

An Active Learning Based Robot Kinematic Calibration Framework Using Gaussian Processes

Ersin Daş¹ and Joel W. Burdick¹

Abstract—Future NASA lander missions to icy moons will require completely automated, accurate, and data efficient calibration methods for the robot manipulator arms that sample icy terrains in the lander’s vicinity. To support this need, this paper presents a Gaussian Process (GP) approach to the classical manipulator kinematic calibration process. Instead of identifying a corrected set of Denavit-Hartenberg kinematic parameters, a set of GPs models the residual kinematic error of the arm over the workspace. More importantly, this modeling framework allows a Gaussian Process Upper Confident Bound (GP-UCB) algorithm to efficiently and adaptively select the calibration’s measurement points so as to minimize the number of experiments, and therefore minimize the time needed for recalibration. The method is demonstrated in simulation on a simple 2-DOF arm, a 6 DOF arm whose geometry is a candidate for a future NASA mission, and a 7 DOF Barrett WAM arm.

I. INTRODUCTION

Kinematic Calibration (KC) of robot manipulators is a classic problem [1]–[3]. Calibration was originally motivated by the need for precise positioning of factory robot tools and end-effectors. With limited real-time sensing, factory robots depend upon their open-loop positioning precision to reliably accomplish sophisticated tasks. As further discussed below, there are several approaches to kinematic calibration, methods to deploy calibration in a variety of situations, and approaches to better automate the calibration process.

This paper revisits the calibration problem with new methods because of proposed NASA missions to the icy moons of Europa and Enceladus, which may host subsurface oceans. Cryovolcanic activity might transport water/ice from a subsurface ocean through surface cracks. As part of a Europa lander mission [4], a robotic arm will gather surface material samples from the vicinity of these cracks, and analyze the local terrain using tools (scoops, drills, penetrometers, and bevometers). The icy deposits may contain molecules that indicate the presence of life in the subsurface oceans.

These missions will likely be battery powered, which severely limits their duration. Round trip communication delays make remote manipulator operation impractical. Hence, highly autonomous and efficient science operations are crucial. While Europa is not a factory environment, kinematic accuracy of the sampling robot arm is still crucial because proper science interpretation of the samples requires accurate localization of the their origins. Furthermore, since deep space missions must rigorously minimize mass, the sampling

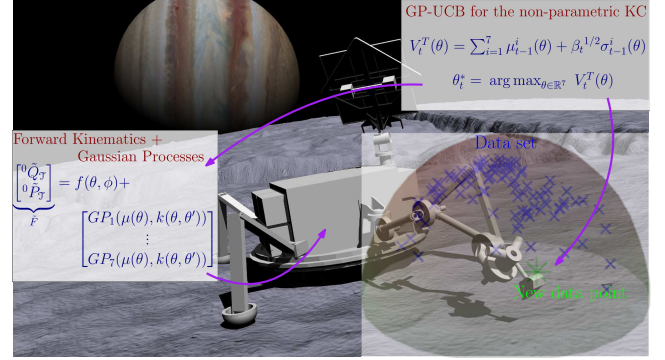


Fig. 1: Visualization of the proposed active learning-based kinematic calibration framework on a space robotic arm.

arm and its tools will likely be very lightweight, and thus more deformable than a factory robot.

Clearly, a data-efficient arm self-calibration process is indispensable for reliable icy moon sample collection and high mission efficiency. At a minimum, upon landing and first deployment, the sampling arm must be calibrated in-situ to accommodate any distortions that occur during the multi-year space flight to the moon. Moreover, these highly automated missions should be resilient to distortions of the robot and tool geometries that arise from unexpectedly forceful interactions with the terrain, joint freeze-ups in the 50°K surface temperature, and improper loading of the sampling tools. Practically speaking, icy moon missions require an automated calibration process that (1) provides sufficient accuracy, especially for the workspace regions where critical science occurs; (2) requires very few samples of arm data to reach a high recalibration accuracy, since in-situ recalibration consumes valuable mission time that should otherwise be devoted to science; and (3) can handle miscalibrations that are larger than typically experienced by industrial robots.

This paper develops an approach to manipulator kinematic calibration based on Gaussian Processes (GP) [5]. Gaussian Process Regression (GPR) can be used to learn kinematic residual errors, and allows for arm recalibration after large arm deformations. More importantly, Gaussian Process Optimization [6] adaptively and autonomously selects a small set of arm configurations that provide accurate recalibration. We demonstrate the utility of the proposed method on a 2-DOF planar manipulator, a 7-DOF Barrett WAM robot arm, and 6-DOF robot arm used in the OceanWATERS testbed [7].

This paper is organized as follows. After some preliminaries in Sect. II, Sect. III introduces an active learning-based kinematic calibration framework. Simulations are presented in Sect. IV, while Sect. V concludes the paper.

*“This work was supported by NASA Grant 80NSSC21K1032.”

¹Ersin Daş and Joel Burdick are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125, USA. {ersindas, jburdick}@caltech.edu

Related Work: Kinematic calibration of a robot manipulator has been extensively studied. Typically, the calibration process requires expert inputs for experiment design and numerous measurements to optimize an observability index. After data acquisition, offline optimization-based techniques, such as iterative least squares regression with linearized kinematic equations [8], Quadratic-Programming (QP) [9], and the Levenberg-Marquardt algorithm [10] are used to identify the kinematic parameters. Although these approaches improve robot positioning accuracy, they are neither data-efficient or capable of completely autonomous implementation. Both characteristics are needed for a deep-space application.

Machine learning-based frameworks can reduce positioning errors by considering issues, such as joint elasticities and backlash, that affect calibration accuracy. A Product-Of-Exponential kinematic model [11] has been combined with a GP to compensate for residual deflections [12]. In [13], an Artificial Neural Network directly learns robot position and orientation without nominal forward kinematic equations. The post-calibration accuracy of these methods highly depends upon the quality of the input data, whose selection relies upon the calibration operator's knowledge. An unsupervised learning-based recalibration, which also optimizes the kinematic parameters for manipulator visual servoing, is proposed in [14].

Many previous works use observability indexes [8] to optimize the choice of sampling locations. An experimental comparison of the indexes is studied in [15]. An active calibration algorithm is developed in [16] to find the most efficient set of calibration poses for the linearization-based techniques. A convex optimization approach to choosing configurations from a pool of candidate data points is proposed in [17]. The calibration experiment design methods in [18], [19], based on partial pose measurement, is performed before the experiments. Although these methods are efficient for linearization-based calibration, they do not consider how to choose sampling points in the presence of larger inaccuracies. In [20], an online calibration approach combines an unscented Kalman filter and iterative particle filter with a probabilistic sampling strategy. A Kalman filter-based online calibration method is presented in [21] for legged robots.

II. PRELIMINARIES

We use the standard notation: \mathbb{R} and \mathbb{R}^+ represent the set of real and positive real numbers, respectively. The Euclidean norm of a matrix is denoted by $\|\cdot\|$. $f \sim GP(\mu, k)$ denotes that the function $f(\cdot)$ is sampled from a Gaussian Process (which is reviewed in section II-D).

A. Forward Kinematics

A forward kinematic function maps a robot's joint space to its workspace, and it is a crucial part of closed-loop kinematic motion analysis. We adopt the widely used D-H convention that defines the relative position and orientation of the robot links and end effector using a four-parameter representation.¹ Given an n jointed robot and the D-H

parameters for the i^{th} link: joint angle θ_i , twist angle α_i , link length a_i , link offset d_i , we define the transformation matrix ${}^{i-1}T_i \in SE(3)$, which consists of a rotation matrix ${}^{i-1}R_i \in SO(3)$ and a translation vector ${}^{i-1}P_i \in \mathbb{R}^3$, between neighboring robot links, link $i-1$ and link i , as

$${}^{i-1}T_i = \left[\begin{array}{c|c} {}^{i-1}R_i & {}^{i-1}P_i \\ \hline 0 & 1 \end{array} \right] = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

for $i = 1, 2, \dots, n$, where θ_i is variable if the joint is revolute, while d_i is variable if the joint is prismatic. Then, the forward kinematics function is given by

$${}^0T_{\mathcal{T}} = {}^0T_1(\theta_1, \phi_1) {}^1T_2(\theta_2, \phi_2) \dots {}^{n-1}T_n(\theta_n, \phi_n) {}^nT_{\mathcal{T}}, \quad (2)$$

where $\theta = [\theta_1 \dots \theta_n]^T \in \mathbb{R}^n$, $\phi = [\alpha_i \ d_i \ a_i]^T \in \mathbb{R}^{3n}$, and \mathcal{T} represents the tool frame. This function gives the orientation and position of the tool frame with respect to the base frame, i.e., ${}^0R_{\mathcal{T}} \in SO(3)$, ${}^0P_{\mathcal{T}} \in \mathbb{R}^3$, and can be written more compactly as

$$\underbrace{[{}^0Q_{\mathcal{T}} \quad {}^0P_{\mathcal{T}}]^T}_F = f(\theta, \phi), \quad (3)$$

where ${}^0Q_{\mathcal{T}}$ is a four-parameter quaternion representation of the rotation matrix [22]. Note that the quaternion conversion from a rotation matrix may be discontinuous, and therefore complex to model by Gaussian Processes [23]. To circumvent this problem, we utilize the continuous quaternion conversion framework proposed in [24].

B. Kinematic Calibration Problems

We now formalize two different calibration problems with respect to the kinematic uncertainty structure. In the following definitions, let $\tilde{F} = [{}^0\tilde{Q}_{\mathcal{T}} \quad {}^0\tilde{P}_{\mathcal{T}}]^T$ represent the actual (uncertain, after distortion) kinematic function.

The Parametric calibration problem: Consider a perturbed D-H convention parameter set

$$[\theta \ \phi]^T = [\theta_i + \Delta\theta_i \ \alpha_i + \Delta\alpha_i d_i + \Delta d_i \ a_i + \Delta a_i]^T, \quad (4)$$

$$\Delta = [\Delta\theta_i \ \Delta\alpha_i \ \Delta d_i \ \Delta a_i]^T \in \mathbb{R}^{4n}, \quad (5)$$

where Δ is the difference between the nominal (idealized) and actual D-H parameters. The parametric calibration problem is

$$[\phi^*]^T = \arg \min_{[\phi]^T \in \mathbb{R}^{4n}} \|\tilde{F} - F\|^2 \quad (6)$$

The Non-parametric calibration problem: Consider the nominal forward kinematic function $f_n(\theta, \phi)$ with an unstructured additive uncertainty term $\Delta f(\theta, \phi)$ such that $f(\theta, \phi) = f_n(\theta, \phi) + \Delta f(\theta, \phi)$. Then, the non-parametric calibration problem is defined as

$$\Delta f^*(\theta) = \arg \min_{\Delta f(\theta)} \|\tilde{F} - F\|^2 \quad (7)$$

¹The methods of this paper can be readily adapted to a product-of-exponentials representation of manipulator kinematics.

That is, we want to find the *residual* function $\Delta f^*(\theta)$ that minimizes the calibration error with respect to the forward kinematic function that uses idealized D-H parameters.

Kinematic calibration methods are driven by error between the measured end-effector data and the computed forward kinematics data. Therefore, an expert proposes enough experiments to provide sets of input joint angles θ and measurements \tilde{F} to solve the given optimal error reduction problem. Then, the objective function of the optimization problems in (6), (7) is redefined on a discrete set of data as

$$\hat{\Delta}F = [\tilde{F}_1 - F_1 \quad \dots \quad \tilde{F}_m - F_m]^T \in \mathbb{R}^{7m}, \quad (8)$$

where m is the number of measurements.

C. Classical Linearized Recalibration

The calibration cost function given in (6) is generally nonlinear and non-convex depending on the robot kinematic structure. Thus, most classical recalibration techniques are based on a linearized version of the parametric calibration problem. By assuming relatively small parametric deflections, one can linearize the calibration function (3) as

$$\underbrace{\tilde{F} - F}_{\Delta F \in \mathbb{R}^7} = \underbrace{\begin{bmatrix} \frac{\partial f}{\partial \alpha_i} & \frac{\partial f}{\partial d_i} & \frac{\partial f}{\partial a_i} \end{bmatrix}}_{J_\phi \in \mathbb{R}^{7 \times 3n}} \underbrace{[\Delta \alpha_i \quad \Delta d_i \quad \Delta a_i]^T}_{\Delta \phi \in \mathbb{R}^{3n}}, \quad (9)$$

where J_ϕ is the Jacobian matrix. Therefore, when the nominal D-H parameters and their possible lower bound $\Delta \phi_{lb} \in \mathbb{R}^{3n}$ and upper bound $\Delta \phi_{ub} \in \mathbb{R}^{3n}$ are given, the linearization-based KC problem can be solved via the following constrained QP:

$$\begin{aligned} \Delta \phi^* = \arg \min_{\Delta \phi \in \mathbb{R}^{3n}} \quad & \|\tilde{F} - F - J_\phi \Delta \phi\|^2 \\ \text{s.t.} \quad & \Delta \phi_{lb} \leq \Delta \phi \leq \Delta \phi_{ub} \end{aligned}$$

Then, we obtain $\phi^* = \phi + \Delta \phi^*$. This QP is iteratively solved until an acceptable error convergence is achieved [9].

D. Gaussian Processes (GPs)

Gaussian Processes provide a stochastic, data-driven, supervised machine learning approach to specify the relations between input and output data sets of smooth functions through Bayesian inference [5]. A GP function, $f(x) : X \rightarrow \mathbb{R}$, defined on a data domain X , is fully specified by its mean function $\mu(x) : X \rightarrow \mathbb{R}$ and a covariance/kernel function $k(x, x') : X \times X \rightarrow \mathbb{R}$:

$$\mu(x) \triangleq \mathbb{E}[f(x)], \quad (10)$$

$$k(x, x') \triangleq \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]. \quad (11)$$

We say that $f(\cdot)$ is distributed according to:

$$f(x) \sim GP(\mu(x), k(x, x')). \quad (12)$$

Given a collection of training data $\mathcal{D} \triangleq \{X_N \in X, Y_N \in \mathbb{R}^N\}$ with inputs $X_N = [x_1, \dots, x_N]$, outputs $Y_N = [f(x_1) + \epsilon_1, \dots, f(x_N) + \epsilon_N]^T$ and i.i.d zero-mean Gaussian noises $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$, a GP predicts a set of outputs based on an input test data set

$\tilde{X} = [\tilde{x}_1, \dots, \tilde{x}_M] \in X$. The posterior of the GP over the observations, $\tilde{f}|\tilde{X}, \mathcal{D}$, is also a GP distribution with a mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$ given by

$$\tilde{f}|\tilde{X}, \mathcal{D} \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2), \quad (13)$$

$$\tilde{\mu} = K_{\tilde{X}X_N}(K_{X_NX_N} + \sigma_\epsilon^2 I)^{-1} Y_N, \quad (14)$$

$$\tilde{\sigma}^2 = K_{\tilde{X}\tilde{X}} - K_{\tilde{X}X_N}(K_{X_NX_N} + \sigma_\epsilon^2 I)^{-1} K_{X_N\tilde{X}}, \quad (15)$$

where I is the identity matrix, $K_{X_NX_N} \in \mathbb{R}^{N \times N}$, $K_{\tilde{X}\tilde{X}} \in \mathbb{R}^{M \times M}$, $K_{\tilde{X}X_N} \in \mathbb{R}^{M \times N}$, $K_{X_N\tilde{X}} \in \mathbb{R}^{N \times M}$ are the covariance matrices for the set of points, which measure the correlations between the inputs with $[K_{X_NX_N}]_{i,j} = k(x_i, x_j)$, $[K_{\tilde{X}\tilde{X}}]_{i,j} = k(\tilde{x}_i, \tilde{x}_j)$, $[K_{\tilde{X}X_N}]_{i,j} = k(\tilde{x}_i, x_j)$, $[K_{X_N\tilde{X}}]_{i,j} = k(x_i, \tilde{x}_j)$.

Briefly, the kernel function reflect the features of the functions to be learned, such as smoothness and input correlations. Our experiments use the commonly used squared-exponential (SE) kernel given by

$$k(x_i, x_j) = \sigma_f^2 \exp\left(\frac{-\|x_i - x_j\|^2}{2l^2}\right) + \sigma_n^2 \delta(x_i, x_j), \quad (16)$$

where $\theta_h = [l \ \sigma_f \ \sigma_n]^T$ are the tunable hyperparameters and $\delta(x_i, x_j)$ is the Kronecker delta function. The optimal hyperparameters can be found by minimizing the negative log marginal likelihood function [5]. For data set \mathcal{D} , this function is defined as

$$\begin{aligned} \log p(Y_N | X_N, \theta_h) = & -\frac{1}{2} Y_N^T (K_{X_NX_N}(\theta_h) + \sigma_\epsilon^2 I)^{-1} Y_N \\ & - \frac{1}{2} \log |(K_{X_NX_N}(\theta_h) + \sigma_\epsilon^2 I)| - \frac{N}{2} \log 2\pi. \end{aligned} \quad (17)$$

Finally, the optimal hyperparameters θ_h^* , and σ_ϵ^* can be computed via the gradient-descent based solution of the following typically non-convex optimization problem:

$$[\theta_h^* \ \sigma_\epsilon^*]^T = \arg \min_{\theta_h, \sigma_\epsilon} -\log p(Y_N | X_N, \theta_h). \quad (18)$$

III. ACTIVE LEARNING BASED KINEMATIC CALIBRATION APPROACH

A. GPs for Non-parametric Calibration

In this section, the residuals representing the inaccuracies in the forward kinematics are modelled via GPs. Consequently, we will be able to achieve effective non-parametric robot kinematic learning and calibration using noisy sensor measurements in a stochastic framework.

The kinematic residual errors can be captured with seven GPs along each orientation and translation direction. In particular, each component of matrix F in (3) is modelled as a single-output GP with a mean and kernel function as

$$\underbrace{\begin{bmatrix} {}^0\tilde{Q}_T \\ {}^0\tilde{P}_T \end{bmatrix}}_{\tilde{F}} = f(\theta, \phi) + \begin{bmatrix} GP_1(\mu(\theta), k(\theta, \theta')) \\ \vdots \\ GP_7(\mu(\theta), k(\theta, \theta')) \end{bmatrix}, \quad (19)$$

where θ is the vector of joint angles, and \tilde{F} is the measured tool-frame poses. The regressed GP functions can predict the residual error and its variance in each direction of motion, at a new query input θ^* , by using (14) and (15).

Our GP-based modeling of the residual errors in each motion direction is simplistic. One should expect coupling in the errors across these directions. Multi-output Gaussian Processes (MOGPs) [25] and vector-valued GPs [26] are suitable solutions to this problem. However, due to the computational scaling of these methods, we assume single-output GPs. As shown in Section IV, the Gaussian distribution property holds in our applications. Thus, the decoupled GP assumption is reasonable and computationally efficient.

B. GPs for Parametric Calibration

We briefly note that GP regression can be used for parametric calibration, all input parameters of the kinematic function in (3) can be modelled as a single-output GP. To this end, we modify (3) as

$$[\theta \ \phi]^T = f^{-1}({}^0Q_{\mathcal{T}}, {}^0P_{\mathcal{T}}), \quad (20)$$

where the D-H parameters, including joint angles, are outputs, and the tool frame poses are inputs. By assuming that each component in $[\theta \ \phi]^T \in \mathbb{R}^{4n}$ can be modeled by a GP, the parametric calibration problem can be written as

$$\begin{bmatrix} \tilde{\theta} \\ \tilde{\phi} \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \phi_0 \end{bmatrix} + \begin{bmatrix} GP_1(\mu(F), k(F, F')) \\ \vdots \\ GP_{4n}(\mu(F), k(F, F')) \end{bmatrix}, \quad (21)$$

where $\tilde{\theta}$, $\tilde{\phi}$, F are the measurements, θ_0 , ϕ_0 are the nominal values of the D-H parameters.

As compared to the non-parametric approach in (19), the joint angle vector is also assumed to be uncertain. Therefore, Eq. (21) can model the possible joint angle offsets. But, the number of GPs needed in this approach increases with the number of robot joints. For example, for a 7 DOF robot, 28 different GPs must be defined. Because of these drawbacks, we leave the parametric approach to our future studies.

C. GP-UCB for Non-parametric Experiment Design

We are concerned not only with fitting a GP model to calibration data, but also with designing a calibration experiment that minimizes the number of samples. such optimization is a classical experimental design problem that aims to find the best set of measurement points [27], [28]. When using GP regression-based calibration alone, a user must select the training and test data points, which are essential for the method's accuracy. This section adopts a Gaussian Process Upper Confidence Bound (GP-UCB) [6] active learning algorithm to adaptively select a sequence of informative measurements to minimize the residual error.

Modeling the residual kinematic error with a GP, as in Section III-A, brings a significant advantage for experiment design, since it encodes residual function uncertainty in the confidence bounds [29]. With this feature, the GP-UCB algorithm uses currently available measurements and the residual function confidence bounds to decide the next test point that optimizes the utility function $V_t(x) : X \rightarrow \mathbb{R}$

$$V_t(x) = \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x), \quad (22)$$

$$x_t^* = \arg \max_{x \in X} V_t(x), \quad (23)$$

where $\beta_t \in \mathbb{R}^+$ is an iteration-varying parameter [6] that weights residual function uncertainty in the selection of the next sampling point, $\sigma_{t-1}(x)$ is the square root of the variance, and $\mu_{t-1}(x)$ is the predicted value of the mean function at input x . Hence, GP-UCB chooses sampling points that trade off exploitation of joint space regions with high calibration performance against exploration of regions with high uncertainty about the residual function.

The GP-UCB algorithm has an appealing *no regret* property [6]. I.e., the algorithm is *guaranteed* to find the globally optimal residual functions given sufficient examples. Our experiments below show that some other calibration experiment design approaches become stuck in local minima. Moreover, its *instantaneous regret* is proportional to $T^{-1/2}$, where T is the number of experiments. This property implies that the calibration error steadily decreases with each sample. While this paper focuses on sequential sampling selection, a variation of this approach, entitled GP-BUCB [30], proposes *batches* of experiments, while still retaining a no-regret property and similar instantaneous regret performance.

Since there are seven different GPs with the same input vector, we modify the utility function as the sum of UCBs:

$$V_t^T(x) = \sum_{i=1}^7 \mu_{t-1}^i(x) + \beta_t^{1/2} \sigma_{t-1}^i(x), \quad (24)$$

where $\mu_{t-1}^i(x)$, $\sigma_{t-1}^i(x)$ are the mean and variance of the i^{th} output given in (19). Hence, the resulting new putatively optimal data sampling point is obtained by considering all GPs simultaneously. The authors in [31] show that this approximation does not violate the no-regret property of the UCB algorithm under mild conditions.

The basic pseudo-code of the proposed GP-UCB-based calibration approach is presented in Algorithm 1, which starts with any available GP prior, and the next measurement is chosen to maximize the utility function (24). The GP residual is updated from the new measurement (using Eqs. (14) and (15)). This iteration continues until a stopping criterion is satisfied. For the algorithm parameter β_t , which is defined in Theorem 2 of [6], we use a strategy proposed in [6] as

$$\beta_t = 2 \log(t^2 2\pi^2 / (3\delta_\beta)) + 2d_\beta \log(t^2 d_\beta a_\beta b_\beta r_\beta \sqrt{\log(4d_\beta / \delta_\beta)}). \quad (25)$$

Algorithm 1 GP-UCB for non-parametric arm calibration

Input: Input set $\theta \in X$

GP prior μ_0 , σ_0 , k

Kernel function $k(\cdot, \cdot)$

Parameter β_t

Output: θ^*

for $t = 1, 2, \dots$, **do**

Choose $\theta_t^* = \arg \max_{\theta \in X} V_t^T(\theta)$

Compute σ_t via Equation (15)

Obtain $\Delta F_t = (\tilde{F}(\theta_t) - F(\theta_t)) + \epsilon_t$

Compute μ_t via Equation (14)

end for

IV. EXPERIMENTS

We evaluate the performance of the GP-based residual regression and GP-UCB-based sampling selection in three simulated robotic arms with perturbed kinematic parameters. We compare our results with three other techniques: 1) A gridding-based *D-optimal design algorithm* with a convex approximation proposed in [17]. This algorithm utilizes a pool of possible joint angle sets. Then, the D-optimality criteria is adopted using the Jacobian matrix J_ϕ given in (9) and this pool. In [17], it is shown that D-optimal experimental design leads to better performance than random sampling and the DEMAX algorithm. 2) *Expected improvement (EI)* based sampling algorithm [32]. Like the UCB utility function in (24), we modify the EI algorithm's utility to sum the expected improvements in each direction. 3) *Random sampling*.

A. 2-DOF Manipulator

We start with an easy-to-understand simulated calibration of a 2R planar robot (Fig. 2-(a)). We seek to learn the forward kinematic residuals while choosing a small set of sampling points. The uncertain D-H parameters of the arm are $d_1 = d_2 = 0 \pm 0.1 \text{ m}$, $a_1 = a_2 = 1 \pm 0.2 \text{ m}$, $\alpha_1 = \alpha_2 = 0 \pm 0.1 \text{ rad}$, $\theta_1, \theta_2 \in [-3, 3] \text{ rad}$. Even in this simple example, linearization-based methods may not yield an accurate calibration in the presence of additive uncertainties, since these approaches are formulated to mainly address constant inaccuracies, i.e., $\phi = \phi_0 + e_\phi$.

We assume precise measurements of the joint angles (θ_1, θ_2) and the orientation and position of tool frame ($F \in \mathbb{R}^7$) (e.g., by a calibrated high-resolution camera). We run Algorithm 1 for $t = 1, 2, \dots, 30$ iterations, and choose the β_t in (25) and an SE kernel (16) with the hyperparameters optimized using (18) for all simulation examples.

Fig. 2-(d) plots the calibration error, $\|\tilde{F} - F\|$, versus the number of samples selected by GP-UCB. As more samples are collected, the calibration error norm converges to zero, confirming the no-regret property of this approach. EI-based sampling initially concentrates on the exploration of the calibration error functions. But after 10 steps, it encounters a local minimum, and does not continue to explore. For the D-optimal sampling, we start with 10 randomly selected data points, since this method requires as many data points as the number of unknowns. Although this method initially offers better performance, the convergence rate is slow. While random sampling also provides high-precision kinematic error modelling, it requires significantly more sampling points than the active learning approach, as shown in Fig. 2-(d). This confirms that GP-UCB is intelligently selecting its experiments. The post-calibrated performance of GP-UCB is shown in Fig. 2-(g) for the x position error.

To demonstrate our methods ability to handle larger kinematic errors, and noisy measurements, the planar robot's D-H parameters are perturbed at a mean level starting from 10% up to 200%. Secondly, a normally distributed noise with a variance of 0.01 has been added to each output. For comparison, we used the Kalibrot toolbox [9], which relies on classical linearized recalibration. Fig. 3-(a) shows that

our method learns and tolerates increasing kinematic uncertainties and is sufficiently robust to sensor noise. Although Kalibrot is robust against sensor noise, its calibration error increases roughly linearly with parameter uncertainty.

B. 7-DOF Barrett WAM Robot Arm

The OWLAT testbed was developed to evaluate autonomous techniques for *Ocean Worlds* icy moon lander missions [33]. A 7-DOF Barrett WAM manipulator simulates robotic sampling operations (Fig. 2-(b)) that might take place on an icy moon mission. The arm's kinematic parameters are given in Table I with additive uncertainty bounds.

Fig. 2-(e) shows the simulated convergence of the forward kinematic error estimate versus GP-UCB sample number. The results in this figure suggest that the calibration performance and data efficiency of our framework are better than other sampling methods for this 7-DOF arm, especially for small sample number. Although the EI algorithm gave the second best results for this robot arm, this method's lack of design parameters is a weakness. The gridding-based D-optimal algorithm fails in this simulation. Fig. 2-(h) shows the post-calibrated kinematic accuracy on 50 test points linearly spaced along the x position.

C. 6-DOF OceanWATERS Lander Robot Arm

We additionally demonstrate the proposed algorithm on the open-source OceanWATERS simulation testbed [7] (see Fig. 2-(c) and Fig. 1), which is a likely design for a future lander mission. The arm's nominal kinematic parameters and our tested uncertainty bounds are given in Table I.

Since our goal is to fit a GP to the residuals, we present the distribution of forward kinematic errors for 1500 sample points, chosen by the GP-UCB algorithm, in Fig. 3-(b). The figure demonstrates that the kinematic error has a nearly normal distribution with zero mean. This result confirms that the use of a Gauss Process framework is warranted.

Fig. 2-(f) compares the calibration performance of GP-UCB and the other approaches over 600 sampling iterations. The figure shows that GP-UCB is an efficient sampling method for this robot, since fewer data points are needed for accurate recalibration. The EI and D-optimal algorithms reduce the norm of the calibration error; but they become "stuck" on one sample point after ~ 300 samples. The post-calibration accuracy of the GP model is demonstrated on 50 linearly spaced x -axis test sample points in Fig. 2-(i).

V. CONCLUSIONS AND FUTURE WORK

Motivated by the proposed use of manipulators in future icy moon sampling missions, this study aimed to accurately learn the manipulator forward kinematic error residual in an online, non-parametric, and data-efficient way. GP regression models the kinematic inaccuracies and their confidences, while an online sequential GP-UCB algorithm efficiently selects the optimal measurement in a way that adapts to the specific errors experienced by the robot. Our numerical studies on three different arm geometries found that this framework generally outperformed other methods in the limit

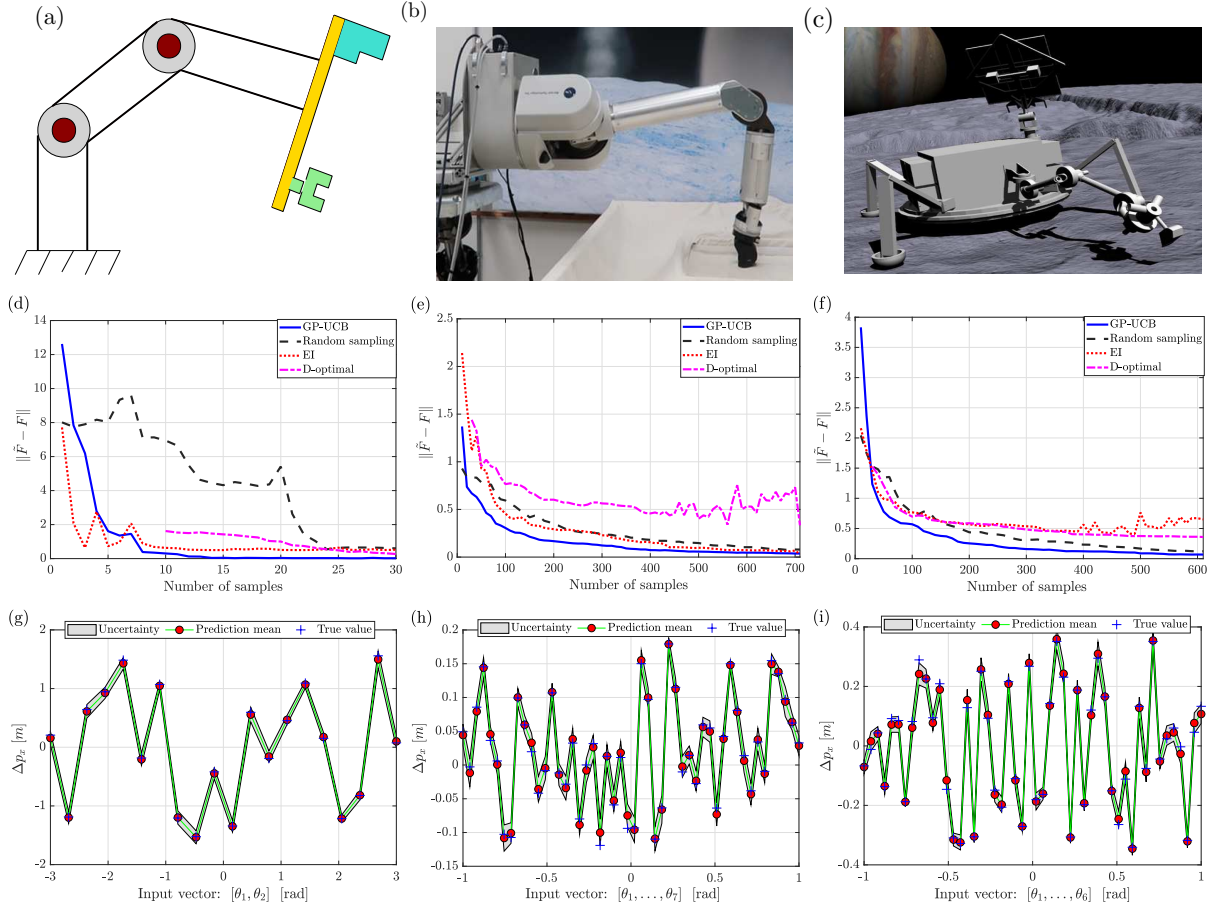


Fig. 2: Simulation examples. (a) 2-DOF Manipulator. (b) 7-DOF Barrett WAM manipulator arm in OWLAT. (c) 6-DOF OceanWATERS Lander Robot Arm. (d), (e), (f) The evaluations of the calibration errors with respect to the number of iterations for 2-DOF, 7-DOF, and 6-DOF robots, respectively. (g), (h), (i) The errors in position x obtained from the test samples for 2-DOF, 7-DOF, and 6-DOF robots, respectively.

7-DOF Barrett WAM Robot Arm							
Joint	1	2	3	4	5	6	7
d	0 ± 0.01	0 ± 0.02	0.55 ± 0.2	0 ± 0.03	0.3 ± 0.2	0 ± 0.04	0.06 ± 0.06
a	0 ± 0.01	0 ± 0.03	0.045 ± 0.01	-0.045 ± 0.01	0 ± 0.07	0 ± 0.1	0 ± 0.01
α	$-\pi/2 \pm 0.2$	$\pi/2 \pm 0.2$	$-\pi/2 \pm 0.3$	$\pi/2 \pm 0.2$	$-\pi/2 \pm 0.1$	$\pi/2 \pm 0.1$	0 ± 0.3
θ	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
6-DOF OceanWATERS Lander Robot Arm							
d	0 ± 0.01	0 ± 0.02	0 ± 0.06	-0.15 ± 0.01	0 ± 0.04	0 ± 0.06	-
a	0.16 ± 0.1	0.37 ± 0.1	0.05 ± 0.02	0.463 ± 0.1	-0.238 ± 0.1	0.225 ± 0.02	-
α	$\pi/2 \pm 0.2$	0 ± 0.2	$\pi \pm 0.3$	0 ± 0.2	0 ± 0.1	$\pi/2 \pm 0.1$	-
θ	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	-

TABLE I: DH parameters for the 7-DOF Barrett WAM Robot and 6-DOF OceanWATERS Lander Robot Arm.

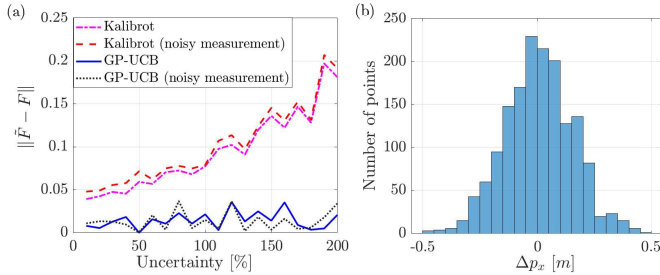


Fig. 3: (a) Kalibrot vs GP-UCB. (b) Histograms of calibration error distribution along x -axis before the calibration process.

of a small number of calibration experiments, which is vital for deep space missions. In future work, we will investigate specific GP kernels that better capture the uncertainty properties of manipulator kinematics, and study the effects of measurement noise on calibration accuracy under GP and classical calibration frameworks. We will also evaluate if coupled Multivariate Gaussian Process models [26], [34], [35] may further improve the calibration accuracy and sample efficiency.

REFERENCES

- [1] J. M. Hollerbach, "A survey of kinematic calibration," in *The robotics review 1*, 1989, pp. 207–242.
- [2] W. Khalil and E. Dombre, *Modeling identification and control of robots*. CRC Press, 2002.
- [3] A. Nubiola and I. A. Bonev, "Absolute calibration of an ABB IRB 1600 robot using a laser tracker," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 236–245, 2013.
- [4] K. P. Hand, C. B. Phillips, A. Murray, J. Garvin, E. Maize, R. Gibbs, G. Reeves, A. San Martin, G. Tan-Wang, J. Krajewski, *et al.*, "Science goals and mission architecture of the Europa lander mission concept," *The Planetary Science Journal*, vol. 3, no. 1, p. 22, 2022.
- [5] C. R. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. The MIT Press, 2006.
- [6] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.
- [7] D. Catanoso, A. Chakrabarty, J. Fugate, U. Naal, T. M. Welsh, and L. J. Edwards, "OceanWATERS lander robotic arm operation," in *2021 IEEE Aerospace Conference (50100)*. IEEE, 2021, pp. 1–11.
- [8] J. M. Hollerbach and C. W. Wampler, "The calibration index and taxonomy for robot kinematic calibration methods," *The international journal of robotics research*, vol. 15, no. 6, pp. 573–591, 1996.
- [9] F. Cursi, W. Bai, and P. Kormushev, "Kalibrot: A simple-to-use MATLAB package for robot kinematic calibration," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8852–8859.
- [10] T. S. Lembono, F. Suárez-Ruiz, and Q.-C. Pham, "SCALAR: Simultaneous calibration of 2-D laser and robot kinematic parameters using planarity and distance constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1971–1979, 2019.
- [11] F. C. Park, "Computational aspects of the product-of-exponentials formula for robot kinematics," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 643–647, 1994.
- [12] W. Jing, P. Y. Tao, G. Yang, and K. Shimada, "Calibration of industry robots with consideration of loading effects using Product-Of-Exponential (POE) and Gaussian Process (GP)," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4380–4385.
- [13] F. Cursi, W. Bai, W. Li, E. M. Yeatman, and P. Kormushev, "Augmented neural network for full robot kinematic modelling in SE(3)," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7140–7147, 2022.
- [14] N. Gothoskar, M. Lázaro-Gredilla, Y. Bekiroglu, A. Agarwal, J. B. Tenenbaum, V. K. Mansinghka, and D. George, "DURableVS: Data-efficient Unsupervised Recalibrating Visual Servoing via on-line learning in a structured generative model," *arXiv preprint arXiv:2202.03697*, 2022.
- [15] A. Joubair and I. A. Bonev, "Comparison of the efficiency of five observability indices for robot calibration," *Mechanism and Machine Theory*, vol. 70, pp. 254–265, 2013.
- [16] Y. Sun and J. M. Hollerbach, "Observability index selection for robot calibration," in *2008 IEEE international conference on robotics and automation*. IEEE, 2008, pp. 831–836.
- [17] K. Kamali and I. A. Bonev, "Optimal experiment design for elasto-geometrical calibration of industrial robots," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2733–2744, 2019.
- [18] Y. Wu, A. Klimchik, S. Caro, B. Furet, and A. Pashkevich, "Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments," *Robotics and Computer-Integrated Manufacturing*, vol. 35, pp. 151–168, 2015.
- [19] H. Wang, T. Gao, J. Kinugawa, and K. Kosuge, "Finding measurement configurations for accurate robot calibration: Validation with a cable-driven robot," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1156–1169, 2017.
- [20] G. Du, Y. Liang, C. Li, P. X. Liu, and D. Li, "Online robot kinematic calibration using hybrid filter with multiple sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 7092–7107, 2020.
- [21] S. Yang, H. Choset, and Z. Manchester, "Online kinematic calibration for legged robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8178–8185, 2022.
- [22] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [23] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.
- [24] J. Wu, "Optimal continuous unit quaternions from rotation matrices," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 4, pp. 919–922, 2019.
- [25] W. Bruinsma, E. Perim, W. Tebbutt, S. Hosking, A. Solin, and R. Turner, "Scalable exact inference in multi-output Gaussian processes," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1190–1201.
- [26] M. Hutchinson, A. Terenin, V. Borovitskiy, S. Takao, Y. Teh, and M. Deisenroth, "Vector-valued Gaussian processes on Riemannian manifolds via gauge independent projected kernels," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 160–17 169, 2021.
- [27] Y. Sun and J. M. Hollerbach, "Active robot calibration algorithm," in *2008 IEEE international conference on robotics and automation*. IEEE, 2008, pp. 1276–1281.
- [28] J. Hollerbach, W. Khalil, and M. Gautier, "Model identification," in *Springer handbook of robotics*. Springer, 2016, pp. 113–138.
- [29] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.
- [30] T. Desautels, A. Krause, and J. W. Burdick, "Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization," *Journal of Machine Learning Research*, vol. 15, pp. 3873–3923, 2014.
- [31] B. Shen, R. Gnanasambandam, R. Wang, and Z. J. Kong, "Multi-task Gaussian process upper confidence bound for hyperparameter tuning and its application for simulation studies of additive manufacturing," *IIEE Transactions*, pp. 1–13, 2022.
- [32] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
- [33] H. Nayar, A. Goel, A. Boettcher, M. Hans, J. Sawoniewicz, A. Gaut, S. Higa, H. Ono, A. Jain, C. Lim, *et al.*, "Development of a lander autonomy testbed for ocean worlds missions," in *Earth and Space 2021*, 2021, pp. 531–540.
- [34] T. de Wolff, A. Cuevas, and F. Tobar, "MOGPTK: The multi-output Gaussian process toolkit," *Neurocomputing*, vol. 424, pp. 49–53, 2021.
- [35] R. Moriconi, M. P. Deisenroth, and K. Sesh Kumar, "High-dimensional Bayesian optimization using low-dimensional feature spaces," *Machine Learning*, vol. 109, no. 9, pp. 1925–1943, 2020.