Deterministic Prior Sample Set Prediction Approaches for Newton-Flow Particle Filters

Sukkeun Kim and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
sukkeun.kim@kit.edu, uwe.hanebeck@kit.edu

***Abstract*—In Bayesian estimation, the prediction step (time propagation) is performed with a known noisy process model. We propose three approaches to perform the prediction step for deterministic sample sets to maximise the performance of Newton-Flow (NF) Particle Filters (PFs) in discrete-time stochastic dynamic systems. The evaluation in a radar target tracking scenario with discontinuity demonstrates significant improvements compared to the Extended Kalman Filter (EKF) and the PF. In addition, the evaluation provides insights into different prediction approaches.**

***Index Terms*—Bayesian state estimation, density estimation, Localised Cumulative Distribution (LCD), Newton-Flow (NF), Particle Filter (PF).**

I. Introduction

Particle flow-based filters are approaches for Bayesian estimation that migrate particles toward the posterior distribution instead of using importance weighting [1], [2], [3], [4]. The Newton-Flow (NF) Particle Filter (PF) is a state-of-the-art approach in this category, capable of precise posterior density estimation [5]. In our previous work, we demonstrated that the NF can estimate the posterior density with nonlinear measurements and multi-modal prior densities [1]. The NF was tested with both random and deterministic prior sample sets; the result showed that the update using deterministic prior samples yields significantly better performance. This improvement stems from the fact that deterministic samples can represent the state density better than random samples because the samples are optimally selected to represent the underlying distribution [6], [7], [8].

In dynamic systems with a process model, the situation is different because we need to propagate samples while accounting for the uncertainty introduced by process model noise. In other words, a prediction step that can yield a deterministic prior sample set is required to maximise the performance of the NF. However, the NF has not yet been fully investigated regarding the prediction step for dynamic systems. To fill this gap and to maximise the potential of the NF, we propose three approaches to perform the prediction step (time propagation) that yield deterministic prior sample sets. An overview of three proposed approaches is shown in Fig. 1. In addition, we propose a simple radar target tracking scenario with a discontinuity, or a "jump", which is similar to the kidnapped robot problem. The main contributions of this study are listed as follows:

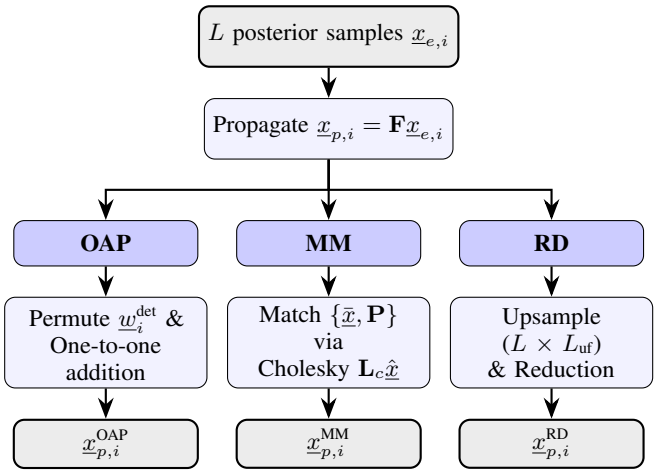

Fig. 1: **The three proposed prediction approaches for deterministic prior sample sets.** One-to-one addition with permutation (OAP, left), moment matching (MM, centre), and reduction from a deterministic upsampled set (RD, right).

1) Three prediction approaches for deterministic prior sample sets are proposed.
2) A simple radar target tracking scenario with discontinuity, "jump", is proposed.
3) The proposed multiple prediction approaches are tested and compared in the tracking scenario.

The rest of this paper is composed as follows: Section II formulates the dynamic system we are interested in. Section III proposes three prediction approaches for deterministic prior sample sets, and Section IV presents a comparative simulation result of the proposed approaches with other filters. Section V discusses the results and the characteristics of each approach. Finally, we conclude this study with future work in Section VI.

II. Problem Formulation

In a discrete-time dynamic system, a random N-dimensional state vector at time step k, $\underline{\boldsymbol{x}}_k \in \mathbb{R}^N$, is propagated using the deterministic prediction function $\underline{a}_k$ as shown in (1)[1]:

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}_k(\underline{\boldsymbol{x}}_k, \underline{u}_k, \underline{\boldsymbol{w}}_k) \ , \tag{1}$$

[1]Underlines denote vectors, small boldface indicates random variables and capital boldface indicates matrices. The subscripts p and e denote the predicted and estimated states, respectively.

where $\underline{\boldsymbol{x}}_k$ is the state vector at time step k, $\underline{u}_k$ is a control input, and $\underline{\boldsymbol{w}}_k$ is the system noise. This step, to propagate the state to the next time step $k+1$, is called the "prediction step". However, the state is usually hidden and is observable indirectly via measurements. The relation between the measurement vector $\underline{\boldsymbol{y}}_k$, with the hidden state, $\underline{\boldsymbol{x}}_k$, can be mapped using the deterministic measurement function $\underline{h}_k$ as shown in (2):

$$\underline{\boldsymbol{y}}_k = \underline{h}_k(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{v}}_k) \ . \tag{2}$$

The measurement update, or "filter step", uses Bayes' rule to estimate the state as follows:

$$f_k(\underline{x}_k|\mathbf{Z}_k) \propto f_k(\underline{z}_k|\underline{x}_k) \cdot f_k(\underline{x}_k|\mathbf{Z}_{k-1}) \ , \tag{3}$$

where $f_k(\underline{x}_k|\mathbf{Z}_k)$ is the posterior density of the state given measurements $\mathbf{Z}_k = \{\underline{z}_1, \underline{z}_2, \ldots, \underline{z}_k\}$, $f_k(\underline{z}_k|\underline{x}_k)$ is the likelihood function f_k^L with measurement $\underline{z}_k$, and $f_k(\underline{x}_k|\mathbf{Z}_{k-1})$ is the prior density of the state.

The NF is a state-of-the-art particle flow-based filter which can formulate posterior density better than classical approaches [5]. This approach outperforms classical methods, particularly in scenarios involving highly informative measurements. In such cases, standard PFs often suffer from particle degeneracy, a challenge that the NF can mitigate [1]. The main idea of NF is the progressive inclusion of the measurement, also called homotopy enabled using artificial time (or homotopy parameter), defined as $\gamma \in [0, 1]$. The progressive inclusion of the measurement can be realised by two main approaches – discrete progression steps, such as in [9] and continuous progression, which is the case for NF. The continuous homotopy equation can be formulated from (3) as shown below:

$$f_e(\underline{x}, \gamma) \propto f_L(\underline{x})^\gamma \cdot f_p(\underline{x}) \ . \tag{4}$$

The NF utilises infinitesimal steps of the flow so that the approximated density remains within the permissible space during the measurement update step with homotopy. The NF employs a reference density $\tilde{f}_e(\underline{x}, \gamma)$, which depends on the artificial time γ for these infinitesimal steps. In order to induce a "flow" that migrates the samples, the distance between the reference density and the true posterior density $f_e(\underline{x}, \underline{\eta}(\gamma))$, parametrised by $\underline{\eta}(\gamma)$, is defined as

$$D\big(\underline{\eta}(\gamma), \gamma\big) = D\Big(\tilde{f}_e\big(\underline{x}, \gamma\big), f_e\big(\underline{x}, \underline{\eta}(\gamma)\big)\Big) \ . \tag{5}$$

The distance utilised in the NF is the generalised Cramér–von Mises Distance (CvMD), which is based on the Localised Cumulative Distribution (LCD) defined in [10]. The change of the parameter of the posterior density can be obtained by differentiating the distance specified in (5) twice with respect to γ. This yields the "flow" of samples toward the posterior. Refer to [5], [1] for further details.

In order to apply the NF in dynamic systems, it is important to propagate the state; however, this prediction step has not been considered in the original paper [5] or the survey paper [1]. This step must be performed precisely, as the NF has been shown to perform better with a deterministic prior sample set than with a random prior sample set [1].

III. Prediction Approaches for Deterministic Prior Sample Sets

The primary challenge in the prediction step with a sample set is uncertainty propagation. In the case of Extended Kalman Filter (EKF), where the first two moments are tracked throughout the filtering steps, uncertainty is propagated via the state covariance. However, this cannot be directly applied in the case of sample-based filters such as PF. In these cases, the standard approach is random noise injection, which can result in clustered samples if the number of samples is too small. To tackle this challenge, we propose three approaches for the realisation of the prediction step that provide deterministic prior sample sets. We consider the additive zero-mean Gaussian noise case and no control input in this study. This simplifies (1) to

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}_k(\underline{\boldsymbol{x}}_k) + \underline{\boldsymbol{w}}_k \ . \tag{6}$$

A. One-to-One Addition with Permutation

The first approach is One-to-one Addition with Permutation (OAP). This approach is similar to the standard process noise injection of the PF but utilises a deterministic sample set. At the beginning of the process, a deterministic noise sample set with L samples is generated to account for the process noise covariance $\mathbf{Q}_w = \mathbf{G}(q^2\mathbf{I})\mathbf{G}^\top$, where $\mathbf{G}$ maps the noise to the state space. This is performed in two steps: 1) a standard Gaussian deterministic set is sampled using the LCD-based sampling [6], and 2) this set is transformed using the Cholesky factor $\mathbf{L}_c$, where $\mathbf{L}_c\mathbf{L}_c^\top = \mathbf{Q}_w + \epsilon\mathbf{I}$, ensuring the noise samples reflect the system uncertainty q. In every prediction step, the state samples are propagated using (6), and each noise sample is added to a corresponding propagated state sample (one-to-one addition). In subsequent time steps, a random permutation is performed on the noise sample set to ensure variability. A visualisation of the steps for the OAP approach is shown in Fig. 2 and the pseudocode is provided in Algorithm 1.

Algorithm 1 One-to-one addition with permutation

Initialisation:
Compute process noise covariance: $\mathbf{Q}_w \leftarrow \mathbf{G}\big(q^2\mathbf{I}\big)\mathbf{G}^\top$
Ensure symmetry: $\mathbf{Q}_w \leftarrow \frac{1}{2}\big(\mathbf{Q}_w + \mathbf{Q}_w^\top\big)$
Compute Cholesky factor $\mathbf{L}_c$ such that $\mathbf{L}_c\mathbf{L}_c^\top = \mathbf{Q}_w + \epsilon\mathbf{I}$
Sample L standardised noise samples $\underline{\hat{w}}_i$ via LCD [6]
Generate noise samples: $\underline{w}_i^{\text{det}} \leftarrow \mathbf{L}_c\underline{\hat{w}}_i$
▷ Noise samples now considers q
In each prediction step k**:**
Input: Posterior samples $\underline{x}_{e,i}$, noise samples $\underline{w}_i^{\text{det}}$
Propagate samples: $\underline{x}_{p,i} \leftarrow \mathbf{F}\underline{x}_{e,i}$
if $k > 1$ **then** ▷ Permute noise for variability
    $\pi \leftarrow$ random permutation of $\{1, \ldots, L\}$
    $\underline{w}_i^{\text{eff}} \leftarrow \underline{w}_{\pi(i)}^{\text{det}}$
else
    $\underline{w}_i^{\text{eff}} \leftarrow \underline{w}_i^{\text{det}}$
end if
$\underline{x}_{p,i}^{\text{OAP}} \leftarrow \underline{x}_{p,i} + \underline{w}_i^{\text{eff}}$ ▷ One-to-one addition
return $\underline{x}_{p,i}^{\text{OAP}}$

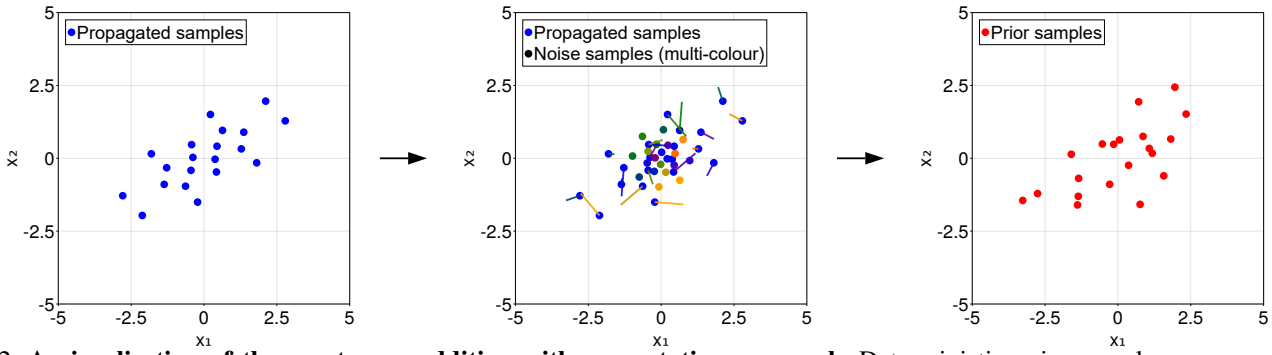

Fig. 2: **A visualisation of the one-to-one addition with permutation approach.** Deterministic noise samples are permuted and added to each propagated state sample (one-to-one addition). The multi-colour lines rooting from the propagated state samples indicate which noise sample was added and the resulting direction.

Although this approach behaves similarly to random noise injection in the short term, in the long term, it is closer to optimal than random additive noise. This approach is suboptimal but has the advantage of being fast and easy to implement. In addition to OAP, the Additive Random (AR) noise case will be considered in the numerical simulation.

B. Moment Matching

The second approach is Moment Matching (MM). This approach is similar to the standard noise covariance propagation of the EKF but utilises a deterministic sample set. At the beginning of the process, a deterministic sample set with L samples, which will serve as the prior set, is generated using LCD-based deterministic sampling [6]. In every prediction step, the state samples are propagated using (6), and then, the mean and the covariance, with the process noise $\mathbf{Q}_w = \mathbf{G}(q^2\mathbf{I})\mathbf{G}^\top$ injected, are approximated from the samples as follows:

$$\bar{\underline{x}} = \frac{1}{L}\sum_{i=1}^{L} \underline{x}_{p,i} \ , \tag{7}$$

$$\mathbf{P} = \frac{1}{L-1}\sum_{i=1}^{L} \left(\underline{x}_{p,i} - \bar{\underline{x}}\right)\left(\underline{x}_{p,i} - \bar{\underline{x}}\right)^\top + \mathbf{G}(q^2\mathbf{I})\mathbf{G}^\top \ . \tag{8}$$

Then, we perform MM to match the mean and covariance of the pre-sampled set to those in (7) and (8), respectively. This is performed using the Cholesky factor $\mathbf{L}_c$: $\underline{x}_{p,i}^{\text{MM}} = \bar{\underline{x}} + \mathbf{L}_c\hat{\underline{x}}$. Since the LCD-based samples are invariant to translation and scaling, the moment-matched sample set remains fully deterministic under the Gaussian assumption. A visualisation of the steps for the MM approach is shown in Fig. 3 and the pseudocode is provided in Algorithm 2.

C. Reduction from a Deterministic Upsampled Set

The third approach is Reduction from a Deterministic upsampled set (RD). This approach increases the number of samples (upsampling) by adding every noise sample from a deterministic set to each propagated state sample, subsequently reducing the set back to the original size. At the beginning of the process, we follow the same sampling strategy as OAP but generate a smaller set of L_{uf} (upsampling factor) noise samples.

Algorithm 2 Moment matching

Initialisation:
Sample standardised Gaussian deterministic samples $\hat{\underline{x}}$ using LCD [6]
In each prediction step k**:**
Input: Posterior samples $\underline{x}_{e,i}$, process noise std q, mapping matrix $\mathbf{G}$, deterministic samples $\hat{\underline{x}}$
Propagate samples: $\underline{x}_{p,i} \leftarrow \mathbf{F}\underline{x}_{e,i}$
Compute sample statistics:
$\bar{\underline{x}} \leftarrow \frac{1}{L}\sum_{i=1}^{L}\underline{x}_{p,i}$ ▷ Sample mean
$\mathbf{P} \leftarrow \frac{1}{L-1}\sum_{i=1}^{L}\left(\underline{x}_{p,i} - \bar{\underline{x}}\right)\left(\underline{x}_{p,i} - \bar{\underline{x}}\right)^\top + \mathbf{G}\left(q^2\mathbf{I}\right)\mathbf{G}^\top$ ▷ Covariance with noise
Moment matching via Cholesky:
Ensure symmetry: $\mathbf{P} \leftarrow \frac{1}{2}\left(\mathbf{P} + \mathbf{P}^\top\right)$
Compute Cholesky factor $\mathbf{L}_c$ such that $\mathbf{L}_c\mathbf{L}_c^\top = \mathbf{P} + \epsilon\mathbf{I}$ ▷ Cholesky decomposition
Transform deterministic samples:
$\underline{x}_{p,i}^{\text{MM}} \leftarrow \bar{\underline{x}} + \mathbf{L}_c\hat{\underline{x}}$ ▷ Re-instantiate samples
return $\underline{x}_{p,i}^{\text{MM}}$

In every prediction step, the state samples are propagated using (6), and all L_{uf} noise samples are added to each of the L propagated state samples, resulting in an upsampled set of $L \times L_{\text{uf}}$ samples. Following this, we perform LCD-based optimal reduction [11] to reduce the $L \times L_{\text{uf}}$ samples back to the original number L. For example, for an original $L = 300$ and upsampling factor $L_{\text{uf}} = 5$, the set of $L \times L_{\text{uf}} = 1500$ samples is reduced back to $L = 300$. The reduction is performed by minimising the generalised CvMD between the LCDs of the upsampled high-resolution density, $\tilde{F}(\underline{m}, b)$ and the new, lower-resolution density, $F(\underline{m}, b)$, as shown below:

$$D = \int_{\mathbb{R}_+} w(b) \int_{\mathbb{R}^N} \left(\tilde{F}(\underline{m}, b) - F(\underline{m}, b)\right)^2 \, d\underline{m}\, db \ , \tag{9}$$

where $w(b) : \mathbb{R}_+ \to [0, 1]$ is a suitable weighting function. Refer to [11] for further details on the optimal reduction. A visualisation of the steps for the RD approach is shown in Fig. 4 and the pseudocode is provided in Algorithm 3.

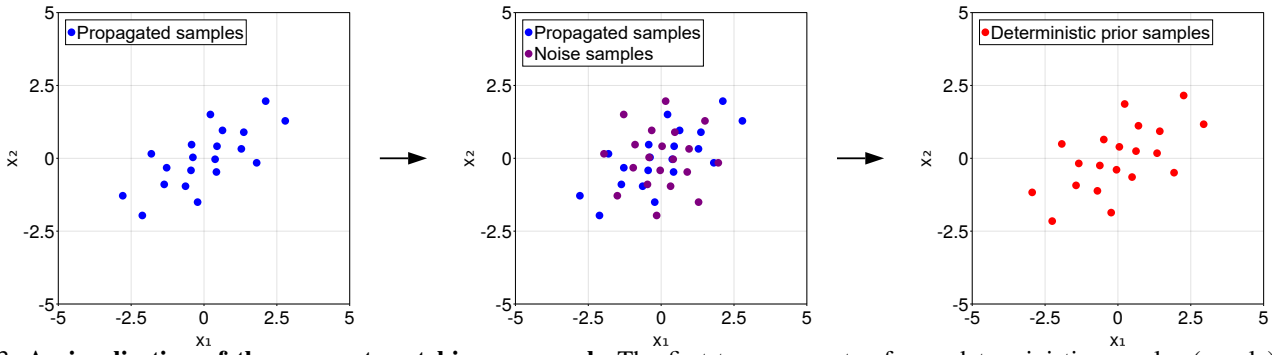

Fig. 3: **A visualisation of the moment matching approach.** The first two moments of new deterministic samples (purple) are matched with those of the propagated state samples (blue) by accounting for the process noise covariance.

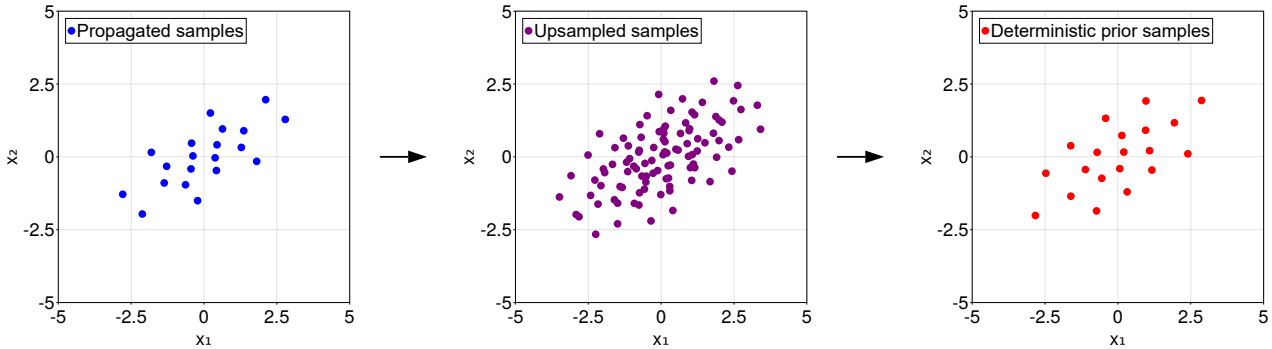

Fig. 4: **A visualisation of the reduction from a deterministic upsampled set approach.** The propagated state samples are upsampled with L_{uf} deterministic noise samples (purple) and then reduced back to L samples using optimal reduction.

Algorithm 3 Reduction from a deterministic upsampled set

Initialisation:
Compute process noise covariance: $\mathbf{Q}_w \leftarrow \mathbf{G}(q^2\mathbf{I})\mathbf{G}^\top$
Ensure symmetry: $\mathbf{Q}_w \leftarrow \frac{1}{2}\left(\mathbf{Q}_w + \mathbf{Q}_w^\top\right)$
Compute Cholesky factor $\mathbf{L}_c$ such that $\mathbf{L}_c\mathbf{L}_c^\top = \mathbf{Q}_w + \epsilon\mathbf{I}$
Sample L_{uf} standardised noise samples $\underline{\hat{w}}_j$ via LCD [6]
Generate upsampling noise samples: $\underline{w}_j^{\text{uf}} \leftarrow \mathbf{L}_c\underline{\hat{w}}_j$
▷ Noise samples now considers q
In each prediction step k**:**
Input: Posterior samples $\underline{x}_{e,i}$, noise samples $\underline{w}_j^{\text{uf}}$
Propagate samples: $\underline{x}_{p,i} \leftarrow \mathbf{F}\underline{x}_{e,i}$
Upsampling step:
Construct upsampled set $\underline{x}_{i,j}^{\text{up}}$ of size $L \times L_{\text{uf}}$:
$\underline{x}_{i,j}^{\text{up}} = \{\underline{x}_{p,i}^- + \underline{w}_j^{\text{uf}} \mid i \in \{1,\ldots,L\}, j \in \{1,\ldots,L_{\text{uf}}\}\}$
Optimal reduction:
Assign uniform weights: $w_y \leftarrow 1/(L \cdot L_{\text{uf}})$
$\underline{x}_{p,i}^{\text{RD}} \leftarrow \text{Reduction}(w_y, \underline{x}_{i,j}^{\text{up}}, L)$ [11]
▷ Reduce back to original size L
return $\underline{x}_{p,i}^{\text{RD}}$

This approach provides an optimal solution at the cost of increased computational load. Two approaches will be considered in the numerical example: Reduction Random (RR) where upsampling is done using random noise samples $\mathbf{G}\underline{w}_j$ with $\underline{w}_j \sim \mathcal{N}(\underline{0}, q^2\mathbf{I})$ and RD where upsampling is done using a pre-sampled deterministic noise samples $\underline{w}_j^{\text{uf}}$.

IV. Comparative Numerical Simulation

In this section, we propose a simple radar target tracking scenario with a discontinuity. Such discontinuities are common in real-world applications and are similar to the well-known kidnapped robot problem [12]. The target trajectory is divided into three segments: 1) the target moves in two-dimensional (2D) space following the Constant Acceleration (CA) model, 2) it "jumps" to a new position, and 3) it continues following the CA model after the jump with a new initial velocity and acceleration. This scenario makes tracking significantly more complicated and produces a small overlap between the prior density and the likelihood. This trajectory and the jump are illustrated in Fig. 5.

The state vector $\underline{x}_k = [x_k, \dot{x}_k, \ddot{x}_k, y_k, \dot{y}_k, \ddot{y}_k]^\top$, comprising position, velocity, and acceleration in 2D, is propagated using a CA model as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{F}_1 \end{bmatrix}, \text{ where } \mathbf{F}_1 = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

and the process noise mapping matrix $\mathbf{G}$ maps the 2D jerk noise into the six-dimensional (6D) state space:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & \underline{0}_{3\times1} \\ \underline{0}_{3\times1} & \mathbf{G}_1 \end{bmatrix} \text{ where } \mathbf{G}_1 = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}. \quad (11)$$

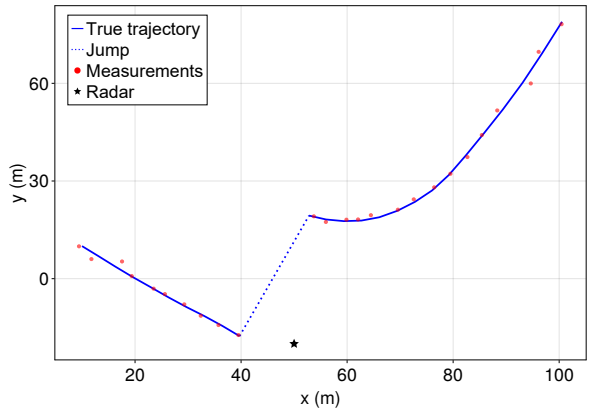

Fig. 5: **True trajectory of the radar target tracking scenario.** The target follows the CA model starting from the left-hand side for the first 10 time steps. Afterward, it "jumps" (shown with a dashed line) to a new position and continues following the CA model with a new initial velocity and acceleration.

For the EKF, the analytic covariance is updated using the discrete-time process noise covariance $\mathbf{Q}_w$:

$$\mathbf{P}_{k+1} = \mathbf{F}\mathbf{P}_k\mathbf{F}^\top + \mathbf{Q}_w, \quad \text{where} \quad \mathbf{Q}_w = \mathbf{G}(q^2\mathbf{I}_{2\times 2})\mathbf{G}^\top \ , \tag{12}$$

and stochastic sample prediction for the PF is performed via:

$$\underline{x}_{k+1,i} = \mathbf{F}\underline{x}_{k,i} + \mathbf{G}\underline{w}_{k,i} \ , \tag{13}$$

where $\underline{w}_{k,i} \sim \mathcal{N}\big(\underline{0}, q^2\mathbf{I}_{2\times 2}\big)$ represents the stochastic noise samples. For NF, the proposed approaches in Section III are used.

The measurement at time step k is defined as $\underline{z}_k = [r_{m,k}, \theta_{m,k}]^\top$, representing the measured range and azimuth from the radar located at (x_r, y_r). The mapping from the state $\underline{x}_k$ to the measurement space is given by $h(\underline{x}_k)$ as follows:

$$h(\underline{x}_k) = \begin{bmatrix} r_{p,k} \\ \theta_{p,k} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - x_r)^2 + (y_k - y_r)^2} \\ \operatorname{atan}(y_k - y_r, x_k - x_r) \end{bmatrix} \ , \tag{14}$$

where x_k and y_k denote the Cartesian position of the target at time step k. The Gaussian likelihood f_L for a given state is computed as follows:

$$f_L(\underline{z}_k|\underline{x}_k) \propto \exp\left(-\frac{1}{2}\left[\frac{(r_{m,k} - r_{p,k})^2}{\sigma_r^2} + \frac{(\theta_{m,k} \ominus \theta_{p,k})^2}{\sigma_\theta^2}\right]\right) \ , \tag{15}$$

where $\ominus$ denotes the subtraction operator wrapped to the interval $[-\pi, \pi]$ to account for angular periodicity.

The simulation is conducted over 25 time steps with an interval of $T = 1\,\mathrm{s}$. The true trajectory is initialised with the initial state $\underline{x}_0 = [10.0, 10.0, 3.0, -3.0, 0.02, 0.05]^\top$, where the values represent the position, velocity, and acceleration for the x and y dimensions, respectively. The corresponding units are m, $\mathrm{m\,s^{-1}}$, and $\mathrm{m\,s^{-2}}$. The true trajectory is generated by following a CA model with process noise standard deviation $q = 0.3\,\mathrm{m\,s^{-2}}$. At time step $k = 10$, a discrete state jump is introduced by incrementing the position by $[\Delta x, \Delta y]^\top = [10.0, 40.0]^\top\,\mathrm{m}$. Post-jump, the motion parameters are modified to $\ddot{x} = 0.1\,\mathrm{m\,s^{-2}}$, $\dot{y} = -1.5\,\mathrm{m\,s^{-1}}$, and $\ddot{y} = 1.0\,\mathrm{m\,s^{-2}}$ to simulate an accelerated movement toward the northeast. The radar is stationary at $(x_r, y_r) = (50.0, -20.0)\ \mathrm{m}$ for the duration of the scenario. The measurements are assumed to be corrupted by additive zero-mean Gaussian noise $\underline{v}_k \sim \mathcal{N}(\underline{0}, \mathbf{R})$, with a diagonal covariance matrix, $\mathbf{R} = \operatorname{diag}\big(\sigma_r^2, \sigma_\theta^2\big)$ where the standard deviations are $\sigma_r = 10.0\,\mathrm{m}$ and $\sigma_\theta = 2^\circ$. For the evaluated filters, the initial state estimate $\hat{\underline{x}}_0$ is set using the first measurement, with an initial estimation error covariance set to $\mathbf{P}_0 = \mathbf{I}_{6\times 6}$. To account for model uncertainty, the process noise standard deviation is set to $q = 0.5\,\mathrm{m\,s^{-2}}$. The PF is tested with $L = 3000$ samples, while all implementations of the NF are tested with $L = 300$ samples. For NF, a deterministic sample set is used for the initialisation, and the upsampling factor L_{uf} for RR and RD is set to $L_{\text{uf}} = 5$.

The NF variants, incorporated with proposed prediction approaches, are compared against the EKF and PF in the proposed scenario. It is worth noting that the CA model is known to be a suboptimal solution in similar scenarios, and an Interacting Multiple-Model (IMM)-EKF would typically perform better than a standard EKF. However, we do not consider the IMM-EKF in this comparison because the main contribution of this study focuses on the error propagation for the NF and intrinsic filtering performance within a dynamic system, rather than the performance gain provided by multiple-model frameworks. All filters are evaluated via a Monte Carlo (MC) simulation with 1000 runs to randomise the process and measurement noises. The simulations are performed using Julia on a laptop equipped with a 2.20 GHz Intel Core Ultra 7 CPU and 32 GB of RAM. The tracking trajectories of the first MC run are shown in Figs. 6 to 12. For evaluation, the position root mean square error (RMSE) is used as defined below:

$$\text{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}[(x_k^{\text{true}} - \bar{x}_{e,k})^2 + (y_k^{\text{true}} - \bar{y}_{e,k})^2]} \ , \tag{16}$$

where K is the total duration of one MC run, and $\bar{x}_{e,k}$ and $\bar{y}_{e,k}$ are the estimated means of the 2D position. For the final evaluation, the average RMSE (ARMSE) is calculated by averaging the RMSE of each MC run. In addition, the average time per single prediction step (prediction time) and the average computation time per MC run (run time) are evaluated, with the results summarised in Table I.

TABLE I: Performance comparison of EKF, PF, and NF with multiple prediction approaches for 1000 MC runs.

Method	No. of Particles	ARMSE (m)	Prediction Time (ms)	Run Time (s)
EKF	—	31.7531	**0.0058**	**0.0012**
PF	3000	76.0016	0.1479	0.0111
NF-AR		21.2955	0.0732	8.9768
NF-OAP		**19.0510**	0.1094	8.8460
NF-MM	300	24.4100	0.1907	6.5874
NF-RR		21.6168	6447.5334	161.6601
NF-RD		23.8318	5488.4503	138.8228

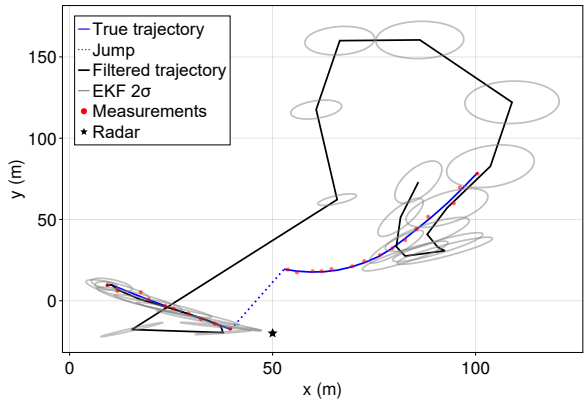

Fig. 6: **Tracking results of the EKF displaying the filtered trajectory and 2σ contours.** The EKF fails to track the true trajectory after the "jump".

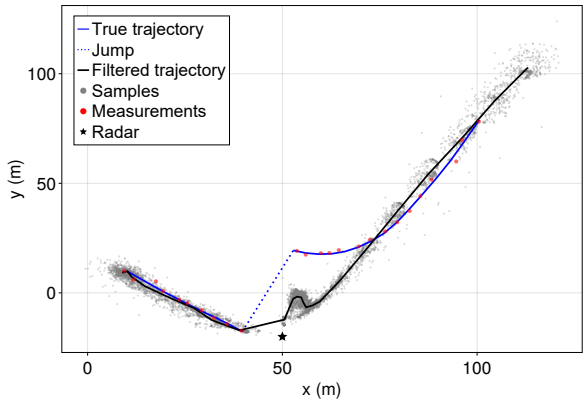

Fig. 8: **Tracking results of the NF-AR displaying the filtered trajectory and samples.** The filter loses the true track after the "jump" and shows error while tracking the true trajectory.

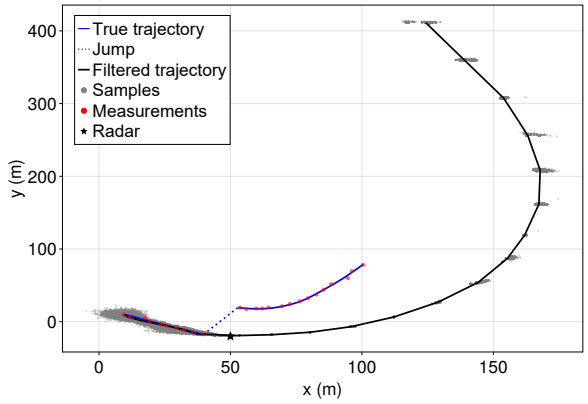

Fig. 7: **Tracking results of the PF displaying the filtered trajectory and samples.** The PF fails to track the true trajectory after the "jump" and undergoes divergence.

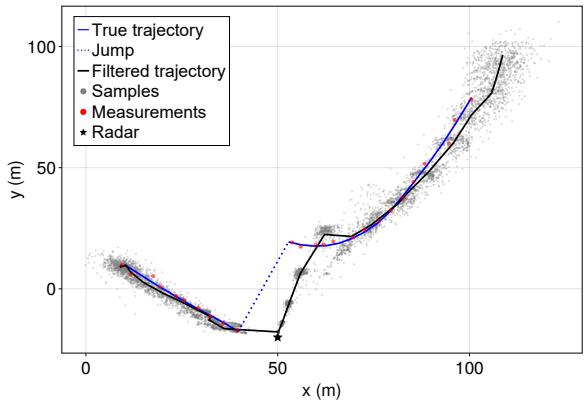

Fig. 9: **Tracking results of the NF-OAP displaying the filtered trajectory and samples.** The filter experiences a moderate tracking error for several steps following the "jump" before converging back to the true trajectory.

V. Discussion

From the simulation, all NF variants incorporated with the proposed prediction approaches outperform the EKF and PF, which indicates that the NF can handle scenarios with discontinuity better than the compared filters. In detail, the one-to-one addition with permutation (NF-OAP) yields the smallest ARMSE, followed by the additive random noise (NF-AR) and the reduction from a random upsampled set (NF-RR). On the other hand, the NF variants incorporated with fully deterministic approaches, such as the moment matching (NF-MM) and the reduction from a deterministic upsampled set (NF-RD), yield similar but slightly larger ARMSE compared to the other proposed approaches. This result provides the very interesting insight that a method which is close to deterministic in the long term but exhibits behaviour similar to randomness in the short term, as is the case with the NF-OAP, can achieve a smaller ARMSE than fully deterministic approaches. Furthermore, the randomness in the NF-AR and NF-RR also leads to a smaller ARMSE compared to the fully deterministic approaches.

One possible explanation for this tendency is that the introduction of "randomness" helps to mitigate the discontinuity considered in this scenario compared to fully deterministic approaches, such as the NF-MM and NF-RD. If we compare the NF-RR and NF-RD side-by-side, both utilise optimal reduction to produce a deterministic sample set that best represents the density. However, upsampling with random samples (NF-RR) achieves a smaller ARMSE than using deterministic samples (NF-RD), which suggests the introduction of randomness can help achieve a smaller ARMSE. By introducing this randomness, even if it results in a less-than-optimal representation of the density, the filter may experience the same effect as increasing the process noise in the prediction step. Nevertheless, two caveats should be noted here: 1) the introduction of randomness may only be helpful in cases where "jump" exist, and 2) a smaller ARMSE does not necessarily mean that randomness improves the representation of the underlying density. The second statement, the relation between

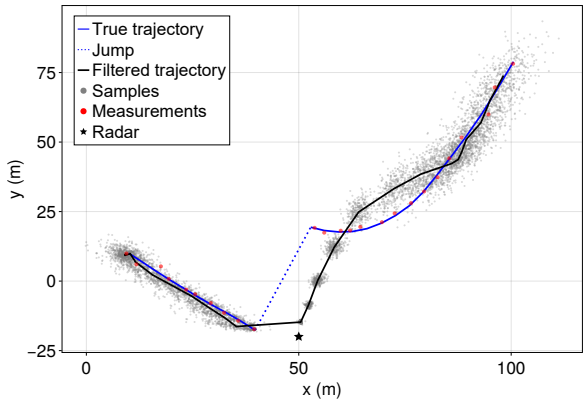

Fig. 10: **Tracking results of the NF-MM displaying the filtered trajectory and samples.** The filter experiences a tracking error for several steps following the "jump".

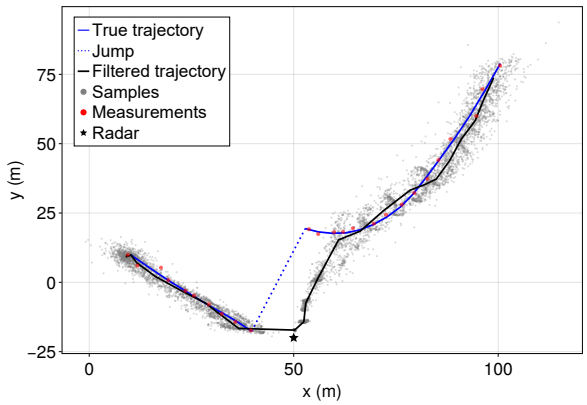

Fig. 11: **Tracking results of the NF-RR displaying the filtered trajectory and samples.** The filter experiences a moderate tracking error for several steps following the "jump" but successfully tracks the true trajectory.

ARMSE and the underlying density representation, is somewhat analogous to the comparison between EKF and Unscented Kalman Filter (UKF). One might observe that the EKF produces a smaller ARMSE while simultaneously underestimating the uncertainty compared to the UKF. This highlights the necessity of investigating different scenarios and introducing additional metrics that can better evaluate uncertainty.

Computation time is another critical factor, particularly for future real-time applications. In this study, the average time per single prediction step (prediction time) measures only the iterative prediction step, excluding the filter step and initial deterministic sample generation. In addition, the average computation time per MC run (run time) measures the total execution time from the first measurement through the end of the simulation, incorporating all prediction and filter steps but excluding the initial sample generation. As expected, the EKF shows the lowest prediction time and the lowest run time. Among the NF variants, the NF-AR exhibits

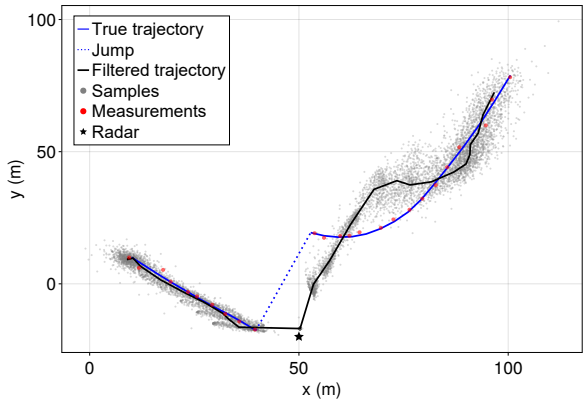

Fig. 12: **Tracking results of the NF-RD displaying the filtered trajectory and samples.** The filter loses the true track after the "jump" and shows error while tracking the true trajectory.

the lowest prediction time, while the NF-MM demonstrates the lowest run time. The NF-OAP achieves the second-lowest time for both metrics among the proposed approaches. In contrast, the two approaches utilising optimal reduction, the NF-RR and NF-RD, are multiple orders of magnitude slower in both metrics, primarily due to the internal optimisation required for the reduction step. Consequently, while the other filters are shown to be applicable for real-time applications, these two variants are unsuitable for such constraints. It is worth noting that although all NF variants utilise the exact same filter step, the ordinary differential equation (ODE) solving time varies significantly depending on the prior sample set. This can result in a different tendency for the run time compared to that of the prediction time. For instance, among the variants that do not use optimal reduction, the NF-AR shows the lowest prediction time but the highest run time. Conversely, the NF-MM shows the highest prediction time but the lowest run time.

VI. Conclusion

This study proposed three approaches for producing deterministic prior sample sets during the prediction step: 1) One-to-one Addition with Permutation (OAP), 2) Moment Matching (MM), and 3) Reduction from a Deterministic upsampled set (RD). The numerical simulation demonstrated that the NF variants incorporated with these proposed approaches outperformed classical methods in a radar target tracking scenario involving discontinuity. Furthermore, these multiple prediction approaches showed that the performance of the NF depends on the prior density. Specifically, the NF-OAP exhibited the smallest ARMSE in the tracking scenario, significantly outperforming the EKF and PF. While this study provided a baseline, some parameters were not fully investigated, and the evaluation was limited to a single scenario. Future research will extend this work through an ablation study, a sensitivity analysis, and evaluations across diverse, high-dimensional scenarios using additional metrics.

ACKNOWLEDGEMENT

Sukkeun Kim would like to thank Dr.-Ing. Daniel Frisch and Dominik Prossel for their valuable discussions.

REFERENCES

[1] Sukkeun Kim and Uwe D. Hanebeck. "The Hitchhiker's Guide to Particle Flow-based Filters: A Survey". 2026.

[2] Uwe D. Hanebeck and Martin Pander. "Progressive Bayesian estimation with deterministic particles". In: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, pp. 2028–2034.

[3] Tao Yang, Prashant G. Mehta, and Sean P. Meyn. "Feedback Particle Filter". In: *IEEE Transactions on Automatic Control* 58.10 (2013), pp. 2465–2480.

[4] Frederick Daum and Jim Huang. "Nonlinear filters with particle flow induced by log-homotopy". In: *Signal Processing, Sensor Fusion, and Target Recognition XVIII*. Ed. by Ivan Kadar. Vol. 7336. International Society for Optics and Photonics. SPIE, 2009, p. 733603.

[5] Uwe D. Hanebeck. "Newton-Flow Particle Filters based on Generalized Cramér Distance". In: *arXiv preprint: 2509.00182* (Aug. 2025). DOI: 10.48550/arXiv.2509.00182. URL: https://arxiv.org/abs/2509.00182.

[6] Uwe D. Hanebeck. "Deterministic Sampling of Multivariate Densities based on Projected Cumulative Distributions". In: *Proceedings of the 54th Annual Conference on Information Sciences and Systems (CISS 2020)*. Princeton, New Jersey, USA, Mar. 2020. DOI: 10.1109/CISS48834.2020.1570617413. URL: https://ieeexplore.ieee.org/document/9086228/.

[7] Daniel Frisch and Uwe D. Hanebeck. "Deterministic Gaussian Sampling With Generalized Fibonacci Grids". In: *Proceedings of the 24th International Conference on Information Fusion (Fusion 2021)*. Sun City, South Africa, Nov. 2021. DOI: 10.23919/FUSION49465.2021.9626975. URL: https://ieeexplore.ieee.org/document/9626975/.

[8] Daniel Frisch and Uwe D. Hanebeck. "The Generalized Fibonacci Grid as Low-Discrepancy Point Set for Optimal Deterministic Gaussian Sampling". In: *Journal of Advances in Information Fusion* 18.1 (June 2023), pp. 16–34. ISSN: 1557-6418. URL: https://isif.org/media/generalized-fibonacci-grid-low-discrepancy-point-set-optimal-deterministic-gaussian-sampling.

[9] Uwe D. Hanebeck. "PGF 42: Progressive Gaussian Filtering with a Twist". In: *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*. Istanbul, Turkey, July 2013.

[10] Uwe D. Hanebeck and Vesa Klumpp. "Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance". In: *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*. Seoul, Republic of Korea, Aug. 2008, pp. 33–39. DOI: 10.1109/MFI.2008.4648104. URL: http://ieeexplore.ieee.org/document/4648104/.

[11] Uwe D. Hanebeck. "Optimal Reduction of Multivariate Dirac Mixture Densities". In: *arXiv preprint: Systems and Control (cs.SY)* (Nov. 2014). DOI: 10.48550/arXiv.1411.4586. URL: https://arxiv.org/abs/1411.4586.

[12] S.P. Engelson and D.V. McDermott. "Error correction in mobile robot map learning". In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*. 1992, 2555–2560 vol.3. DOI: 10.1109/ROBOT.1992.220057.