

Edge-Enabled Adaptive Shape Estimation of 3-D Printed Soft Actuators With Gaussian Processes and Unscented Kalman Filters

Kaige Tan¹, Member, IEEE, Qinglei Ji¹, Lei Feng¹, Member, IEEE,
and Martin Törngren², Senior Member, IEEE

I. INTRODUCTION

Abstract—Soft actuators have the advantages of compliance and adaptability when working with vulnerable objects, but the deformation shape of the soft actuators is difficult to measure or estimate. Soft sensors made of highly flexible and responsive materials are promising new approaches to the shape estimation of soft actuators, but suffer from highly nonlinear, hysteresis, and time-variant properties. A nonlinear and adaptive state observer is essential for shape estimation from soft sensors. Current state estimation methods rely on complex nonlinear data-fitting models, and the robustness of the estimation methods is questionable. This study investigates the soft actuator dynamics and the soft sensor model as a stochastic process characterized by the Gaussian process (GP) model. The unscented Kalman filter is applied to the GP model for more reliable variance adjustment during the sequential state estimation process than conventional methods. In addition, a major limitation of the GP model is its computational complexity during online inference. To improve the real-time performance while guaranteeing accuracy, we introduce an edge server to decrease the onboard computational and memory overhead. The experiments showcase a significant improvement in estimation accuracy and real-time performance compared to baseline methods.

Index Terms—Hydraulic/pneumatic actuators, modeling, control, and learning for soft robots, soft sensors and actuators.

Manuscript received 5 December 2022; revised 24 February 2023 and 11 April 2023; accepted 17 April 2023. Date of publication 1 May 2023; date of current version 14 September 2023. This work was supported in part by the TECoSA Vinnova Competence Center for Trustworthy Edge Computing Systems and Applications at the KTH Royal Institute of Technology, in part by the InSecTT (which was supported by the ECSEL Joint Undertaking (JU) through European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, and Turkey under Grant 876038) and IRIS Area II. The document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains. (Corresponding author: Lei Feng.)

Kaige Tan, Lei Feng, and Martin Törngren are with the Department of Machine Design, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: kaiget@kth.se; lfeng@kth.se; martint@kth.se).

Qinglei Ji was with the Department of Production Engineering and the Department of Machine Design, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden. He is now with Volvo Car Corporation, 418 78 Gothenburg, Sweden (e-mail: qinglei@kth.se).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIE.2023.3270505>.

Digital Object Identifier 10.1109/TIE.2023.3270505

SOFT robots made of compliant materials have shown their advantages for adaptability, safety, and novel actuation [1]. Nonetheless, the continuum and hysteresis properties of soft robots pose a challenge in analyzing their movement and kinematics compared to the dynamics of conventional rigid robots [2]. To address this issue, researchers study and apply continuum mechanics to analyze the kinematics of the soft material, primarily using the visco-hyperelastic model [3], piecewise constant curvature models [4], and finite-element method (FEM) [5]. Those kinematic analysis studies establish the soft actuators' plant models and are used to predict the shape deformation under external actuation. A high-precision plant model is essential for the accuracy of robot's actuator control. However, given the gap between the model and the real physical device, an increasing uncertainty will evolve if only the state prediction model is used. Therefore, soft sensors that measure the shape of the soft actuators and have minimal interference to their free movement are emerging and gaining interest within the field of soft robotics [6].

3-D printing has been intensively applied to fabricate soft actuators directly integrated with printed soft sensors [1], which either use commercial conductive filaments [7] or self-developed materials [8] that exhibit deformation-responsive properties. These sensors usually have piezoresistive or capacitive effects, which provides feedback through the resistance or capacitance variations. With the external electrical circuit, sensor feedback can be captured to reflect the deformation of the soft actuator. Such a soft actuator–sensor system establishes a state estimator [9] and enjoys the proprioception property [10], i.e., the system itself has the sensing capacity without the necessity of external sensors, e.g., a camera module with image feedback [11], where its detection accuracy can be largely influenced by the line of sight of the camera in a multiactuator intersection scenario. Thus, the applicability and flexibility of the actuator system are enhanced.

Despite the advantage of proprioception, piezoresistive and capacitive materials cause soft sensors to manifest substantial hysteresis and nonlinearities [12], which confine their usage since basic calibration techniques with regression cannot adequately identify their features. Compensation strategies are desired to resolve this issue. Operator-based techniques identify

the hysteresis behavior using the *hysteresis operator* [13], [14], [15], which models hysteresis as a cumulative memory effect of delayed relay components. These models share a similar operating principle, in which the parameters of highly nonlinear basis functions are fitted through a complex identification process. Furthermore, hysteresis modeling of soft sensors also utilizes data-driven methods. The dynamics of the soft system is treated as a black box model, where the hysteresis is regarded as an inherent property of the sensor [16].

A general limitation of the aforementioned methods is that they essentially construct a deterministic model to characterize the system dynamics, which fails to consider model uncertainty or environmental noise. Moreover, the soft sensor is susceptible to external noise [12], and the dynamics drift in the soft actuator due to polymer aging [17] is likely to be nonnegligible. To overcome these problems, a probability distribution model can be used to characterize the entire soft actuator–sensor system. Considering the dynamics of soft actuators and sensors as a random process allows the state estimation problem to be turned into a Bayesian filtering problem, but only a few studies [18], [19], [20] have been conducted in this area [19], mainly owing to the modeling complexity of nonlinearity and hysteresis with probability models. In addition, during the filtering process, an extended Kalman filter (EKF) linearizes the nonlinear dynamic model, but it is challenging to guarantee system approximation accuracy with the highly nonlinear system.

This study enhances our previous result [20] by characterizing the soft actuator–sensor system with a more accurate representation. Specifically, the Gaussian process (GP) model is employed to identify the nonlinearity and hysteresis in the soft material. The Bayesian nonparametric formulation of GP makes it ideal for modeling the uncertainty and noise in the dynamics of soft sensors and actuators. Utilizing GP regression, researchers can construct the time-serial dependence in soft materials [21], [22], [23]. Moreover, owing to its natural Bayesian interpretation and predictive variance, the GP model can also be used to combine the state transition model and the measurement function into the Bayesian recursive estimation [19]. By utilizing GP regression, the Bayesian estimation approach can automatically learn the model and noise process from the training data. In light of this, we incorporate the GP models for the soft actuator and sensor into the unscented Kalman filter (UKF) to perform the recursive state estimation. The UKF applies the unscented transform to realize a more accurate approximation than the EKF [9] and exhibits enhanced accuracy for highly nonlinear functions.

The integration framework of GP and UKF was initially proposed by Ko et al. [24]. Recently, it has been applied in various fields, such as grip force estimation [25], rock movement prediction [26], and lithium-ion battery state of charge estimation [27]. However, its application in soft robotics is still unexplored. Specifically, the hysteresis property induced by soft materials brings high nonlinearity and strong time dependence, which challenges the GP model training and its integration with the UKF. A standard GP-UKF approach cannot accurately estimate the states of soft robots, and the tailored design of the system model for hysteresis characterization of a soft actuator–sensor system remains an open issue.

Another limitation of the GP approach is its computational complexities for model training and online inference, which are cubic and quadratic in the number of training points, respectively [24]. The increase in the number of training points also raises a memory burden during online implementation, since the storage capacity of a microcontroller is usually limited. To resolve this issue, we resort to the concept of edge computing, which offloads the computational and memory overhead to the edge server [28]. We set up wireless communication and only transmit a small amount of necessary data for executing computationally heavy estimation tasks, by which the communication latency is reduced. With the offloading scheme, the real-time performance of the entire solution is improved without any significant loss of accuracy.

The contributions of our article are summarized as follows.

- 1) A GP model is formulated to identify the dynamics of soft actuators and soft sensors, which can characterize the hysteresis and uncertainty in the soft material without using parametric models. Consequently, the GP model is more flexible and not limited by the conjectured parametric function.
- 2) We integrate the GP model with the UKF and implement the GP-UKF approach to improve the state estimation accuracy. The state-space model for the soft actuator–sensor system is designed and adapted to characterize the time dependence induced by hysteresis. The developed method is evaluated in real time to test the estimation precision. The GP-UKF showcases the advantages compared to the pure GP and the previous multihypothesis extended Kalman filter (MH-EKF) approach in estimation accuracy and computational resource efficiency.
- 3) To circumvent the problem of high computational and memory overhead in the GP inference, we offload the online estimation task from the microcontroller to an edge server. The offloading architecture guarantees real-time performance and high estimation accuracy. The edge computing scheme via wireless communication provides more flexibility for future applications of the soft actuator.

The rest of this article is organized as follows. Section II first constructs the probabilistic model to represent the hysteresis, with which the dynamics models of soft actuators and soft sensors are identified. Then, the GP-UKF approach is derived and used as a filtering approach to improve the estimation accuracy. We introduce the experimental setup and offloading scheme in Section III. Section IV shows the evaluation and results of our approach. Finally, Section V concludes this article.

II. MODELING OF SOFT ACTUATOR–SENSOR SYSTEM BASED ON GP REGRESSION

A. System Modeling by the GP

This article uses the same type of soft actuator–sensor system in our previous study [20], which consists of a soft actuator and the proprioceptive soft sensor, as illustrated in Fig. 1. The soft sensor is printed onto the soft actuator directly via multimaterial 3-D printing methods. Its electric resistance is deformation responsive owing to the changed distribution of carbon black

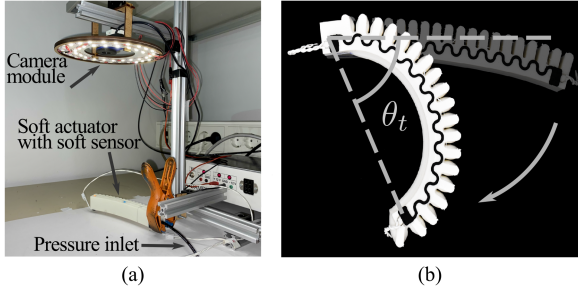


Fig. 1. Experimental setup. (a) Test rig and 3-D printed soft actuator. (b) 3-D printed soft actuator with the integrated soft sensor. The soft actuator is deformed with the bending angle θ_t .

during deformation. This proprioceptive soft actuator–sensor system constructs a state estimator.

We first elaborate the dynamic models of the soft actuator and sensor, whose state variables are $\mathbf{x}_{a,t}$ and $\mathbf{x}_{s,t}$, respectively. The subscript a/s represents actuator/sensor, and subscript t represents the time step. Inspired by the *probability box* method proposed in [19], we formulate the hysteresis of the soft actuator and sensor by considering their output defined by a probability distribution function. The sensor state $\mathbf{x}_{s,t}$ consists of three elements. To express the time dependence, we consider that the sensor output is influenced by values of the bending angle in two consecutive periods of time (i.e., θ_{t-1} and θ_t). In addition, owing to the memory effect of hysteresis [12], we include the sensor value of resistance in the last time step r_{t-1} as a state variable, which leads to the sensor model function $f_s: \mathbb{R}^3 \rightarrow \mathbb{R}$, and

$$r_t = f_s(\mathbf{x}_{s,t}) \quad (1a)$$

$$\mathbf{x}_{s,t} = [r_{t-1}, \theta_{t-1}, \theta_t]^\top. \quad (1b)$$

Similarly, the soft actuator $f_a: \mathbb{R}^3 \rightarrow \mathbb{R}$ can be modeled with

$$\theta_{t+1} = f_a(\mathbf{x}_{a,t}) \quad (2a)$$

$$\mathbf{x}_{a,t} = [\theta_t, P_{t-1}, P_t]^\top \quad (2b)$$

where P_t is the input pressure.

Due to the reality gap, the model functions f_s and f_a have modeling errors. Therefore, this study applies GP models to describe the probabilistic distribution of functions f_s and f_a . Due to their similarity, we only show the derivation of GP regression of the soft actuator as an example. The GP model of the soft actuator is characterized by the mean function $m_a(\mathbf{x}_{a,t})$ and covariance function $k_a(\mathbf{x}_{a,t}, \mathbf{x}_{a,t'})$, i.e., $f_a(\mathbf{x}_{a,t}) \sim \mathcal{GP}(m_a(\mathbf{x}_{a,t}), k_a(\mathbf{x}_{a,t}, \mathbf{x}_{a,t'}))$. In the formula, $m_a(\mathbf{x}_{a,t}) = \mathbb{E}[f_a(\mathbf{x}_{a,t})] \in \mathbb{R}$, and $k_a(\mathbf{x}_{a,t}, \mathbf{x}_{a,t'}) = \mathbb{E}[(f_a(\mathbf{x}_{a,t}) - m_a(\mathbf{x}_{a,t}))(f_a(\mathbf{x}_{a,t'}) - m_a(\mathbf{x}_{a,t'}))] \in \mathbb{R}_+$, where t and t' are two time indexes. The mean function is normally initialized as zero with $m_a(\mathbf{x}_{a,t}) = 0$. Moreover, this study selects the squared exponential kernel as the kernel function for the covariance function, with the expression of [29]

$$k_a(\mathbf{x}_{a,t}, \mathbf{x}_{a,t'}) = \lambda_f^2 \exp \left[-\frac{1}{2} (\mathbf{x}_{a,t} - \mathbf{x}_{a,t'})^\top \Lambda^{-1} (\mathbf{x}_{a,t} - \mathbf{x}_{a,t'}) \right] + \lambda_\omega^2 \delta(\mathbf{x}_{a,t}, \mathbf{x}_{a,t'}) \quad (3)$$

where λ_f^2 is the amplitude hyperparameter denoting signal variance, λ_ω^2 is the hyperparameter representing the measurement noise variance, $\Lambda = \text{diag}([\lambda_1^2, \lambda_2^2, \lambda_3^2])$ is the scaling factor along each dimension of the input space, and $\delta(\cdot, \cdot)$ is the Kronecker delta function. The kernel function in (3) follows the form by considering the noisy observations (see, e.g., [30, Sec. 2.2]). The kernel hyperparameters $\boldsymbol{\lambda} = [\lambda_f^2, \lambda_\omega^2, \Lambda]$ have a significant impact on the shape of regression functions, and they can be optimally defined by log marginal likelihood [29]. With a training dataset $\mathcal{T}_a = \{X_a, \boldsymbol{\theta}\}$ of n training input samples $X_a = [\mathbf{x}_{a,1}, \dots, \mathbf{x}_{a,n}]$ and their corresponding outputs $\boldsymbol{\theta} = [\theta_2, \dots, \theta_{n+1}]$, the optimal values $\boldsymbol{\lambda}_a^*$ can be acquired by maximizing the likelihood of the observed outputs through

$$\begin{aligned} \boldsymbol{\lambda}_a^* &= \arg \max_{\boldsymbol{\lambda}} \log p(\boldsymbol{\theta} | X_a, \boldsymbol{\lambda}) \\ &= -\frac{1}{2} \boldsymbol{\theta} K_a^{-1} \boldsymbol{\theta}^\top - \frac{1}{2} \log |K_a| - \frac{n}{2} \log 2\pi \end{aligned} \quad (4)$$

where K_a is the $n \times n$ covariance matrix and

$$K_a(i, j) = k_a(\mathbf{x}_{a,i}, \mathbf{x}_{a,j}). \quad (5)$$

Note that K_a is symmetric and positive semidefinite because of the influence of λ_ω^2 . Then, given an arbitrary test state $\mathbf{x}_{a,t}$, the function value $f_a(\mathbf{x}_{a,t})$ is jointly Gaussian with the training inputs X_a , with the prior as

$$\begin{aligned} p \left(\begin{bmatrix} f_a(X_a) \\ f_a(\mathbf{x}_{a,t}) \end{bmatrix} \middle| \mathbf{x}_{a,t}, \boldsymbol{\lambda}_a^*, X_a \right) \\ = \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} K_a & k_a(X_a, \mathbf{x}_{a,t}) \\ k_a(X_a, \mathbf{x}_{a,t})^\top & k_a(\mathbf{x}_{a,t}, \mathbf{x}_{a,t}) \end{bmatrix} \right) \end{aligned} \quad (6)$$

where $f_a(X_a) = [f_a(\mathbf{x}_{a,1}), \dots, f_a(\mathbf{x}_{a,n})]^\top$. $k_a(X_a, \mathbf{x}_{a,t})$ is an $n \times 1$ vector, and each entry denotes the kernel function value between $\mathbf{x}_{a,t}$ and $\mathbf{x}_{a,t'} \in X_a$. $\mathbf{0}$ is a column vector of zeros with the length $n+1$, as we consider the prior mean function as zero (i.e., $m_a(\mathbf{x}_{a,t}) = 0$). Based on (6), the distribution for the predicted output θ_{t+1} is Gaussian, yielding

$$p(\theta_{t+1} | \mathbf{x}_{a,t}, \boldsymbol{\lambda}_a^*, \mathcal{T}_a) = \mathcal{N}(\mu_{\theta,t+1}, \sigma_{\theta,t+1}) \quad (7)$$

and

$$\mu_{\theta,t+1} = k_a(X_a, \mathbf{x}_{a,t})^\top K_a^{-1} \boldsymbol{\theta}^\top \quad (8a)$$

$$\sigma_{\theta,t+1} = k_a(\mathbf{x}_{a,t}, \mathbf{x}_{a,t}) - k_a(X_a, \mathbf{x}_{a,t})^\top K_a^{-1} k_a(X_a, \mathbf{x}_{a,t}) \quad (8b)$$

where $\mu_{\theta,t+1}$ and $\sigma_{\theta,t+1}$ are the mean and variance of θ_{t+1} , respectively. The sensor model estimation f_s through GP regression can be derived similarly. We omit its derivation here for simplicity but will give the necessary formulations in the next section when integrating with the UKF.

B. UKFs With GP Models

Note that when we derive the GP formulation in the previous section, the input state $\mathbf{x}_{a,t}$ is regarded as deterministic points without considering uncertainty. Nonetheless, the GP model outputs (i.e., θ_{t+1}) are expressed in the stochastic Gaussian

form. During the online estimate where the long-term prediction is performed, if we directly incorporate the stochastic state θ_t generated by the GP model into the state expression in (2b), the nondeterministic state $\mathbf{x}_{a,t}$ will become a non-Gaussian prediction of θ_{t+1} during the iterations. Such implementation will be problematic in the long run, as already demonstrated in previous studies [19], [31]. Kim et al. [19] circumvent the variance and only utilize the mean value from the GP model, by which $\mathbf{x}_{a,t}$ can be expressed deterministically. However, the method does not consider the distribution of the input signal and may lead to a growing estimation error. In addition, Deisenroth et al. [31] propose the assumed density filter, where the Gaussian form is preserved through moment matching, and it can capture the true mean and the true covariance of the posterior distribution [9]. However, its high computational complexity restricts the real-time implementation.

These problems motivate us to investigate a new method to perform the online estimation with stochastic states in the field of soft robotics. Since the GPs naturally consider the uncertainty and noise in the model, we can improve the state prediction accuracy through the sequential state estimation. Specifically, this study leverages the UKF to retain the Gaussian distribution with high efficiency. Compared to the EKF approach [19], [20], the UKF performs *unscented transform*, which is a stochastic approximation by a weighted statistical linear regression process. In addition, the UKF is a derivative-free filter that does not require the computation of Jacobians. By contrast, EKF implementation will bring difficulties in our case, since the dynamic functions (i.e., f_a and f_s) are not explicitly expressed in the GP approach. Also, compared to other sample-based filters, e.g., particle filters, the UKF adopts the Gaussian distribution assumption, which matches the GP posterior output and improves the computational efficiency in real-time applications.

In this section, we present the unscented transform by referring to the definitions in [9] and [24]. Same to the last section, we continue to use the actuator model as the representative. In addition, note that the GP models derived in (1a) and (2a) are regarded as the state prediction function in the Kalman filter, and we shall denote their outputs by \hat{r}_t and $\hat{\theta}_{t+1}$ in the following.

To fit in the UKF scheme, we express a nonlinear actuator-sensor state-space model in the discrete-time domain by

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, u_t) \quad (9a)$$

$$z_{t+1} = g(\mathbf{x}_{t+1}) \quad (9b)$$

where f and g denote the state transition and observation model function, \mathbf{x}_t is the system state at time step t , and u_t is the input signal. z_{t+1} is the observation value, which differentiates \hat{r}_t since the latter is the ideal estimation. We regard the soft actuator-sensor system as an entirety and define the state by $\mathbf{x}_t = [\theta_t \ r_t \ P_{t-1}]^\top$ and the input by $u_t = P_t$ to reflect the hysteresis influence. The state \mathbf{x}_t is viewed as a random model, which is represented by the posterior probabilities based on the past measurements and controls. We denote by $\theta_t \sim \mathcal{N}(\mu_{\theta,t}, \sigma_{\theta,t})$ and $r_t \sim \mathcal{N}(\mu_{r,t}, \sigma_{r,t})$ the estimated bending angle and sensor value in \mathbf{x}_t . Thus, the state is $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ with $\boldsymbol{\mu}_t = [\mu_{\theta,t}, \mu_{r,t}, P_{t-1}]^\top$. Σ_t is diagonal (i.e., $\Sigma_t = \text{diag}(\sigma_{\theta,t}, \sigma_{r,t}, \sigma_P)$) only during initialization (i.e., $t = 1$).

The off-diagonal elements are nonzero after the measurement update in the UKF.

Moreover, considering the influence of external disturbances, the input pressure is also stochastic with time-invariant variance σ_P . Thus, the system model (9) can be expanded with

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, u_t) = \begin{bmatrix} f_a([\theta_t, P_{t-1}, P_t]^\top) \\ f_s([r_t, \theta_t, \hat{\theta}_{t+1}]^\top) \\ P_t \end{bmatrix} \quad (10a)$$

$$z_{t+1} = g(\mathbf{x}_{t+1}) = C\mathbf{x}_{t+1} + \epsilon_z \quad (10b)$$

where $\hat{\theta}_{t+1}$ is the estimation value from (2), $C = [0 \ 1 \ 0]$, and ϵ_z is the sensor measurement noise with $\epsilon_z \sim \mathcal{N}(0, \sigma_z)$.

The UKF applies deterministic sampling to build a small set of sample points and propagates them through the nonlinear model functions, from which the new Gaussian estimation is formed. Given the posterior distribution from the time step t , we first extract *sigma points* from this distribution and pass them through the model transition function f . The sigma points are located at the mean and symmetrically along the main axes of the covariance. The dimension of the sigma points depends on the number of random variables. Since the uncertainty is considered in u_t , note that herein the dimension of multivariate Gaussian distribution is $n = |\mathbf{x}_t| + |u_t| = 4$. We denote $\mathbf{q}_t = [\mathbf{x}_t; u_t] \in \mathbb{R}^4$ to represent the random variable in unscented transform. There are total $2n + 1$ sigma points $\mathcal{Q}_t = \{\mathbf{q}_t^{[i]} | i = 0, \dots, 2n\}$, and

$$\mathbf{q}_t^{[0]} = \boldsymbol{\mu}_t^{\mathbf{q}} \quad (11a)$$

$$\mathbf{q}_t^{[i]} = \boldsymbol{\mu}_t^{\mathbf{q}} + \left(\sqrt{(n + \tilde{\lambda})\Sigma_t^{\mathbf{q}}} \right)_i, \quad i = 1, \dots, n \quad (11b)$$

$$\mathbf{q}_t^{[i]} = \boldsymbol{\mu}_t^{\mathbf{q}} - \left(\sqrt{(n + \tilde{\lambda})\Sigma_t^{\mathbf{q}}} \right)_{i-n}, \quad i = n + 1, \dots, 2n \quad (11c)$$

where $\boldsymbol{\mu}_t^{\mathbf{q}} = [\boldsymbol{\mu}_t; P_t]$, $\Sigma_t^{\mathbf{q}} = \text{diag}(\Sigma_t, \sigma_P)$ is a block diagonal matrix, $(\cdot)_i$ selects the i th column of the matrix, and $\tilde{\lambda}$ is a scaling parameter that determines how far the sigma points are spread from the mean [32]. The sigma points are given as the inputs for the model function f , and the corresponding output is the set $\hat{\mathcal{X}}_{t+1} = \{\hat{\mathbf{x}}_{t+1}^{[i]} | i = 0, \dots, 2n\}$, where $\hat{\mathbf{x}}_{t+1}^{[i]} = f(\mathbf{x}_t^{[i]}, u_t^{[i]})$. $\hat{\mathbf{x}}_{t+1}^{[i]}$ follows Gaussian distribution and $\hat{\mathbf{x}}_{t+1}^{[i]} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{t+1}^{[i]}, \hat{\Sigma}_{t+1}^{[i]})$, where $\hat{\boldsymbol{\mu}}_{t+1}^{[i]}$ and $\hat{\Sigma}_{t+1}^{[i]}$ are estimated from (8) and (10a). Since the set \mathcal{Q}_t is transferred through the nonlinear function f , the output $\hat{\mathcal{X}}_{t+1}$ will not likely be distributed in the Gaussian form. In order to possess the Gaussian distribution property, we extract the mean $\bar{\boldsymbol{\mu}}_{t+1}$ and variance $\bar{\Sigma}_{t+1}$ of the assumed Gaussian from the mapped sigma points $\hat{\mathcal{X}}_{t+1}$ with

$$\bar{\boldsymbol{\mu}}_{t+1} = \sum_{i=0}^{2n} w_m^{[i]} \hat{\boldsymbol{\mu}}_{t+1}^{[i]} \quad (12a)$$

$$\bar{\Sigma}_{t+1} = \sum_{i=0}^{2n} w_c^{[i]} \left[(\hat{\boldsymbol{\mu}}_{t+1}^{[i]} - \bar{\boldsymbol{\mu}}_{t+1})(\hat{\boldsymbol{\mu}}_{t+1}^{[i]} - \bar{\boldsymbol{\mu}}_{t+1})^\top + \hat{\Sigma}_{t+1}^{[i]} \right] \quad (12b)$$

where $\mathcal{W}_m = \{w_m^{[i]} | i = 0, \dots, 2n\}$ and $\mathcal{W}_c = \{w_c^{[i]} | i = 0, \dots, 2n\}$ are the weights for sigma points when recovering the mean and covariance of the Gaussian distribution (see, e.g., [9, Sec. 3.4.1]). Equations (11) and (12) establish the essence of unscented transfer that preserves the Gaussian property after the nonlinear function computation. Instead of performing linearization around the mean point as in the EKF, the unscented transform attains the state prediction with Gaussian form without the calculation of Jacobians.

By performing unscented transfer based on (11) and (12), we obtain the prior distribution of the state at time step $(t+1)$ following Gaussian distribution with $\bar{\mathbf{x}}_{t+1} \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_{t+1}, \bar{\boldsymbol{\Sigma}}_{t+1})$, where $\bar{\mathbf{x}}_{t+1} = [\bar{\theta}_{t+1} \quad \bar{r}_{t+1} \quad P_t]^\top$. Note that $\bar{\theta}_{t+1}$ is different from $\hat{\theta}_{t+1}$, where $\hat{\theta}_{t+1}$ is obtained from a single point (i.e., $\boldsymbol{\mu}_t$) projection through f_a , while $\bar{\theta}_{t+1}$ is the unscented transform on f when taking the posterior distribution $p(\mathbf{x}_t)$ as the input, which enables us to capture the overall uncertainty during the state prediction.

We use the observation model g and measurement z_{t+1} for the state estimation update in the following step. Note that in (10b), the observation model is linear; thus, we can skip the unscented transform on g as in the conventional UKF, and the predicted observation $\bar{z}_{t+1} \sim \mathcal{N}(\bar{\mu}_{z,t+1}, \bar{\sigma}_{z,t+1})$ can be expressed directly with

$$\bar{\mu}_{z,t+1} = C\bar{\boldsymbol{\mu}}_{t+1} = \mu_{r,t+1} \quad (13a)$$

$$\bar{\sigma}_{z,t+1} = C\bar{\boldsymbol{\Sigma}}_{t+1}C^\top + \sigma_z = \sigma_{r,t+1} + \sigma_z. \quad (13b)$$

Accordingly, the Kalman gain $K_{t+1} \in \mathbb{R}^3$ is calculated by

$$K_{t+1} = \bar{\boldsymbol{\Sigma}}_{t+1}C^\top(\bar{\sigma}_{z,t+1})^{-1}. \quad (14)$$

Subsequently, with the measurement z_{t+1} , the posterior state estimation $\mathbf{x}_{t+1} \sim \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$ can be updated by

$$\boldsymbol{\mu}_{t+1} = \bar{\boldsymbol{\mu}}_{t+1} + K_{t+1}(z_{t+1} - \bar{\mu}_{z,t+1}) \quad (15a)$$

$$\boldsymbol{\Sigma}_{t+1} = \bar{\boldsymbol{\Sigma}}_{t+1} - K_{t+1}\bar{\sigma}_{z,t+1}K_{t+1}^\top. \quad (15b)$$

Thus, shape estimation $p(\theta_{t+1})$ can be derived from (15). Note that the off-diagonal elements in $\boldsymbol{\Sigma}_{t+1}$ are nonzero during update due to (15b). Algorithm 1 presents the pseudocode for the GP-UKF online implementation. The initialized variance values in Line 1 are estimated based on the observations and experiment, and more details about the numerical parameter initialization are presented in Section IV. Note that lines 2 and 3 can be estimated offline based on the collected training data before the real-time implementation.

As mentioned previously, one main advantage of integrating GP models with the UKF is that the values of estimation uncertainty (i.e., $\hat{\sigma}_{\theta,t}^{[i]}, \hat{\sigma}_{r,t}^{[i]}$ in $\hat{\boldsymbol{\Sigma}}_t^{[i]}$) can be calculated adaptively by leveraging GP regression, whereas in the conventional UKF, they are manually tuned and often static. With a higher density of training data around the model state, a lower level of state uncertainty variance can be determined, thus achieving the automatic adaptation of the noise variance value.

Algorithm 1: GP-UKF: Soft Actuator State Estimation.

Data: Training data sets for actuator and sensor $\mathcal{T}_a, \mathcal{T}_s$, weights for sigma points $\mathcal{W}_m, \mathcal{W}_c$

Result: Gaussian posterior $p(\theta_t)$

- 1 Initialize $\boldsymbol{\mu}_1 = [0, 0, 0]^\top$, $\boldsymbol{\Sigma}_1 = \text{diag}(1, 1.8, 0.001)$, $\sigma_z = 1.8$, $z_0 = 0$, $t = 1$;
- 2 Calculate $K_a, K_s, X_a, X_s, \boldsymbol{\theta}$ based on $\mathcal{T}_a, \mathcal{T}_s$;
- 3 Get hyperparameters λ_a^*, λ_s^* of GP models by (4);
- 4 **while true do**
- 5 Receive the pressure input P_t ;
- 6 Build a sigma points set \mathcal{Q}_t by (11);
- 7 **foreach** $q_t^{[i]} \in \mathcal{Q}_t$ **do**
- 8 Perform GP regression to get $\hat{\mathbf{x}}_{t+1}^{[i]}$ by (7), (8) and (10a);
- 9 **end**
- 10 Combine sigma points to get $p(\bar{\theta}_{t+1})$ by (12);
- 11 Calculate the predicted observation $p(\bar{z}_{t+1})$ by (13);
- 12 Calculate Kalman gain K_t by (14);
- 13 Obtain the posterior $p(\mathbf{x}_{t+1})$ by (15);
- 14 Update time step: $t \leftarrow t + 1$;
- 15 **end**

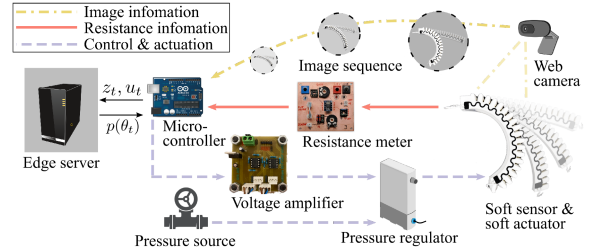


Fig. 2. Overview of the experiment system.

III. EXPERIMENT SETUP

A. Data Acquisition System

A thermoplastic polyurethane (TPU)-printed soft actuator is fabricated for the validation of the proposed method. Meanwhile, we print the soft sensor onto the actuator directly via multimaterial 3-D printing methods with NinjaTek EEL™ material, which is a mixture of TPU and carbon black [33].

Fig. 2 gives an overview of the experiment setup. The soft actuator deformation is induced by the air pressure from the proportional pressure regulator (Festo VEAB-L-26-D9-Q4-V1-1R1), which is connected to the pressure source of 6 bar at maximum and controlled by an Arduino Due microcontroller board. In addition, to guarantee the accuracy of pressure outputs from the regulator, we calibrate and verify the pressure output with a pressure sensor (Festo SPAU-P6R-W-Q4D-L-PNLK-PNVBA-M12U) before the experiments. We design a resistance meter circuit using a Wheatstone bridge to convert the varied sensor resistances during bending into analog voltage signals, which are then read by the microcontroller board as

the sensor measurement. A compiled program is downloaded to the microcontroller, which performs the estimation and control tasks. We design a proportional–integral (PI) controller with the feedforward part to realize the bending angle control [20]. After receiving the control signal calculated by the microcontroller, the voltage amplifier module sends the corresponding electric voltage level signal ranging from 0 to 10 V to the pressure regulator to drive the soft actuator with deformation.

In addition, we monitor the system status and send reference control commands to the board in real time through MATLAB Simulink external mode. As illustrated in Fig. 1(a), we use a web camera to detect the bending angle of the soft actuator by the image feedback module, which is regarded as the ground truth signal for comparison. The image feedback module uses a Logitech C270 HD web camera, which is located on top of the soft actuator and captures real-time pictures at 30 frames/s. The image capture and processing modules are carefully calibrated before conducting the experiments, and the environment of the camera module is static once we finish the calibration, to ensure that the bending angle accuracy captured by the web camera will not be influenced during the training and testing phases. The gray area on the left of Fig. 2 represents the alternative offloading schemes, where the estimation task is offloaded to an edge server. Its implementation is elaborated in the next section.

B. Offloading Scheme

According to our previous study [20], the step response rise time for the same type of soft actuator is around 800 ms. Thus, we define the estimation and control period in this task as 50 ms to ensure that the control frequency is ten times faster. However, the computational complexity poses a challenge to execute the estimation task with the GP approach on the microcontroller in real time. The microcontroller board has limited capacity in both computational speed and memory size, featuring 84-MHz clock speed, 96-kB SRAM, and 512-kB flash memory. Recall that in (8), the calculation of posterior estimation requires the information of $K_a \in \mathbb{R}^{n \times n}$, $X_a \in \mathbb{R}^{3 \times n}$, and $\theta \in \mathbb{R}^n$, which leads to the quadratic space complexity $\mathcal{O}(n^2)$ to the number of training data n during the online GP inference. With the growth of training data, the periodic task may not be schedulable for computation or affordable to write in the on-chip memory. We keep the data as the double-precision type both in the storage and memory to guarantee the accuracy of estimation and bending angle control. Thus, when the number of training points is 64, the memory overhead for the GP parameters alone is over 70 kB, posing a big challenge for SRAM capacity. In light of this, we set the maximal number of training points to 64 in this study.

To further resolve this resource deficiency issue, we resort to an edge computing scheme and offload the estimation task from the microcontroller to the edge. The edge server is a desktop PC with Intel(R) Core(TM) i7-8700 CPU at 3.20 GHz, six cores, and 8.0-GB installed memory (RAM). We build the wireless communication between the microcontroller and the edge server through Wi-Fi. Due to the lack of wireless communication capacity on the microcontroller, we connect it with an ESP82-01S board as the Wi-Fi serial transceiver module, which

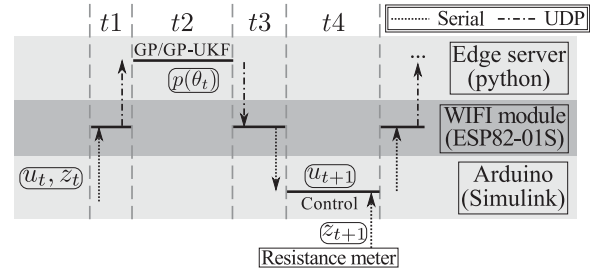


Fig. 3. Overview of the communication path.

supports Wi-Fi at 2.4 GHz and up to 72.2 Mb/s of data rate. We select the user datagram protocol (UDP) as the protocol due to its simplicity and efficiency. ESP82-01S board transmits UDP packets between the microcontroller and the edge server through Wi-Fi.

The communication path in the edge-enabled approach can be found in Fig. 3. The microcontroller handles sensor reading and pressure control tasks, where z_t and u_t are sent to the Wi-Fi module through Serial communication. After that, the data are sent to the edge server by the Wi-Fi module, where the GP/GP-UKF are computed. After the edge computation, the posterior estimation $p(\theta_t)$ in (15) is then sent back to the microcontroller for shape control in the next period, via the Wi-Fi module for transferring the UDP packets to Serial messages. To avoid the buffering of Serial data, the data transmission frequency is set to the same as the control frequency with 20 Hz. Since the estimation and control tasks are accomplished on different nodes, the end-to-end delay in the closed-loop control system can be attributed to four segments, as denoted by $t1-t4$ in Fig. 3. $t1$ and $t3$ represent the communication delays between the microcontroller and the edge server, which contain both the Serial and UDP transmissions. $t2$ and $t4$ denote the computation delays from the estimation and control tasks, respectively. To meet the schedulability requirement, the sum of $t1-t4$ should be less than 50 ms, thus enabling the latest control signal u_t to drive the actuator in each period. Based on the signal ranges, we apply *fixed point representation* and keep two decimal places for each data point, which limits the communication overhead by sending each data point with 2 bytes.

IV. RESULTS AND VALIDATION

We compare the performance of GP-UKF with the following methods for evaluation.

- 1) *GP* [30], [22]: The bending angle θ_t is predicted by multistep GP regression through (2) without sensor feedback.
- 2) *MH-EKF* [20]: The actuator is modeled by a second-order state-space function, and the sensor is modeled by several quadratic functions. The Kalman filter is performed through multihypothesis correspondence.
- 3) *Bayes filter (BF)* [9], [19]: The models are still represented by GP regression. The posterior distribution is calculated by the BF without the unscented transform. The BF is only used to compare the variance estimation in Section IV-B.

TABLE I
HYPERPARAMETERS FOR THE GP MODELS

	λ_f^2	λ_w^2	λ_1^2	λ_2^2	λ_3^2
actuator (λ_a^*)	3278.3	0.0575	61.2754	28193	87528
sensor (λ_s^*)	2221.2	0.0044	5503.9	3986.8	1634.6

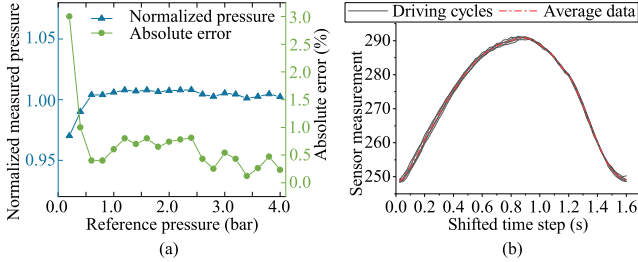


Fig. 4. Variance verification of measurement for input pressure and output resistance. (a) Pressure input verification. (b) Output resistance verification.

In addition, a web camera is applied to provide the ground truth of the estimation by image processing. We use **Camera** feedback as the true value to calculate the estimation error.

The training data for GP models are collected by driving the actuator with sine waves with positive offsets so that the minimal value is 0° . The maximal values of the sine waves change from 10° to 50° , and the frequency of all waves is 2 rad/s. The hyperparameters of GP models are identified offline through optimizing (4) with the GPML toolbox.¹ We present the identified kernel hyperparameters of GP models for the actuator (λ_a^*) and the sensor (λ_s^*) in Table I.

To initialize the parameters in the UKF, we estimate the values of σ_P and σ_z through experiments. As can be seen in Fig. 4(a), different pressure references P_r are set to the pressure regulator, and the readings from the pressure sensor are recorded to calculate the error. The left vertical axis shows the measured pressure P normalized by the pressure reference P_r , and the right axis shows the error calculated by $|(P - P_r)/P_r|$. The figure shows that the directly regulated pressure by the pressure regulator is higher for low pressures less than 0.5 bars, but the errors are still at low levels less than 3%. For higher pressures, the errors are always less than 1%. The verification result shows the high accuracy of the pressure regulator. For simplicity, we consider σ_P as a constant, and assume the average absolute error sampled in Fig. 4(a) as one standard deviation, by which we calculate $\sigma_P \approx 0.001$.

The verification of the output resistance value is performed similarly. We measure its stabilized value when driving the actuator with the pressure of a sine wave multiple times. In addition, we shift the time step by offsetting each driving cycle with the start time at the origin and plot the corresponding sensor output in Fig. 4(b), where the average value is also plotted for comparison. The output resistance value in different driving cycles has a standard deviation of 1.3340, and a variance of 1.7796. Thus, we determine the sensor measurement and initial resistance variance with $\sigma_z = \sigma_{r,0} \approx 1.8$.

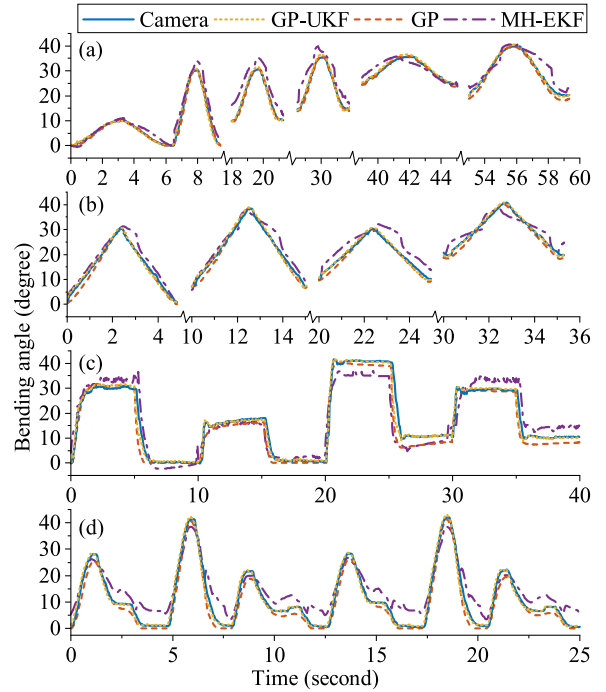


Fig. 5. Estimation accuracy comparison with (a) sine waves, (b) triangle waves, (c) square waves, and (d) a random signal.

A. Comparison With Benchmarks

As shown in Fig. 5, the shape estimation accuracy is assessed by driving the actuator with different positive reference signals. Note that we keep the maximal bending angles of the soft actuator at 40° to ensure that it is within the domain of the training dataset. In Fig. 5(a) and (b), we compare the estimation performance with sine and triangle waves of different amplitudes, biases, and frequencies. All three approaches can retain high accuracy with 0° as the origin. However, performance deterioration can be observed in the MH-EKF with the angle biases, which is an open issue identified in [20] due to the model limitation. Compared to the MH-EKF, the GP-class approaches improve the accuracy by adopting the data-driven method with higher model complexity. In Fig. 5(c) and (d), when the bending angle is static, the GP-UKF outperforms the GP with a more noticeable advantage as a result of incorporating sensor feedback. The reason can be attributed to the imperfect coverage of input space in the training data. Since the training data are composed of sine waves, the pressure is continuously varied, where the stationary input state is not included in the training; thus, the model accuracy is influenced. The GP-UKF has a better performance owing to the additional sensor feedback for the estimation adjustment.

Table II lists the estimation error of different types of signals, where the normalized root-mean-square error (NRMSE) is calculated by the range of data from root-mean-square error (RMSE). The square wave has a much higher max error than the other signals because the response time latency incurs a large error with the sudden change in reference, even though this large error is not that noticeable in Fig. 5(c). The GP-UKF

¹[Online]. Available: <http://gaussianprocess.org/gpml/code/matlab/doc/>

TABLE II
ESTIMATION ERROR COMPARISON WITH DIFFERENT TYPES OF SIGNAL

		RMSE	NRMSE	mean error	max error
sine	GP	0.959	0.024	0.698	3.130
	GP-UKF	0.917	0.022	0.692	3.830
	MH-EKF	3.273	0.081	2.535	8.780
triangle	GP	1.304	0.032	1.072	3.170
	GP-UKF	0.837	0.021	0.702	2.050
	MH-EKF	2.971	0.073	2.457	7.870
square	GP	3.301	0.079	2.082	18.670
	GP-UKF	1.876	0.045	0.847	13.480
	MH-EKF	4.818	0.116	3.610	27.130
random	GP	2.154	0.050	1.755	6.460
	GP-UKF	1.318	0.031	0.958	4.900
	MH-EKF	4.456	0.106	3.909	9.180

TABLE III
ESTIMATION ERROR COMPARISON WITH A RANDOM REFERENCE SIGNAL

		RMSE	NRMSE	mean error	max error
num64	GP	2.049	0.053	1.544	6.750
	GP-UKF	1.311	0.027	0.931	5.230
num13	GP	2.902	0.078	2.354	8.220
	GP-UKF	2.188	0.058	1.816	7.150
num7	GP	6.236	0.168	4.850	18.120
	GP-UKF	4.695	0.127	3.680	14.900

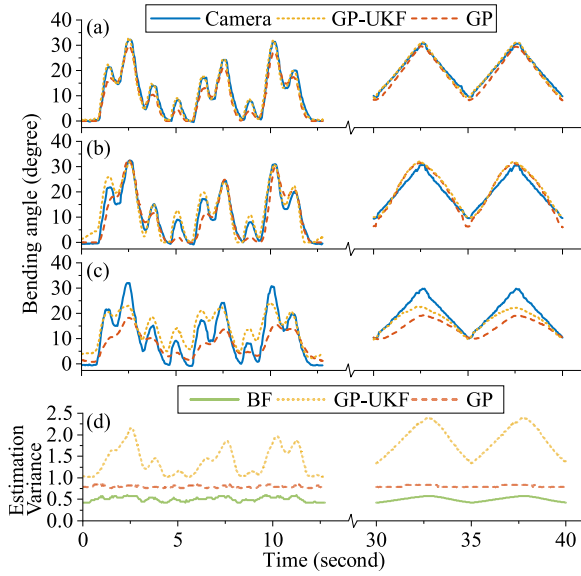


Fig. 6. Estimation accuracy comparison by different sampling densities with (a) num64, (b) num13, (c) num7, and (d) estimation variance comparison.

enjoys a consistent higher accuracy among the three approaches, where the NRMSE value is reduced by 30.937% and 57.614% compared to the GP and the MH-EKF on average.

B. Performance With Different Sampling Density

As mentioned in Section III-B, the GP approach involves a much higher computational and memory overhead compared to the traditional methods. Besides the offloading approach for handling the computationally heavy tasks at the edge, another circumvention is to sacrifice the accuracy by reducing the number of training data and saving computation and memory overhead. We compare the tradeoff between the estimation accuracy and the overhead by checking the performance when the number of training data varies.

Fig. 6(a), (b), and (c) compare the estimation performance under three training data sampling densities, i.e., num64, num13, and num7. As mentioned in Section III-B, the maximal number of training datasets is limited to 64 due to memory capacity, which corresponds to the num64 scenario. The numbers of training points in num13 and num7 are 13 and 7, which are

five and ten times less, respectively, compared to the num64 scenario.

The reference signals in Fig. 6 are a random signal and a triangle wave. Note that since two models need to be trained (i.e., actuator and sensor), the amount of training data in the GP-UKF is doubled compared to the GP with the same sampling density level. It can be seen that, with less training data, the estimations of both methods show an increasing deviation from the ground truth Camera curve. Table III gives the estimation error comparison of the random reference signal in Fig. 6. Similar to Table II, the GP-UKF performs better than the GP under different densities. In addition, the GP-UKF at num13 maintains a similar level of accuracy compared to the GP at num64. Even though two separate models are required in the GP-UKF, it can achieve the same level of accuracy when the training data density is five times less, where the memory overhead is only 8% compared to its GP counterpart because of the quadratic space complexity $\mathcal{O}(n^2)$. This implies that the GP-UKF benefits from sensor feedback and is more resource efficient than the pure GP approach.

We compare the variance estimation from different methods at num13 in Fig. 6(d). The GP-UKF has a consistent higher variance compared to the GP, mainly owing to the unscented transfer contributing to more uncertainties during the prior state estimation. It can be seen that, when the estimation error varies in the num13 scenario in Fig. 6(b), the variance changes accordingly in Fig. 6(d), thus realizing the adaptive variance estimation. To further verify that the posterior belief should have a lower level of uncertainty compared to the prior distribution, we design a BF with GP models and exclude the unscented transfer step. Compared to the GP, the BF has a lower variance value. However, both methods show little ability to adjust the variance adaptively compared to the GP-UKF.

C. Hysteresis Loop Analysis

Since the hysteresis phenomenon exists in both the soft sensor and actuator, we compare the estimated hysteresis loop determined by different approaches as a criterion to evaluate the estimation accuracy. As shown in Fig. 7, we focus on the actuator hysteresis estimation by utilizing the sine wave as the reference signal and drawing the relationship between input pressure P_t and estimated angle θ_{t+1} . Besides, the GP-UKFs with two different sampling densities are plotted separately for comparison. It can be seen that the GP-UKF at num64 characterizes the hysteresis loop with the highest precision, while the GP at num64 has a similar trend to that of the GP-UKF at num13, which verifies the results in Table II. The MH-EKF fails to retain the

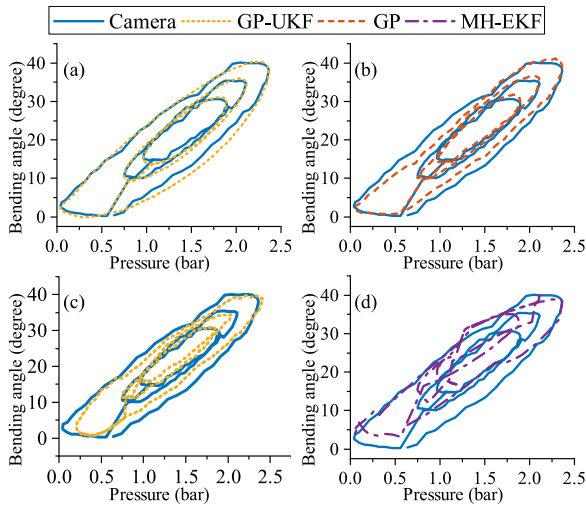


Fig. 7. Soft actuator hysteresis loop estimation comparison. (a) GP-UKF, num64. (b) GP, num64. (c) GP-UKF, num13. (d) MH-EKF, num64.

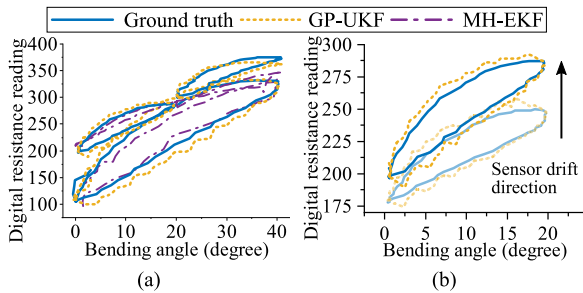


Fig. 8. Soft sensor hysteresis loop estimation. (a) Different amplitudes and biases. (b) Temporal variation of the loop due to the sensor drift.

shape of hysteresis loops in multiple scenarios, mainly due to its inability to handle the angle bias.

Fig. 8 illustrates the hysteresis loop of the soft sensor. Only the GP-UKF and the MH-EKF are compared since the GP is excluded from sensor feedback. Fig. 8(a) shows the sensor hysteresis loop in different sine waves of amplitudes and biases, where the GP-UKF has a significant advantage over the MH-EKF. Fig. 8(b) plots the temporal variation of the hysteresis loop owing to the sensor drift [12]. It can be seen that the GP-UKF achieves an accurate estimation against the time dependence issue.

D. Computational Efficiency Comparison

Fig. 9 analyzes the relationship between the computational overhead and the estimation accuracy. We use *response time* [34] to represent the estimation task overhead. The response time of the task is evaluated with both onboard and offloading schemes, where the latter enjoys a higher computational capacity but includes the communication delay. With the increase in the number of training datasets, the values of NRMSE gradually decrease, and a higher estimation accuracy is obtained. Meanwhile, the response time of the onboard approach climbs nearly in a quadratic shape. When the number of datasets exceeds 16, the periodic task of 50 ms becomes overdue for the GP-UKF. Due to its memory limitation, the microcontroller fails to handle the task when the

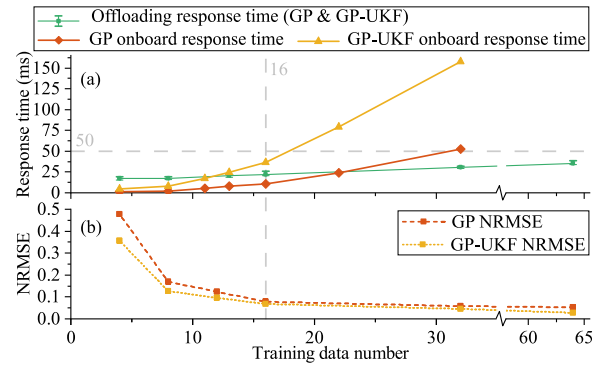


Fig. 9. Comparison by the number of training datasets. (a) Response time. (b) Estimation error (NRMSE).

TABLE IV

COMPARISON OF THE EXECUTION TIMES FOR EVERY TASK SEGMENT

	t_1	t_2	t_3	t_4
average time (ms)	3.566	10.054	1.449	4.338
maximal time (ms)	4.274	11.828	2.057	4.790

number of training datasets exceeds 32. Nevertheless, due to the high-computational capacity in the edge-based scheme, the offloading approach remains schedulable with the increase of training datasets.

As indicated by Fig. 3, the end-to-end delay in the closed-loop control system is decomposed into four parts. During the experiments, we compare the piecewise execution time for each part and demonstrate them in Table IV, in which t_1 – t_4 are defined in Fig. 3. The data shown in Table IV are collected by performing the GP-UKF approach in the num13 scenario, where the soft actuator is driven by a random signal. We show data by calculating the average and maximal execution times among the task periods in 5 min. The onboard execution times of the control algorithms are measured by Simulink profiling tools, and the execution times on the edge server are estimated by Python Profilers. Compared t_1 – t_4 , the most computationally demanding task is offloaded to the edge server, and the microprocessor only needs to handle the PI control and actuation jobs, which significantly reduces the onboard execution time and ensures real-time efficiency. In addition, the wireless data transmission enabled by UDP brings low communication latency with an average receiving time of $t_1 = 3.566$ ms and sending time of $t_3 = 1.449$ ms. The communication stability is also examined throughout the experiment by recording the number of lost packets, which results in a packet loss rate of 0.137% and guarantees the control accuracy of the offloading approach. The sum of execution times for all task segments in the closed-loop system is less than one task period, which makes sure that the control signal is calculated and actuated based on the latest sensor feedback and will not suffer any deadline misses.

V. CONCLUSION

This study investigated the probabilistic shape estimation problem in a soft actuator–sensor system. The data-driven model, enabled by GP regression, was applied to characterize both the soft actuator and soft sensor dynamics. In addition, to

further improve the accuracy of the bending angle estimation, a UKF was designed and collaborated with the GP models. A higher precision estimation system was achieved with the adaptive variance while addressing the hysteresis issue of soft materials. Moreover, the GP approach required a high computation and memory overhead in the online implementation. To enable real-time performance and guarantee the precision of estimation, we circumvented this issue by leveraging the edge computing scheme. The heavy computational estimation task was offloaded to the edge server through wireless communication, and the edge server approach clearly improved the real-time estimation performance through the verification with different types of reference signals. In future work, we will study soft sensor feedback for external force detection. Also, the drift property of the soft sensor will be characterized.

ACKNOWLEDGMENT

This research has been carried out as part of the TECoSA Vinnova Competence Center for Trustworthy Edge Computing Systems and Applications at KTH Royal Institute of Technology.

REFERENCES

- [1] T. Wallin, J. Pikul, and R. Shepherd, "3D printing of soft robotic systems," *Nature Rev. Mater.*, vol. 3, no. 6, pp. 84–100, 2018.
- [2] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [3] S. M. Mustaza, Y. Elsayed, C. Lekakou, C. Saaj, and J. Fras, "Dynamic modeling of fiber-reinforced soft manipulator: A visco-hyperelastic material-based continuum mechanics approach," *Soft Robot.*, vol. 6, no. 3, pp. 305–317, 2019.
- [4] H. Gu, H. Wang, F. Xu, Z. Liu, and W. Chen, "Active fault detection of soft manipulator in visual servoing," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9778–9788, Oct. 2021.
- [5] M. Luo et al., "A single-chamber pneumatic soft bending actuator with increased stroke-range by local electric guidance," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8455–8463, Sep. 2021.
- [6] G. Gerboni, A. Diodato, G. Ciuti, M. Cianchetti, and A. Menciassi, "Feedback control of soft robot actuators via commercial flex bend sensors," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 4, pp. 1881–1888, Aug. 2017.
- [7] D. Zhou, Y. Liu, X. Tang, and J. Zhao, "Differential sensing method for multidimensional soft angle measurement using coiled conductive polymer fiber," *IEEE Trans. Ind. Electron.*, vol. 68, no. 1, pp. 401–411, Jan. 2021.
- [8] T. Hainsworth, L. Smith, S. Alexander, and R. MacCurdy, "A fabrication free, 3D printed, multi-material, self-sensing soft actuator," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4118–4125, Jul. 2020.
- [9] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.
- [10] Q. Zhao, J. Lai, and H. K. Chu, "Reconstructing external force on the circumferential body of continuum robot with embedded proprioceptive sensors," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13111–13120, Dec. 2022.
- [11] Q. Ji, X. Zhang, M. Chen, X. V. Wang, L. Wang, and L. Feng, "Design and closed loop control of a 3D printed soft actuator," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng.*, 2020, pp. 842–848.
- [12] M. Schouten, G. Wolterink, A. Dijkshoorn, D. Kosmas, S. Stramigioli, and G. Krijnen, "A review of extrusion-based 3D printing for the fabrication of electro-and biomechanical sensors," *IEEE Sens. J.*, vol. 21, no. 11, pp. 12900–12912, Jun. 2021.
- [13] X. Liu, M. Huang, R. Xiong, J. Shan, and X. Mao, "Adaptive inverse control of piezoelectric actuators based on segment similarity," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5403–5411, Jul. 2019.
- [14] X. Zhang, B. Li, X. Chen, Z. Li, Y. Peng, and C.-Y. Su, "Adaptive implicit inverse control for a class of discrete-time hysteretic nonlinear systems and its application," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 4, pp. 2112–2122, Aug. 2020.
- [15] X. Zhang, H. Xu, X. Chen, Z. Li, and C.-Y. Su, "Modeling and adaptive output feedback control of butterfly-like hysteretic nonlinear systems with creep and their applications," *IEEE Trans. Ind. Electron.*, vol. 70, no. 5, pp. 5182–5191, May 2023.
- [16] Y.-D. Tao, H.-X. Li, and L.-M. Zhu, "Rate-dependent hysteresis modeling and compensation of piezoelectric actuators using Gaussian process," *Sens. Actuators A. Phys.*, vol. 295, pp. 357–365, 2019.
- [17] D. Kim, J. Kwon, B. Jeon, and Y.-L. Park, "Adaptive calibration of soft sensors using optimal transportation transfer learning for mass production and long-term usage," *Adv. Intell. Syst.*, vol. 2, no. 6, 2020, Art. no. 1900178.
- [18] M. Li, R. Kang, D. T. Branson, and J. S. Dai, "Model-free control for continuum robots based on an adaptive Kalman filter," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 1, pp. 286–297, Feb. 2018.
- [19] D. Kim, M. Park, and Y.-L. Park, "Probabilistic modeling and Bayesian filtering for improved state estimation for soft robots," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1728–1741, Oct. 2021.
- [20] K. Tan, Q. Ji, L. Feng, and M. Törngren, "Shape estimation of a 3D printed soft sensor using multi-hypothesis extended Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8383–8390, Jul. 2022.
- [21] X. Luo, M. Xiao, Y. Ding, and H. Ding, "Hysteresis modeling and compensation of a pneumatic end-effector based on Gaussian process regression," *Sens. Actuators A. Phys.*, vol. 315, 2020, Art. no. 112227.
- [22] Z. Q. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "A probabilistic model-based online learning optimal control algorithm for soft pneumatic actuators," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1437–1444, Apr. 2020.
- [23] Y. Cao, J. Huang, H. Ru, W. Chen, and C.-H. Xiong, "A visual servo-based predictive control with echo state Gaussian process for soft bending actuator," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 1, pp. 574–585, Feb. 2021.
- [24] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "GP-UKF: Unscented Kalman filters with Gaussian process prediction and observation models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 1901–1907.
- [25] Y. Li and B. Hannaford, "Gaussian process regression for sensorless grip force estimation of cable-driven elongated surgical instruments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1312–1319, Jul. 2017.
- [26] F. E. Sotiropoulos and H. H. Asada, "Autonomous excavation of rocks using a Gaussian process model and unscented Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2491–2497, Apr. 2020.
- [27] X. Chen, X. Chen, and X. Chen, "A novel framework for lithium-ion battery state of charge estimation based on Kalman filter Gaussian process regression," *Int. J. Energy Res.*, vol. 45, no. 9, pp. 13238–13249, 2021.
- [28] M. Törngren, H. Thompson, E. Herzog, R. Inam, J. Gross, and G. Dán, "Industrial edge-based cyber-physical systems-application needs and concerns for realization," in *Proc. IEEE/ACM Symp. Edge Comput.*, 2021, pp. 409–415.
- [29] M. Liu, G. Chowdhary, B. C. Da Silva, S.-Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Syst. Mag.*, vol. 38, no. 5, pp. 53–86, Oct. 2018.
- [30] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA, USA: MIT Press, 2006.
- [31] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 408–423, Feb. 2015.
- [32] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adaptive Syst. Signal Process., Commun., Control Symp.*, 2000, pp. 153–158.
- [33] Q. Ji, J. Jansson, M. Sjöberg, X. V. Wang, L. Wang, and L. Feng, "Design and calibration of 3D printed soft deformation sensors for soft actuator control," *Mechatron.*, vol. 92, 2023, Art. no. 102980.
- [34] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. Cambridge, MA, USA: MIT Press, 2016.



Kaige Tan (Member, IEEE) received the B.Sc. degree in mechatronics from the Harbin Institute of Technology, Harbin, China, in 2018, and the M.Sc. degree in mechatronics in 2020 from the KTH Royal Institute of Technology, Stockholm, Sweden, where he is currently working toward the Ph.D. degree in mechatronics with the Department of Machine Design.

His research interests include optimization, reinforcement learning, and optimal filtering.



Qinglei Ji received the B.Eng. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015, and the M.Sc. degree from the Ecole Nationale Supérieure de Mécanique et d'Aérotechnique, Poitiers, France, in 2018, both in aerospace engineering, and the Ph.D. degree in production engineering from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2022.

He is currently a Solution Engineer and Technical Expert in System Resilience with Volvo Car Corporation, Gothenburg, Sweden. His research interests include autonomous driving, robotics, and artificial intelligence.



Lei Feng (Member, IEEE) received the B.S. and M.S. degrees in mechatronics from Xi'an Jiaotong University, Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the Systems Control Group, University of Toronto, Toronto, ON, Canada, in 2007.

In 2012, he joined the Mechatron. and Embedded Control System Division, KTH Royal Institute of Technology, Stockholm, Sweden, where he is currently an Associate Professor.

His main research interests include energy management control of mechatronic systems, autonomous driving, verification and control synthesis of cyber-physical systems, and supervisory control of discrete event systems.



Martin Törngren (Senior Member, IEEE) received the Ph.D. degree in mechatronics from the KTH Royal Institute of Technology, Stockholm, Sweden, in 1995.

Since 2002, he has been a Professor of Embedded Control Systems with the Division of Mechatronics and Embedded Control Systems, KTH Royal Institute of Technology. He is a pioneer in bridging the gaps between the fields of automatic control and real-time distributed systems, with research in recent years focused on architectural design and safety of autonomous trustworthy cyber-physical systems. He has spent several periods abroad as a visiting scholar, including with University of California, Berkeley, Berkeley, CA, USA, and the Stevens Institute of Technology, Hoboken, NJ, USA. Networking and multidisciplinary research have been characteristic throughout his career. He created the Innovative Center for Embedded Systems, a KTH-industry competence network launched in 2008. He is the Director of the Swedish Research Center on Trustworthy Edge Computing Systems and Applications.