral model. The Approximate model does not do this adjustment, but the tracks are nearly indistinguishable. The effect of adjusting the initial position distribution does not seem worth the effort or computational load, based on the performance over the evaluation set. It does not seem likely that it is useful in many other scenarios either, except perhaps if the measurement noise is substantial. It requires intimate knowledge of birth intensities, which introduces another error mode.

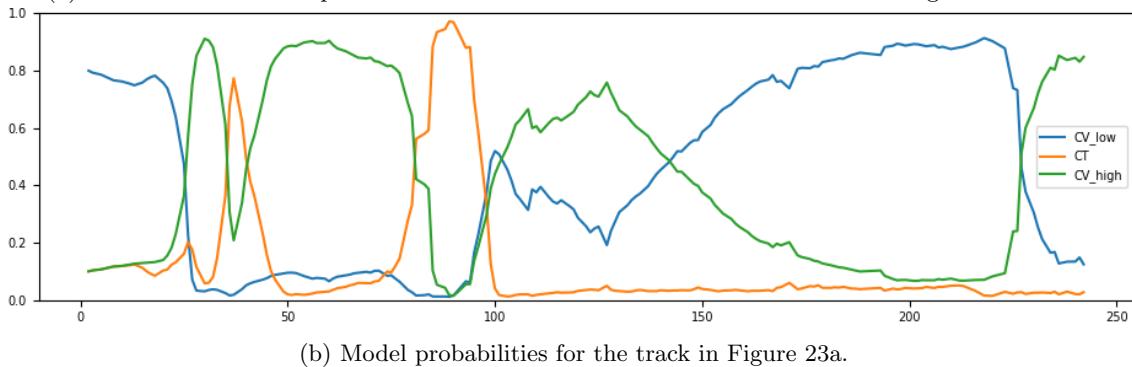
5.4 Case 3 - Kinematic models

The IMM framework in VIMMJIPDA is a complicating factor, so it should be justified. Hypothetically, a single high-covariance CT model could do the work of the three models used in this project. For the most part, this is also the case, but in some tracks the IMMs make a difference.

Figure 23a shows a track from the evaluation set where all models are used. In the straight parts of the track, the low-covariance model ensures smooth trajectories, while the CT model dominates in the turns. The high-covariance model is mostly influential in the transitions between the two other models in this scenario. This is not the general case; when the measurements are more noisy, the high-covariance model also dominates. Figure 23b shows the model probabilities at each time step for the track.

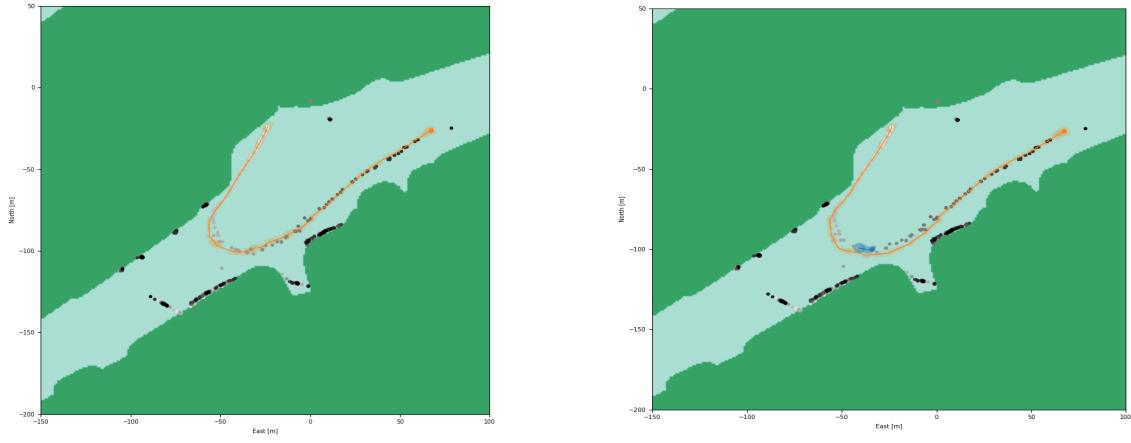


(a) This track is an example where all kinematic models dominate at different stages in the track.



(b) Model probabilities for the track in Figure 23a.

Figure 23: An example of a track where all models of the IMM are used.



(a) With IMM the CT model allows the track to follow reasonably close to the measurements.

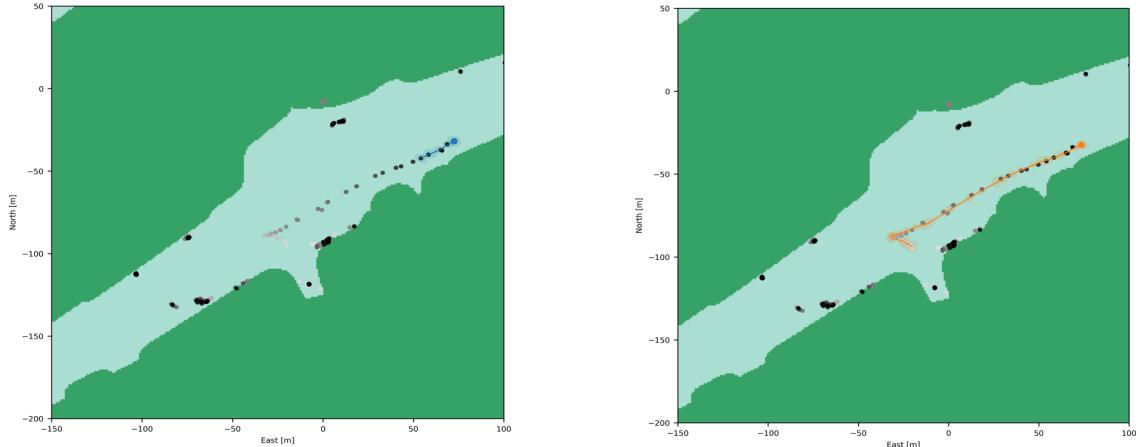
(b) Without IMM the single CV model is not able to follow the measurements closely, so a new target spawns in the ungated measurements.

Figure 24: The IMM makes a difference in some scenarios.

In this case, similar performance can be achieved by a single high covariance CV model. Figure 24 shows a track where this is not the case. Here, the CV model of Figure 24b is unable to make a tight enough turn, leaving some measurements unclaimed and free to spawn a new track. The IMM enabled tracker in Figure 24a follows the trajectory much closer, claiming all relevant measurements. Over the entire evaluation set, the difference in performance between using IMM or not is negligible. There are certain scenarios where it makes a difference, but even in these cases the consequences are unimportant.

5.5 Case 4 - Initial existence probability of the Integral and Approximate models

As previously hinted, the extra existence probability provided by the integral $\mathbb{I}(z)$ relative to the approximation can be decisive in track formation. Figure 25 shows an example of such a track. In Figure 25a the Approximate model is used, while the Integral model is used in Figure 25b. The Approximate model does not give enough existence probability to surpass the confirmation threshold on the hypothesis of the track found by the Integral model.



(a) Late initialization in the case of using the Approximation model.

(b) The initialization is good in the Integral model.

Figure 25: When the tracks are less dense, the extra existence probability can be decisive.

The affected tracks are tracks with initial measurements just outside a high-intensity birth zone. With the Approximate model, the existence probability is calculated by following Equation 58 with U being the low intensity. The Integral model calculates the existence probability according to Equation 64, with the integral of Equation 59. This means the existence probability is influenced by all close unknown target intensities. Since the target in Figure 25 spawns in a low-intensity zone bordering on a high-intensity zone, the integral considers some of the high-intensity zone, giving a higher initial existence probability. Figure 26a shows the difference in the approximated integral of Equation 57 and the numerical solution to Equation 59 over the surveillance area. That is, the difference between the unknown intensity $U(\mathbf{z})$ and the integral $\mathbb{I}(\mathbf{z})$. Clearly, they differ at the edge of the unknown intensity zones.

Inspecting the figure, the integral values seem to be roughly an average of the values around. This should be not surprising, as one way to view the integral $\mathbb{I}(\mathbf{z})$ is as Gaussian weighting of the unknown intensity. To illustrate this, Figure 26b is the unknown intensity plotted together with a smoothed unknown intensity. The smoothing is done by a convolution with a Gaussian kernel. Using the smoothed unknown intensity in place of the unknown intensity in the Approximate model gives the performance of the Integral model on the track in Figure 25.

Gaussian kernels of two sizes were tried.

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

The difference between the unknown intensity integral $\mathbb{I}(\mathbf{z})$ and the smoothed unknown intensity is summarized in Table 11. Smoothing with a kernel of size 3×3 more than halves the error made by the approximation in the Approximate model.

	No smoothing	5×5	3×3
Norm [1/m ²]	5.08×10^{-5}	2.43×10^{-5}	2.24×10^{-5}

Table 11: Euclidean norm of the error in the smoothing approach, compared to the numerical solution of the integral $\mathbb{I}(\mathbf{z})$.

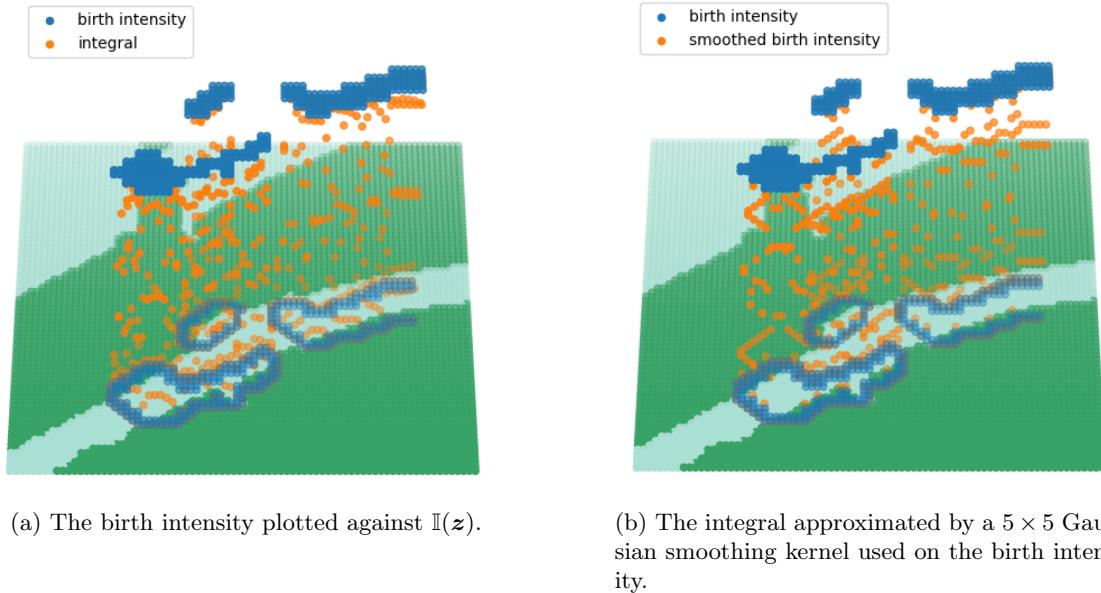


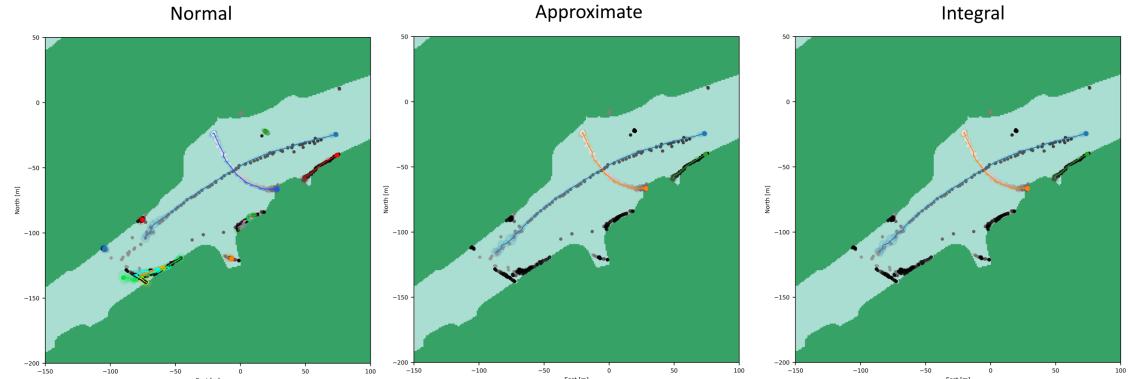
Figure 26: Scatter plot of the birth intensity integral values, and the smoothed birth intensity. As expected, the points on the border between high- and low-intensity differ from the birth intensity. Points where the integral and approximate equal the low intensity are not shown.

The result of the insight above is a computationally efficient way to compute a more accurate approximation of $\mathbb{I}(z)$ in the Approximate model. This closes the narrow performance gap between the Approximate and Integral model further, without the significant extra numerical cost of the Integral model. This variant of the Approximate model is also implemented in the code of this project.

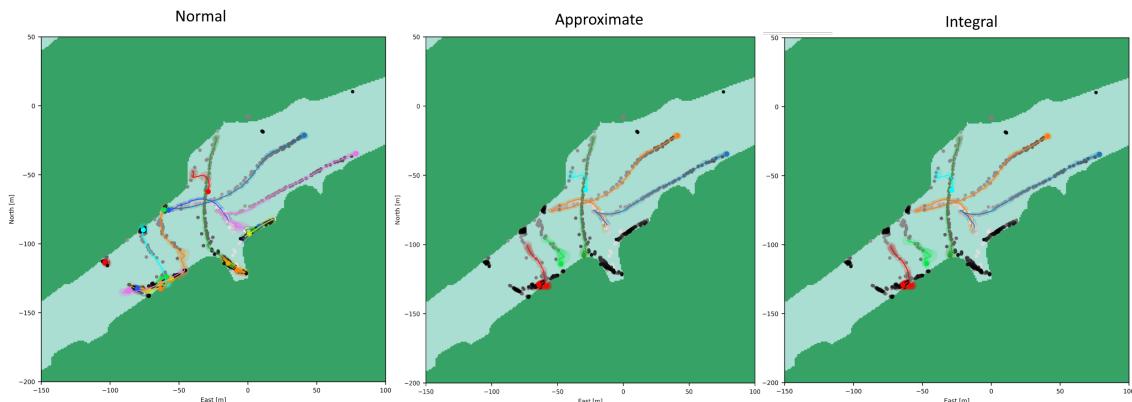
5.6 Case 5 - Early detection

The final evaluation criteria is earlier detection of tracks. This is partly successful. When a track starts in a low-clutter area, the amended birth zones facilitate earlier detection. Figure 27a shows an example of this. The blue track is detected earlier in the Approximate and Integral models compared to the baseline normal VIMMJIPDA. The Approximate and Integral models perform roughly equally well on this criterion, especially with the amendment to the Approximate model discussed in Section 5.5.

However, when targets originate in high clutter areas, they are rarely detected before they have left the area. The same equations that allow tracks to ignore measurements in high-clutter areas hinder detection in these areas. When a track is initialized in a high-clutter area, the existence probability is low. The high clutter intensity of the area denies any continuation of the track by claiming all measurements as clutter in the association probability formula. This happens because of the division by the clutter intensity of the update track weights. All hypotheses of updating the track with a new measurement then gets a low probability, making the marginal hypothesis dominated by the missed detection hypothesis. This means the existence probability of



(a) The blue track is detected earlier in the Approximate and Integral models compared to the normal VIMMJIPDA.



(b) The two tracks crossing the canal in the lower left corner are detected later in the Approximate and Integral models compared to the normal VIMMJIPDA.

Figure 27: Comparisons of when the trackers detect targets.

the marginalized hypothesis is lower than the initial existence probability. The result is that the track is pruned. Only tracks exiting clutter areas on their first measurement after initialization have a chance of reaching the confirmation threshold. In practice, tracks are mostly detected several time steps after leaving a high-clutter area. Figure 27b shows an example of this. Here the baseline VIMMJIPDA, with constant low clutter density, detects the two tracks crossing the canal in the lower left corner earlier than the other models.

To summarize, the spatially variable birth intensity facilitates earlier detection in low-clutter areas, but is dominated by the clutter intensity in high-clutter areas. Tracks are only detected after leaving high-clutter areas. Criteria E2 and E4 are therefore only partially successful.

5.7 Discussion

Based on the evidence presented in Table 10, criterion E1 is successful. E3 is also accomplished by the models that incorporate spatially varying intensities. The last two criteria, E2 and E4, were not met to a higher degree by the proposed models compared to the baseline. The early detection works in some cases, but is dominated by the clutter in other. Using the discussed amendment to the Approximate model of Section 5.5, the performance gain by using the Integral model is negligible. Perhaps the initial state distribution adjustment can have significant effect in certain high noise scenarios, but it seems unlikely to be decisive. The performance of the Approximation model is clearly the better choice taking the computational expense of the Integral model into account. The performance is generally good over the data set of this project, an example of this is shown in Figure 28.

In total, the performance of the Approximate model is a step towards more accurate multitarget tracking. Natural extensions could be to analyze the effect of varying survival and detection probabilities. Targets are more likely to "die" at the docks and the edges of the surveillance area.

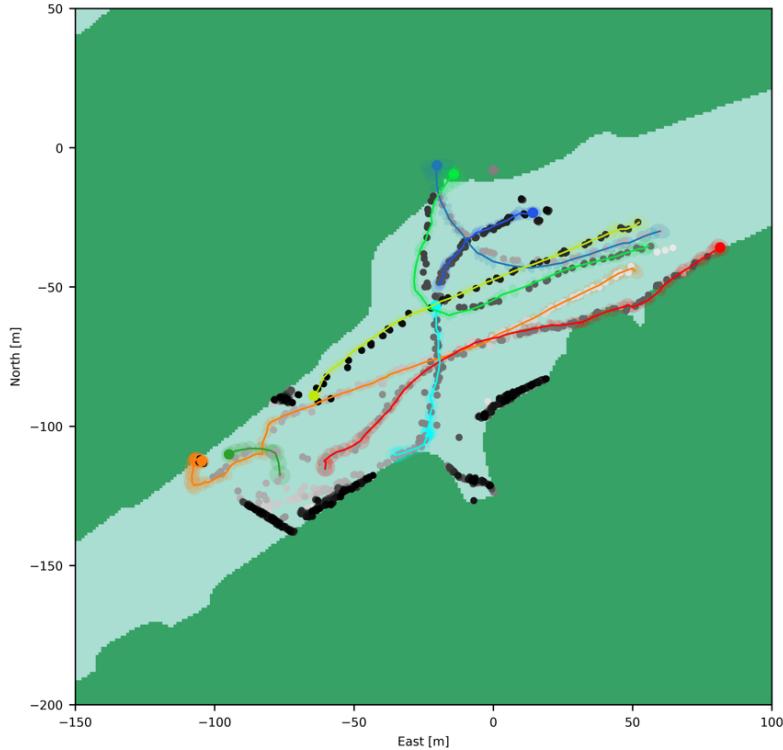


Figure 28: The tracking of the approximate model is good. The dark green track in the lower left corner is a false track caused by very noisy measurements in this area. The turquoise track was initialized late and should have been continued instead of a new blue track being created.

These areas coincide with the high-intensity zones of the birth intensity. The varying detection probabilities is in practice what the visual state of the VIMMJIPDA does adaptively. Extending the initial state distribution to also adjust the velocity distribution based on the initial position is another thing that can be considered. Targets detected at the edge of the surveillance area usually have a velocity towards the center of the canal. This could be beneficial for detecting targets leaving cluttered docks. Taking this one step further, the radar data could be used to predict trajectories in the canal giving more informed kinematic models than the simple CV and CT models. Trajectory prediction based on AIS has been done in [11]. The confirmation and termination thresholds were determined largely by trial and error in this project. This is a nontrivial problem that deserves more consideration but was outside the scope of this project and left for future work.

There are a few things about the setup of this project to point out. Mainly, the data the analysis of the intensities is based on is very clearly biased towards traffic. The radar data used was only recorded when there were known targets in the canal, meaning there are targets present in all the data. In reality, the canal is empty most of the time. A reasonable assumption is that the clutter is independent of traffic, so this bias does not affect the clutter intensity. This is a reasonable assumption because the vast majority of clutter is caused by land returns. The birth intensity, however, is obviously not independent of traffic. Had the data shown less traffic in the canal, the birth intensities would have had a lower estimate. A bias towards traffic is arguably better than the opposite. Recall is more important than precision for collision avoidance, missing one target might lead to a crash if the tracker is used in an autonomous setting. On the other hand, too many false tracks hinder good route planning. The Approximate and Integral model seem to find a decent compromise between precision and recall over the data set. Precision and recall are not inherently contradictory metrics, but in practice, prioritizing higher recall often comes at the expense of precision.

The trackers have not been thoroughly evaluated over scenarios without much traffic, but it is reasonable to believe the performance will be comparable to the one seen in this project. Much of the data recorded by the trackers only have targets in one part of the canal. Due to the single-linkage clustering in the VIMMJIPDA, this is effectively equivalent to one tracker handling this part of the canal and other trackers handling clusters of measurements in other parts of the canal. Since the performance of the "trackers" surveying the untrafficked part of the canal is good, this is reason to believe that the tracker would perform well overall in periods of time with less traffic, like night-time.

There is a huge dependence in the performance on the predefined zones in the spatially varying models. With erroneously defined zones, the results of the trackers are easily worse than that of the baseline, especially for the clutter estimation, since it is much larger than the unknown intensity. This motivates online estimation of the intensity zones, in addition to the prior estimates, to adapt to any changes or correct any errors made. [12], [13], and [14] explore estimators for clutter estimation. A temporally varying unknown target intensity is discussed in [15].

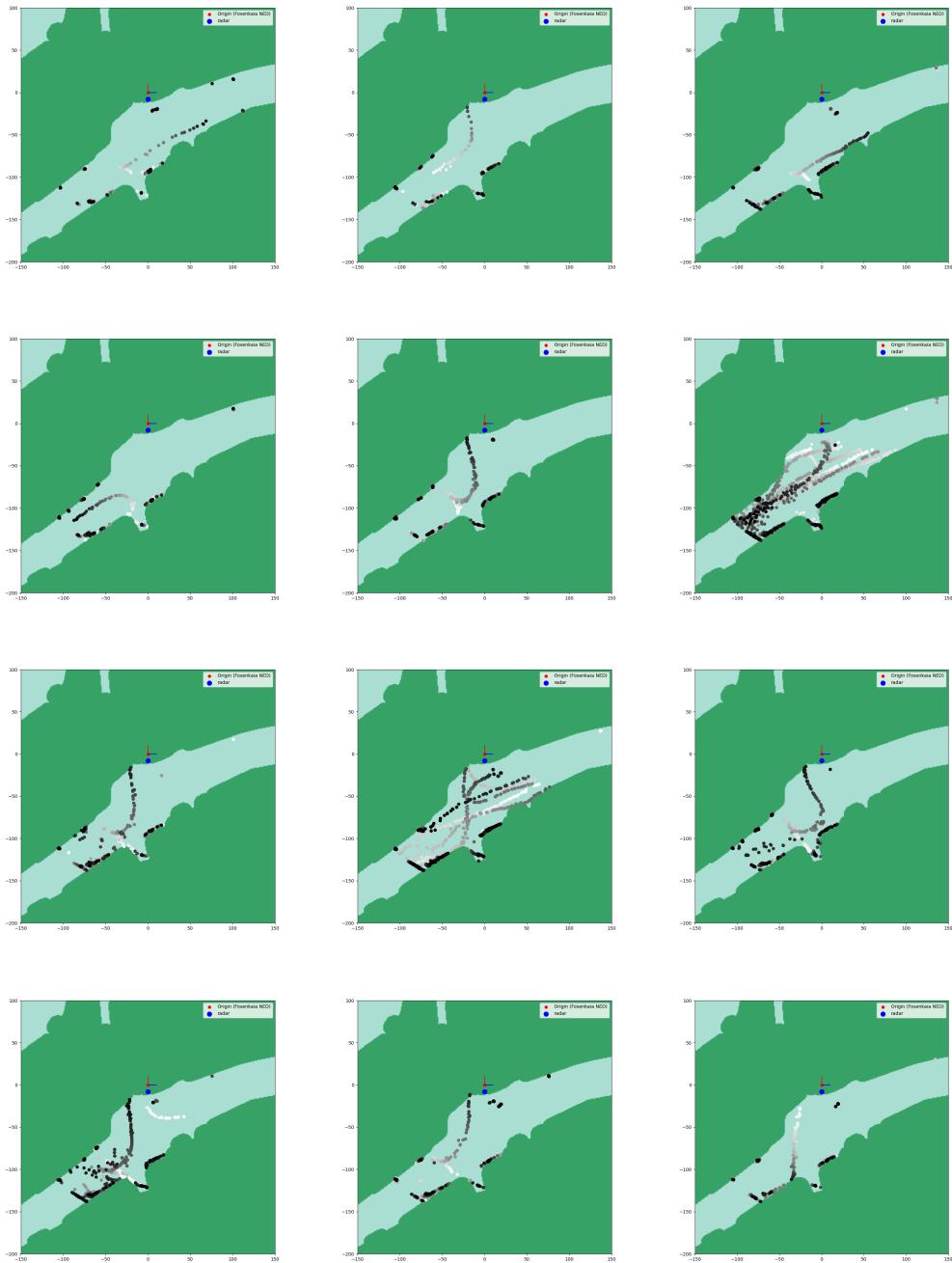
6 Conclusion

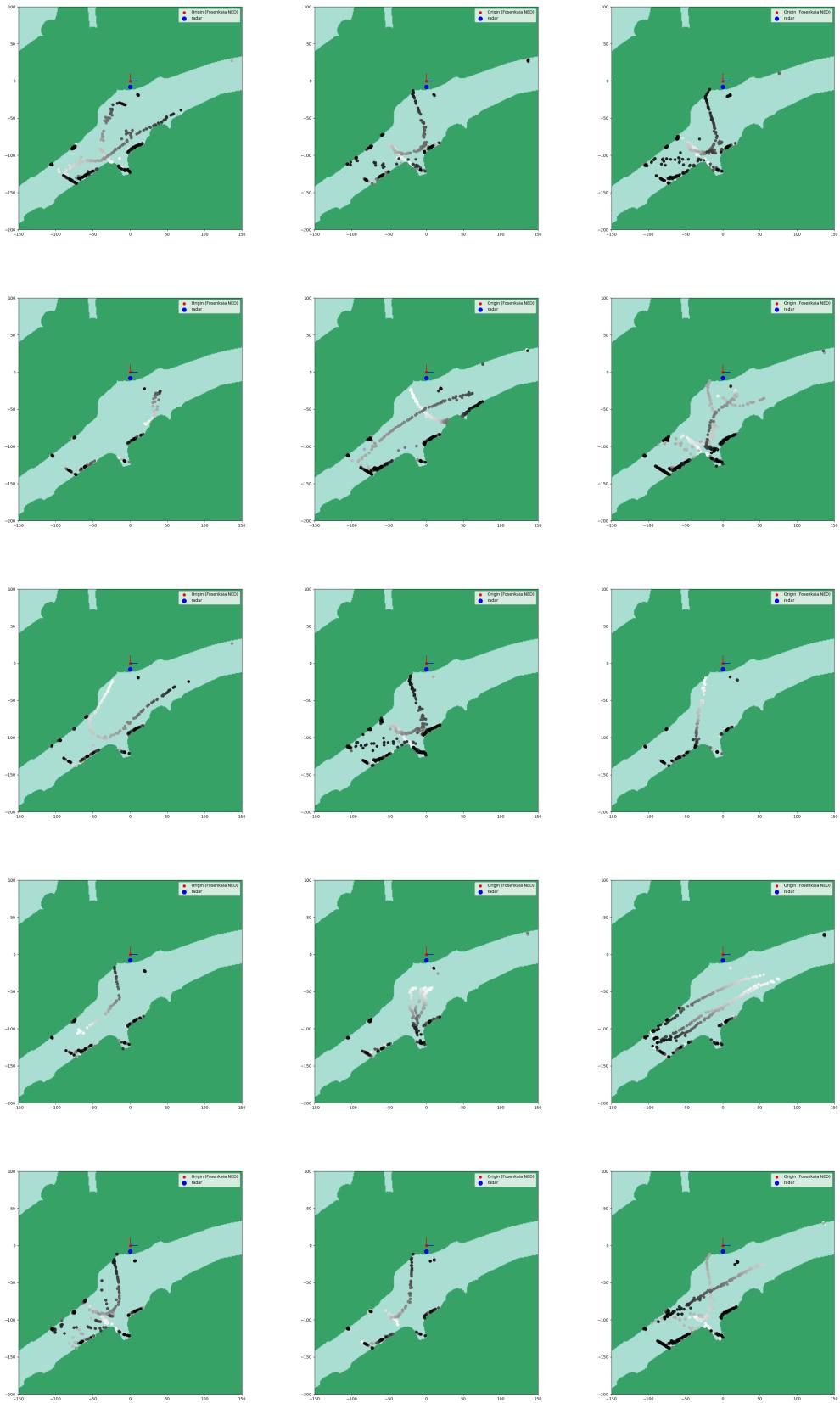
Totalling a reduction in false tracks of roughly 91% without losing correct tracks, the models incorporating spatially variable intensities clearly outperform the baseline VIMMJIPDA. The incorporation of spatial dependence in the clutter and birth intensity clearly helps reject many false tracks. However, the models did not outperform the baseline on all criteria. Some tracks are detected earlier, but the high clutter intensity denies the possibility of detecting tracks before they leave high-clutter areas. In low-clutter areas with high birth intensity, the birth intensity facilitates earlier detection. Overall, correctly defined intensities help tracker performance, but erroneously defined zones can hurt performance. For this reason, prior information about an area should be paired with an intensity estimation scheme for long-running trackers.

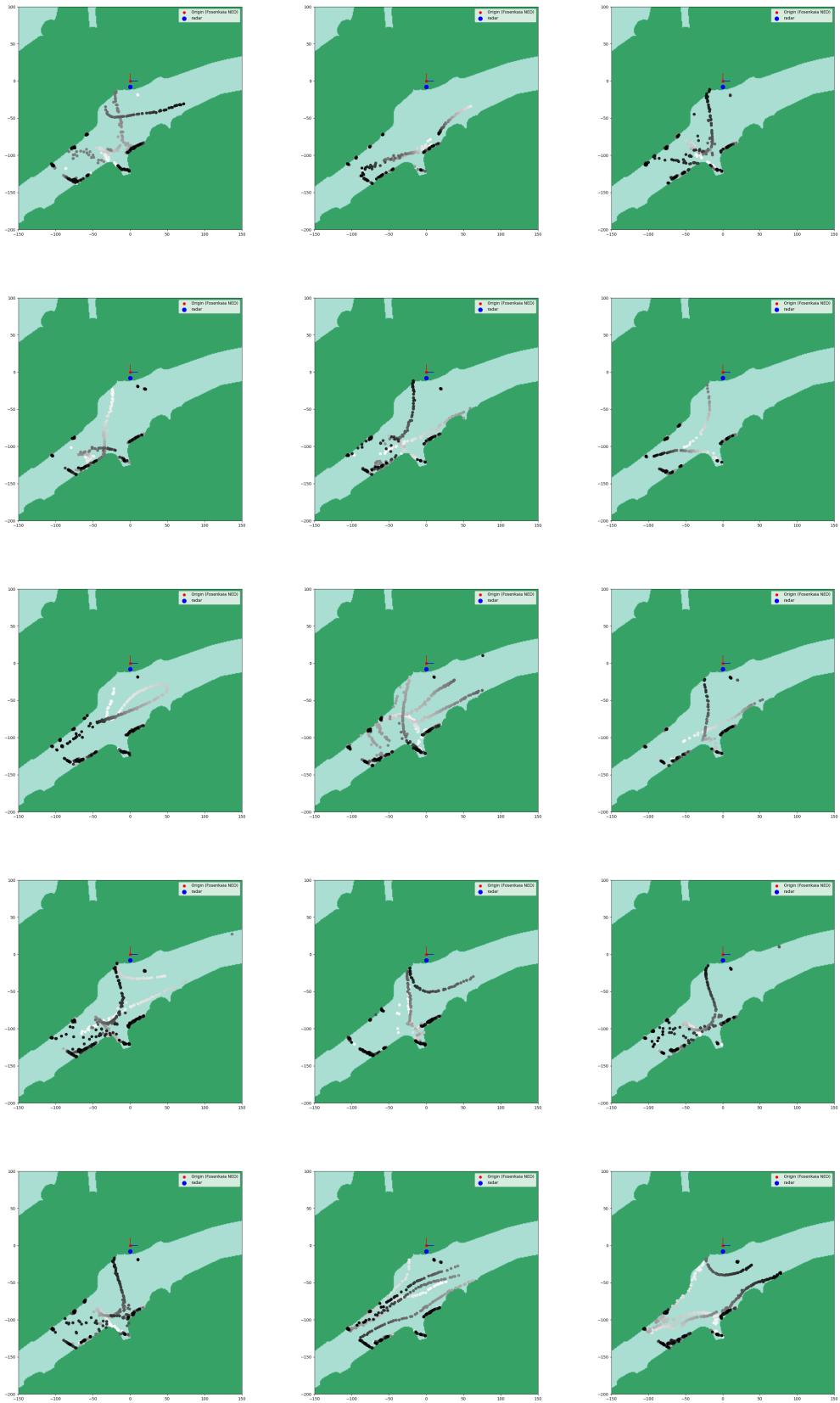
Two different variations of the VIMMJIPDA with spatially varying intensities were derived and tested. The Approximate model made several approximations to avoid the need for numerical integration. By precomputing, the Integral model allowed efficient accurate approximations of the numerical integrations needed by moving the extra computational cost ahead of time. However, this technique is not transferable to natural extensions, such as tracking on a moving platform. Over the evaluation set, the performance of the Approximate model proved almost identical to the Integral model. Averaging the birth-intensity in the Approximate model using a Gaussian kernel further closes the gap between the two trackers, making the Approximate model the natural choice.

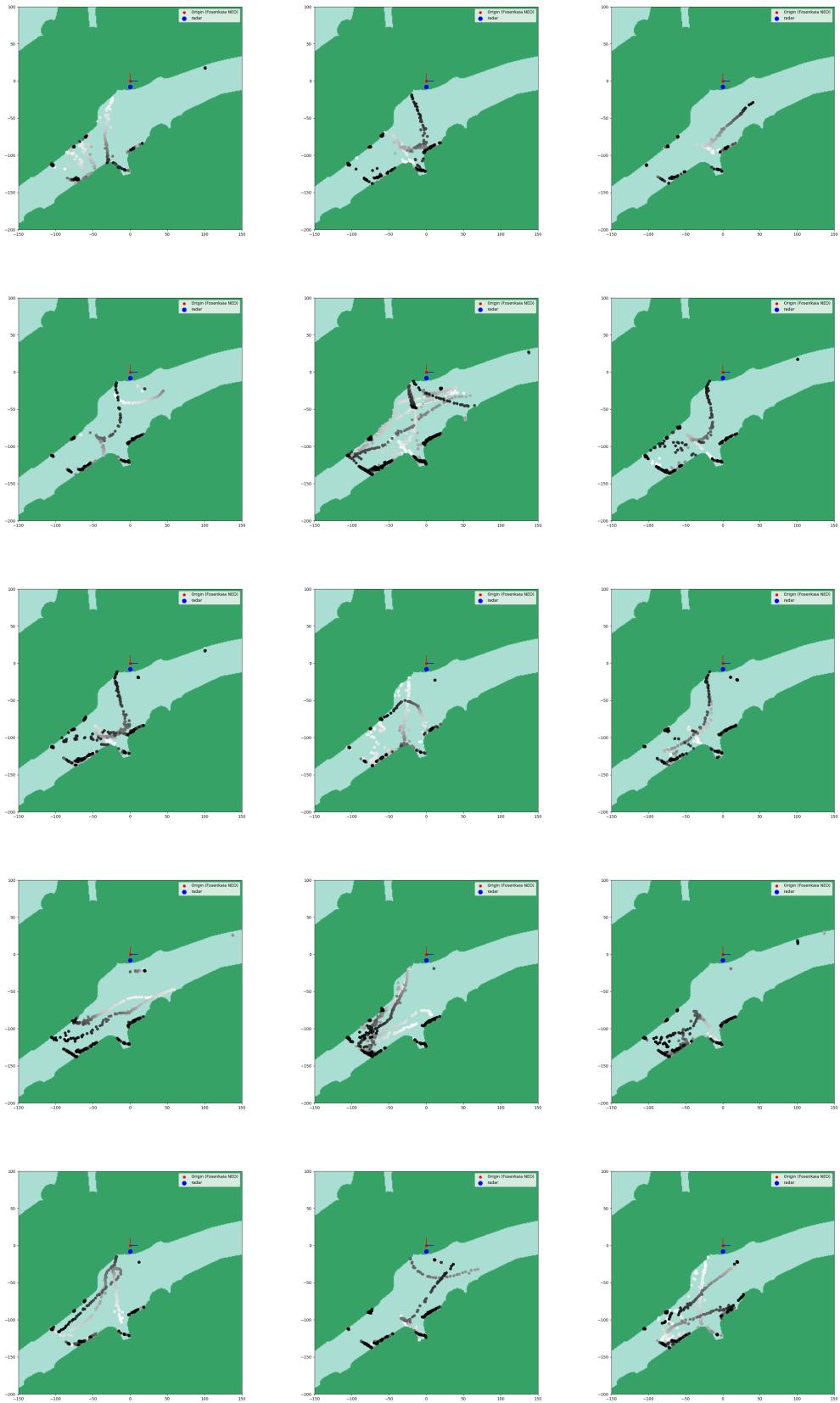
Appendix

A All scenarios used for evaluation









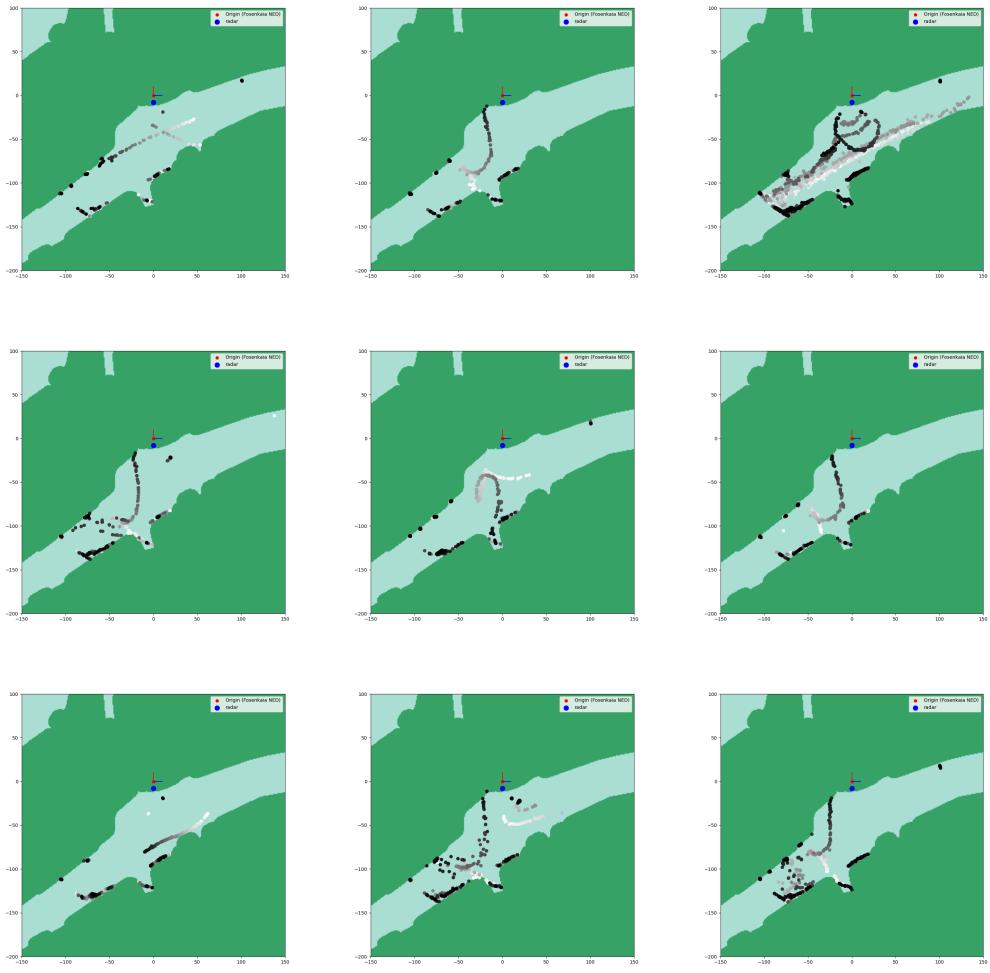


Figure 29: All scenarios used for evaluation of the trackers. Newer measurements are darker.

Bibliography

- [1] E. F. Brekke, A. G. Hem and L.-C. N. Tokle, ‘Multitarget tracking with multiple models and visibility: Derivation and verification on maritime radar data’, *IEEE Journal of Oceanic Engineering*, vol. 46, no. 4, pp. 1272–1287, 2021. DOI: 10.1109/JOE.2021.3081174.
- [2] S. Dissanayake and J. J. Lu, ‘Traffic control device comprehension: Differences between domestic and international drivers in USA’, *IATSS Research*, vol. 25, no. 2, pp. 80–87, 2001, ISSN: 0386-1112. DOI: [https://doi.org/10.1016/S0386-1112\(14\)60072-8](https://doi.org/10.1016/S0386-1112(14)60072-8). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0386111214600728>.
- [3] G. Yannis, J. Golias and E. Papadimitriou, ‘Accident risk of foreign drivers in various road environments’, *Journal of Safety Research*, vol. 38, no. 4, pp. 471–480, 2007, ISSN: 0022-4375. DOI: <https://doi.org/10.1016/j.jsr.2007.01.014>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022437507000850>.
- [4] E. F. Wilthil, E. Brekke and O. B. Asplin, ‘Track initiation for maritime radar tracking with and without prior information’, in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 1–8. DOI: 10.23919/ICIF.2018.8455260.
- [5] E. F. Brekke, *Fundamentals of Sensor Fusion*. n.p., 2023, Lecture notes in course TTK4250 Sensor Fusion at NTNU. Can be obtained by contacting the author.
- [6] J. L. Williams, ‘Marginal multi-bernoulli filters: RFS derivation of MHT, JIPDA, and association-based member’, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1664–1687, 2015. DOI: 10.1109/TAES.2015.130550.
- [7] M. A. Sirnes, ‘Joint localization and tracking: Experiments and preliminary work on active sensors’, M.S. thesis, Department of Engineering Cybernetics, 2020.
- [8] P. Jäckel, *A note on multivariate gauss-hermite quadrature*, <http://www.jaeckel.org/ANoteOnMultivariateGaussHermiteQuadrature.pdf>, 2005.
- [9] Wikipedia contributors, *Gauss–Hermite quadrature — Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/wiki/Gauss%20%93Hermite_quadrature, 2022.
- [10] Wikipedia contributors, *Matrix norm — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/wiki/Matrix_norm#Frobenius_norm, 2023.
- [11] S. Hexeberg, ‘AIS-based vessel movement prediction for asv collision avoidance’, M.S. thesis, Department of Engineering Cybernetics, 2016.
- [12] O. B. Asplin, ‘Near-shore target tracking with clutter maps’, M.S. thesis, Department of Engineering Cybernetics, 2018.
- [13] X. Chen, R. Tharmarasa, M. Pelletier and T. Kirubarajan, ‘Integrated bayesian clutter estimation with jipda/mht trackers’, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 395–414, 2013. DOI: 10.1109/TAES.2013.6404111.
- [14] D. Musicki, S. Suvorova, M. Morelande and B. Moran, ‘Clutter map and target tracking’, in *2005 7th International Conference on Information Fusion*, vol. 1, 2005. DOI: 10.1109/ICIF.2005.1591838.
- [15] J. L. Williams, ‘Hybrid poisson and multi-bernoulli filters’, in *2012 15th International Conference on Information Fusion*, 2012, pp. 1103–1110.