IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

A Data-Driven Approach to Prediction and Optimal Bucket-Filling Control for Autonomous Excavators

Ryan J. Sandzimier and H. Harry Asada, *Member, IEEE*

***Abstract*—We develop a data-driven, statistical control method for autonomous excavators. Interactions between soil and an excavator bucket are highly complex and nonlinear, making traditional physical modeling difficult to use for real-time control. Here, we propose a data-driven method, exploiting data obtained from laboratory tests. We use the data to construct a nonlinear, non-parametric statistical model for predicting the behavior of soil scooped by an excavator bucket. The prediction model is built for controlling the amount of soil collected with a bucket. An excavator collects soil by dragging the bucket along the soil surface and scooping the soil by rotating the bucket. It is important to switch from the drag phase to the scoop phase with the correct timing to ensure an appropriate amount of soil has accumulated in front of the bucket. We model the process as a heteroscedastic Gaussian process (GP) based on the observation that the variance of the collected soil mass depends on the scooping trajectory, i.e. the input, as well as the shape of the soil surface immediately prior to scooping. We develop an optimal control algorithm for switching from the drag phase to the scoop phase at an appropriate time and for generating a scoop trajectory to capture a desired amount of soil with high confidence. We implement the method on a robotic excavator and collect experimental data. Experiments show promising results in terms of being able to achieve a desired bucket fill factor with low variance.**

***Index Terms*—Data-driven control, Gaussian process, robotics in construction, mining robotics, field robots, model learning for control, optimization and optimal control, probability and statistical methods**

Fig. 1. Typical hydraulic excavator (backhoe) used in construction and mining industries. In many excavation tasks, the soil medium is fine and homogeneous.

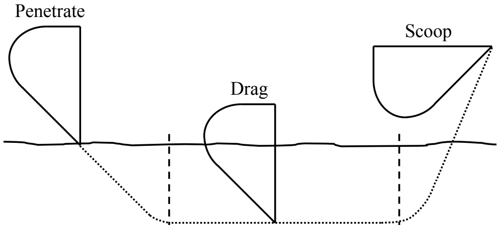

Fig. 2. The three phases of an excavation cycle (penetrate, drag, scoop). First, the bucket penetrates the soil to reach a desirable depth. Next, the bucket moves forward in a dragging motion to accumulate soil inside and in front of the bucket. When a desirable amount of soil has accumulated, the excavator scoops the soil by rotating and translating the bucket to collect the desired amount of soil.

I. Introduction

THERE is a growing need in the construction and mining industries for autonomous excavators that can operate with increased productivity and fuel efficiency [1]. The worldwide shortage of skilled workers to operate these excavators is a major driver behind the development of intelligent excavators for performing various earthmoving tasks.

A single excavation cycle consists of three sequential phases: penetrate, drag, and scoop. Structuring the problem in this way is common practice and breaks the excavation cycle into manageable sub-tasks. Fig. 2 illustrates these phases. While loading a dump truck with soil, an experienced operator can fully load the truck in just a few excavation cycles without overloading the truck. This requires the operator to collect a desired amount of soil in the bucket in a limited amount of time. Unskilled operators are unable to precisely fill the bucket, resulting in additional loading cycles and underloading or overloading of the truck. Thus, the amount of soil collected in the bucket, referred to as the bucket fill factor, is a fundamental metric for quantifying productivity and operator skill level [2]. Cycle time is another important factor for maximizing productivity. To reduce cycle time and increase fuel efficiency, experienced operators often only fill the bucket to 80% of capacity [1]. Scooping a desired amount of soil is thus a critical requirement for autonomous excavators.

Early work by Bernold [3] proposed force feedback and impedance control as an effective method for controlling the path of the bucket as it drags through soil. This work focuses mainly on control of the bucket during the drag phase. It does not consider directly controlling the bucket fill factor, where the scoop phase plays a crucial role. In order to control the bucket fill factor, we must have a model of the bucket-soil interactions. Bucket-soil interactions are very complex and difficult to model. There has been some work using the Fundamental Equation of Earthmoving (FEE) [4] to predict resistive forces during excavation. The FEE relies on parameters that are often unknown a priori. Singh [5] used

This paper was recommended for publication by Editor Youngjin Choi upon evaluation of the Associate Editor and Reviewers' comments.

Ryan J. Sandzimier and H. Harry Asada are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, USA. rsandz@mit.edu, asada@mit.edu

the FEE to motivate a choice of basis functions to model the resistive forces. They used global regression on a set of training data to determine the coefficients for each of these basis functions. Luengo et. al [6] used a similar data-driven approach, but used gradient descent optimization to directly solve for the nonlinearly involved parameters in the FEE. These data-driven approaches focus on predicting resistive forces. In order to predict the bucket fill factor, we need to be able to predict the actual soil motion during excavation.

Attempts at modeling bucket-soil interactions using discrete-element methods (DEM) are met with computational challenges that make these models infeasible for real-time applications. BLOKS3D [7] is an efficient library for simulating granular particle flow for three-dimensional polyhedral particles of any size. Nezami et. al [8] used this library to simulate bucket-soil interactions during the excavation of coarse gravel. While the results of the simulations were consistent with experimental measurements, the required computation time makes these simulations infeasible for real-time applications with current computer hardware. The simulation required a minimum of 2 hours per second of real time. When using more fine soils, these simulations become even more demanding.

Homma et. al [9] proposed a simple heuristic soil model to apply to the excavation of sand. They discretize the soil volume into a three-dimensional occupancy grid and impose a set of production rules to predict changes to the soil volume's shape as cells become occupied or vacant. Singh and Simmons [10] used this model for planning actions to take during an excavation cycle. They assume the bucket collects the volume of soil swept by the bucket during an excavation cycle. They then use the production rules to simulate the resettling of the soil after the soil displacement. We use this heuristic model and approach as the baseline when comparing the performance of our own method.

Singh and Simmons develop a task planning approach to excavation. Placing constraints on the allowable actions of the excavator and forces at the bucket, they choose actions that satisfy a bound on a cost criterion. We propose a similar approach, restricting bucket trajectories to a low-dimensional parameterization that makes the problem tractable. For our approach, the proposed cost criterion to be minimized is related to the bucket fill factor.

Terramechanics modeling based on fundamental physical principles is, in general, difficult to use for real-time control due to computational complexity. Model validity is also limited due to unmodeled dynamics. Furthermore, the use of complex feedback control with sophisticated sensors and instrumentation technology is not practical, considering the harsh environment where excavators have to work. In exploring an alternative approach and a new methodology, we exploit the data. Excavation consists mostly of repetitive operations. Although the operations are performed under diverse conditions, we can obtain a large amount of data from both laboratory tests and field operations to handle these conditions. This allows us to use the data for intelligent control of excavators. In a statistical modeling framework, we can deal with highly nonlinear, distributed behaviors of soil without going through terramechanics-based parametric representations. We obtain a non-parametric, nonlinear model directly from the data. It is possible to derive novel control methods from the statistical model.

We aim to apply this data-driven statistical model and control method to bucket fill factor control. The new method takes into account both the expected bucket fill factor as well as the associated variance so we can control the excavator to collect a desired amount of soil with high confidence. Through the estimation of the variance, the data-driven approach can produce a reliable solution to autonomous operation in unstructured environments under diverse conditions.

II. Problem Statement

The phases of an excavation cycle (penetrate, drag, and scoop) occur sequentially and are dependent on each other. The final conditions after the penetrate and drag phases serve as the initial conditions for the scoop phase. While the scoop phase occurs last chronologically, we choose to focus on it first to better motivate the discussion about the preceding phases later. An excavator can scoop a certain amount of soil by controlling the bucket motion during the scoop phase. However, the amount of scooped soil highly depends on the accumulated soil within the bucket and in front of the bucket just before the scooping action begins. Therefore, the shape of the accumulated soil both inside and in front of the bucket is a critical factor in predicting the bucket fill factor. In the current work, we describe the soil profile with a set of soil state variables, $\boldsymbol{s} \in \mathbb{R}^{n_s}$. As detailed later, we measure the accumulated soil with a depth camera and compress the image data to a representation using a small number of variables.

The input command to the bucket is described with a vector $\boldsymbol{u} \in \mathbb{R}^{n_u}$ consisting of the rotational as well as translational movements of the bucket. In the current work, a single input command defines the entire trajectory of the bucket throughout the scoop phase. The soil profile at the beginning of the scoop phase, $\boldsymbol{s}$, and the bucket input command, $\boldsymbol{u}$, are the two sets of variables that determine the bucket fill factor at the end of the scooping phase. Collectively, these variables are combined as $\boldsymbol{x} = [\boldsymbol{s}, \boldsymbol{u}] \in \mathbb{R}^{n}$, where $n = n_s + n_u$. The resultant bucket fill factor at the end of scooping is denoted by output y.

The bucket fill factor y is stochastic. We treat the scoop phase as a stochastic process where the inputs are the combined soil state and bucket command variables, $\boldsymbol{x}$, and the output is random variable y. The bucket fill factor y has a probability density conditioned on the soil profile and bucket command, $\boldsymbol{x}$.

Fig. 3 illustrates three examples of typical soil states. In the top case, not enough soil has accumulated. The bucket fill factor y is likely to be too low if we initiate the scoop phase immediately. The bottom case is likely to overfill the bucket. In each case, the soil profile immediately prior to scooping serves as the initial conditions for the scoop phase. As the excavator drags the bucket through the soil, the soil

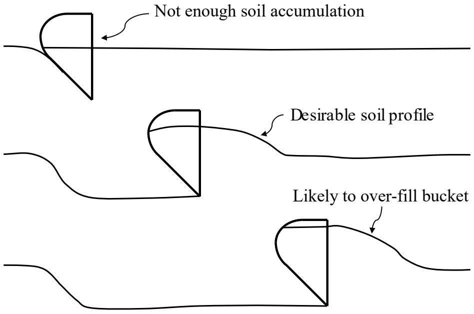

Fig. 3. If the scoop is initiated too early (top), we are likely to underfill the bucket. If the scoop is initiated too late (bottom), we will almost surely (low variance) fill the bucket near its capacity for most choices of scoop actions. However, if we only want to fill the bucket partially, say to 80% of its capacity, we are likely to overfill the bucket. For some appropriate transition point (middle), the soil that has accumulated inside and in front of the bucket is desirable and we can likely achieve the desired bucket fill factor.

profile continually changes. The excavator controller makes a decision when to initiate the scoop phase by observing the soil profile throughout the drag phase.

To make this control decision, we must construct a prediction model that can predict the bucket fill factor y_* in response to the current soil profile $\boldsymbol{s}_*$ and a bucket control command $\boldsymbol{u}_*$. We follow the notation convention from the Gaussian process regression (GPR) literature by using a $*$ subscript to denote the test inputs and output that we want to make predictions about. If the predicted bucket fill factor is significantly lower than its desired value, the drag phase should continue in hopes of achieving a more desirable $\boldsymbol{s}_*$ in the future. To this end, we must be able to make predictions about how the soil state will change if the drag action continues. We represent the soil state during the drag phase with a vector $\boldsymbol{s}'$ using the same soil state parameterization as before. We represent the drag action as a vector $\boldsymbol{v}$ where the elements are the parameter values of the drag action. To condense our notation, we refer to $\boldsymbol{\xi} = [\boldsymbol{s}', \boldsymbol{v}]$ as the input vector for the drag phase. Using $\boldsymbol{\xi}_* = [\boldsymbol{s}'_*, \boldsymbol{v}_*]$ to represent a new test input, we want to make predictions about the corresponding test output $\boldsymbol{s}_*$. Using such a prediction model, we can make decisions about whether it is more desirable to initiate the scoop phase using the current soil state or to continue with another drag action and wait to initiate the scoop phase in the future.

III. Prediction Model

We use a data-driven approach to learn a prediction model. That is, we collect a training data set $\mathcal{D}_s = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ consisting of pairs of inputs and outputs for N scoop trials. We model y as the sum of some unknown latent function $f(\boldsymbol{x})$ and additive independent noise.

$$y(\boldsymbol{x}_i) = f(\boldsymbol{x}_i) + \epsilon_i \tag{1}$$

Note that the additive noise ϵ_i depends on input $\boldsymbol{x}_i$ because of the heteroscedastic nature of scooping soil, as discussed previously. We will examine and verify this assumption of heteroscedasticity later in section V-C.

Given the current test input $\boldsymbol{x}_*$ and the training data set $\mathcal{D}_s$, we must predict the probability distribution $P(y_* \,|\, \boldsymbol{x}_*, \mathcal{D}_s)$ for the corresponding bucket fill factor y_*. Gaussian processes (GPs) are an effective framework for making this type of prediction. Unlike traditional GPs [11], which assume homoscedasticity, our process is heteroscedastic. Goldberg et. al [12] dealt with heteroscedastic GPR (HGPR) using Markov chain Monte Carlo (MCMC) approximate inference. While effective, MCMC is very slow compared to traditional GPR. Lázaro-Gredilla and Titsias [13] developed an effective method called variational HGPR (VHGPR) using a variational approximation which maximizes an analytically tractable lower bound on the maximum likelihood. Lázaro-Gredilla and Titsias have shown that the performance of VHGPR is similar to that of HGPR using MCMC and is comparable in speed to homoscedastic GPR. We find VHGPR used for other robotics applications. Planar pushing, for example, is another problem which exhibits heteroscedastic properties. Bauza and Rodriguez [14] use VHGPR to model and predict the motion of an object being pushed on a planar surface. For a given object state and choice of pushing action, they predict the most likely motion of the object as well as the variability in this motion.

Under the VHGPR framework, we place a GP prior on the latent function $f(\boldsymbol{x})$ and Gaussian priors on the noise terms ϵ_i.

$$f(\boldsymbol{x}) \sim \mathcal{GP}(0, k_f(\boldsymbol{x}, \boldsymbol{x}')) \tag{2}$$

$$\epsilon_i \sim \mathcal{N}(0, e^{g(\boldsymbol{x}_i)}) \tag{3}$$

where $g(\boldsymbol{x}_i)$ is the log-variance at every input $\boldsymbol{x}_i$. We also place a GP prior on $g(\boldsymbol{x})$.

$$g(\boldsymbol{x}) \sim \mathcal{GP}(\mu_0, k_g(\boldsymbol{x}, \boldsymbol{x}')) \tag{4}$$

where $k_f(\boldsymbol{x}, \boldsymbol{x}')$ and $k_g(\boldsymbol{x}, \boldsymbol{x}')$ are covariance functions and μ_0 is the noise mean hyperparameter. We will explain how we choose μ_0 as well as other hyperparameters later in this section. The covariance functions describe the spatial covariance between any two input vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$. In practice, we use the automatic relevance determination squared exponential (ARD-SE) kernel for both $k_f(\boldsymbol{x}, \boldsymbol{x}')$ and $k_g(\boldsymbol{x}, \boldsymbol{x}')$.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{n} \frac{([\boldsymbol{x}]_i - [\boldsymbol{x}']_i)^2}{l_i^2}\right) \tag{5}$$

where $[\boldsymbol{x}]_i$ and $[\boldsymbol{x}']_i$ are the i^{th} elements of $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively. σ_0^2 and each l_i^2 are hyperparameters for the ARD-SE covariance function. We represent these hyperparameters as a vector $\boldsymbol{\Theta} = [\sigma_0^2, l_1^2, ..., l_n^2]$. Covariance functions $k_f(\boldsymbol{x}, \boldsymbol{x}')$ and $k_g(\boldsymbol{x}, \boldsymbol{x}')$ have hyperparameters $\boldsymbol{\Theta}_f$ and $\boldsymbol{\Theta}_g$, respectively. The predictive distribution for y_* according to the VHGPR [13] framework is:

$$P(y_* \,|\, \boldsymbol{x}_*, \mathcal{D}_s) = \int \mathcal{N}(y_* | a_*, c_*^2 + e^{g_*}) \times \mathcal{N}(g_* | \mu_*, \sigma_*^2) dg_* \tag{6}$$

$$a_* = \boldsymbol{k}_{f*}^T (\boldsymbol{K}_f + \boldsymbol{R})^{-1} \boldsymbol{y} \tag{7}$$

$$c_*^2 = k_{f**} - \boldsymbol{k}_{f*}^T (\boldsymbol{K}_f + \boldsymbol{R})^{-1} \boldsymbol{k}_{f*} \tag{8}$$

$$\mu_* = \boldsymbol{k}_{g*}^T(\boldsymbol{\Lambda} - \frac{1}{2}\boldsymbol{I})\boldsymbol{1} + \mu_0 \quad (9)$$

$$\sigma_*^2 = k_{g**} - \boldsymbol{k}_{g*}^T(\boldsymbol{K}_g + \boldsymbol{\Lambda}^{-1})^{-1}\boldsymbol{k}_{g*} \quad (10)$$

$$\boldsymbol{\mu} = \boldsymbol{K}_g(\boldsymbol{\Lambda} - \frac{1}{2}\boldsymbol{I})\boldsymbol{1} + \mu_0\boldsymbol{1} \quad (11)$$

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{K}_g^{-1} + \boldsymbol{\Lambda} \quad (12)$$

where $\boldsymbol{y}$ is a column vector with elements $[\boldsymbol{y}]_i = y_i$. $\boldsymbol{K}_f$ is a matrix with elements $[\boldsymbol{K}_f]_{ij} = k_f(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{k}_{f*}$ is a column vector with elements $[\boldsymbol{k}_{f*}]_i = k_f(\boldsymbol{x}_i, \boldsymbol{x}_*)$, and $k_{f**} = k_f(\boldsymbol{x}_*, \boldsymbol{x}_*)$ for covariance function $k_f(\boldsymbol{x}, \boldsymbol{x}')$. We construct $\boldsymbol{K}_g$, $\boldsymbol{k}_{g*}$, k_{g**} in the same way using covariance function $k_g(\boldsymbol{x}, \boldsymbol{x}')$. $\boldsymbol{R}$ is a diagonal matrix with elements $[\boldsymbol{R}]_{ii} = e^{g(\boldsymbol{x}_i)}$ where $g(\boldsymbol{x}_i) = [\boldsymbol{\mu}]_i - [\boldsymbol{\Sigma}]_{ii}/2$. a_* and c_*^2 are the mean and variance of the predicted latent function $f(\boldsymbol{x})$ at the test input $\boldsymbol{x}_*$. μ_* and σ_*^2 are the mean and variance of the predicted log-variance function $g(\boldsymbol{x})$ at the test input $\boldsymbol{x}_*$. $\boldsymbol{\Lambda}$ is a positive semidefinite diagonal matrix whose elements represent the free parameters in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. We optimize the hyperparameters $\boldsymbol{\Lambda}$, μ_0, $\boldsymbol{\Theta}_f$, and $\boldsymbol{\Theta}_g$ simultaneously to maximize the variational bound.

The predictive distribution in (6) is not analytically tractable. However, its mean and variance are computable analytically.

$$\mathbb{E}[y_*|\boldsymbol{x}_*, \mathcal{D}_s] = a_* \quad (13)$$

$$\mathbb{V}[y_*|\boldsymbol{x}_*, \mathcal{D}_s] = c_*^2 + e^{\mu_* + \sigma_*^2/2} \quad (14)$$

We can use this same framework for making predictions about the soil state $\boldsymbol{s}$ throughout the drag phase. Specifically, given a drag test input $\boldsymbol{\xi}_*$, we want to make predictions about each component $[\boldsymbol{s}_*]_j$ of the corresponding output $\boldsymbol{s}_*$. In general, the soil state $\boldsymbol{s}$ is vector-valued. GPR is traditionally formulated only for scalar outputs except for some work [15] that formulates homoscedastic GPs with multiple dependent outputs. To our knowledge, there is no prior work successfully handling heteroscedastic GPs with multiple dependent outputs. In the current work, we make a simplifying assumption that the components of vector $\boldsymbol{s}$ are statistically independent and use multiple VHGPR models in parallel to make predictions about each output independently. By making this assumption, we do lose information about potential correlation among outputs. However, we show in Section V that even with this loss of information we can still make appropriate predictions.

We can make use of (1)-(14) by substituting $\boldsymbol{\xi}_*$ for $\boldsymbol{x}_*$, $\mathcal{D}_d = \{(\boldsymbol{\xi}_i, \boldsymbol{s}_i)\}_{i=1}^N$ for $\mathcal{D}_s$, and each $[\boldsymbol{s}_*]_j$ for y_* and independently optimizing the hyperparameters for each VHGPR model.

IV. Optimal Control

In this section, we consider how to optimize our choice of actions given the current soil state and a desired bucket fill factor. To do this, we make use of the prediction models discussed in Section III. First, we focus on the scoop phase. Given the current soil state $\boldsymbol{s}_*$ that we want to test, we choose a trajectory $\boldsymbol{u}_{opt}$ that minimizes a cost function $C_s(\boldsymbol{s}_*, \boldsymbol{u}_*)$.

$$\boldsymbol{u}_{opt} = \arg\min_{\boldsymbol{u}_*} C_s(\boldsymbol{s}_*, \boldsymbol{u}_*) \quad (15)$$

The goal of the scoop phase is to achieve a specified bucket fill factor. A natural choice of cost function is the expected squared error between the bucket fill factor y_* and some desired bucket fill factor y_d.

$$C_s(\boldsymbol{s}_*, \boldsymbol{u}_*) = \mathbb{E}[(y_* - y_d)^2 \,|\, \boldsymbol{x}_*, \mathcal{D}_s] \quad (16)$$

Expanding (16) and using the relationship between the second moment, variance, and mean, $\mathbb{E}[y_*^2|\boldsymbol{x}_*, \mathcal{D}_s] = \mathbb{V}[y_*|\boldsymbol{x}_*, \mathcal{D}_s] + \mathbb{E}[y_*|\boldsymbol{x}_*, \mathcal{D}_s]^2$ yields:

$$C_s(\boldsymbol{s}_*, \boldsymbol{u}_*) = \mathbb{V}[y_*|\boldsymbol{x}_*, \mathcal{D}] + (\mathbb{E}[y_*|\boldsymbol{x}_*, \mathcal{D}] - y_d)^2 \quad (17)$$

From (13) and (14), we have closed-form expressions for the mean and variance that we can plug into (17).

$$C_s(\boldsymbol{s}_*, \boldsymbol{u}_*) = c_*^2 + e^{\mu_* + \sigma_*^2/2} + (a_* - y_d)^2 \quad (18)$$

To minimize the cost function, we take the derivative with respect to each of the components of the action and use gradient descent methods to search for a minimum. Taking the derivative with respect to $[\boldsymbol{x}_*]_j$, the j^{th} component of $\boldsymbol{x}_*$ corresponding to an action input:

$$\frac{\partial C_s(\boldsymbol{s}_*, \boldsymbol{u}_*)}{\partial [\boldsymbol{x}_*]_j} = \frac{\partial c_*^2}{\partial [\boldsymbol{x}_*]_j} + 2\frac{\partial a_*}{\partial [\boldsymbol{x}_*]_j}(a_* - y_d) + \left(\frac{\partial \mu_*}{\partial [\boldsymbol{x}_*]_j} + \frac{1}{2}\frac{\partial \sigma_*^2}{\partial [\boldsymbol{x}_*]_j}\right) e^{\mu_* + \sigma_*^2/2} \quad (19)$$

$$\frac{\partial a_*}{\partial [\boldsymbol{x}_*]_j} = \frac{\partial \boldsymbol{k}_{f*}^T}{\partial [\boldsymbol{x}_*]_j}(\boldsymbol{K}_f + \boldsymbol{R})^{-1}\boldsymbol{y} \quad (20)$$

$$\frac{\partial c_*^2}{\partial [\boldsymbol{x}_*]_j} = \frac{\partial \boldsymbol{k}_{f**}}{\partial [\boldsymbol{x}_*]_j} - 2\boldsymbol{k}_{f*}^T(\boldsymbol{K}_f + \boldsymbol{R})^{-1}\frac{\partial \boldsymbol{k}_{f*}}{\partial [\boldsymbol{x}_*]_j} \quad (21)$$

$$\frac{\partial \mu_*}{\partial [\boldsymbol{x}_*]_j} = \frac{\partial \boldsymbol{k}_{g*}^T}{\partial [\boldsymbol{x}_*]_j}(\boldsymbol{\Lambda} - \frac{1}{2}\boldsymbol{I})\boldsymbol{1} \quad (22)$$

$$\frac{\partial \sigma_*^2}{\partial [\boldsymbol{x}_*]_j} = \frac{\partial \boldsymbol{k}_{g**}}{\partial [\boldsymbol{x}_*]_j} - 2\boldsymbol{k}_{g*}^T(\boldsymbol{K}_g + \boldsymbol{\Lambda}^{-1})\frac{\partial \boldsymbol{k}_{g*}}{\partial [\boldsymbol{x}_*]_j} \quad (23)$$

where the derivatives of $\boldsymbol{k}_{f*}$, $\boldsymbol{k}_{g*}$, k_{f**}, k_{g**} with respect to $[\boldsymbol{x}_*]_j$ depend on the derivatives of the covariance functions. For the ARD-SE covariance functions, we have:

$$\left[\frac{\partial \boldsymbol{k}_*}{\partial [\boldsymbol{x}_*]_j}\right]_i = \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_*)}{\partial [\boldsymbol{x}_*]_j} = \frac{[\boldsymbol{x}_i]_j - [\boldsymbol{x}_*]_j}{l_j^2} k(\boldsymbol{x}_i, \boldsymbol{x}_*) \quad (24)$$

$$\frac{\partial \boldsymbol{k}_{**}}{\partial [\boldsymbol{x}_*]_j} = \frac{\partial k(\boldsymbol{x}_*, \boldsymbol{x}_*)}{\partial [\boldsymbol{x}_*]_j} = 0 \quad (25)$$

Using gradient descent methods, we search for the optimal scoop action $\boldsymbol{u}_{opt}$ and the corresponding cost. To initialize the optimization, we choose a representative sample from the training data set according to a distance metric. Given a current test soil state $\boldsymbol{s}_*$ and a desired bucket fill factor y_d, we choose an optimal scoop action. However, if the soil state at the start of the scoop phase is undesirable, the cost associated with the optimal scoop may still be high. Therefore, we next consider how to decide if it is better to initiate the scoop phase with the current soil state or to continue the drag action in hopes of making the soil state more desirable prior to scooping.

Given a drag input $\boldsymbol{\xi}_*$, we can predict the distribution of the corresponding soil state output $\boldsymbol{s}_*$. Rewriting (6) in terms of the drag phase inputs and outputs:

$$P([\boldsymbol{s}_*]_j \mid \boldsymbol{\xi}_*, \mathcal{D}_d) = \int \mathcal{N}([\boldsymbol{s}_*]_j | a_*, c_*^2 + e^{g_*}) \times \mathcal{N}(g_* | \mu_*, \sigma_*^2) dg_* \tag{26}$$

which is analytically intractable. However, we can approximate the distribution up to several digits using Gauss-Hermite quadrature [13], which is computationally inexpensive. The accuracy of this approximation is sufficient for our application. In Section V-F, we discuss the impact of the approximation in further detail.

Using the assumption that the components of the predicted soil state $\boldsymbol{s}_*$ are statistically independent, we can calculate the predicted distribution of the full soil state:

$$P(\boldsymbol{s}_* | \boldsymbol{\xi}_*, \mathcal{D}_d) = \prod_{j=1}^{n_s} P([\boldsymbol{s}_*]_j | \boldsymbol{\xi}_*, \mathcal{D}_d) \tag{27}$$

where $\boldsymbol{s}_* \in \mathbb{R}^{n_s}$

In order to make a decision about whether to initiate the scoop phase or continue with another drag action, we must compare the cost to initiate the scoop phase now to the expected cost to initiate the scoop phase in the future. The cost to continue dragging for some drag input $\boldsymbol{\xi}_*$ is:

$$C_d(\boldsymbol{\xi}_*) = \mathbb{E}[C_s(\boldsymbol{s}_*, u_{opt}) | \boldsymbol{\xi}_*, \mathcal{D}_d] \tag{28}$$

$$= \int C_s(\boldsymbol{s}_*, \boldsymbol{u}_{opt}) P(\boldsymbol{s}_* | \boldsymbol{\xi}_*, \mathcal{D}_d) d\boldsymbol{s}_* \tag{29}$$

This integral is not analytically tractable. However, we can approximate this expected cost using importance sampling. While it is difficult to sample directly from $P(\boldsymbol{s}_* | \boldsymbol{\xi}_*, \mathcal{D}_d)$, we can approximate the probability density at a given sample $\boldsymbol{s}_*$. By sampling from some distribution $q(\boldsymbol{s}_*)$ that is easy to sample from, we can approximate the expectation as follows:

$$C_d(\boldsymbol{\xi}_*) \approx \sum_{i=1}^{k} \frac{w(\boldsymbol{s}_{*,i}) C_s(\boldsymbol{s}_{*,i}, \boldsymbol{u}_{opt})}{\sum_{i'=1}^{k} w(\boldsymbol{s}_{*,i'})} \tag{30}$$

where

$$w(\boldsymbol{s}_{*,i}) = \frac{P(\boldsymbol{s}_{*,i} | \boldsymbol{\xi}_*, \mathcal{D}_d)}{q(\boldsymbol{s}_{*,i})} \tag{31}$$

By taking this k-sample weighted average of the optimal scoop cost, we approximate the cost to continue dragging. The approximation is more accurate for large k and in the limit as k approaches infinity, the estimated cost approaches the expected cost. In practice, we choose k large enough that the approximation is sufficient. In Section V-F, we discuss this approximation in further detail.

We compare the cost to continue dragging for every possible drag action $\boldsymbol{v}$ and compare it to the cost to initiate the scoop phase now. If the cost to initiate scooping is smaller for every choice of $\boldsymbol{v}$, we decide to initiate the scoop phase now. Otherwise, we continue dragging by performing one step of a drag action. We then repeat these steps until the scoop phase is initiated.

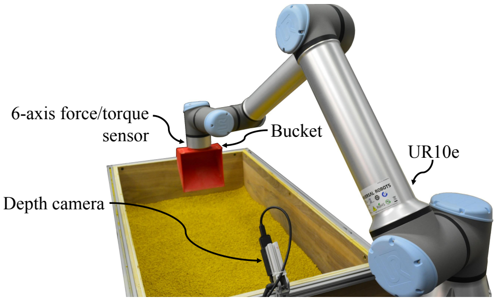

Fig. 4. Experimental setup

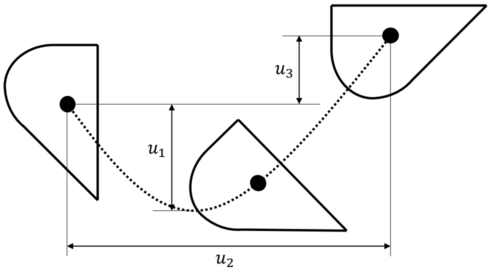

Fig. 5. Scoop action parameterization. During the scoop phase, the bucket rotates about the bucket center (shown as black circle) at a constant velocity. While the bucket rotates, the bucket center moves along a parabolic trajectory. We parameterize the parabolic trajectory $\boldsymbol{u} = [u_1, u_2, u_3]$.

V. Performance and Experimental Results

In this section, we discuss our experimental implementation of the proposed method and present the experimental results. In addition, we address the assumption of heteroscedasticity in more detail.

A. Experimental Setup

The experimental setup is shown in Fig. 4. We use a Universal Robots UR10e to control the excavator bucket. The robot is equipped with a 6-axis force/torque sensor. We use this sensor to measure the mass of soil collected in the bucket after scooping (bucket fill factor). Throughout the dragging phase and immediately prior to scooping, a stereo camera captures a depth image which we use to represent the soil state. The soil medium is homogeneous, low-density, and has particles between 1-3mm in size.

B. Parameterization of Soil State and Actions

To represent the soil state, we capture a depth image using a stereo camera. After isolating pixels corresponding to the soil inside and in front of the bucket, we project the depth image onto an orthogonal plane (top). This new depth map contains pixels representing the depth of the soil relative to the bucket center. Although the image is high-dimensional due to the large number of pixels, we use principal component analysis and find that the first two principal components explain 96% of the variance in the depth images. We then represent the soil state $\boldsymbol{s} \in \mathbb{R}^2$ as a vector with the projection of the depth image onto the first two principal component directions.

To represent the scoop actions, we parameterize the allowable scoop trajectories the bucket follows during scooping. This parameterization is illustrated in Fig. 5. During the scoop phase, the bucket rotates about the bucket center to collect the soil. While rotating, we allow the bucket center to follow a parabolic trajectory, which takes three parameters to represent. We find that this parameterization is simple enough to keep the input space relatively small, while also rich enough to allow a sufficient range of scoop actions for controlling the bucket fill factor. We control the bucket along this reference trajectory using impedance control to account for high reaction forces at the bucket that may not allow the robot to follow the reference trajectory exactly.

While operators sometimes vary the depth of the bucket throughout the drag, these variations are typically small relative to the forward motion of the bucket. Therefore, we prescribe a constant depth during the drag phase to reduce the dimensionality of the drag action parameterization. The only parameter necessary to represent a drag action is the drag length, or the distance to move the bucket in the forward direction. The maximum allowable drag length is limited by the workspace of the excavator. When optimizing the transition point between the drag phase and scoop phase, we search over the remaining possible drag lengths.

C. Demonstration of Heteroscedasticity

We observe that the variance in the bucket fill factor is dependent on the initial soil state and the chosen scoop action. Fig. 6 illustrates this heteroscedastic property. By measuring the bucket fill factor for repeated trials using the same initial soil state and scoop action, we can test for homoscedasticity. While setting up the exact same initial soil state for repeated trials is not practical, we can achieve approximately the same initial soil state by repeatedly performing a drag trajectory. This is an approximation because repeating a drag trajectory does not guarantee the same soil state. However, we find the approximation is appropriate.

We measure the bucket fill factor for a combination of two separate drag actions and two separate scoop actions. For one of the drag actions, the bucket penetrates deep into the soil and drags for a long length. We refer to this as the long drag. For the other drag action, which we refer to as the short drag, the bucket penetrates to a shallow depth and drags for a short length. During a long drag, a lot of soil accumulates inside and in front of the bucket. On the other hand, for a short drag, the bucket barely moves any soil. We refer to the two choices of scoop actions as the deep scoop and the shallow scoop. For the deep scoop, the bucket moves deep into the soil while also moving forward to collect soil. For the shallow scoop, the bucket only rotates. There is no forward or downward movement during the shallow scoop.

We measure the bucket fill factor for four groups: long drag and deep scoop (LD-DS), long drag and shallow scoop (LD-SS), short drag and deep scoop (SD-DS), and short drag and shallow scoop (SD-SS). The null hypothesis is that the bucket fill factor samples came from populations which each have the same variance. We use an F-test to test this hypothesis for each pair of groups. Table I shows the p-values for each of these tests. We determine with statistical significance that the bucket fill factor is heteroscedastic.

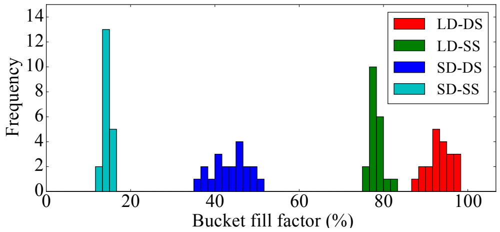

Fig. 6. Measured bucket fill factor for four groups: long drag and deep scoop (LD-DS), long drag and shallow scoop (LD-SS), short drag and deep scoop (SD-DS), and short drag and shallow scoop (SD-SS).

TABLE I
P-VALUES OF F-TESTS FOR COMPARING SAMPLE VARIANCES

Samples	***p*-value**
LD-DS vs. LD-SS	0.0036
LD-DS vs. SD-DS	0.1363
LD-DS vs. SD-SS	1.837×10^{6}
LD-SS vs. SD-DS	2.795×10^{5}
LD-SS vs. SD-SS	0.0280
SD-DS vs. SD-SS	4.816×10^{9}

D. Training the Model

We collect training data sets $\mathcal{D}_s$ and $\mathcal{D}_d$ by performing random drag and scoop actions, measuring the soil state throughout the drag phase and immediately prior to the scoop phase, and measuring the resulting bucket fill factor. The use of random drag and scoop actions provides a rich set of soil states, ensuring the training data set spans the entire input space.

The prediction models depend on the training data sets and the hyperparameters $\mathbf{\Theta}_f$ and $\mathbf{\Theta}_g$. In general, we do not know the best choice of hyperparameters, so we must infer them from the training data. We find the hyperparameters that maximize the variational bound on the log-likelihood [13] using the L-BFGS-B optimization algorithm [16].

We set aside a testing data set to cross-validate the trained prediction models. We use the normalized mean squared error (NMSE) and normalized log-probability density (NLPD) as performance measures. The NMSE and NLPD are defined as follows:

$$NMSE = \frac{\sum_{j=1}^{M}(y_{*,j} - a_{*,j})^2}{\sum_{j=1}^{M}(y_{*,j} - \bar{y})^2} \tag{32}$$

$$NLPD = -\frac{1}{M}\sum_{j=1}^{M}\log(p(y_{*,j}|\boldsymbol{x}_{*,j}, \mathcal{D}_s)) \tag{33}$$

where $\boldsymbol{x}_{*,j}$ and $y_{*,j}$ are the j^{th} test input and output, respectively, M is the number of test inputs and outputs, $a_{*,j}$ is the mean of the predictive distribution corresponding to the j^{th} test input, and $\bar{y}$ is the mean of the training data outputs.

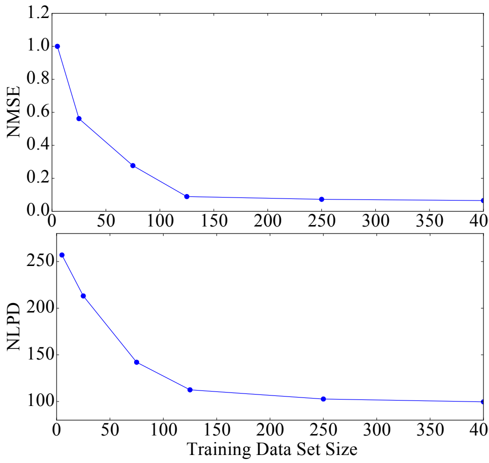

Fig. 7. Normalized mean squared error (NMSE) and normalized log-probability density (NLPD) for the bucket fill factor prediction model.

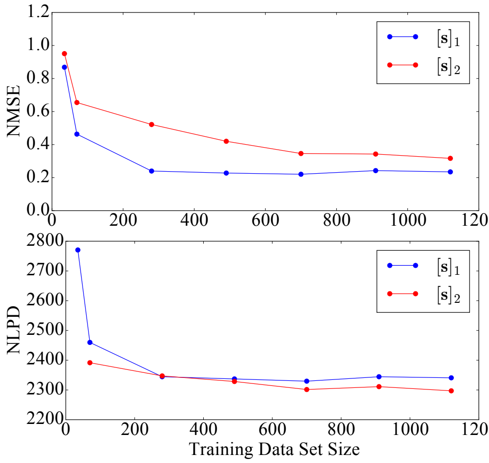

Fig. 8. Normalized mean squared error (NMSE) and normalized log-probability density (NLPD) for the soil state prediction models.

Fig. 7 and 8 show how the performance measures change with varying training data set sizes for the bucket fill factor prediction model and soil state prediction models, respectively. We find that the performance saturates around 250 training data points for the bucket fill factor prediction model and around 1000 training data points for the soil state prediction models. Beyond this amount of training data, the performance of the models does not seem to improve. Therefore, we train our bucket fill factor prediction model using 250 training data points and our soil state prediction models using 1000 training data points. We observe that the second component of the soil state requires more training data points to saturate than the first component and has a higher NMSE. Since the components of the soil state represent the principal components, it makes sense that the second component would be more difficult to train.

E. Performance

To evaluate the performance of the drag transition decision-making, we measure the cost to initiate the scoop phase throughout the drag phase. Using the prediction model, we decide the optimal transition point. We can then compare this transition point to the actual optimal transition point, which we determine retroactively. Fig. 9 shows these results for a single drag phase. We find that the predicted cost is relatively close to the actual measured cost. We illustrate a single drag phase as an example, however, we find these results to be consistent and typical for any choice of target bucket fill factor.

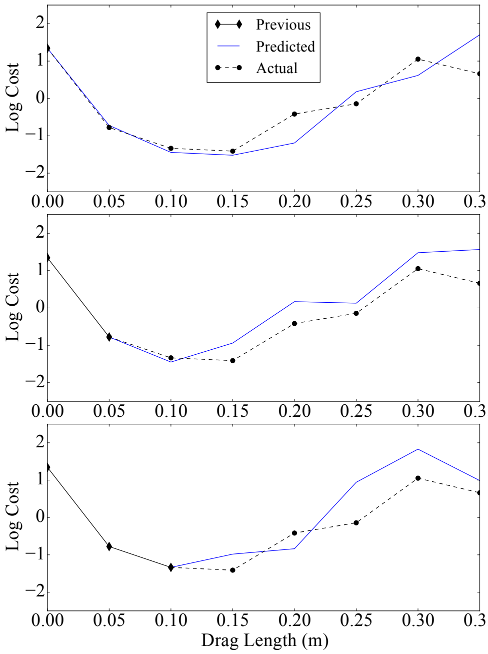

Fig. 9. Predicted cost and actual cost at the beginning of the drag phase (top), after executing one drag action (middle), and after executing a second drag action (bottom).

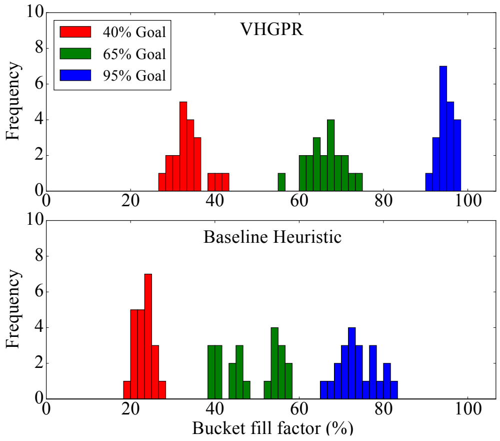

Fig. 10. Results controlling the bucket fill factor using the VHGPR (top) and baseline heuristic (bottom) prediction models to decide when to initiate the scoop phase and which scoop action to perform. There were three target bucket fill factors: 40%, 65%, and 95%.

Next, we discuss the performance while controlling the bucket fill factor. We vary the drag depth and target bucket fill factor for each trial and the controller must decide when to transition to the scoop phase and which scoop action to perform to minimize the cost. The varied drag depth provides a variety of initial scenarios which the controller must compensate for. The results of this experiment are shown in Fig. 10. We compare our method to a baseline heuristic model [10] that predicts the bucket fill factor by considering the volume of soil swept by the bucket during the excavation cycle. We use the same optimization technique to determine the optimal actions when using the heuristic prediction model. For both the VHGPR and heuristic prediction models, as we increase the desired bucket fill factor, the experimentally measured bucket fill factor increases. However, we notice that the results using VHGPR have higher accuracy and lower variance than when using the baseline heuristic model.

F. Discussion

In Section IV, we present two intractable integrals (26) and (29) and propose approximations for computing these integrals numerically. We now discuss the effect and suitability of these approximations. When calculating the cost to continue dragging, we approximate the expected cost using a k-sample weighted average of the optimal scoop cost. In the limit as k approaches infinity, this sampling-based approach converges to the actual expected cost. One design decision is choosing sufficiently large k such that the approximation is suitable. In practice, we choose k large enough that the approximation is accurate to several digits. In this way, any error introduced from this approximation is negligible compared to errors introduced by process noise, sensor noise, and any model inaccuracies. Similarly, the approximation of probability distributions using Gauss-Hermite quadrature is accurate up to several digits [13], which is more than enough accuracy for our application.

In our experimental implementation, we use a relatively simple parameterization of the action space. It is important to consider how our method scales to more complex parameterizations. As the action space grows, the required training data size tends to increase. Traditional GPs scale as $\mathcal{O}(N^3)$ with the size of the training data [11]. However, there are sparse approximations [17] of GPs, which are more scalable. Determining the optimal transition point between the drag and scoop phases is susceptible to the curse of dimensionality, as is typical for dynamic programming.

Accurate bucket filling is one of the major challenges in autonomous excavation. There is a clear and significant difference even between experienced and novice human operators' abilities to control the bucket fill factor accurately. Poor bucket-filling accuracy directly influences the overall productivity. The proposed method is the first attempt to fill the gap, and will drive autonomous excavator development to a new level.

VI. Conclusion

We develop a data-driven model for predicting the bucket fill factor during an excavation cycle. Using this prediction model, we develop an optimal control algorithm to determine the optimal transition point between the drag and scoop phases as well as the optimal scoop action to achieve a desired bucket fill factor with high certainty. The experimental results are promising and demonstrate that a data-driven model can be suitable for making predictions about soil-bucket interactions and controlling the bucket fill factor. We present a specific parameterization of the problem, however, the approach extends to other parameterizations if provided an appropriately sized training data set.

There are some key areas for future work on this problem. For the experiment, we used soil of uniform grain size. For real world applications, it is important to consider various soil grain sizes and properties. This requires a large amount of data collected for many different conditions. The proposed GP model is an effective framework. It is flexible enough to apply to the diverse conditions. It is also efficient since it converges with a relatively small training data set size. We use a simple method for parameterizing drag actions, however, considering more complex trajectories may increase the performance.

References

[1] F. Ng, J. A. Harding, and J. Glass, "An eco-approach to optimise efficiency and productivity of a hydraulic excavator," *Journal of cleaner production*, vol. 112, pp. 3966–3976, 2016.

[2] S. Dadhich, U. Bodin, and U. Andersson, "Key challenges in automation of earth-moving machines," *Automation in Construction*, vol. 68, pp. 212–222, 2016.

[3] L. E. Bernold, "Motion and path control for robotic excavation," *Journal of Aerospace Engineering*, vol. 6, no. 1, pp. 1–18, 1993.

[4] A. R. Reece, "Paper 2: The fundamental equation of earth-moving mechanics," *Proc. Inst. Mech. Eng.*, vol. 179, no. 6, pp. 16–22, 1964.

[5] S. Singh, "Learning to predict resistive forces during robotic excavation," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1995, pp. 2102–2107.

[6] O. Luengo, S. Singh, and H. Cannon, "Modeling and identification of soil-tool interaction in automated excavation," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, vol. 3. IEEE, 1998, pp. 1900–1906.

[7] D. Zhao, E. G. Nezami, Y. M. Hashash, and J. Ghaboussi, "Three-dimensional discrete element simulation for granular materials," *Engineering Computations*, vol. 23, no. 7, pp. 749–770, 2006.

[8] E. G. Nezami, Y. M. Hashash, D. Zhao, and J. Ghaboussi, "Simulation of front end loader bucket–soil interaction using discrete element method," *International journal for numerical and analytical methods in geomechanics*, vol. 31, no. 9, pp. 1147–1162, 2007.

[9] K. Homma, T. Nakamura, T. Arai, and H. Adachi, "Spatial image model for manipulation of shape variable objects and application to excavation," in *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*. IEEE, 1990, pp. 645–650.

[10] S. Singh and R. G. Simmons, "Task planning for robotic excavation." in *IROS*, vol. 92, 1992, pp. 1284–1291.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[12] P. W. Goldberg, C. K. Williams, and C. M. Bishop, "Regression with input-dependent noise: A gaussian process treatment," in *Advances in neural information processing systems*, 1998, pp. 493–499.

[13] M. Lázaro-Gredilla and M. K. Titsias, "Variational heteroscedastic gaussian process regression." in *ICML*, 2011, pp. 841–848.

[14] M. Bauza and A. Rodriguez, "A probabilistic data-driven model for planar pushing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3008–3015.

[15] P. Boyle and M. Frean, "Dependent gaussian processes," in *Advances in neural information processing systems*, 2005, pp. 217–224.

[16] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[17] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable GPs," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.