Adaptive Kernel Kalman Filter for Magnetic Anomaly Detection-based Metallic Target Tracking

Citation for published version:
Sun, M, Hodgskin-Brown, R, Davies, ME, Proudler, I & Hopgood, JR 2023, Adaptive Kernel Kalman Filter for Magnetic Anomaly Detection-based Metallic Target Tracking. in *2023 Sensor Signal Processing for Defence Conference (SSPD)* . IEEE, 2023 Sensor Signal Processing for Defence Conference , Edinburgh, United Kingdom, 12/09/23. https://doi.org/10.1109/SSPD57945.2023.10256968

Digital Object Identifier (DOI):
10.1109/SSPD57945.2023.10256968

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
2023 Sensor Signal Processing for Defence Conference (SSPD)

Adaptive Kernel Kalman Filter for Magnetic Anomaly Detection-based Metallic Target Tracking

Mengwei Sun, Richard Hodgskin-Brown, Mike E. Davies, Ian K. Proudler, James R. Hopgood

***Abstract*—This paper proposes the use of the adaptive kernel Kalman filter (AKKF) to track metallic targets using magnetic anomaly detection (MAD). The proposed AKKF-based approach enables accurate tracking of moving metallic targets using magnetometer sensors, even in the presence of dynamic and unknown magnetic moments. The experimental results demonstrate that the proposed method exhibits favourable tracking and estimation performance with reduced computational complexity compared with the bootstrap particle filter (PF). For example, in magnetic moment strength estimation, the relative root mean square error (RRMSE) of the proposed algorithm using 50 particles can approach 2.5% with a computation time of 0.18 seconds, whereas the RRMSE of the PF using 2000 particles is 4.5% with a computation time of 1.4 seconds. This study highlights the potential of AKKF in MAD for metallic target tracking using magnetometer sensors.**

***Index Terms*—Adaptive kernel Kalman filter, magnetic anomaly detection, metallic target tracking**

I. Introduction

Detecting and tracking targets are critical in automated surveillance and security systems that aim to keep up with evolving safety and security risks. In recent years, magnetic anomaly detection (MAD) has been widely studied for various applications in military and civilian contexts [1], such as airborne maritime surveillance [2], shipwrecks [3], access control [4], and tracking of moving metallic vehicles [5], [6]. The magnetic field is an intrinsic characteristic of many objects. The ability to detect and track magnetic fields provides a non-invasive and contactless method for monitoring and analysing these objects. Tracking techniques based on MAD typically utilise magnetic sensors, such as magnetometers [5], [6], to detect and measure the magnetic field generated by the objects. The position and orientation of the target can then be estimated based on the measured magnetic field [2], [5], [6]. Unlike other tracking technologies, such as optical or ultrasonic methods, magnetic tracking is emerging as an occlusion-free tracking scheme for estimating the position and orientation of the target [7].

The tracking problem can be formulated under the Bayesian framework by understanding the relationship between the magnetic moment of the target and its kinematic parameters. In [5], [6], magnetometer sensor models for tracking metallic point targets and extended targets are proposed and validated. The suitability of magnetometer sensors for tracking is analysed regarding local observability and the Cramér–Rao lower bound (CRLB). The extended Kalman filter (EKF) and the weighted least squares algorithm, by minimising the cost function, are used for estimating the kinematic parameters and magnetic moment, respectively. However, the time-varying magnetic dipole moment, which arises due to the moving vehicle's heading, is ignored and set to be constant. In [2], the authors investigate the use of various nonlinear filters for kinematic and magnetic dipole tracking applications and compare their performances. The nonlinear filters that are compared include the EKF, unscented Kalman filter (UKF), generic particle filter (GPF), auxiliary particle filter (APF), a combination of EKF and GPF and a combination of UKF and GPF. Sithiravel et al. [2] also include the derivation of the posterior CRLB to quantify the possible best estimation accuracy for MAD.

The proposed sensor model in [2], [5], [6] results in a sequential Bayesian estimation problem that is both highly nonlinear and high-dimensional. Choosing a Bayesian filter involves balancing between accuracy and computation complexity. While the EKF is computationally efficient, its accuracy may suffer when the system's nonlinearity is high. In contrast, the UKF and particle filter (PF) can provide better accuracy for highly nonlinear problems. However, the computational cost of the UKF can increase for high-dimensional systems, while the PF can suffer from the curse of dimensionality. Recently, the adaptive kernel Kalman filter (AKKF) has been proposed [8]–[11], which demonstrates significant improvement in estimation performance compared to other nonlinear Kalman filters (KFs) while reducing computation complexity and avoiding resampling, as is often required with most PFs in tracking systems. This paper investigates the potential of using the AKKF within MAD-based vehicle tracking with the following contributions:

- Exploring a new application for the AKKF. While previous work focused on utilising the AKKF for object tracking problems, this paper uses the AKKF for joint tracking and magnetic parameters estimation, which are high-dimensional and high nonlinear problems.

M. W. Sun, M. E. Davies and J. R. Hopgood are with the Institute of Digital Communications, University of Edinburgh, Edinburgh, EH9 3FG, U.K. E-mail: (msun; mike.davies; james.hopgood)@ed.ac.uk.

I. Proudler is with the Centre for Signal & Image Processing (CeSIP), Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, U.K. E-mail: ian.proudler@strath.ac.uk.

Richard Hodgskin-Brown is with the Electromagnetic Sensing Group, Department of Electronics & Electrical Engineering, University of Manchester, Manchester, M13 9PL, UK. E-mail: richard.hodgskin-brown@postgrad.manchester.ac.uk

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing; EPSRC Impact Acceleration Accounts (IAA) Project number PV022.

- The simulations evaluate the tracking and estimation performance of the AKKF and demonstrate improved computation efficiency in vehicle tracking and magnetic parameters estimation. For example, compared with the PF, the relative root mean square error (RRMSE) of the magnetic moment strength estimation achieved by the AKKF can be improved from 7% to 2% when using 100 particles.

The paper is structured as follows: Section II describes the system model, Section III presents the AKKF-based algorithm, Section IV provides the simulation results, and Section V draws the conclusions.

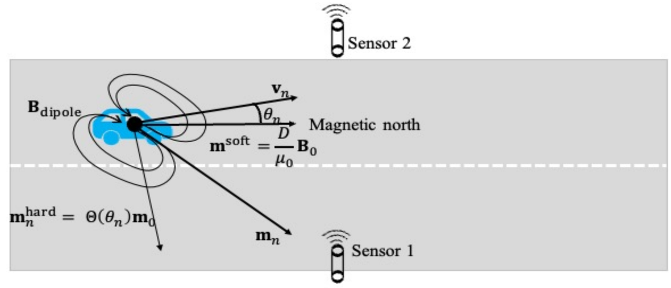

Fig. 1: System setup: Two magnetometer sensors, the vehicle is moving to pass them.

II. System model

The system for MAD-based vehicle tracking is shown in Fig. 1, where two magnetometer sensors are positioned close to a straight road with vector coordinates denoted as $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. The vehicle is moving to pass the stationary magnetometer sensors. The vehicle is approximated as a point magnetic dipole. The dynamic state-space model (DSSM) comprises a motion model that describes the target's position and velocity over time and a measurement model that relates the target's magnetic field to the measurements obtained by the sensors. The target evolves as

$$\mathbf{x}_n = F\mathbf{x}_{n-1} + \mathbf{u}_n = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{n-1} + \mathbf{u}_n. \tag{1}$$

Here, ΔT is the sampling interval and is set as $\Delta T = 1$, $\mathbf{u}_n$ is the process noise vector, the time index n is defined as $n = 1, \ldots, N$, where N represents the number of time steps. The hidden states are $\mathbf{x}_n = [\xi_n, \dot{\xi}_n, \eta_n, \dot{\eta}_n, \zeta_n]^{\mathrm{T}}$, where (ξ_n, η_n, ζ_n) represents the target dipole position in X-axis, Y-axis and Z-axis, and $(\dot{\xi}_n, \dot{\eta}_n)$ represent the corresponding velocity in X-axis and Y-axis. We only consider the vehicle's motion in the X-Y 2D plane and ignore the velocity in the Z-axis, as the vehicle is constrained to move on a flat surface and cannot move up or down.

The measurement at the k-th magnetometer sensor is based on a nonlinear model that can be described as follows [6]

$$\begin{aligned} \mathbf{y}_{n,k} &= h_k(\mathbf{x}_n, \mathbf{m}_n) + \mathbf{e}_{n,k} \\ &= \mathbf{B}_0 + \frac{\mu_0}{4\pi} \frac{3\,(\mathbf{r}_{n,k} \cdot \mathbf{m}_n)\,\mathbf{r}_{n,k} - \|\, \mathbf{r}_{n,k} \,\|^2\, \mathbf{m}_n}{\|\, \mathbf{r}_{n,k} \,\|^5} + \mathbf{e}_{n,k}. \end{aligned} \tag{2}$$

Here, the constant $\mathbf{B}_0$ is the Earth's magnetic field, $\mathbf{r}_{n,k} = [\xi_n, \eta_n, \zeta_n]^{\mathrm{T}} - \mathbf{s}_k$ is the target position relative to the k-th sensor at time n, and $\cdot$ denotes the dot product. The magnetic dipole moment of the target is $\mathbf{m}_n$, and the additive white Gaussian noise (AWGN) associated with the measurement is $\mathbf{e}_{n,k} \sim \mathcal{N}(\mathbf{0}, R_k)$. The magnetic field of the metallic objects, as shown in Fig. 1, is induced partly due to the ferromagnetic content (hard iron) and partly due to the deflection of the Earth's magnetic field (soft iron). Hence, the magnetic moment of the metallic objects is modelled as [6]:

$$\mathbf{m}_n = \mathbf{m}_n^{\text{hard}} + \mathbf{m}_n^{\text{soft}} = \Theta(\theta_n)\mathbf{m}_0 + \frac{D}{\mu_0}\mathbf{B}_0, \tag{3}$$

The rotation matrix $\Theta(\theta_n)$ is used to model the effect of the heading on the magnetic field refers to the magnetic north, and it can be expressed as [6]:

$$\Theta(\theta_n) = \begin{bmatrix} \cos\theta_n & -\sin\theta_n & 0 \\ \sin\theta_n & \cos\theta_n & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

The magnetic dipole moment of the target, denoted by $\mathbf{m}_0$, is assumed to be independent of the external magnetic field. The scalar constant D accounts for the magnetic field induced by the target's ferromagnetic content and deflection of the Earth's magnetic field. The permeability of the vacuum, represented by μ_0, is a fundamental physical constant that describes the magnetic properties of free space.

III. AKKF-based tracking and estimation algorithms

The purpose of metallic target tracking is to precisely track the target's movement and simultaneously estimate its magnetic moment. This is accomplished through the utilisation of a posterior probability density function (pdf), which explains the joint distribution of the target's hidden states $\mathbf{X}_n = \left[\mathbf{x}_n^{\mathrm{T}}, \mathbf{m}_n^{\mathrm{T}}, \mathbf{m}_0, D\right]^{\mathrm{T}}$, considering the observations $\mathbf{y}_{1:n,1:2}$ at two sensors which are located at $\mathbf{s}_{1:2}$. The joint posterior pdf is decomposed in Equation (5). In this section, we will discuss how to use the PF and the AKKF to sequentially approximate the joint posterior pdf.

A. PF-based algorithm

The PF approximates the joint posterior pdf by using a weighted set of particles. Each particle represents a possible value of the joint state variables $\mathbf{X}_n$ at each time step $n = 1, 2, ..., N$. The joint posterior distribution in (5) can be estimated as follows:

$$\begin{aligned} &p(\mathbf{X}_n \mid \mathbf{y}_{1:n,1:2}) \\ &\approx \frac{1}{M} \sum_{i=1}^{M} w_n^{\{i\}}\, \delta(\mathbf{x}_n - \mathbf{x}_n^{\{i\}}, \mathbf{m}_n - \mathbf{m}_n^{\{i\}}, \mathbf{m}_0 - \mathbf{m}_{0,n}^{\{i\}}, D - D_n^{\{i\}}). \end{aligned} \tag{6}$$

$$p(\mathbf{X}_n \mid \mathbf{y}_{1:n,1:2}) = p(\mathbf{x}_n, \mathbf{m}_n, \mathbf{m}_0, D \mid \mathbf{y}_{1:n,1:2}) = p\left(\mathbf{y}_{n,1:2} \mid \mathbf{x}_n, \mathbf{m}_n, \mathbf{m}_0, D\right) \times \frac{\iiiint p\left(\mathbf{x}_n|\mathbf{x}_{n-1}\right) p\left(\mathbf{m}_n|\mathbf{x}_n, \mathbf{m}_{n-1}, \mathbf{m}_0, D\right) p\left(\mathbf{m}_0, D\right) p\left(\mathbf{x}_{n-1}, \mathbf{m}_{n-1}, \mathbf{m}_0, D \mid \mathbf{y}_{1:n-1,1:2}\right) d\mathbf{x}_{n-1} d\mathbf{m}_{n-1} d\mathbf{m}_0 dD}{p\left(\mathbf{y}_{n,1:2} \mid \mathbf{y}_{1:n-1,1:2}\right)} \tag{5}$$

Here, $w_n^{\{i\}}$ represents the weight of the i-th particle at time step n, δ denotes the Dirac delta function, and M is the number of particles. At each time step n, the weight $w_n^{\{i\}}$ is updated based on the likelihood of the observation $\mathbf{y}_{1:n,1:2}$ given the particle's state variables $\{\mathbf{x}_n^{\{i\}}, \mathbf{m}_n^{\{i\}}, \mathbf{m}_0^{\{i\}}, D^{\{i\}}\}$, i.e., $w_n^{\{i\}} = w_{n-1}^{\{i\}} p\left(\mathbf{y}_{1:n,1:2} \mid \mathbf{x}_n^{\{i\}}, \mathbf{m}_n^{\{i\}}, \mathbf{m}_0^{\{i\}}, D^{\{i\}}\right)$. The state variables of each particle are updated using the transition probabilities as (7), where $\theta_n^{\{i\}} = \arctan2(\dot{\eta}_n^{\{i\}}, \dot{\xi}_n^{\{i\}})$, and $\mathbf{u}_n^{\{i\}}$ represents a process noise sample drawn from the process noise distribution.

$$\mathbf{x}_n^{\{i\}} = F\mathbf{x}_{n-1}^{\{i\}} + \mathbf{u}_n^{\{i\}} \tag{7a}$$

$$\mathbf{m}_n^{\{i\}} = \Theta(\theta_n^{\{i\}})\mathbf{m}_{0,n}^{\{i\}} + \frac{D_n^{\{i\}}}{\mu_0}\mathbf{B}_0 \tag{7b}$$

$$\mathbf{m}_{0,n}^{\{i\}} = \mathbf{m}_{0,n-1}^{\{i\}} \tag{7c}$$

$$D_n^{\{i\}} = D_{n-1}^{\{i\}}. \tag{7d}$$

After updating the particles and their weights, the particles are resampled to obtain a new set of particles for the next time step. The resampling process involves randomly selecting particles from the current set with probability proportional to their weights, with replacement.

However, the computational cost of the PF grows exponentially with the number of state variables, making it impractical for high-dimensional problems. In high-dimensional problems, it is difficult to obtain a sufficient number of particles to represent the posterior pdf accurately, leading to particle degeneracy, where only a small subset of particles have non-zero weights, and the rest are effectively ignored. This can result in poor estimation accuracy and instability in the estimates. To address this issue, we investigate the use of the AKKF to solve high-dimensional problems with low computational costs and favourable accuracy.

B. AKKF-based algorithm

The proposed AKKF [8] enables us to obtain the empirical kernel mean embedding (KME) of the posterior pdf of the hidden state in (5). This is accomplished using a set of feature mappings of generated particles and their corresponding kernel weights. The particles are updated and propagated in the data space based on the parametric DSSMs, and the corresponding kernel weights are predicted and updated linearly. Common kernel functions used for KMEs include linear, quadratic, quartic, and Gaussian kernels. The quartic kernel can be used when the data is highly nonlinear and complex. Considering the system setup and the DSSM in equations (1) and (2), the nonlinearity of the measurement model is highly nonlinear. Therefore, we apply the quartic kernel to approximate the predictive and posterior pdfs in this paper. The quartic kernel function $k(\mathbf{X}, \mathbf{Y})$ and its corresponding feature mapping $\phi_{\mathbf{X}}(\mathbf{X})$ are defined as:

Algorithm 1 AKKF-based metallic target tracking algorithm

Require: DSSM: motion model and measurement model.

1: **Initialisation**: Set the initial particles in the data space $\tilde{\mathbf{x}}_0^{\{i=1:M\}} \sim P_{\text{init}}$, $\mathbf{w}_0 = 1/M\,[1, \dots, 1]^{\mathrm{T}}$.

2: **for** $n = 1 : N$ **do**

3: Prediction:
- In the data space, propagate proposal particles following (7),
- $\Rightarrow$ In the kernel feature space with basis Φ_n: $\mathbf{w}_n^- = \Gamma_n \mathbf{w}_{n-1}^+$, $\quad S_n^- = \Gamma_n S_{n-1}^+ \Gamma_n^{\mathrm{T}} + V_n$.

4: Update:
- In the data space: $\mathbf{y}_n^{\{i\}} = h(\mathbf{X}_n^{\{i\}}, \mathbf{e}_n^{\{i\}})$,
- $\Rightarrow$ In the kernel feature space with basis Φ_n: $\mathbf{w}_n^+ = \mathbf{w}_n^- + Q_n\left(G_{:,\mathbf{y}_n} - G_{\mathbf{yy}}\mathbf{w}_n^-\right)$, $S_n^+ = S_n^- - Q_n G_{\mathbf{yy}} S_n^-$.

5: Proposal particles draw:
- In the data space: $\tilde{\mathbf{X}}_n^{\{i=1:M\}} \sim \mathcal{N}\left(\mathbb{E}\left(\mathbf{X}_n\right), \mathrm{Cov}\left(\mathbf{X}_n\right)\right)$,
- $\Rightarrow$ Get the kernel feature space with basis Ψ_n.

6: **end for**

$$k(\mathbf{X}, \mathbf{Y}) = (\mathbf{X}^{\mathrm{T}}\mathbf{Y} + c)^4 \tag{8a}$$

$$\phi_{\mathbf{X}}(\mathbf{X}) = \left[a_1, \dots, a_j, \dots, a_d\right]^{\mathrm{T}}, \tag{8b}$$

where $c \geq 0$ is a free parameter that trades off the influence of higher-order versus lower-order terms in the polynomial, and the element in the quartic kernel feature mapping is

$$a_j = \frac{\sqrt{4!}}{\sqrt{\varrho_1! \dots \varrho_K! \varrho_{K+1}!}} x_1^{\varrho_1} \dots x_k^{\varrho_K} \sqrt{c}^{\varrho_{K+1}}, \quad \varrho_1 + \cdots + \varrho_{K+1} = 4.$$

Here, $\varrho_1, \dots, \varrho_{K+1}$ are non-negative integers representing the powers of the corresponding input dimensions. The dimension of $\phi_{\mathbf{X}}(\mathbf{X})$ is $d = (K+4)!/(4!K!)$, where K is the dimension of the hidden state $\mathbf{X} = [x_1, \dots, x_k, \dots, x_K]^{\mathrm{T}}$.

The proposed AKKF-based algorithm is realised sequentially by embedding the pdf $p(\mathbf{X}_n \mid \mathbf{y}_{1:n,1:2})$ into an reproducing kernel Hilbert space (RKHS) as an empirical KME,

$$p(\mathbf{X}_n \mid \mathbf{y}_{1:n,1:2}) \to \hat{\mu}_{\mathbf{X}_n}^+ = \Phi_n \mathbf{w}_n^+, \tag{9}$$

where Φ_n represents the kernel feature mappings of particles and $\mathbf{w}_n^+$ is the updated kernel weight. The AKKF-based algorithm consists of three main steps, which we will further explain in the following subsections. The algorithm is summarised in Algorithm I. See [8] for details of the AKKF.

1) Draw Proposal Particles at Time $n-1$: The posterior distribution pdf at time $n-1$, i.e., $p(\mathbf{X}_{n-1} \mid \mathbf{y}_{1:n-1,1:2})$ is empirically as approximated by an element $\hat{\mu}^{+}_{\mathbf{X}_{n-1}}$ in the RKHS based on the AKKF, resulting in $p(\mathbf{X}_{n-1} \mid \mathbf{y}_{1:n-1,1:2}) \rightarrow \hat{\mu}^{+}_{\mathbf{X}_{n-1}} = \Phi_{n-1}\mathbf{w}^{+}_{n-1}$. Here, $\Phi_{n-1} = \left[\phi_{\mathbf{x}}(\mathbf{X}^{\{1\}}_{n-1}), \ldots, \phi_{\mathbf{x}}(\mathbf{X}^{\{M\}}_{n-1})\right]$ represents the kernel feature mappings of the particles $\mathbf{X}^{\{1:M\}}_{n-1}$ using the quartic kernel function, and $\mathbf{w}^{+}_{n-1}$ is the weight vector with a positive definite weight covariance matrix denoted as S^{+}_{n-1}. Then, $\mathbb{E}(X_{n-1})$ and $\mathrm{Cov}(\mathbf{X}_{n-1})$ from $\hat{\mu}^{+}_{\mathbf{X}_{n-1}}$ are extracted and passed to the data space following [8]. Next, proposal particles are generated according to the importance of distribution as $\tilde{\mathbf{X}}^{\{i=1:M\}}_{n-1} \sim \mathcal{N}\left(\mathbb{E}\left(\mathbf{X}_{n-1}\right), \mathrm{Cov}\left(\mathbf{X}_{n-1}\right)\right)$, and mapped to the RKHS as $\Psi_{n-1} = \left[\phi_{\mathbf{x}}(\tilde{\mathbf{X}}^{\{1\}}_{n-1}), \ldots, \phi_{\mathbf{x}}(\tilde{\mathbf{X}}^{\{M\}}_{n-1})\right]$.

2) Prediction from Time $n-1$ to Time n: The empirical KME of the predictive probability at time n is approximated using a linear conditional operator in the RKHS:

$$\begin{aligned} p(\mathbf{X}_n|\mathbf{y}_{1:n-1,1:2}) &\mapsto \hat{\mu}^{-}_{\mathbf{X}_n} = \hat{C}_{\mathbf{X}_n|\tilde{\mathbf{X}}_{n-1}}\hat{\mu}^{+}_{\mathbf{X}_{n-1}} \\ &= \Phi_n \underbrace{(K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} + \lambda_{\tilde{K}} I)^{-1} K_{\tilde{\mathbf{x}}\mathbf{x}}}_{\Gamma_n} \mathbf{w}^{+}_{n-1} = \Phi_n \mathbf{w}^{-}_{n}. \end{aligned} \tag{10}$$

Here, $\Phi_n = \left[\phi_{\mathbf{x}}(\mathbf{X}^{\{1\}}_{n}), \ldots, \phi_{\mathbf{x}}(\mathbf{X}^{\{M\}}_{n})\right]$ represent the feature mappings of the state particles at time n, which are obtained by propagating $\tilde{\mathbf{X}}^{\{i=1:M\}}_{n-1}$ through the process function following (7). The Gram matrices $K_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} = \Psi^{\mathrm{T}}_{n-1}\Psi_{n-1}$ and $K_{\tilde{\mathbf{x}}=\mathbf{x}}\Psi^{\mathrm{T}}_{n-1}\Phi_{n-1}$. And Γ_{n-1} represents the change of sample representation from Φ_{n-1} to Ψ_{n-1}. The regularisation parameter, $\lambda_{\tilde{K}}$, ensures that the inverse is well-defined, and I is the identity operator matrix. Following the derivation in [8], the kernel weight covariance matrix, S^{-}_{n}, is calculated as $S^{-}_{n} = \Gamma_n S^{+}_{n-1}\Gamma^{\mathrm{T}}_{n} + V_n$, where V_n is the finite matrix representation of the transition residual matrix [8].

3) Update at Time n: The observation particles are updated based on the observation models in (2). The kernel mappings of observation particles in the kernel feature space are $\Upsilon_n = \left[\phi_{\mathbf{y}}(\mathbf{y}^{\{1\}}_{n,1:2}), \ldots, \phi_{\mathbf{y}}(\mathbf{y}^{\{M\}}_{n,1:2})\right]$. Based on the derivations in [8], the KME vector, the weight vector, and the kernel weight covariance matrix are updated as shown in Equations (11a) to (11c), respectively.

$$\hat{\mu}^{+}_{\mathbf{x}_n} = \hat{\mu}^{-}_{\mathbf{x}_n} + Q_n\left[\phi_{\mathbf{y}}(\mathbf{y}_n) - \hat{C}_{\mathbf{y}_n|\mathbf{x}_n}\hat{\mu}^{-}_{\mathbf{x}_n}\right] = \Phi_n\mathbf{w}^{+}_{n}, \tag{11a}$$

$$\mathbf{w}^{+}_{n} = \mathbf{w}^{-}_{n} + Q_n\left(G_{:,\mathbf{y}_n} - G_{\mathbf{yy}}\mathbf{w}^{-}_{n}\right) \tag{11b}$$

$$S^{+}_{n} = S^{-}_{n} - Q_n G_{\mathbf{yy}} S^{-}_{n}. \tag{11c}$$

Here, Q_n is the kernel Kalman gain, $G_{:,\mathbf{y}_n} = \Upsilon^{\mathrm{T}}_{n}\phi_{\mathbf{y}}(\mathbf{y}_n)$, and the Gram matrix of the observation at time n is $G_{\mathbf{yy}} = \Upsilon^{\mathrm{T}}_{n}\Upsilon_n$ [8].

IV. Simulation Results

The simulation parameters are set as follows: the initial state of the vehicle is set to $\mathbf{x}_1 = [-7.56, 3.75, 6.75, 0.4]^{\mathrm{T}}$, and the hard iron dipole moment of the vehicle is $\mathbf{m}_0 = [-203, 124, 267]^{\mathrm{T}}\mathrm{Am}^2$ [6]. The soft iron scalar is $D = 1\mathrm{m}^3$ [6]. The sensors' axes are $\mathbf{s}_1 = [0, 0, 0.3]^{\mathbf{T}}$ and $\mathbf{s}_2 = [0, 9, 0.7]^{\mathbf{T}}$, and the measurement noise covariance matrices are [6]

$$R_1 = 10^{-15}\begin{bmatrix} 0.1303 & -0.0073 & -0.0114 \\ -0.0073 & 0.1112 & 0.0117 \\ -0.0114 & 0.0117 & 0.1558 \end{bmatrix}$$

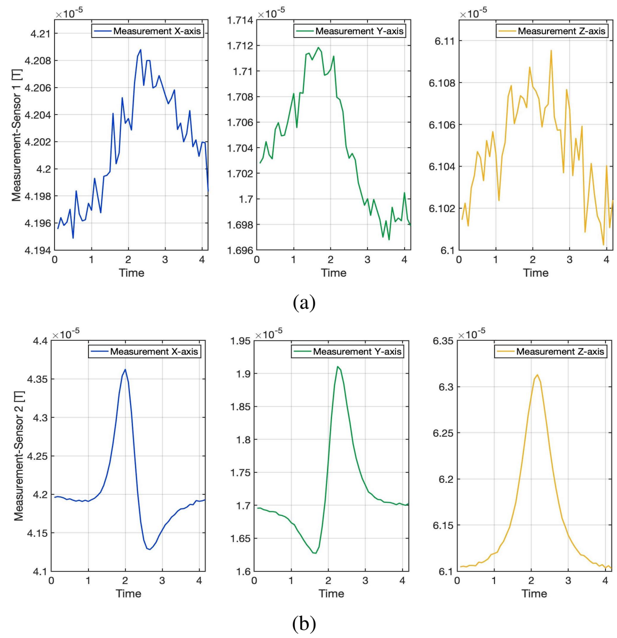

Fig. 2: Measured magnetic field strength in X/Y/Z axes at two sensors. (a) Sensor 1; (b) Sensor 2.

$$R_2 = 10^{-15}\begin{bmatrix} 0.1500 & 0.0205 & 0.0215 \\ 0.0205 & 0.1937 & 0.0310 \\ 0.0215 & 0.0310 & 0.1483 \end{bmatrix}.$$

Here, the unit of measurement is Telsa. The magnetic field strength measured in X-axis, Y-axis, and Z-axis at two sensors is shown in Fig. 2. The initial prior distribution of the hidden states for particles is drawn following the settings as $\xi^{\{i=1:M\}}_{0} \sim \mathcal{U}(-7.6, -7.4)$, $\eta^{\{i=1:M\}}_{0} \sim \mathcal{U}(6, 8)$, $\dot{\xi}^{\{i=1:M\}}_{0} \sim \mathcal{N}(\dot{\xi}_0, 10^{-2})$, $\eta^{\{i=1:M\}}_{0} \sim \mathcal{N}(\eta_0, 10^{-2})$, $z^{\{i=1:M\}}_{0} \sim \mathcal{N}(z_0, 10^{-1})$, $D^{\{i=1:M\}}_{0} \sim \mathcal{N}(D_0, 10^{-2})$, $\mathbf{m}^{\{i=1:M\}}_{0} \sim \mathcal{N}(\mathbf{m}_0, 10^3 I)$.

Fig. 3 displays a representative trajectory and the tracking performance obtained by the AKKF and the PF. Fig. 4 and Fig. 5 display the estimation performance of the hard iron dipole moment $\mathbf{m}_0$ and the soft iron scalar D, obtained from these two filters. The AKKF uses $M^{\mathrm{AKKF}} = 100$ particles, while $M^{\mathrm{PF}} = 2000$ particles are used for the PF. From Fig. 3 to Fig. 5, we can see that the AKKF with a smaller number of particles achieved favourable tracking and estimation performance compared to the PF with a large number of particles. We then compare the average root mean square error (RMSE) of the AKKF and the PF using the same number of particles, along with its standard deviation for tracking performance. RMSE is defined in (12). We obtain 100 Monte Carlo (MC) realisations with an increasing number of particles, specifically $M = [50, 100, 200]$, while the bootstrap PF with 2000 particles is considered as the benchmark performance, as shown in 6(a).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{N}(\xi_n - \hat{\xi}_n)^2 + (\eta_n - \hat{\eta}_n)^2}{N}}. \tag{12}$$

We also compared the RRMSE and its standard deviation

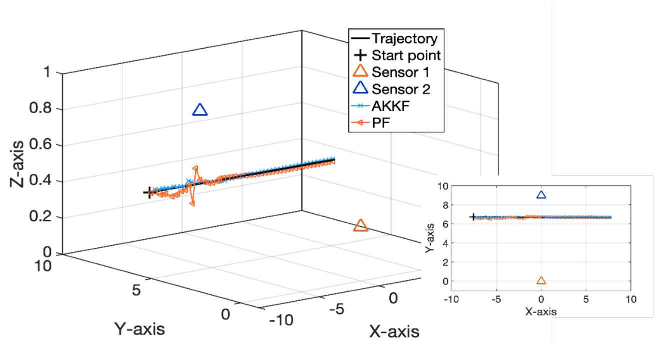

Fig. 3: Ground truth trajectory versus tracking performance achieved by the AKKF and the PF.

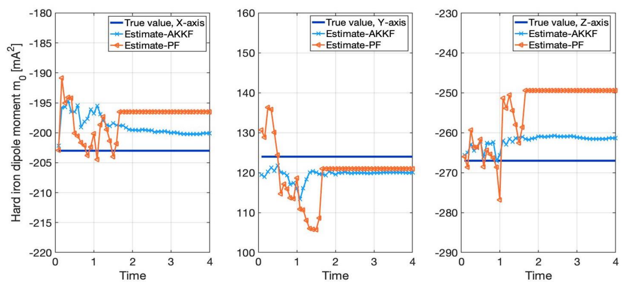

Fig. 4: True hard iron dipole moment $\mathbf{m}_0$ versus estimated values.

for the estimation performance of $\mathbf{m}_0$ and D, as well as the computation time, as shown in Figures 6(b) to 6(d), respectively.

Based on the simulation results, we draw the following conclusions: the proposed AKKF demonstrates significantly improved performance with the same number of particles compared to the PF, especially for trajectory tracking and magnetic moment strength estimation. For example, with 200 particles, the tracking accuracy can be improved by 0.13m, and magnetic moment strength estimation accuracy can be improved by 5%. Moreover, compared with the benchmark performance achieved by the PF with 2000 particles, the AKKF shows satisfactory tracking and estimation performance with significantly reduced computational complexity when dealing with high nonlinear and high-dimensional problems. This improved performance and reduced computational complexity are due to the ability of the AKKF to efficiently represent high-dimensional data using kernels, which can capture more information about the data in the rich feature space of the kernel. The feature mappings can then be used to perform computations more efficiently. In contrast, the PF works with the data directly and may struggle to handle high-dimensional data.

V. Conclusions

This paper explores a new application for the AKKF by utilising it for joint tracking and magnetic parameters estimation in high-dimensional and high nonlinear problems. The simulations presented demonstrate improved computational efficiency in vehicle tracking and magnetic parameter estimation.

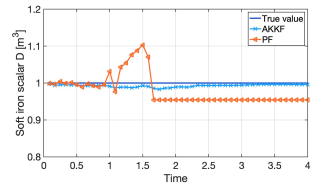

Fig. 5: True soft iron scalar D versus estimated values.

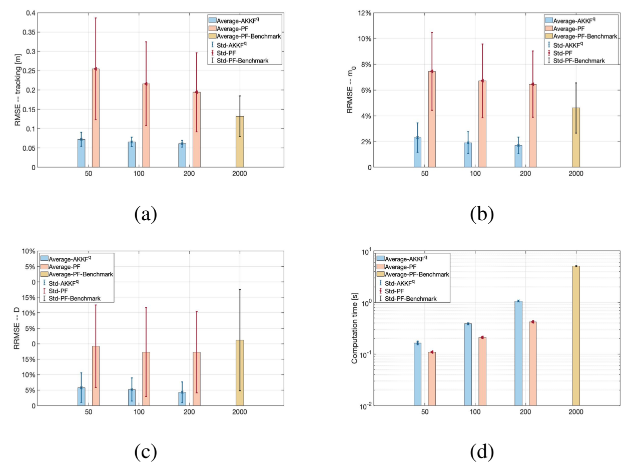

Fig. 6: Average and standard derivation of tracking RMSE. RMSE and computation performance (a) Tracking; (b) Hard iron dipole moment estimation; (c) Soft iron scalar estimation; (d) Computation time.

References

[1] J. Lenz and S. Edelstein, "Magnetic sensors and their applications," *IEEE Sensors Journal*, vol. 6, no. 3, pp. 631–649, 2006.

[2] R. Sithiravel, B. Balaji, B. Nelson, M. K. McDonald, R. Tharmarasa, and T. Kirubarajan, "Airborne maritime surveillance using magnetic anomaly detection signature," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3476–3490, 2020.

[3] P. Lin, N. Zhang, M. Chang, and L. Xu, "Research on the model and the location method of ship shaft-rate magnetic field based on rotating magnetic dipole," *IEEE Access*, vol. 8, pp. 162 999–163 005, 2020.

[4] C. Asci, W. Wang, and S. Sonkusale, "Security monitoring system using magnetically-activated rfid tags," in *2020 IEEE SENSORS*, 2020, pp. 1–4.

[5] N. Wahlström, J. Callmer, and F. Gustafsson, "Magnetometers for tracking metallic targets," in *2010 13th International Conference on Information Fusion*, 2010, pp. 1–8.

[6] N. Wahlström and F. Gustafsson, "Magnetometer modeling and validation for tracking metallic targets," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 545–556, 2014.

[7] V. Biancalana, R. Cecchi, P. Chessa, M. Mandalà, G. Bevilacqua, Y. Dancheva, and A. Vigilante, "Validation of a fast and accurate magnetic tracker operating in the environmental field," *Instruments*, vol. 5, no. 1, p. 11, mar 2021.

[8] M. Sun, M. E. Davies, I. K. Proudler, and J. R. Hopgood, "Adaptive kernel kalman filter," *IEEE Trans. Signal Process.*, vol. 71, pp. 713–726, 2023.

[9] ——, "Adaptive kernel kalman filter based belief propagation algorithm for maneuvering multi-target tracking," *IEEE Signal Process. Lett.*, vol. 29, pp. 1452–1456, 2022.

[10] ——, "Adaptive kernel kalman filter multi-sensor fusion," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, 2021, pp. 1–8.

[11] ——, "Adaptive kernel kalman filter," in *2021 Sensor Signal Processing for Defence Conference (SSPD)*, 2021, pp. 1–5.