Interval Particle Filter for LiDAR-Based Object Tracking

Towards Robust Tracking via Probabilistic and Interval Methods

Mohamed FNADI and Régis LHERBIER

LISIC – UR4491 – Université du Littoral Côte d'Opale

mohamed.fnadi@univ-littoral.fr

SCAN 2025 Oldenburg – Germany

September 22-26, 2025

Research Context and Objectives

Floating debris and marine waste

Small vessels and navigation hazards

Scientific Challenges

- **Diverse targets**: Various floating objects (debris, nets, vessels) with different properties
- **Harsh environments**: Dynamic marine conditions with waves and weather effects
- **Perception Sensors Uncertainties**: Noisy LiDAR measurements

Research Context and Objectives

Floating debris and marine waste

Small vessels and navigation hazards

Research Objectives

- Develop **robust LiDAR-based detection** for floating objects in marine environments
- Implement **interval-based tracking** for reliable trajectory estimation of floating objects

LiDAR-based Detection and Tracking of Floating Objects

LiDAR Data → Floating Object Detection (not treated in work)

Ouster OS1-128

Velodyne HDL-32

Floating Object Detection
(not treated in this work)

↓

Interval Particle Filter
Robust Tracking

↓

Target Trajectography

Outline

LiDAR-Based Detection: Principles and Models

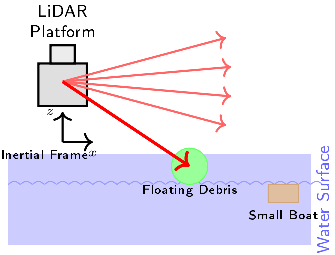
LiDAR Measurement Parameters

- **Distance** (r): Radial distance to object barycenter
- **Orientation** (ϕ): Relative orientation between LiDAR and object frame

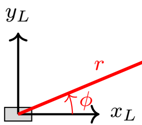
Key Assumptions

- Object detection available at each sampling interval
- LiDAR sensor position known and fixed in inertial frame
- Nonlinear uncertain system dynamics and observation models
- Measurements subject to bounded uncertainties:
$$[\mathbf{Y}_k] = [[\boldsymbol{r}_k]\ ,\ [\boldsymbol{\phi}_k]]$$
$$= [[r_k - \Delta r_k, r_k + \Delta r_k]\ ,\ [\phi_k - \Delta\phi_k, \phi_k + \Delta\phi_k]]$$

Marine Detection Scenario



Coordinate System



Bayes Filter: Prediction and Correction

Prediction Relation: Chapman-Kolmogorov equation

$$
\begin{aligned}
\texttt{pred}(\boldsymbol{x}_k) &= p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k-1}) \\
&= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{0:k-1})\, d\boldsymbol{x}_{k-1} \quad \text{(total prob.)} \\
&= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot \texttt{bel}(\boldsymbol{x}_{k-1})\, d\boldsymbol{x}_{k-1} \quad \text{(def. of bel)}
\end{aligned}
$$

Upadte

$$
\begin{aligned}
\texttt{bel}(\boldsymbol{x}_k) &= p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k}) \\
&= p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{y}_{0:k-1}) \quad \text{(Markov)} \\
&= \frac{1}{p(\boldsymbol{y}_k|\boldsymbol{y}_{0:k-1})}\, p(\boldsymbol{y}_k|\boldsymbol{x}_k) \cdot p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k-1}) \quad \text{(Bayes)} \\
&= \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k) \cdot \texttt{pred}(\boldsymbol{x}_k)}{p(\boldsymbol{y}_k|\boldsymbol{y}_{0:k-1})} \quad \text{(def. of pred)}
\end{aligned}
$$

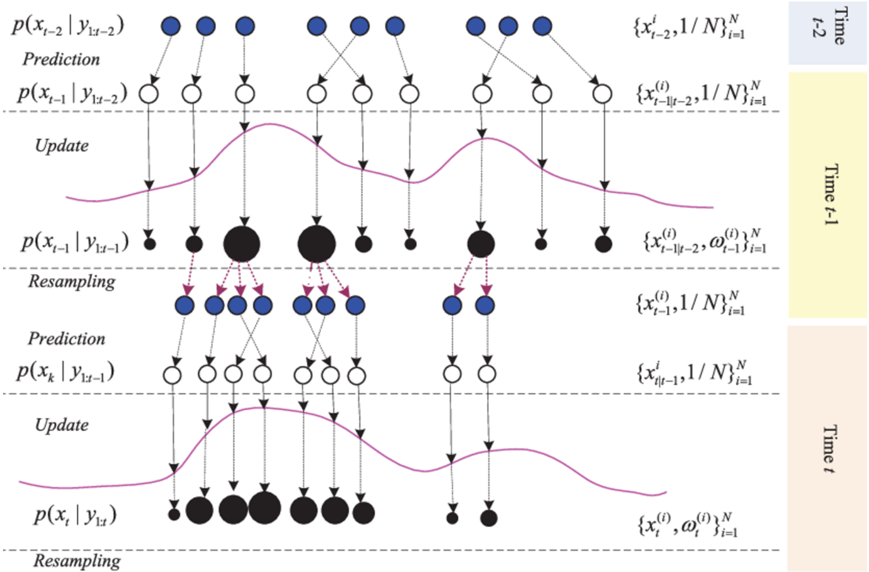
Bayesian Filter Algorithm

Algorithm 1: Bayesian Filter

Input: $\texttt{bel}(\mathbf{x}_{k-1})$, u_k, y_k
Output: $\texttt{bel}(\mathbf{x}_k)$
for *each* $\mathbf{x}_k$ **do**
  $\texttt{pred}(\mathbf{x}_k) \leftarrow \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, u_k) \cdot \texttt{bel}(\mathbf{x}_{k-1})\, d\mathbf{x}_{k-1}$
  $\texttt{bel}(\mathbf{x}_k) \leftarrow h \cdot p(y_k|\mathbf{x}_k) \cdot \texttt{pred}(\mathbf{x}_k)$
end
return $\texttt{bel}(\mathbf{x}_k)$

Difficult to implement exactly because there isn't always an analytical solution for the integral (line 4) and product (line 5)

Classical Particle Filter



Classical Particle Filter Algorithm

Algorithm 2: Classical Particle Filter

Input: $X_i(k)$, $w_i(k)$ (particles and weights at time k)
Output: $X_i(k+1)$, $w_i(k+1)$ (particles and weights at time $k+1$)

```
while explore do
    y(k+1) = measure();                                         LiDAR sensor observation
    for each particle i from 1 to N do
        X_i(k+1) ← f(X_i(k), u(k), σ)                           prediction using noisy model
        w_i(k+1) ← p(y(k+1) | X_i(k+1))                         weight update
    end for
    for each particle i from 1 to N do
        w_i(k+1) = w_i(k+1) / Σ_{j=1}^{N} w_j(k+1)              Normalization
        N_eff ← 1 / Σ_{i=1}^{N} w_i^2(k+1)                      calculation of N_eff
    end for
    if N_eff < N_threshold then
        for each particle i from 1 to N do
            X_i(k+1) ← resample(X_i(k+1), w_i(k+1))             resampling
            w_i(k+1) ← 1/N                                      weight normalization
        end for
    end if
    k ← k+1
end while
```

Where:

$$w_i(k+1) = \frac{w_i(k+1)}{\sum_{j=1}^{N} w_j(k+1)}$$

$$N_{\text{eff}} \leftarrow \frac{1}{\sum_{i=1}^{N} w_i^2(k+1)}$$

Why Interval Particle Filter?

Limitations of Classical Approaches

- **Kalman Filter**: Optimal for linear Gaussian systems but limited in complex marine environments
- **Particle Filter (PF)**: Effective for linear and nonlinear non-Gaussian noise but requires accurate noise distribution models
- **Set-Membership Methods**: Provide guaranteed bounds but often yield overly conservative estimates

Interval Particle Filter Advantage

Hybrid approach combining the probabilistic robustness of particle filters with the guaranteed bounded-error properties of interval analysis

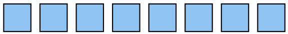
F. Abdallah, A. Gning, and P. Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3), pp. 807–815, 2008.

IPF Algorithm Overview

Step 1: Initialization

❶ Initialization

$$\left\{ \left[\mathbf{x}_0^{(\eta)} \right],\ \omega_0^{(\eta)} = \frac{1}{N_p} \right\}_{\eta=1}^{N_p}$$



N_p *initial particle boxes*

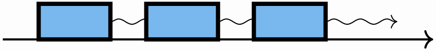
- Particle initialization with bounded boxes
- Uniform distribution of initial weights
- Each particle represents an initial state hypothesis of the target

IPF Algorithm Overview

Step 2: Prediction

❷ Prediction

$$\left\{ \left[\mathbf{x}_{k+1}^{(\eta)}\right] = [\mathbf{f}] \left(\left[\mathbf{x}_{k}^{(\eta)}\right], [\mathbf{u}_k], \left[\mathbf{w}_{k}^{(\eta)}\right] \right) \right\}_{\eta=1}^{N_p}$$



Propagation with system uncertainty

- Particle propagation via the target dynamic model – inclusion function $[\mathbf{f}]$
- Incorporation of system uncertainties
- Expansion of uncertainty intervals using inclusion function $[\mathbf{f}]$

❸ Forward-Backward Contraction

Forward propagation: $\left\{\left[\mathbf{y}_{k+1}^{(\eta)}\right] = [\mathbf{g}]\left(\left[\mathbf{x}_{k+1}^{(\eta)}\right]\right)\right\}_{\eta=1}^{N_p}$

Observation error contraction: $\left\{\left[\boldsymbol{v}_{k+1}^{(\eta)}\right] = \left[\boldsymbol{v}_{k+1}^{(\eta)}\right] \cap \left(\color{red}{[\mathbf{y}_{k+1}]} - \left[\mathbf{y}_{k+1}^{(\eta)}\right]\right)\right\}_{\eta=1}^{N_p}$

Measurement update: $\left\{\left[\mathbf{y}_{k+1}^{(\eta)}\right] = \left[\mathbf{y}_{k+1}^{(\eta)}\right] \cap \left([\mathbf{g}]\left(\left[\mathbf{x}_{k+1}^{(\eta)}\right]\right) + \left[\boldsymbol{v}_{k+1}^{(\eta)}\right]\right)\right\}_{\eta=1}^{N_p}$

Innovation calculation: $\left\{\left[\tilde{\mathbf{y}}_{k+1}^{(\eta)}\right] = \left[\mathbf{y}_{k+1}^{(\eta)}\right] \cap \left(\color{red}{[\mathbf{y}_{k+1}]} - \left[\boldsymbol{v}_{k+1}^{(\eta)}\right]\right)\right\}_{\eta=1}^{N_p}$

Backward propagation (Overlap): $\left\{\color{blue}{\left[\tilde{\mathbf{x}}_{k+1}^{(\eta)}\right]} = \left[\mathbf{x}_{k+1}^{(\eta)}\right] \cap \mathbf{g}^{-1}\left(\left[\tilde{\mathbf{y}}_{k+1}^{(\eta)}\right]\right)\right\}_{\eta=1}^{N_p}$

- Integration of LiDAR sensor measurements (in red)
- Interval contraction to eliminate over-estimation using observation inclusion function $[\mathbf{g}]$

❹ Correction

Likelihood	$\left\{ Likelihood_k^{(\eta)} = \prod_{j=1}^{n_x} \frac{\left\| \left[\tilde{\mathbf{x}}_{k+1}^{(\eta)}(j) \right] \right\|}{\left\| \left[\mathbf{x}_{k+1}^{(\eta)}(j) \right] \right\|} \right\}_{\eta=1}^{N_p}$
Weight update	$\left\{ \omega_{k+1}^{(\eta)} = Likelihood_k^{(\eta)} \times \omega_k^{(\eta)} \right\}_{\eta=1}^{N_p}$

- Likelihood calculation based on interval volumes
- Particle weight update
- Particles well-aligned with measurements receive higher weights

IPF Algorithm Overview

Step 5: Estimation and Resampling

5 Estimation and Resampling

Weighting by box centers

$$\hat{\mathbf{x}}_{k+1} = \sum_{\eta=1}^{N_p} \omega_{k+1}^{(\eta)} \text{mid}\left(\left[\mathbf{x}_{k+1}^{(\eta)}\right]\right) = \sum_{\eta=1}^{N_p} \omega_{k+1}^{(\eta)} \mathbf{c}_{k+1}^{(\eta)}$$

If $N_{\text{eff}} < \textit{threshold}$: resampling

$$N_{\text{eff}} = \frac{1}{\sum_{\eta=1}^{N_p} \left(\omega_{k+1}^{(\eta)}\right)^2}$$

- State estimation by weighted average of centers
- Calculation of sampling efficiency
- Resampling when necessary to avoid degeneracy

Algorithm 3: Interval Particle Filter for Object Tracking

Initialization: $\left\{\left[\mathbf{x}_0^{(\eta)}\right], \omega_0^{(\eta)} = \frac{1}{N_p}\right\}_{\eta=1}^{N_p}$ — N_p initial particle boxes

for $k = 1$ **to** T **do**

  $[\mathbf{y}_k] = [[r_k]\ ,\ [\phi_k]]$ — LiDAR measurements boxes

  for $\eta = 1$ **to** N_p **do**

    Prediction: $\left[\mathbf{x}_k^{(\eta)}\right] \leftarrow [\mathbf{f}]\left(\left[\mathbf{x}_{k-1}^{(\eta)}\right], [\mathbf{u}_{k-1}], \left[\boldsymbol{w}_{k-1}^{(\eta)}\right]\right)$ — Propagation with system uncertainty

    Forward-Backward Contraction:

      $\left[\tilde{\mathbf{x}}_k^{(\eta)}\right] \leftarrow \text{Contraction}\left([\mathbf{y}_k], [\mathbf{x}_k], \left[\mathbf{y}_k^{(\eta)}\right], \left[\boldsymbol{v}_k^{(\eta)}\right]\right)$

    Correction: $\mathsf{L}_k^{(\eta)} \leftarrow \prod_{j=1}^{n_x} \frac{\left\|\left[\tilde{\mathbf{x}}_k^{(\eta)}(j)\right]\right\|}{\left\|\left[\mathbf{x}_k^{(\eta)}(j)\right]\right\|}$ and $\omega_k^{(\eta)} \leftarrow \mathsf{L}_k^{(\eta)} \times \omega_k^{(\eta)}$ — Likelihood and Weight update

  end for

  $\omega_k^{(\eta)} \leftarrow \frac{\omega_k^{(\eta)}}{\sum_{\eta=1}^{N_p} \omega_k^{(\eta)}}$ — Normalize weights

  Output $\hat{\mathbf{x}}_k \leftarrow \sum_{\eta=1}^{N_p} \omega_k^{(\eta)} \text{mid}\left(\left[\mathbf{x}_k^{(\eta)}\right]\right)$ — State estimation

  if $\frac{1}{\sum_{\eta=1}^{N_p}\left(\omega_k^{(\eta)}\right)^2} < N_{\text{thresh}}$ **then**

    Resample$\left(\left\{\left[\mathbf{x}_k^{(\eta)}\right], \omega_k^{(\eta)}\right\}_{\eta=1}^{N_p}\right)$ — Resample if needed

  end if

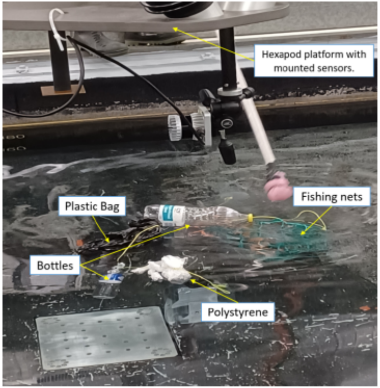
end for

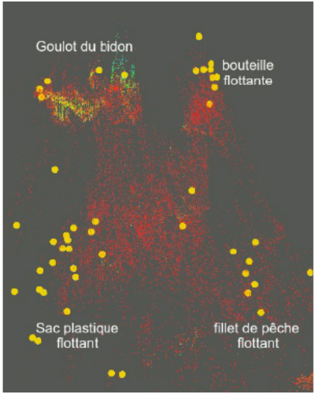
Outline

Experimental Validation: LiDAR Detection at IFREMER, France

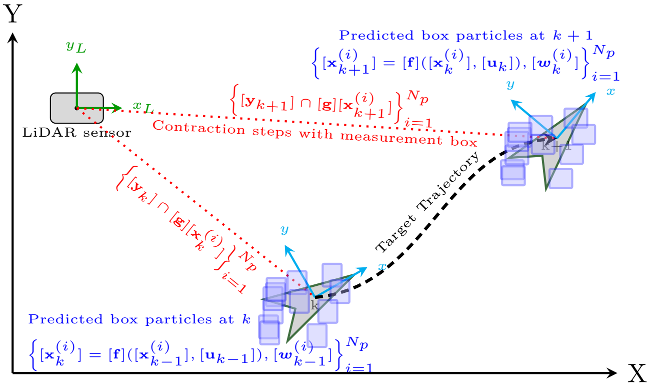
Objective: Validate LiDAR capability for floating object detection in marine environments

- Field experiments conducted using **OS1** and **VLP-16** 3D-LiDAR systems
- Multiple floating targets under various environmental conditions





LiDAR technology proves effective for marine floating object detection despite environmental challenges



- Objective: Track ASV position and heading using noisy LiDAR measurements
- Challenge: Significant sensor uncertainty and limited observations
- Solution: Interval Particle Filter for robust state estimation

System Models for ASV Tracking

Kinematic Model

State vector: $\mathbf{X}_k = [x_k, y_k, \theta_k]^T$ (position and heading)

$$\mathbf{X}_{k+1} = f(\mathbf{X}_k, \mathbf{U}_k) = \begin{bmatrix} x_k + \Delta x \cos(\theta_k) - \Delta y \sin(\theta_k) \\ y_k + \Delta x \sin(\theta_k) + \Delta y \cos(\theta_k) \\ \theta_k + \Delta\theta \end{bmatrix}$$

Control input: $\mathbf{U}_k = [\Delta x, \Delta y, \Delta\theta]^T$ (measured displacement)

LiDAR Observation Model

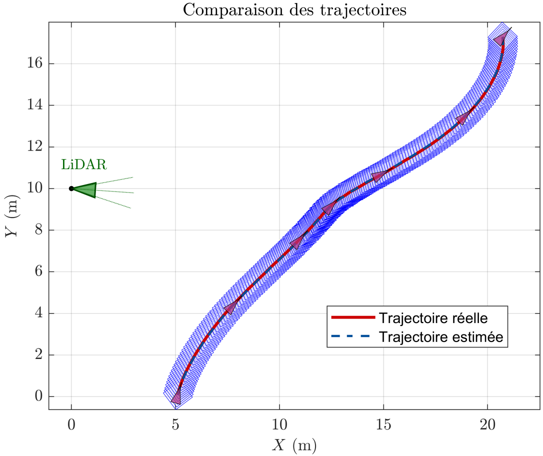
Measurements: range r_k and orientation ϕ_k to known target (x_k, y_k)

$$\mathbf{Y}_k = g(\mathbf{X}_k) = \begin{bmatrix} \sqrt{(x_k - x_L)^2 + (y_k - y_L)^2} \\ \text{atan2}(y_k - y_L, x_k - x_L) - \theta_k \end{bmatrix}$$

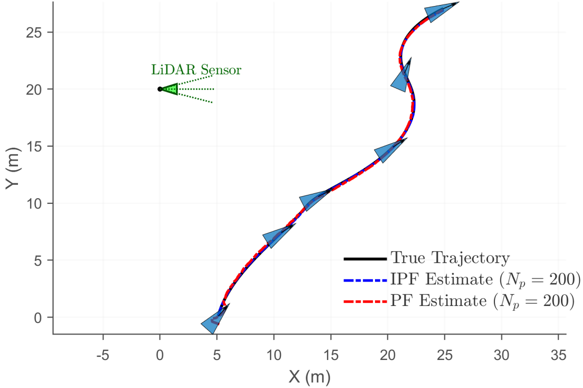
Experimental Setup - INTLAB

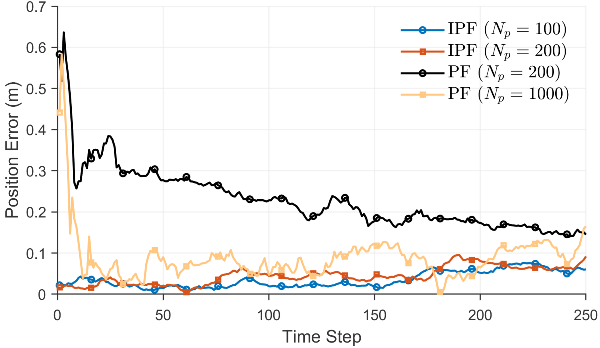
Implementation Framework

- **Implementation**: MATLAB with INTLAB toolbox
- **Comparison**: IPF vs. conventional Particle Filter (PF)
- **Application**: ASV trajectory tracking
- **Sensor**: LiDAR from static known position
- **Metrics**: RMSE and computational time



Qualitative Results: Trajectory Estimation





- IPF maintains tighter bounds around true trajectory
- Significant error reduction compared to standard PF
- Consistent performance with fewer particles

Performance Evaluation

Comparative Analysis: IPF vs. Standard PF

Method	Particles (N_p)	Time (s)	Pos. RMSE (m)	Angle RMSE (°)
IPF	**100**	0.1401	**0.0354**	**0.8327**
IPF	200	0.3409	0.0246	0.7964
PF	200	0.0213	0.2298	0.9023
PF	1000	0.0682	0.0823	0.8254

- IPF demonstrates robust performance
  - **High Accuracy**: IPF achieves lower RMSE under high-noise conditions
  - **Particle Efficiency**: Better performance with fewer particles
  - **Guarantees**: State always within computed bounds
- **Limitation**
  - Higher computational efficiency
  - Requires optimization for real-time

Outline

Conclusion and Future Work

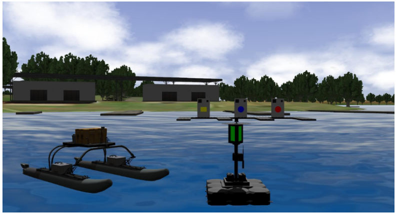
Conclusion

Interval Particle Filtering enables robust and efficient object tracking using LiDAR data:

- **Superior accuracy** with fewer particles compared to conventional methods
- **Guaranteed state enclosures** under significant uncertainty
- **Enhanced robustness** in challenging sensor conditions

Future Work

- **Advanced simulations**: Zonotope and ellipsoid PF, Mobile LiDAR on ASV, and multi-target tracking scenarios, dynamic simulations under VRX Gazebo.
- **Real-world validation**: Experimental testing in rivers, harbors, and coastal areas



VRX Gazebo Simulator

References

Thank you for your attention

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp.
A tutorial on particle filters for online nonlinear non-Gaussian Bayesian tracking.
IEEE Transactions on signal processing, 50(2), pp. 174–188, 2002.

B. Fortin, J. C. Noyer, and R. Lherbier.
A particle filtering approach for joint vehicular detection and tracking in lidar data.
2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings, pp. 391–396, 2012.

A. Rauh, M. Lahme, S. Rohou, L. Jaulin, T. N. Dinh, T. Raissi, and M. Fnadi.
Offline and Online Use of Interval and Set-Based Approaches for Control and State Estimation: A Review of Methodological Approaches and Their Application.
Logical Methods in Computer Science, 2024.

F. Abdallah, A. Gning, and P. Bonnifait.
Box particle filtering for nonlinear state estimation using interval analysis.
Automatica, 44(3), pp. 807–815, 2008.

M. Fnadi, and A. Rauh.
Exponential State Enclosure Techniques for the Implementation of Validated Model Predictive Control.
Acta Cybernetica, 26(4), 839–854, 2024.

Thank you for your attention

mohamed.fnadi@univ-littoral.fr