A Taylor Series Approach to Correction of Input Errors in Gaussian Process Regression

Muzaffar Qureshi[1] Tochukwu Elijah Ogri[1] Zachary I. Bell[2] Wanjiku A. Makumi[2] Rushikesh Kamalapurkar[1]

***Abstract*— Gaussian Processes (GPs) are widely recognized as powerful non-parametric models for regression and classification. Traditional GP frameworks predominantly operate under the assumption that the inputs are either accurately known or subject to zero mean noise. However, several real-world applications such as mobile sensors have imperfect localization, leading to inputs with biased errors. These biases can typically be estimated through measurements collected over time using, for example, Kalman filters. To avoid recomputation of the entire GP model when better estimates of the inputs used in the training data become available, we introduce a technique for updating a trained GP model to incorporate updated estimates of the inputs. By leveraging the differentiability of the mean and covariance functions derived from the squared exponential kernel, a second-order correction algorithm is developed to update the trained GP models. Precomputed Jacobians and Hessians of kernels enable real-time refinement of the mean and covariance predictions. The efficacy of the developed approach is demonstrated using two simulation studies, with error analyses revealing improvements in both predictive accuracy and uncertainty quantification.**

I. Introduction

Accurate mapping of unknown fields is critical in various scientific and engineering domains, including environmental monitoring, search and rescue missions, and autonomous underwater exploration [1]–[3]. In many applications, a mobile agent equipped with sensors navigates through the environment to gather field measurements, which are used to construct a representation of the underlying field. Among a variety of available mapping techniques, Gaussian Processes (GPs) have emerged as a powerful approach due to their ability to capture complex spatial dependencies and to provide uncertainty quantification [4], [5].

The accuracy of the input data influences the performance of the GP regression [5]–[7]. In conventional GP-based field mapping, a mobile agent is directed to specific measurement locations to collect field values using onboard sensors [8]. However, due to localization errors caused by unmodeled agent dynamics, environmental disturbances, or calibration mismatches, the inputs to the GP, i.e., the position of the agent at the time measurements were collected, are not exactly known [9], [10]. As a result, the accuracy and reliability of the resulting GP model is degraded.

In field mapping problems, GPS sensors are commonly employed to mitigate these uncertainties by providing precise localization during the measurement process. However, GPS signals may be unavailable or unreliable in environments with occlusions, signal interference, or other challenging conditions [11], [12]. In the absence of GPS corrections, uncertainties in input locations propagate through the GP model, leading to degraded predictive performance.

In [13] and [14], Girard developed an analytical method for computing the mean and variance of predictions under input uncertainty, leveraging the squared exponential (SE) covariance function with Gaussian noise. Candela et al. [15] extended this framework by focusing on iterative forecasting for discrete-time non-linear dynamic systems. However, the applicability of these methods is constrained by their reliance on the assumption that the distributions of the input parameters are known. Furthermore, accurately estimating the parameters of the Gaussian distribution is challenging when dealing with complex or high-dimensional datasets [16].

Heteroscedastic Gaussian process (GP) regression offers a more flexible approach for handling varying noise distributions across the input space. By accounting for input-dependent noise (heteroscedasticity), this framework dynamically adjusts output variance, enabling more precise uncertainty quantification and improved predictive accuracy. The work of Kersting et al. [17] demonstrated the effectiveness of this approach in datasets with non-uniform noise levels. However, these methods primarily focus on inflating the GP covariance predictions while relying on deterministic input measurements for training the mean function.

The noisy input Gaussian process (NIGP) framework [18] extends the standard heteroscedastic GP by explicitly incorporating input uncertainty into the GP covariance structure. It achieves this by utilizing a local linear expansion around each input point, effectively treating input noise as output noise that is proportional to the squared gradient of the GP posterior mean. While this method effectively captures input noise in many scenarios, its accuracy hinges on the validity of the local linear approximation. This approximation may break down when dealing with highly nonlinear functions, leading to potential inaccuracies in the resulting predictions.

Another approach is to use the extended Kalman filter (EKF) to recursively estimate disturbance parameters contributing to input uncertainty [19], [20]. Unlike local linearization of the NIGP, the EKF captures the temporal evolution of states and disturbances, improving the input

This research was supported in part by the Air Force Research Laboratories under contract numbers FA8651-24-1-0019 and FA8651-23-1-0006 and the Office of Naval Research under contract number N00014-21-1-2481. Any opinions, findings, or recommendations in this article are those of the author(s), and do not necessarily reflect the views of the sponsoring agencies.

[1] Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida, USA, email: {muzaffar.qureshi, tochukwu.ogri, rkamalapurkar} @ufl.edu.

[2] Air Force Research Laboratories, Florida, USA, email: {zachary.bell.10, wanjiku.makumi} @us.af.mil.

accuracy and mapping fidelity. However, the EKF performance may degrade in the presence of strong nonlinearities or disturbances.

In summary, while current research on input errors in GP regression is focused on analyzing the impact of input uncertainty on the GP predictions, methods to update the GP model to account for updated input estimates have not been studied to the best of our knowledge. This paper introduces a novel technique for retroactively refining Gaussian Process (GP) models to account for deterministic input errors, without requiring retraining or revisiting measurement locations. Leveraging the differentiability properties of the GP kernel, our approach utilizes precomputed higher-order derivatives to efficiently adjust the mean and the covariance estimates of the GP model in real-time to incorporate corrected input locations. This two-stage framework comprises of an offline phase for computing the higher-order derivatives of the mean and covariance functions and an online phase for updating these functions when corrected input estimates become available. The contributions of this paper include

- The development of a technique for updating GP models by integrating corrections for localization-induced input errors using precomputed higher-order derivatives of the GP mean and covariance function.
- Providing detailed GP prediction error and computational time analyses.
- Extensive Monte Carlo simulations for two different cases to demonstrate the efficiency of the developed approach.

II. Problem Formulation

Consider a sensor-equipped agent operating in a domain $\mathcal{X} \subset \mathbb{R}^p$ that is tasked with visiting a set of measurement locations to create a map of a scalar field $h : \mathcal{X} \to \mathbb{R}$. The field can be modeled by GP as

$$\bar{h} \sim \text{GP}(\mu, \Sigma), \tag{1}$$

where $\mu : \mathbb{R}^p \to \mathbb{R}$ is the mean function and $\Sigma : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is the covariance function. To develop the GP model, consider a set of planned measurement locations given as $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_n\}$, where $n \in \mathbb{N}$ represents the total number of measurement locations. The corresponding field measurements at these planned locations are denoted as $\hat{\mathbf{Z}} = \{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n\}$, where $\hat{z}_i = h(\hat{\mathbf{x}}_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ denotes the zero-mean Gaussian measurement noise.

A squared exponential covariance kernel is selected to construct the GP, defined as

$$k(\hat{\mathbf{x}}, \hat{\mathbf{x}}') := \alpha^2 \exp\left(-\frac{||\hat{\mathbf{x}} - \hat{\mathbf{x}}'||^2}{2\beta^2}\right), \tag{2}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ represent any two measurement locations, α represents the amplitude, and β is the length scale. The squared exponential kernel is selected due to its smoothness properties and the universal function approximation property of the resulting GP [21].

Given any query location $\mathbf{x}_e$, the training kernel matrix $K_{n-n} \in \mathbb{R}^{n \times n}$, the test-train kernel matrix $K_{e-n} \in \mathbb{R}^{1 \times n}$, and the testing kernel matrix $K_{e-e} \in \mathbb{R}$ can be computed using the kernel function defined in (2) as

$$K_{n-n}(j,k) := k\left(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}_k\right), \tag{3}$$

$$K_{e-e} := k(\mathbf{x}_e, \mathbf{x}_e), \tag{4}$$

$$K_{e-n}(k) := k(\mathbf{x}_e, \hat{\mathbf{x}}_k). \tag{5}$$

Given the set of planned measurement locations $\hat{\mathbf{X}}$, and corresponding measurements $\hat{\mathbf{Z}}$, the GP model is used to predict the mean and covariance functions evaluated at the given test point $\mathbf{x}_e$. The predicted mean value of the GP model at $\mathbf{x}_e$, is given by

$$\hat{\mathbf{M}}(\hat{\mathbf{X}}, \mathbf{x}_e) = K_{e-n} \cdot (K_{n-n})^{-1} \mathbf{Z}. \tag{6}$$

The covariance matrix $\hat{\mathbf{S}}$, which captures the uncertainty in the GP prediction, is computed as

$$\hat{\mathbf{S}}(\hat{\mathbf{X}}, \mathbf{x}_e) = K_{e-e} - K_{e-n} \cdot (K_{n-n})^{-1} \cdot (K_{e-n})^{\top}. \tag{7}$$

In an ideal scenario, the agent would exactly follow a planned trajectory to visit all locations $\hat{\mathbf{X}}$ to collect the field measurements $\hat{\mathbf{Z}}$. However, due to unmodeled dynamics, the trajectory of the agent deviates from the planned trajectory, resulting in measurement locations that differ from planned locations. The actual measurement locations are denoted as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. The corresponding field measurements are denoted by $\mathbf{Z} = \{z_1, z_2, \ldots, z_n\}$, where $z_i = h(\mathbf{x}_i) + \epsilon_i$ represents the true field value at location $\mathbf{x}_i$ corrupted by measurement noise ϵ_i. In the absence of external localization sources such as GPS, the agent trains the GP model using the planned measurement locations $\hat{\mathbf{X}}$ and the collected field measurements $\mathbf{Z}$.

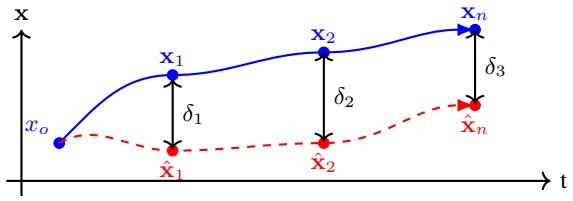

Fig. 1. The schematic of the planned and the actual state trajectory along with the planned and actual measurement locations.

The error between the planned and actual measurement locations at each measurement point can be expressed as $\hat{\mathbf{x}}_i = \mathbf{x}_i + \delta_i$, $\forall i \in \{1, 2, \ldots, n\}$, where $\delta_i \in \mathbb{R}^p$ denotes the localization error at measurement step i. To minimize the impact of these localization errors on the GP model, it is necessary to either acquire measurements $\hat{\mathbf{Z}}$ at the planned locations $\hat{\mathbf{X}}$ or obtain accurate information about the actual measurement locations, $\mathbf{X}$. This work assumes that a state estimation technique, such as visual simultaneous localization and mapping (Visual SLAM) or a disturbance observer, is employed to incrementally refine the agent state estimates over time. As these estimates improve, the localization error at measurement step δ_i also reduces.

The objective is to develop a technique that enables an agent to refine a GP model trained on $\{\hat{\mathbf{X}}, \mathbf{Z}\}$ by leveraging

input error information $\{\delta_i\}$. The goal is to match the prediction accuracy of a model trained on $\{\mathbf{X}, \mathbf{Z}\}$. To facilitate online computation, we aim to enhance the accuracy of the trained GP model without the need for new measurements or recomputation of the entire GP model using new state estimates. By utilizing input error information, this method improves prediction accuracy and reduces the computational costs associated with obtaining new measurements and retraining. The following assumptions are made to keep the problem tractable.

Assumption 1: The field h is a smooth and bounded function with $h(\mathbf{x}) \in [h_{\min}, h_{\max}], \forall \mathbf{x} \in \mathcal{X}$.

Assumption 2: The measurement noise ϵ_i is assumed to i.i.d and uncorrelated, i.e., $\mathbb{E}[\epsilon_i \epsilon_j] = 0, \forall i \neq j$.

Assumption 3: The error between the planned and the actual measurement locations is bounded as $\|\delta_i\| \leq \delta_{\max}, \forall i = 1, 2, \dots n$.

Remark 1: The i.i.d. measurement noise assumption aligns with standard Gaussian process modeling. The bounded localization error assumption reflects practical scenarios, where localization errors are uniformly bounded, to ensure the validity of local approximations and contribute to the stability of corrective methods.

III. Overview

In this paper, a method for correcting a pre-trained GP model using an offline-online approach is developed. In the offline phase, we leverage the differentiability properties of the kernel function to compute higher-order derivatives of the mean and covariance functions with respect to the planned measurement locations. These analytical derivatives are evaluated at the planned measurement locations and stored for use during the online phase.

In the online phase, the stored higher-order derivatives are combined with the input error δ_i to update the GP model. This approach allows the GP model to adapt to localization corrections without requiring new measurements or retraining the entire model.

The effectiveness of the developed method will be evaluated by comparing the corrected GP model to the ideal GP model that would have been trained directly on the actual measurement locations X and the corresponding field values Z. Additionally, the computational complexity of the developed method will be compared with a generic GP model to demonstrate the effectiveness of the approach in real-time adaptation.

IV. GP Prediction with Input Errors

The predictive mean, $\hat{\mathbf{M}}$, in GP regression represents a linear combination of the observed measurements, $\mathbf{Z}$, weighted by the correlation between the query point, $\mathbf{x}_e$, and the training points, $\hat{\mathbf{X}}$. These weighting factors are derived from the kernel matrix K_{e-n} in (5) and the inverse of the kernel matrix K_{n-n} in (3), which both quantify the sensitivity of the GP prediction to changes in measurement locations.

The predictive covariance, $\hat{\mathbf{S}}$, quantifies the uncertainty in the predictions at the query point $\mathbf{x}_e$. The predicted covariance is governed by the matrix K_{e-e}, defined in (4), which represents the inherent uncertainty at the query locations without considering the training data, and a reduction term that reflects how well the training points span the query locations, computed using K_{e-n} and K_{n-n}^{-1}.

To quantify the impact of the error between the true measurement locations, $\mathbf{x}$, and the planned locations, $\hat{\mathbf{X}}$, on the mean and covariance functions, a second order Taylor series expansion is performed around the planned measurement locations $\hat{\mathbf{x}}_i$ for all $i = 1, \dots, n$. This analysis yields the corrected mean vector $\mathbf{M} \in \mathbb{R}$, expressed as

$$\mathbf{M}(\hat{\mathbf{X}}, \mathbf{x}_e) \coloneqq \hat{\mathbf{M}}(\hat{\mathbf{X}}, \mathbf{x}_e) + \sum_{i=1}^{n} \frac{\partial \hat{\mathbf{M}}(\hat{\mathbf{X}}, \mathbf{x}_e)}{\partial \hat{\mathbf{x}}_i} \delta_i + \frac{1}{2} \sum_{i=1}^{n} \frac{\partial^2 \hat{\mathbf{M}}(\hat{\mathbf{X}}, \mathbf{x}_e)}{\partial \hat{\mathbf{x}}_i^2} \delta_i^2, \quad (8)$$

where δ_i represents the perturbation at the i-th measurement location and $\mathbf{M}$ denotes the corrected GP model. The first-order partial derivatives $\frac{\partial \mathbf{M}}{\partial \mathbf{x}_i}$ can be stacked to obtain a Jacobian matrix $\mathbf{J}_M \in \mathbb{R}^{1 \times n}$ denoted as

$$\mathbf{J}_M \coloneqq \begin{bmatrix} \frac{\partial \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_1} & \frac{\partial \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_2} & \dots & \frac{\partial \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_n} \end{bmatrix}. \quad (9)$$

The second-order partial derivatives $\frac{\partial^2 \mathbf{M}}{\partial \mathbf{x}_i \mathbf{x}_j}$ form the elements of the symmetric Hessian matrix $\mathbf{H}_M \in \mathbb{R}^{n \times n}$ given by

$$\mathbf{H}_M \coloneqq \begin{bmatrix} \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_1^2} & \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_1 \partial \hat{\mathbf{x}}_2} & \dots & \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_1 \partial \hat{\mathbf{x}}_n} \\ \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_2 \partial \hat{\mathbf{x}}_1} & \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_2^2} & \dots & \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_2 \partial \hat{\mathbf{x}}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_n \partial \hat{\mathbf{x}}_1} & \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_n \partial \hat{\mathbf{x}}_2} & \dots & \frac{\partial^2 \hat{\mathbf{M}}}{\partial \hat{\mathbf{x}}_n^2} \end{bmatrix}. \quad (10)$$

Remark 2: The higher-order derivatives of the mean function can be computed similarly, following the same approach. The derivatives of the covariance matrix can also be computed using the Taylor series approximation around $\hat{\mathbf{x}}_i$ to yield the second order approximation

$$\mathbf{S}(\hat{\mathbf{X}}, \mathbf{x}_e) \coloneqq \hat{\mathbf{S}}(\hat{\mathbf{X}}, \mathbf{x}_e) + \sum_{i=1}^{n} \frac{\partial \hat{\mathbf{S}}(\hat{\mathbf{X}}, \mathbf{x}_e)}{\partial \hat{\mathbf{x}}_i} \delta_i + \frac{1}{2} \sum_{i=1}^{n} \frac{\partial^2 \hat{\mathbf{S}}(\hat{\mathbf{X}}, \mathbf{x}_e)}{\partial \hat{\mathbf{x}}_i^2} \delta_i^2. \quad (11)$$

Similarly, the derivatives of the covariance function can be defined as

$$\mathbf{J}_s \coloneqq \begin{bmatrix} \frac{\partial \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_1} & \frac{\partial \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_2} & \dots & \frac{\partial \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_n} \end{bmatrix}. \quad (12)$$

$$\mathbf{H}_S \coloneqq \begin{bmatrix} \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_1^2} & \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_1 \partial \hat{\mathbf{x}}_2} & \dots & \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_1 \partial \hat{\mathbf{x}}_n} \\ \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_2 \partial \hat{\mathbf{x}}_1} & \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_2^2} & \dots & \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_2 \partial \hat{\mathbf{x}}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_n \partial \hat{\mathbf{x}}_1} & \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_n \partial \hat{\mathbf{x}}_2} & \dots & \frac{\partial^2 \hat{\mathbf{S}}}{\partial \hat{\mathbf{x}}_n^2} \end{bmatrix}. \quad (13)$$

The differentiability properties of the mean and covariance functions are analyzed in the next section using the differentiability properties of the squared exponential (SE) kernel.

V. Differentiability Properties of SE Kernel

The squared exponential (SE) kernel function, defined in (2), is infinitely differentiable with respect to the input location $\hat{\mathbf{x}}$, guaranteeing the existence and continuity of all its derivatives for all $\hat{\mathbf{x}} \in \mathcal{X}$. This smoothness property is crucial for enabling gradient-based correction methods for GP models.

To generalize the GP prediction framework to accommodate multiple query locations, let $t \in \mathbb{N}$ be the number of query locations and let the query set be denoted as $\mathbf{X}_e = \{\mathbf{x}_{e,1}, \mathbf{x}_{e,2}, \ldots, \mathbf{x}_{e,t}\}$, where each $\mathbf{x}_{e,i} \in \mathbb{R}^p$ denotes a query location within the domain $\mathcal{X}$. Evaluation of the kernel and its derivatives at all points in $\mathbf{X}_e$ yields the matrices $\mathcal{K}_{e-e} \in \mathbb{R}^{t\times t}$ and $\mathcal{K}_{e-n} \in \mathbb{R}^{t\times n}$ with $\mathcal{K}_{n-n} \coloneqq K_{n-n} \in \mathbb{R}^{n\times n}$. The predicted mean and covariance functions can then be evaluated at the test points $\mathbf{X}_e$ as

$$\hat{\mathcal{M}}(\hat{\mathbf{X}}, \mathbf{X}_e) \coloneqq \mathcal{K}_{e-n} \cdot (\mathcal{K}_{n-n})^{-1}\mathbf{Z}, \tag{14}$$

$$\hat{\mathcal{S}}(\hat{\mathbf{X}}, \mathbf{X}_e) \coloneqq \mathcal{K}_{e-e} - \mathcal{K}_{e-n} \cdot (\mathcal{K}_{n-n})^{-1} \cdot (\mathcal{K}_{e-n})^{\top}, \tag{15}$$

where $\hat{\mathcal{M}} \in \mathbb{R}^t$ and $\hat{\mathcal{S}} \in \mathbb{R}^{t\times t}$ denote the higher dimensional versions of (6) and (7), respectively.

Differentiating the kernel function $k(\mathbf{x}, \hat{\mathbf{x}}) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, in (2) with respect to measurement location $\hat{\mathbf{x}}$, and evaluating at $(\mathbf{x}, \hat{\mathbf{x}})$ yields $\nabla k(\mathbf{x}, \hat{\mathbf{x}}) \in \mathbb{R}^p$, where $\nabla k(\cdot, \cdot)$ denotes the gradient operator. Thus, the matrix of kernel derivatives with respect to the training input $\hat{\mathbf{x}}_k$ contains nonzero entries only in the k-th column, and can be expressed in structured form as

$$\nabla\mathcal{K}_{e-n} \coloneqq \begin{bmatrix} 0 & \cdots & 0 & \nabla k(\mathbf{x}_{e,1}, \hat{\mathbf{x}}_k) & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \nabla k(\mathbf{x}_{e,2}, \hat{\mathbf{x}}_k) & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \nabla k(\mathbf{x}_{e,t}, \hat{\mathbf{x}}_k) & 0 & \cdots & 0 \end{bmatrix}, \tag{16}$$

where $\nabla\mathcal{K}_{e-n} \in \mathbb{R}^{t\times n\times p}$ and $\hat{\mathbf{x}}_k \in \hat{\mathbf{X}}$ denotes the k-th training point. Similarly, differentiation of $\mathcal{K}_{n-n}$ with respect to the training point $\hat{\mathbf{x}}_k$ affects both the k-th row and the k-th column. The matrix $\nabla\mathcal{K}_{n-n} \in \mathbb{R}^{n\times n\times p}$ can then be expressed as

$$\nabla\mathcal{K}_{n-n} \coloneqq \begin{bmatrix} 0 & \cdots & \nabla k(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_k) & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ \nabla k(\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_1) & \cdots & 0 & \cdots & \nabla k(\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_n) \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & \nabla k(\hat{\mathbf{x}}_n, \hat{\mathbf{x}}_k) & \cdots & 0 \end{bmatrix}. \tag{17}$$

Since each training point $\hat{\mathbf{x}}_i \in \mathbb{R}^p$ is a vector, we consider perturbing a single component $j \in \{1, \ldots, p\}$ of this vector. For each such dimension, the partial derivatives $(\nabla\mathcal{K}_{e-n})^j \in \mathbb{R}^{t\times n}$ and $(\nabla\mathcal{K}_{n-n})^j \in \mathbb{R}^{n\times n}$ represent the directional derivatives of the corresponding kernel matrices with respect to the j-th coordinate of the training input.

Applying the chain rule to (14), the derivative of the GP mean vector with respect to the j-th component of the training input $\hat{\mathbf{x}}_i$ is given by

$$\frac{\partial\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}_i^j} = (\nabla\mathcal{K}_{e-n})_i^j \cdot \left(\mathcal{K}_{n-n}\right)^{-1}\mathbf{Z} + \mathcal{K}_{e-n} \cdot \left(\nabla\mathcal{K}_{n-n}^{-1}\right)_i^j \cdot \mathbf{Z}. \tag{18}$$

Stacking the resulting column vectors for each dimension $j = 1, \ldots, p$ yields the full Jacobian matrix

$$\frac{\partial\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}_i} = \left[\frac{\partial\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}_i^1}, \frac{\partial\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}_i^2}, \cdots, \frac{\partial\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}_i^p}\right] \in \mathbb{R}^{t\times p}. \tag{19}$$

Leveraging the prior knowledge of $\hat{\mathbf{X}}$ and $\mathbf{X}_e$, these derivative $\frac{\partial\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}_i} \in \mathbb{R}^{t\times p}$ can be computed as a linear function of measurements Z. Specifically, we can write

$$\frac{\partial\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}} = f(\hat{\mathbf{X}}, \mathbf{X}_e, \mathbf{Z}), \qquad \frac{\partial^2\hat{\mathcal{M}}}{\partial\hat{\mathbf{x}}^2} = g(\hat{\mathbf{X}}, \mathbf{X}_e, \mathbf{Z}), \tag{20}$$

where $f : \mathbb{R}^p \times \mathbb{R}^t \times \mathbb{R} \mapsto \mathbb{R}^{t\times p}$ and $g : \mathbb{R}^p \times \mathbb{R}^t \times \mathbb{R} \mapsto \mathbb{R}^{t\times p\times p}$ are operator-valued functions. These functions can be used to update the mean vector online using the update law given in (8). Similarly, the derivative to update the covariance function according to the update law in (15) can be calculated as

$$\begin{aligned}\frac{\partial\hat{\mathcal{S}}}{\partial\hat{\mathbf{x}}_i^j} = &- (\nabla\mathcal{K}_{e-n})_i^j \cdot \mathcal{K}_{n-n}^{-1} \cdot \mathcal{K}_{e-n}^{\top} \\ &- \mathcal{K}_{e-n} \cdot \left(\nabla\mathcal{K}_{n-n}^{-1}\right)_i^j \cdot \mathcal{K}_{e-n}^{\top} \\ &- \mathcal{K}_{e-n} \cdot \mathcal{K}_{n-n}^{-1} \cdot \left((\nabla\mathcal{K}_{e-n})_i^j\right)^{\top}.\end{aligned} \tag{21}$$

The full Jacobian tensor of the covariance with respect to $\hat{\mathbf{x}}_i \in \mathbb{R}^p$ is constructed by stacking the component-wise derivatives

$$\frac{\partial\hat{\mathcal{S}}}{\partial\hat{\mathbf{x}}_i} = \left[\frac{\partial\hat{\mathcal{S}}}{\partial\hat{\mathbf{x}}_i^1}, \frac{\partial\hat{\mathcal{S}}}{\partial\hat{\mathbf{x}}_i^2}, \cdots, \frac{\partial\hat{\mathcal{S}}}{\partial\hat{\mathbf{x}}_i^p}\right] \in \mathbb{R}^{t\times t\times p}. \tag{22}$$

Remark 3: The covariance derivatives $\frac{\partial\hat{\mathcal{S}}}{\partial\hat{\mathbf{x}}_i}$ and $\frac{\partial^2\hat{\mathcal{S}}}{\partial\hat{\mathbf{x}}_i^2}$ depend on the measurement and query locations and can be numerically computed offline without $\mathbf{Z}$.

By precomputing the necessary derivative matrices using prior information from the planned measurement locations, one can derive closed-form update laws for the GP predictive mean vector $\mathcal{M} \in \mathbb{R}^t$ and covariance matrix $\mathcal{S} \in \mathbb{R}^{t\times t}$, as presented in (8) and (11), respectively. These updates are expressed as functions of the measurement vector $\mathbf{Z}$ and the perturbation δ, and can be evaluated online with a computational complexity of $\mathcal{O}(n)$ per test point. These updates represent a significant improvement over full GP recomputation, which requires $\mathcal{O}(n^3)$ operations due to matrix inversion. The complete GP update procedure is summarized in Algorithm 1.

Algorithm 1 GP update algorithm under deterministic input errors

Input: Planned measurement locations $\hat{\mathbf{X}}$, actual measurements $\mathbf{Z}$, initial GP parameters α and β, and kernel function k
Offline Phase:
Compute and store derivatives of mean and covariance functions
$\frac{\partial \hat{\mathcal{M}}}{\partial \mathbf{x}_i}$, $\frac{\partial^2 \hat{\mathcal{M}}}{\partial \mathbf{x}_i^2}$ (Eq. (20))
$\frac{\partial \hat{\mathcal{S}}}{\partial \mathbf{x}_i}$, $\frac{\partial^2 \hat{\mathcal{S}}}{\partial \mathbf{x}_i^2}$ (Eq. (22))
Online Phase:
Initialize: GP model with actual locations $\mathbf{X}$ and field values $\mathbf{Z}$
for each time step $t = 1, 2, \ldots, n$ **do**
Obtain the disturbance information δ_i and $\mathbf{Z}$
Update $\hat{\mathcal{M}}$ and $\hat{\mathcal{S}}$ using Eq. (8)-(11)
end for
Evaluate both GP models $\mathcal{M}$ and $\hat{\mathcal{M}}$ at query locations $\mathbf{X}_e$, and compute error between them
Output: Corrected mean vector $\mathcal{M}$ and covariance matrix $\mathcal{S}$

VI. Convergence Analysis

GP regression relies on the properties of its mean function over the compact input domain $\mathcal{X}$, which can be expressed in terms of the kernel function. The following theorem demonstrates that the GP mean function satisfies the conditions for Taylor series expansion.

Theorem 1: Let $\hat{\mathcal{M}} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ denote the mean function of the GP, defined with corrupted input locations $\hat{\mathbf{x}}$. If the kernel function k is analytic on the input domain $\mathcal{X} \subset \mathbb{R}^p$, the mean function $\hat{\mathcal{M}}$ is also real analytic and can be represented as a Taylor series expansion around $\hat{\mathbf{x}}_0 \in \mathcal{X}$ given as

$$\mathcal{M} = \sum_{N=0}^{\infty} \frac{1}{N!} \nabla^N \hat{\mathcal{M}}(\hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^N, \tag{23}$$

where $\nabla^n \hat{\mathcal{M}}(\hat{\mathbf{x}}_0)$ is the N-th order derivative tensor of $\hat{\mathcal{M}}$ evaluated at $\hat{\mathbf{x}}_0$, and $(\mathbf{x} - \hat{\mathbf{x}}_0)^N$ denotes the n-fold tensor product of $(\mathbf{x} - \hat{\mathbf{x}}_0)$. Furthermore the Taylor series converges uniformly to $\mathcal{M}$ on any compact subset of $\mathcal{X}$.

Proof: The mean function of a Gaussian Process is defined as

$$\hat{\mathcal{M}}(\mathbf{x}_e, \hat{\mathbf{x}}) = \mathcal{K}_{e-n}(\mathbf{x}_e, \hat{\mathbf{x}}) \cdot \mathcal{K}_{n-n}(\hat{\mathbf{x}})^{-1} \mathbf{Z}. \tag{24}$$

Since the kernel function k is analytic and continuously differentiable for all $\hat{\mathbf{x}} \in \mathcal{X}$, each entry in the kernel matrices K_{e-n} and $\mathcal{K}_{n-n}$ is analytic, consequently $\hat{\mathcal{M}}$ is also an analytic function. Therefore, $\mathcal{M}$ can be expressed as a Taylor series around any point $\hat{\mathbf{x}}_0 \in \mathcal{D}$

$$\hat{\mathcal{M}} = \sum_{N=0}^{\infty} \frac{1}{N}! \nabla^N \hat{\mathcal{M}}(\hat{\mathbf{x}}_0) \cdot (\mathbf{x} - \hat{\mathbf{x}}_0)^N, \tag{25}$$

where $\mathcal{D}$ is any neighborhood of $\hat{\mathbf{x}}_0$. Given any compact subset $\mathcal{V} \subset \mathcal{X}$, it holds that $\|(\mathbf{x} - \hat{\mathbf{x}}_0)\| \leq L_f$ for all $\mathbf{x} \in \mathcal{V}$, where $L_f > 0$ is some constant. Using the fact that k is analytic, the Extreme Value Theorem can be used to conclude that there exists a constant $L_M > 0$ such that

$$\|\nabla^{N+1} \hat{\mathcal{M}}(\kappa)\| \leq L_m, \quad \forall \kappa \in \mathcal{V}, \tag{26}$$

where κ lies on the line segment between $\hat{\mathbf{x}}_0$ and $\mathbf{x}$. By the Lagrange Remainder Theorem, we have that for each point $\mathbf{x} \neq \hat{\mathbf{x}}_0$ in $\mathcal{V}$, there is a point κ strictly between $\mathbf{x}$ and $\hat{\mathbf{x}}_0$ such that $R_N(\mathbf{x}) = \mathcal{M}(\mathbf{x}) - \hat{\mathcal{M}}(\mathbf{x})$ is given by

$$R_N(\mathbf{x}) = \frac{1}{(N+1)!} \nabla^{N+1} \hat{\mathcal{M}}(\kappa) \cdot (\mathbf{x} - \hat{\mathbf{x}}_0)^{N+1}. \tag{27}$$

Applying upper bounds to the remainder $R_N(\mathbf{x})$ in (26) yields

$$|R_N(\mathbf{x})| \leq L_M \cdot \frac{L_f^{n+1}}{(N+1)!}. \tag{28}$$

The factorial $(N+1)!$ dominates the polynomial growth of L_f^{N+1}, ensuring that $R_N(\mathbf{x}) \to 0$ as $N \to \infty$. This convergence is uniform on $\mathcal{V}$ because L_m and L_f are constants independent of $\mathbf{x}$. Therefore, the Taylor series converges uniformly to $\mathcal{M}$ on the compact set $\mathcal{V}$. ■

Remark 4: The mean function $\hat{\mathcal{M}}$, expanded around planned points $\hat{\mathbf{x}}$, satisfies the conditions for a Taylor series. This allows efficient local approximations of $\hat{\mathcal{M}}$ using higher-order derivatives, demonstrating that the GP model can be expressed as a Taylor series on compact subsets of $\mathcal{X}$.

Building on Theorem 1, the following theorem determines the bounds on the required number of higher-order derivatives to achieve the desired approximation accuracy within a compact subset of the input domain.

Theorem 2: Let $\mathcal{V} \subset \mathcal{D}$ be a compact subset of the input domain, and assume that for all $\mathbf{x} \in \mathcal{V}$, the distance from the Taylor series evaluation point satisfies $\|\mathbf{x} - \hat{\mathbf{x}}_0\| \leq \beta$, where $\beta > 0$ is a constant. Given a desired approximation accuracy $\epsilon > 0$, the minimum number of derivatives N required in the Taylor series expansion of the GP mean function $\mathcal{M}$ around $\hat{\mathbf{x}}_0$ to ensure that the approximation error does not exceed ϵ on $\mathcal{V}$ is

$$N = \left\lceil \frac{\log\left(\frac{\epsilon}{L_m}\right)}{\log(\beta)} \right\rceil, \tag{29}$$

where $L_m > 0$ is a constant bounding the $(N+1)$-th order derivative tensor $\nabla^{N+1} \mathcal{M}(\kappa)$ for all $\kappa \in \mathcal{V}$.

Proof: From Theorem 1, the remainder term R_N after truncating the Taylor series of $\mathcal{M}$ at order N is given by

$$\|R_N\| \leq \frac{L_m}{\beta}^{N+1} (N+1)!, \tag{30}$$

where L_m is a uniform bound such that $\|\nabla^{N+1} \hat{\mathcal{M}}(\hat{\mathbf{x}})\| \leq L_m$ for all $\kappa \in \mathcal{V}$, and $\|\mathbf{x} - \hat{\mathbf{x}}_0\| \leq \beta$. To achieve the desired approximation accuracy ϵ, we require $\frac{L_m}{\beta}^{N+1}(N+1)! \leq \epsilon$. Using Stirling approximation for the factorial term, $(N+1)! \approx \sqrt{2\pi(N+1)} \left(\frac{N+1}{e}\right)^{N+1}$, the inequality becomes

$$\frac{L_m}{\beta}^{N+1} \sqrt{2\pi(N+1)} \left(\frac{N+1}{e}\right)^{N+1} \leq \epsilon. \tag{31}$$

For sufficiently large N, the dominant terms yield $\beta^{N+1} \leq \epsilon \cdot \frac{(N+1)^{N+1}}{L_m} e^{N+1}$. Taking the natural logarithm of both sides,

$$(N+1)\log(\beta) \leq \log\left(\frac{\epsilon}{L_m}\right) + (N+1)\log(N+1) - (N+1). \tag{32}$$

Simplifying and focusing on the leading terms for large N, we obtain

$$(N+1)\log(\beta) \approx \log\left(\frac{\epsilon}{L_m}\right). \tag{33}$$

Solving for N yields $N \approx \frac{\log\left(\frac{\epsilon}{L_m}\right)}{\log(\beta)} - 1$. Since N must be an integer, we take the ceiling of the right-hand side to ensure the approximation error does not exceed ϵ as

$$N = \left\lceil \frac{\log\left(\frac{\epsilon}{L_m}\right)}{\log(\beta)} \right\rceil. \tag{34}$$

Thus, the number of derivatives N required to achieve the desired accuracy ϵ scales logarithmically with $\frac{\epsilon}{L_m}$ and β. ■

Remark 5: The derived bound in Theorem 2 offers insights into the computational resources needed for accurate local approximations of the GP mean function.

VII. Simulation Results

Two simulations are conducted to validate the developed method for refining GP models trained on noisy measurements. The first simulation, performed in a 1D domain, introduces spatially varying perturbations to measurement locations, simulating random, location-dependent noise. The second simulation, conducted in a 2D domain, applies a constant offset δ to all locations, modeling uniform sensor bias across the field.

A. 1-Dimensional Example

The first simulation study considers a one-dimensional function defined as $h_1(\mathbf{x}) = \sin(2\pi\mathbf{x})$, where $\mathbf{x} \in [0, 1]$. Measurement locations were uniformly distributed over the interval $[0, 1]$, selected as $\hat{\mathbf{X}} = \{0, 0.1, 0.2, \ldots, 1\}$. A squared exponential kernel was used with hyperparameters $\alpha = 0.1$ and $\beta = 0.2$. The GP model was constructed symbolically by treating measurement locations and measurements as symbolic variables. The Jacobians and Hessians with respect to measurement locations in (20) and (22), were then computed using CasADi symbolic framework for numerical differentiation.

A perfect GP model was trained on the original measurement locations and their corresponding values, serving as a baseline for evaluating the error between different GP models. The predictive performance was assessed at 100 uniformly distributed query locations over the interval $[0, 1]$. The resulting mean function and associated confidence intervals are shown in Figure 2. A Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.01^2)$ was added to planned locations to represent actual measurement locations. The corrupted GP model was trained on the planned locations and the updated measurements. The resulting GP model is shown in Figure 3, highlighting the impact of location perturbations on the predictions of the GP. Given the error vector $\delta_i \sim [0, 0.03]$, the corrections were applied to the corrupted mean $\hat{\mathcal{M}}$ and covariance $\hat{\mathcal{S}}$ using the derivative matrices, resulting in the corrected GP model as illustrated in Figure 4. The absolute errors between the estimates of the corrupted and perfect GP model and the corrected and perfect GP model were computed and are shown in Figure 5 and Figure 6 for mean and covariance estimates, respectively. Additionally, the computational time required for the correction process was measured and compared to the time needed to compute the full GP model from scratch. The results are summarized in Table I and Table II.

TABLE I
AVERAGE PERCENTAGE IMPROVEMENTS IN 1000 SIMULATIONS IN 1D CASE

Improvement	1D Sim	2D Sim
Corrupted GP $\|\|e\|\|$	1.2653	1.939
Corrected GP $\|\|e\|\|$	1.055	1.2296
Improvement Percentage	79.27 %	71.785 %

TABLE II
AVERAGE TIME IMPROVEMENTS IN 1000 SIMULATIONS IN 2D CASE

Method	Time (secs) 1D	Time (secs) 2D
Corrupted GP	0.019	0.023
Corrected GP	0.000022	0.000036

B. 2-Dimensional Example

For the second simulation study, a two-dimensional scalar field $h_2(\mathbf{x})$ is defined on a bounded domain $\mathcal{X} \subseteq \mathbb{R}^2$ as $g(\mathbf{x}) = \sin(2\pi x) \cdot \cos(2\pi y)$, $\mathbf{x} = (x, y) \in [0, 1]$. The scalar field can be visualized in Figure 7. Following the same procedure as the one-dimensional case, a baseline GP model is trained on perfect measurement locations as shown in Figure 8. To simulate sensor bias, the x-coordinates of all measurement locations are corrupted with a fixed perturbation $\delta_i = 0.1$ for all $i \in \{1, 2, \ldots, n\}$ and shown in Figure 8. A GP model is trained on these corrupted measurement locations while using the actual scalar field values as measurements. The corrected GP predictions using the corrupted GP model. The error trajectories for the corrupted and corrected mean functions are presented in Figure 9, showing significant error reduction after applying the correction mechanism.

VIII. Discussion

The developed correction method addresses deterministic input errors in scalar field mapping by integrating known perturbations δ_i into the framework. By precomputing Jacobians and Hessians during an offline phase, the developed method enables efficient online updates to the GP mean and covariance estimates without revisiting actual measurement locations or recomputing the entire GP model. Simulation results demonstrate significant error reduction compared to standard GP regression (Table I) and improvements in computational time (Table II), highlighting the

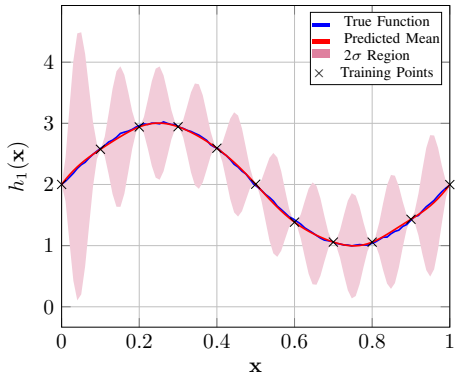

Fig. 2. The sinusoidal test functions are plotted with the GP mean and covariance estimates.

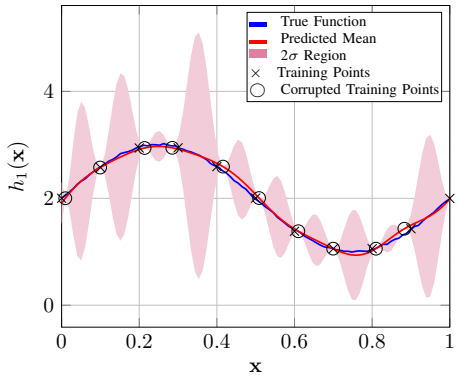

Fig. 3. This figure shows GP mean and covariance estimates with corrupted training points

efficacy of the developed method. The method assumes that the perturbations δ_i are perfectly known, implying full-state observability of the agent. However, in practical scenarios only partial state measurements may be available, requiring additional estimation strategies to infer perturbations from limited observations. Furthermore, while the method enables efficient online updates, the size of the derivative matrices can scale unfavorably for large GP models. To address this, approximation techniques or dimensionality reduction methods may be necessary during the offline phase to ensure computational tractability.

IX. Conclusion

This paper develops a method for refining GP by leveraging deterministic input perturbations to correct the mean and the covariance estimates. The developed method enhances prediction accuracy without requiring re-measurements or full retraining, offering computational efficiency for dynamic environments. Simulation studies conducted in both one and two-dimensional settings confirm the effectiveness of the approach, demonstrating improvements in predictive accuracy and computational efficiency. The developed framework is particularly well-suited for mapping tasks in dynamic environments, where efficient real-time correction is crucial.

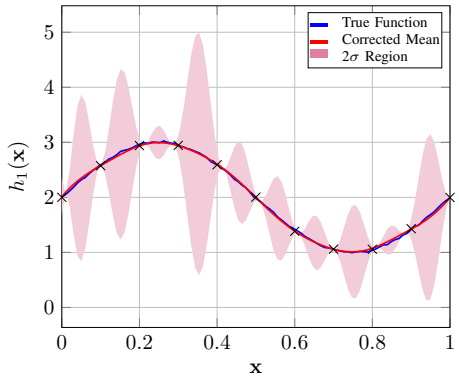

Fig. 4. This figure shows the updated GP mean and covariance estimates with using the developed method.

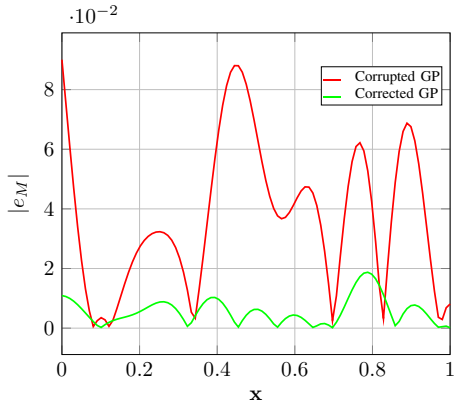

Fig. 5. The absolute error between the mean estimates of true GP with corrupted GP and the corrected GP for simulation 1 are compared in this plot.

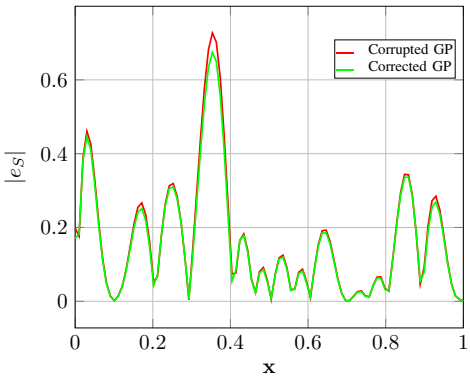

Fig. 6. The absolute error between the covariance estimates of true GP with corrupted GP and the corrected GP are compared in this plot for the simulation 1.

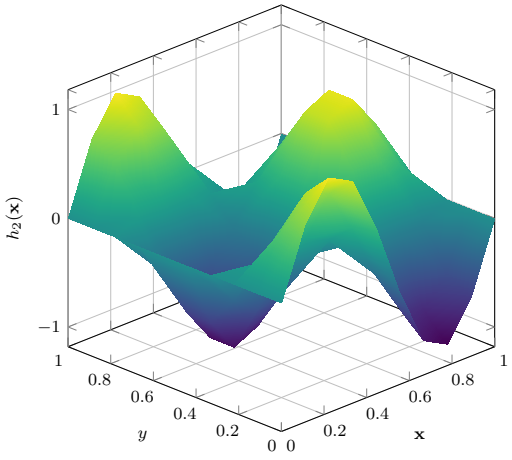

Fig. 7. 3D Surface Plot of the scalar field.

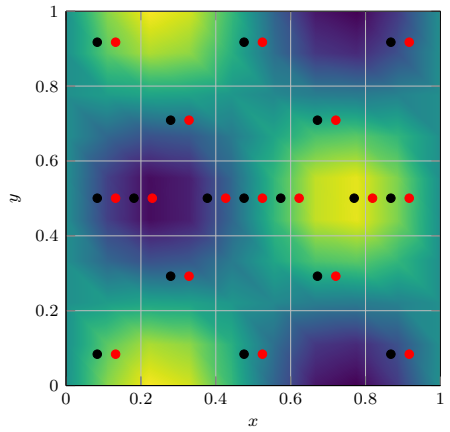

Fig. 8. The top view of the scalar field is shown, with black dots highlighting the actual measurement locations whereas red dots highlight the disturbed measurement locations.

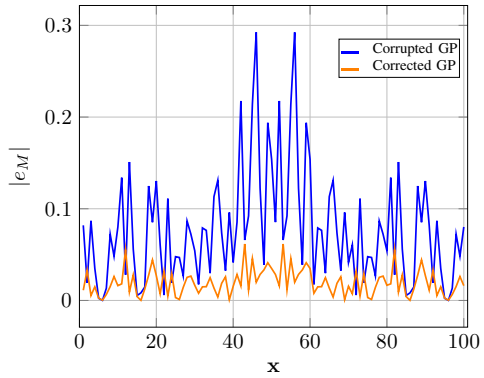

Fig. 9. Error between the mean estimates for 2-D case.

REFERENCES

[1] P. Sabella, "A rendering algorithm for visualizing 3D scalar fields," *Comput. Graph.*, vol. 22, no. 4, pp. 51–58, 1998.

[2] Z. Lin, H. H. T. Liu, and M. Wotton, "Kalman filter-based large-scale wildfire monitoring with a system of UAVs," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 606–615, Jan. 2019.

[3] T. Ren, H. Li, M. F. Modest, and C. Zhao, "Efficient two-dimensional scalar fields reconstruction of laminar flames from infrared hyper-spectral measurements with a machine learning approach," *J. Quant. Spectrosc. Radiat. Transfer*, p. 107724, 2021.

[4] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *J. Math. Psychol.*, pp. 1–16, 2018.

[5] Rasmussen and C. Edward, *Gaussian Processes in Machine Learning*. Springer Berlin Heidelberg, 2004, pp. 63–71.

[6] M. Qureshi, T. E. Ogri, Z. I. Bell, and R. Kamalapurkar, "Scalar field mapping with adaptive high-intensity region avoidance," in *Proc. IEEE Conf. Control Technol. Appl.*, Aug. 2024, pp. 388–393.

[7] N. Uzzaman and H. Bai, "A novel variational Bayesian adaptive kalman filter for systems with unknown state-dependent noise covariance matrices," in *Proc. Amer. Control Conf.* IEEE, Jul. 2024, pp. 1192–1197.

[8] M. Elhashash, H. Albanwan, and R. Qin, "A review of mobile mapping systems: From sensors to applications," *Sensors*, vol. 22, no. 11, 2022.

[9] H. Bai and C. N. Taylor, "Future uncertainty-based control for relative navigation in GPS-denied environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3491–3501, 2020.

[10] M. Qureshi, T. E. Ogri, H. Ramos, Z. I. Bell, and R. Kamalapurkar, "Gaussian process-based scalar field estimation in GPS-denied environments," 2025, arXiv:2502.17584.

[11] A. Bachrach, S. Prentice, R. He, and N. Roy, "RANGE-robust autonomous navigation in GPS-denied environments," *J. Field Robot.*, no. 5, pp. 644–666, 2011.

[12] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slam–based navigation for autonomous micro helicopters in GPS-denied environments," *J. Field Robot.*, vol. 28, no. 6, pp. 854–874, 2011.

[13] A. Girard, C. E. Rasmussen, and R. Murray-Smith, "Gaussian process priors with uncertain inputs: Multiple-step-ahead prediction," in *Proc. Switch. Learn. Feedb. Syst.*, 2002, pp. 158–184.

[14] A. Girard, "Approximate methods for propagation of uncertainty with Gaussian process," Ph.D. dissertation, University of Glasgow (UK), 2004.

[15] J. Candela, A. Girard, J. Larsen, and C. Rasmussen, "Propagation of uncertainty in Bayesian kernel models - application to multiple-step ahead forecasting," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, vol. 2, 2003, pp. II–701.

[16] R. Tripathy, I. Bilionis, and M. Gonzalez, "Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation," *J. Comput. Phys.*, vol. 321, pp. 191–223, 2016.

[17] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *Proc. Int. Conf. Mach. Learn.*, 2007, p. 393–400.

[18] A. Mchutchon and C. Rasmussen, "Gaussian process training with input noise," in *In Proc. Adv. Neural Inf. Proc. Syst.*, vol. 24, 2011.

[19] K. Coleman, H. Bai, and C. N. Taylor, "Invariant-EKF design for a unicycle robot under linear disturbances," in *In Proc. Amer. Control Conf.*, 2020, pp. 3479–3484.

[20] K. Coleman, H. Bai, and C. Taylor, "Extended invariant-EKF designs for state and additive disturbance estimation," *Automatica*, vol. 125, p. 109464, Mar. 2021.

[21] J. Park and I. W. Sandberg, "Universal approximation using Radial-Basis-Function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.