

A Clustered Approach to Adaptive Kriging Particle Filter for Magnetic Field-Aided Navigation

Bastien HUBERT

DTIS, ONERA, Université Paris-Saclay

Palaiseau, France

bastien.hubert@onera.fr

Nicolas MERLINGE

DTIS, ONERA, Université Paris-Saclay

Palaiseau, France

nicolas.merlinge@onera.fr

Karim DAHIA

DTIS, ONERA, Université Paris-Saclay

Palaiseau, France

karim.dahia@onera.fr

Audrey GIREMUS

IMS, Université de Bordeaux, CNRS, Bordeaux INP

Talence, France

audrey.giremus@u-bordeaux.fr

Abstract—Autonomous navigation is a key aspect in unmanned mobile robotics, where accurate and efficient localisation and tracking are crucial. One of the challenges in these situations is the reliable use of sensor data, particularly in environments where GNSS is unavailable. Magnetometric navigation provides a promising alternative by leveraging the anomalies of the Earth's magnetic field. However, accurate estimation in such settings is hindered by noises, non-linearities and scarcity of environmental information. Gaussian process regression-based particle filtering can be considered to alleviate these difficulties.

In this paper, we propose to improve Gaussian process-based particle filtering techniques by introducing a clustering step to select the relevant training data, improving their computational efficiency and accuracy in the presence of spatial non-stationarities. An auxiliary particle filter is considered due to its robustness to particle degeneracy.

The approach was validated on simulated magnetometric data for an unmanned aerial vehicle. Numerical results show that the clustering step greatly reduces the computational cost of the regression while maintaining the accuracy and convergence speed of the particle filter, demonstrating that this technique offers significant advantages for real-time autonomous navigation.

I. INTRODUCTION

Autonomous navigation is a fundamental challenge in aerospace robotics, particularly in scenarios where global navigation satellite system (GNSS) signals are unavailable or unreliable. In such cases, alternative methods are required to ensure accurate positioning. Magnetic field-aided navigation has emerged as a promising approach, correlating spatial variations in the Earth's magnetic field with local measurements from an embedded sensor to estimate position, as presented by [1]. However, the effectiveness of this method is limited by the inherent non-linearities in the navigation model, as well as the noisy and scarce nature of the available magnetic maps.

To address these challenges, particle filtering is commonly used instead of Kalman filters due to its ability to handle non-linearity and multimodal probability distributions. Additionally, Gaussian process (GP) regression offers a probabilistic framework able to grasp underlying spatial dependencies and account for uncertainties, outperforming multilinear interpolation of sparse and degraded measurements.

However, using for regression all the map samples at each time step is not relevant due a prohibitive computational complexity. While [2] alleviates this issue, it considers the GP to be stationary, which cannot be assumed to be the case on larger scales. As an alternative, we presented in [3] an adaptive kriging method, which consisted in dynamically selecting the set of samples to be used for kriging based on their proximity with the swarm of particles. It proved efficient but failed to take into account multi-modality in the posterior probability density function to be estimated.

In this paper, we refine the proposed approach by clustering the particles to select the relevant training data for each obtained subset of particles. By reducing the number of training points, our method significantly enhances computational efficiency while making it possible to handle spatial non-stationarity in the map and preserving estimation accuracy.

II. PROBLEM STATEMENT

Autonomous navigation situations can be formally described in a discrete state-space representation by a pair of equations: at each time step k , the state $x_k \in \mathbb{R}^{d_x}$ follows the dynamic $\mathcal{F}_k : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\nu} \rightarrow \mathbb{R}^{d_x}$ and observes $z_k \in \mathbb{R}^{d_z}$ through the observation function $\mathcal{H}_k : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_z}$. Because of inaccuracies in both the dynamic and the observation models, the two equations are affected by Gaussian white noises of covariance matrices Q_k and R_k respectively:

$$\begin{cases} x_k \sim \mathcal{N}(\mathcal{F}_k(x_{k-1}), Q_k) \\ z_k \sim \mathcal{N}(\mathcal{H}_k(x_k), R_k) \end{cases} \quad (1)$$

Under this representation, Bayesian inference is perfectly suited to provide a state estimate by computing the posterior probability density function (PDF) of the system. The PDF can then be reduced to a single point estimator by computing its expectation:

$$\hat{x}_k \triangleq \mathbb{E}(x_k | z_{1:k}) = \int x_k p(x_k | z_{1:k}) dx_k \quad (2)$$

and its associated covariance:

$$P_k \triangleq \mathbb{V}(\hat{x}_k) = \int (x_k - \hat{x}_k)(x_k - \hat{x}_k)^T p(x_k | z_{1:k}) dx_k \quad (3)$$

A. Particle Filter

While numerous techniques exist to compute the PDF using Bayesian estimation, the non-linearity and non-injectivity of the observation model makes particle filtering methods highly suited to encompass the potential multimodalities of the PDF. As presented by [4], particle filters use importance sampling to approximate the PDF as a mixture of Dirac delta distributions:

$$p(x_k | z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta_{x_k^i}(x_k) \quad (4)$$

where $\{x_k^i, i \in \llbracket 1, N \rrbracket\}$ are the N particles in \mathbb{R}^{d_x} drawn at each time step according to an importance density, usually equal to the transition probability function:

$$x_k^i \sim q(x_k | x_{k-1}^i, z_k) = p(x_k | x_{k-1}^i) \quad (5)$$

Their respective weights w_k^i , updated during the correction step, are expressed as follows:

$$w_k^i \propto w_{k-1}^i p(z_k | x_k^i) \quad \text{with} \quad \sum_{i=1}^N w_k^i = 1 \quad (6)$$

The particle filter estimator corresponding to (2) is thus the weighted sum of the particles:

$$\hat{x}_k = \sum_{i=1}^N w_k^i x_k^i \quad (7)$$

and its covariance can be written as:

$$P_k = \sum_{i=1}^N w_k^i (x_k^i - \hat{x}_k)(x_k^i - \hat{x}_k)^T \quad (8)$$

B. Gaussian Process Regression

In magnetic field-aided navigation, higher map resolutions yield better estimations. In real-world applications however, only noisy and undersampled field data is available instead of a complete map. Furthermore, because the field exhibits complex spatial dependencies, simple multilinear interpolations fail to capture its underlying intercorrelations.

A powerful alternative called kriging is presented in [5], and consists of finding the best linear unbiased estimator (BLUE) from the sampled data, provided they are realisations of the same GP. We assume that the estimation point and the samples all follow a noisy zero-mean GP:

$$\begin{cases} z \sim \mathcal{GP}(m(x), \mathcal{K}(x, x) + R) \\ \forall i \in \llbracket 1, \tilde{N} \rrbracket, \tilde{z}^i \sim \mathcal{GP}(m(\tilde{x}^i), \mathcal{K}(\tilde{x}^i, \tilde{x}^i) + \tilde{R}) \end{cases} \quad (9)$$

where x and \tilde{x}^i are the inputs corresponding to z and \tilde{z}^i respectively, $m : x \mapsto 0$ is the identically zero mean of the GP, \mathcal{K} is its covariance function, and R and \tilde{R} are the respective covariances of the additive white noises on z and \tilde{z}^i . Using $\tilde{X} \triangleq [\tilde{x}^1 \dots \tilde{x}^{\tilde{N}}]$ and $\tilde{Z} \triangleq [\tilde{z}^1 \dots \tilde{z}^{\tilde{N}}]$, the BLUE of z knowing the \tilde{N} samples \tilde{z}^i is:

$$z^* \triangleq \sum_{i=1}^{\tilde{N}} \lambda_i \tilde{z}^i = \Lambda^T \tilde{Z} \quad \text{with} \quad \mathbb{E}(z - z^*) = 0 \quad (10)$$

where the vector of coefficients Λ is chosen to minimise the error covariance:

$$\Lambda = \text{argmin}(\mathbb{V}(z - z^*)) \quad (11)$$

As proven by [5], the minimum of (11) is reached for:

$$z^*(x, \tilde{X}, \tilde{Z}) = \mathcal{K}(x, \tilde{X}) \left(\mathcal{K}(\tilde{X}, \tilde{X}) + \tilde{R} \otimes I_{\tilde{N}} \right)^{-1} \tilde{Z} \quad (12)$$

and the corresponding covariance is:

$$R^*(x, \tilde{X}) = \mathcal{K}(x, x) + R \otimes I_N - \mathcal{K}(x, \tilde{X}) \left(\mathcal{K}(\tilde{X}, \tilde{X}) + \tilde{R} \otimes I_{\tilde{N}} \right)^{-1} \mathcal{K}(\tilde{X}, x) \quad (13)$$

III. CLUSTERED ADAPTIVE KRIGING AUXILIARY PARTICLE FILTER (CAK-APF)

As stated in section II-B, the available knowledge of the environment is too poor to be able to navigate with (1) as is, requiring kriging of the observation function to be usable. The navigation system thus becomes:

$$\begin{cases} x_k \sim \mathcal{N} \left(\mathcal{F}_k(x_{k-1}), Q_k \right) \\ z_k \sim \mathcal{N} \left(z_k^*(x_k, \tilde{X}, \tilde{Z}), R_k^*(x_k, \tilde{X}) \right) \end{cases} \quad (14)$$

The kriged observation function requires choosing an expression for the covariance function \mathcal{K} , and determining its hyperparameters. Since the observation field is not supposed to be stationary, the hyperparameters cannot be determined beforehand and must be re-estimated in flight at each step of the algorithm. Furthermore, not every sample from \tilde{X} is relevant depending on the spatial distribution of the particles $\{x_k^i\}$ at step k .

Section III-A presents an efficient method to choose the relevant samples to perform the regression, while section III-B shows how to accommodate for the non-stationarity of the field. Section III-C improves particle filtering by adding a pre-propagation step, and section III-D joins these components in a single algorithm.

A. Data selection

When choosing a covariance function \mathcal{K} for the Gaussian process modelling the field, one requirement is that it must be of finite integral:

$$\int \int |\mathcal{K}(x, \tilde{x})| dx d\tilde{x} < +\infty \quad (15)$$

This implies that the covariance function of two inputs must tend towards 0 as they get further away in any direction:

$$\mathcal{K}(x, \tilde{x}) \xrightarrow{d(x, \tilde{x}) \rightarrow +\infty} 0 \quad (16)$$

It, in turn, means that samples from \tilde{X} that are too far away from any particle x_k^i only add computational cost to the regression without significantly improving the estimation. To prevent this, we proposed in [3] to only use samples inside a hypersphere centred around the predicted estimated state, with a radius proportional to the largest eigenvalue of the predicted covariance matrix:

$$S_k \triangleq \mathcal{B} \left(\hat{x}_{k|k-1}, \alpha \times \max_{\lambda \in \text{sp}(P_{k|k-1})} (3\sqrt{\lambda}) \right) \quad (17)$$

where $\alpha \in [1, +\infty[$ is a dilatation coefficient, $\hat{x}_{k|k-1}$ is the predicted estimated state:

$$\hat{x}_{k|k-1} \triangleq \sum_{i=1}^N w_{k-1}^i x_{k-1}^i \quad (18)$$

and $P_{k|k-1}$ is the associated predicted covariance matrix:

$$P_{k|k-1} \triangleq \sum_{i=1}^N w_{k-1}^i (x_{k-1}^i - \hat{x}_{k|k-1})(x_{k-1}^i - \hat{x}_{k|k-1})^T \quad (19)$$

With this method, (14) is rewritten under its adaptive kriged form:

$$\begin{cases} x_k \sim \mathcal{N}\left(\mathcal{F}_k(x_{k-1}), Q_k\right) \\ z_k \sim \mathcal{N}\left(z_k^*(x_k, S_k \cap \tilde{X}, S_k \cap \tilde{Z}), R_k^*(x_k, S_k \cap \tilde{X})\right) \end{cases} \quad (20)$$

where $S_k \cap \tilde{Z}$ is an abuse of notation to mean "elements of \tilde{Z} for which the corresponding \tilde{x} are in $S_k \cap \tilde{X}$ ".

B. Clustering

Due to the spatial non-stationarity of the magnetic field, the hyperparameters of \mathcal{K} cannot be assumed constant throughout space. However, particles close to each other can be assumed to share the same GP hyperparameters. For this reason, we propose to divide the particles into clusters to locally adjust the hyperparameters and perform the regression.

In this paper, we use the Mean-shift clustering algorithm, first introduced by [6], as it is a non-parametric method and does not require a predefined number of clusters. It relies on kernel density estimation to find the local maxima of the PDF by gradually shifting each particle towards a maximum, and associates particles that converged to the same maximum in a single cluster. Formally, a cluster \mathcal{C} is an equivalence class of the binary relation:

$$(a \equiv b) \triangleq \left(a_n = b_n \right)_{n \rightarrow +\infty} \quad (21)$$

where $(x_n)_{n \in \mathbb{N}}$ is the mean-shift sequence of a point x_0 , defined by the recursive equation:

$$\forall n \in \mathbb{N}, x_{n+1} = m(x_n) - x_n \quad (22)$$

and $m : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is the weighted mean of the density given some kernel function κ :

$$m : x \mapsto \sum_{i=1}^N \kappa(x^i - x) x^i / \sum_{i=1}^N \kappa(x^i - x) \quad (23)$$

Within \mathcal{C} , (20) is modified such that (17) becomes:

$$S_k^{\mathcal{C}} \triangleq \mathcal{B} \left(\hat{x}_{k|k-1}^{\mathcal{C}}, \alpha \times \max_{\lambda \in \text{sp}(P_{k|k-1}^{\mathcal{C}})} (3\sqrt{\lambda}) \right) \quad (24)$$

with $\hat{x}_{k|k-1}^{\mathcal{C}}$ the clustered predicted estimator:

$$\hat{x}_{k|k-1}^{\mathcal{C}} \triangleq \sum_{x_k^i \in \mathcal{C}} w_{k-1}^{i,\mathcal{C}} x_k^i \quad (25)$$

$P_{k|k-1}^{\mathcal{C}}$ its covariance matrix:

$$P_{k|k-1}^{\mathcal{C}} \triangleq \sum_{x_k^i \in \mathcal{C}} w_{k-1}^{i,\mathcal{C}} \left(x_k^i - \hat{x}_{k|k-1}^{\mathcal{C}} \right) \left(x_k^i - \hat{x}_{k|k-1}^{\mathcal{C}} \right)^T \quad (26)$$

$\text{sp}(M)$ the spectrum of a matrix M , and $w_{k-1}^{i,\mathcal{C}}$ the normalised weights inside \mathcal{C} :

$$w_{k-1}^{i,\mathcal{C}} \propto w_{k-1}^i \quad \text{with} \quad \sum_{x_k^i \in \mathcal{C}} w_{k-1}^{i,\mathcal{C}} = 1 \quad (27)$$

C. Auxiliary Particle Filter

While the particle filter presented in section II-A is optimal for an infinite number of particles, the approximation for a finite number leads to a phenomenon of degeneracy, in which the variance of the weights of the particles increase over the iterations of the algorithm. A common solution, presented by [7], is to add a regularisation step to the filter. However, doing so increases the variance of the particles, which is already inflated by kriging the observation function. It also introduces new parameters that can be difficult to fine-tune.

Instead, [8] and [9] propose to model the joint probability $p(x_k, i | z_{1:k})$, where i is the index of the parent particle x_{k-1}^i , and omitting i in the estimation. The auxiliary particle filter (APF) works by choosing the importance density such that:

$$q(x_k, i | z_{1:k}) \propto p(z_k | \mu_k^i) p(x_k | x_{k-1}^i) w_{k-1}^i \quad (28)$$

where $\mu_k^i \sim p(x_k | x_{k-1}^i)$ are some samples, called auxiliary particles, used to assess the consistency of the parent particles with respect to the new observation $p(z_k | \mu_k^i)$. The auxiliary particles are then resampled to keep those with the highest likelihood, and their parents $x_{k-1}^{i^j}$ are used to compute the PDF.

From Bayes' theorem and using (28), the expressions of λ_k^i and w_k^j , respectively the weights of the auxiliary and the "standard" particles can be established:

$$\lambda_k^i \propto p(z_k | \mu_k^i) w_{k-1}^i \quad (29)$$

and

$$w_k^j \propto p(z_k | x_k^j) / p(z_k | \mu_k^{i^j}) \quad (30)$$

D. CAK-APF

Since only a kriged interpolation is available instead of the true observation function, APF needs to be adapted. While a simple way to proceed would be to add the clustering, data selection and kriging steps to two correction stages, it should be noted that $\mu_k^{i^j}$ is designed to be a good representation of the future x_k^j . This means that $\mu_k^{i^j}$ and x_k^j are likely to belong to the same cluster, and have thus the same hyperparameters of the GP.

Algorithm 1 presents the pseudo-code of CAK-APF using the same clusters and hyperparameter estimations to compute both the auxiliary and the standard weights. Notably, the clustered data selection window defined in 24 exclusively accounts for samples chosen by the current cluster, disregarding selections made by any other clusters.

IV. NUMERICAL SIMULATIONS FOR MAGNETIC FIELD AIDED NAVIGATION

A. Application to Magnetic Field Aided Navigation

The use case chosen in this paper to illustrate CAK-APF is an unmanned aerial vehicle (UAV) flying in a rectilinear uniform motion over Mexico, as shown by figure 1.

The UAV observes the sampled data \tilde{X} , symbolised by the \times markers, of the local magnetic anomaly field, the data for which comes from [10]. Its state is fully described by the

Algorithm 1 CAK-APF

```

for  $i \in \llbracket 1, N \rrbracket$  do
    Draw  $\mu_k^i$  from  $p(\cdot | x_{k-1}^i)$  ▷ Pre-propagation
end for
Compute  $\{C_k^l, l \in \llbracket 1, c_k \rrbracket\}$  from (21) ▷ Clustering
for  $l \in \llbracket 1, c_k \rrbracket$  do
    Compute  $S_k^{C_k^l}$  with (24) ▷ Selection
    for  $\mu_k^i \in C_k^{C_k^l}$  do
        Compute  $z_k^*$  from clustered (20) ▷ Kriging
        Compute  $\lambda_k^i$  from (29) ▷ Pre-correction
    end for
end for
 $s \leftarrow \sum_{i=1}^N \lambda_k^i$ 
for  $i \in \llbracket 1, N \rrbracket$  do
     $\lambda_k^i \leftarrow \lambda_k^i / s$  ▷ Normalisation
end for
 $\{i^j, j \in \llbracket 1, N \rrbracket\} = \text{Resample}(\{\lambda_k^i, i \in \llbracket 1, N \rrbracket\})$ 
for  $j \in \llbracket 1, N \rrbracket$  do
    Draw  $x_k^j$  from  $p(\cdot | x_{k-1}^{i^j})$  ▷ Propagation
    Associate  $x_k^j$  to the cluster containing  $\mu_k^{i^j}$ 
    Compute  $z_k^*$  from clustered (20) ▷ Kriging
    Compute  $w_k^j$  from (30) ▷ Correction
end for
 $t \leftarrow \sum_{j=1}^N w_k^j$ 
for  $j \in \llbracket 1, N \rrbracket$  do
     $w_k^j \leftarrow w_k^j / t$  ▷ Normalisation
end for

```

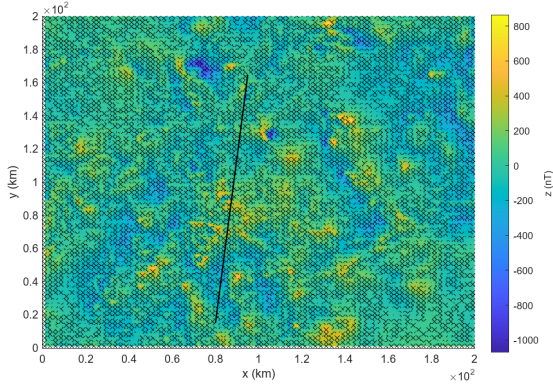


Fig. 1. Trajectory of the UAV over Mexico's magnetic anomaly field

$d_x = 4$ dimension vector $x = [p_x \ p_y \ v_x \ v_y]^T$, with p_x and p_y being the x and y coordinates of the UAV, and v_x and v_y their respective velocities. The dynamical model of the UAV is given by:

$$\mathcal{F}_k : x \in \mathbb{R}^4 \mapsto \begin{bmatrix} I_2 & \Delta_t I_2 \\ 0 & I_2 \end{bmatrix} x \quad (31)$$

where Δ_t is the sampling period of the algorithm, and I_2 is the identity matrix of size 2.

In order to evaluate the different steps of CAK-APF, simulations were performed on four different filters:

- The standard APF with known map instead of kriging;

- A static kriging APF, using every samples to perform the regression (the estimation of the hyperparameters is done once before the start of the Monte-Carlo runs);
- An adaptive kriged APF without clustering;
- CAK-APF as presented by algorithm 1.

N_{MC} Monte-Carlo runs are performed per filter, during which an estimation of the state \hat{x}_k is iteratively computed over N_{iter} iterations from an unknown state drawn from a Gaussian distribution centred around the true initial state x_0 , with covariance P_0 . Table I gives the values of the chosen parameters.

TABLE I
SIMULATION AND CAK-APF PARAMETERS

Parameters	Value
N_{MC}	100
N_{iter}	600
Δ_t	0.5s
map surface	200km \times 200km
map resolution	1km \times 1km
P_0	diag([3000m, 3000m, 3m.s ⁻¹ , 3m.s ⁻¹]) ²
N	5000
Q_k	diag([40m, 40m, 0.4m.s ⁻¹ , 0.4m.s ⁻¹]) ²
R_k	80 ² nT ²
\tilde{N}	80 ²
\tilde{R}	80 ² nT ²
α	2
resampling algorithm	Kitagawa resampling from [4]
κ	Gaussian kernel from [5]
hyperparameter estimation	likelihood gradient ascent from [5]
\mathcal{K}	Epanechnikov function from [5]

B. Numerical Results

For each filter, their root mean square error (RMSE) has been calculated according to:

$$\text{RMSE} : k \in \llbracket 1, N_{iter} \rrbracket \mapsto \sqrt{\frac{\sum_{m=1}^{N_{MC}} \|\varphi(x_k - \hat{x}_{k,m})\|_2^2}{N_{MC}}} \quad (32)$$

where $\varphi : [p_x \ p_y \ v_x \ v_y]^T \mapsto [p_x \ p_y]^T$ is the projector on the positions, x_k is the true state at time k , and $\hat{x}_{k,m}$ is the corresponding estimation by the m -th Monte-Carlo run. A histogram of the number of the samples used to perform the kriging step for each filter has also been computed. Figure 2 gives the RMSEs over time of the four filters, and figure 3 presents their corresponding histogram.

Unsurprisingly, the greatest accuracy is achieved when APF has access to the full and undisturbed map. On the other hand, it must be noted that reconstructing the whole map with a static Gaussian process prior to navigation yields mitigated results. This can be explained by the non-stationarity of the anomaly field, whose hyperparameters seem to have been adequately estimated for iterations 75 to 200, but are less representative outside this range. Similarly, while performing better than its static counterpart, AK-APF exhibits a wide spread in the selected samples, as determined by the covariance matrix from (17). This results in a slower convergence and prevents it from reaching the sub-kilometre accuracy of CAK-APF.

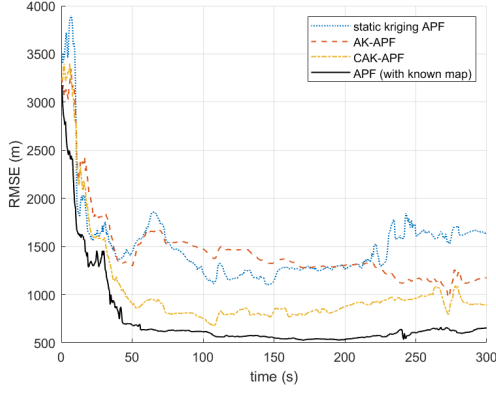


Fig. 2. RMSE over time of the tested filters

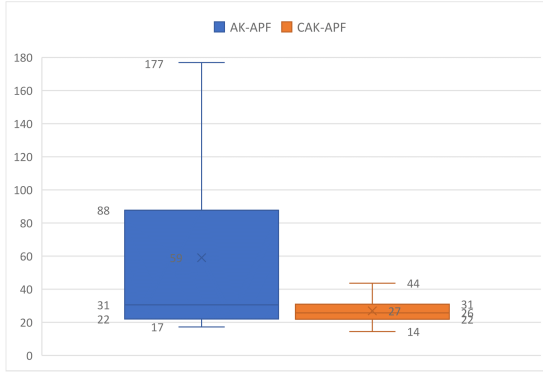


Fig. 3. Histogram of the number of selected samples used in the tested filters

While figure 3 shows that the median numbers of samples for the adaptive kriging methods are relatively close, the upper part of the distribution is much larger for AK-APF than for CAK-APF. This is caused by highly multimodal situations, where AK-APF encompasses irrelevant samples between clusters, in areas where there are no particles. On the other hand, the much narrower distribution of CAK-APF ensures a more consistent computation time, which is crucial in real-time applications.

For N particles, \tilde{N} samples, and $T \ll N$ iterations required by Mean-shift to converge, the complexities of Mean-shift, data selection and hyperparameter learning, and kriging are $\mathcal{O}(TN^2)$, $\mathcal{O}(\tilde{N}^3)$, and $\mathcal{O}(N\tilde{N}^2)$ respectively. For an average of 2 clusters, $T \approx 15$, and considering that CAK-APF uses on average 46% less samples than its AK counterpart, this means that CAK is more than 2.5 faster than AK on average, both of which are more than 6 orders of magnitude faster than static kriging. The gap is even wider considering that the histogram counts the selected data from every cluster of a single iteration, so only a fraction of the samples is simultaneously used, further decreasing the computational cost of CAK-APF.

V. CONCLUSION AND PERSPECTIVES

In this paper, we improved GP-based particle filtering by introducing a clustering step to select the most relevant

training data for the kriging step. By locally estimating the hyperparameters of the GP modelling the environment, we increased both computational speed and accuracy. Through simulations of a UAV navigating with magnetic anomaly data, we showed significant improvements from both static kriging and a simpler adaptive algorithm from which this version is based.

The results show that clustering particles not only optimises the use of available samples but also ensures that the non-stationarity of the magnetic field is adequately addressed. CAK outperformed its counterparts, offering an effective solution for real-time autonomous navigation in environments where global navigation satellite systems are unavailable or unreliable. Furthermore, the use of APF helped mitigating the particle degeneracy problem without introducing additional parameters, thus enhancing the robustness of the algorithm.

Future work will focus on exploring other methods for data selection and computing formal error bounds for the approximated observation model. Additionally, incorporating real-time adaptation of the magnetic field model and introducing a simultaneous localisation and mapping (SLAM) formalism could provide insights into its performance and scalability in operational environments.

REFERENCES

- [1] A. J. Canciani and C. J. Brennan. “An Analysis of the Benefits and Difficulties of Aerial Magnetic Vector Navigation”. In: *IEEE Transactions on Aerospace and Electronic Systems* 56.6 (2020), pp. 4161–4176.
- [2] A. Solin and S. Särkkä. “Hilbert space methods for reduced-rank Gaussian process regression”. In: *Statistics and Computing* 30 (2020), pp. 419–446.
- [3] B. Hubert, K. Dahia, N. Merlinge, and A. Giremus. “Adaptive Kriging Particle Filter and its Application to Terrain-Aided Navigation”. In: *2024 27th International Conference on Information Fusion (FUSION)* (2024), pp. 1–7.
- [4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking”. In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 174–188.
- [5] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Massachusetts Institute of Technology: the MIT Press, 2006.
- [6] K. Fukunaga and L. Hostetler. “The estimation of the gradient of a density function, with applications in pattern recognition”. In: *IEEE Transactions on Information Theory* 21.1 (1975), pp. 32–40.
- [7] C. Musso, N. Oudjane, and F. Le Gland. “Improving Regularised Particle Filters”. In: *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer New York, 2001, pp. 247–271.
- [8] M. K. Pitt and N. Shephard. “Filtering via Simulation: Auxiliary Particle Filters”. In: *Journal of the American Statistical Association* 94.446 (1999), pp. 590–599.
- [9] R. Douc, E. Moulines, and J. Olsson. “Optimality of the auxiliary particle filter”. In: *Probability and Mathematical Statistics* 29.1 (2009), pp. 1–28.
- [10] V. Bankey et al. *Digital Data Grids for the Magnetic Anomaly Map of North America*. 2002. URL: <https://pubs.usgs.gov/publication/ofr02414>.