

# Feature Localization Using Kinematics and Impulsive Hybrid Optimization

Yoke Peng Leong, *Student Member, IEEE*, and Todd Murphey, *Member, IEEE*

**Abstract**—This paper focuses on detecting and localizing a surface feature on an otherwise uniform surface using kinematic data collected during an exploratory procedure. Assuming that characteristics of the feature shape and surface shape are known, a surface feature is detected by performing least squares estimation calculated via impulsive hybrid system optimization. The optimization routine is based on an adjoint formulation which allows the algorithm to be computationally efficient and scalable. This algorithm is also shown to perform well with the presence of measurement noise and model noise, both in simulations and experiments.

**Note to Practitioners**—Traditionally, tactile sensing focuses on using collocated tactile sensors to recognize objects. However, it is not always practical to place tactile sensors at the location of contact. We demonstrate a new way to detect and localize surface features using the dynamical response of a robotic finger or sensor. This approach facilitates tactile object identification in scenarios where tactile sensors might not be practical.

**Index Terms**—tactile estimation, feature detection, feature localization, hybrid optimal control.

## I. INTRODUCTION

**S**URFACE feature detection plays an important role in haptic exploration—the use of tactile data for sensing and estimation—for object identification. People are quite good at detecting and recognizing objects using tactile sensing [1]. For instance, reaching into a bag to find a key among other items is a task that most people can accomplish readily. The same task is challenging for a robot; in addition to the ability to search and take out the key, the robot needs to know what measurements (e.g., collocated tactile sensing in the form of force measurements at the fingertips, kinematic data from joints, etc.) mean in terms a computer can express and analyze. A distinct feature of a key is the row of edges on the key blade. This suggests that an object can be identified using the sense of touch by detecting the object’s geometrical features—e.g., corners, edges, and their ordered relationships to each other. Moreover, just as is often done in image processing, restricting the identification to a subset of physical characteristics detection and localization can be expected to be more robust than identification of a complete physical description of the key. Lastly, restricting to a subset makes identifying equivalence classes easier; it may be desirable to classify two objects that have the same features in the same order as the same *type* of object, regardless of whether they disagree in other aspects of their description.

Y. P. Leong is with the Control and Dynamical Systems of California Institute of Technology, Pasadena, CA 91125, USA (email: ypleong@caltech.edu). T. Murphey is with the Department of Mechanical Engineering, Northwestern University, Evanston, IL 60201, USA (e-mail: t-murphey@northwestern.edu).



Fig. 1. A surface feature is defined by geometric discontinuities. The symbols  $\varphi_i$  represent the surface contours away from discontinuities. The symbols  $\mu_i$  represent the coefficient of friction between changes in friction. The goal is to identify the existence and location of these discrete changes in the surface.

These considerations motivate detecting and localizing features based on measurement data. One of the core questions is how to define a feature in terms of the dynamics of the sensor; these dynamics can be quite complex if the sensor is a hand or some other complex object. A feature such as an edge, corner, ledge, or abrupt change in friction may not have uniquely defined parametric properties. For instance, a ledge may be defined by an abrupt change in height, but how abrupt, how high, etcetera, may not be known a priori. Indeed, identifying all signals that have a detectable change in height over a short spatial distance as ledges is probably preferable for purpose of classification of features. Motivated by this observation, in the present paper we define tactile features in terms of the impulsive and hybrid dynamics they induce in the sensor dynamics and then search through the continuous-time measurements (i.e., either analog signals or interpolated discrete-time measurements) looking for impulses and hybrid transitions that can be mapped to a given feature description. As we will see in Section III, this characterization of features has the advantage of identifying a given feature under many different noise levels.

Given that the amount of uncertainty in a measurement depends on the surface one is interacting with, and given that the surface is likely not known a priori and one would like to use the same calculation regardless of the surface, it is natural—though not necessary—to pose the estimation as a least-squares estimation so that the variance of the estimate does not need to be known ahead of time. There are, of course, other choices one could make, but we use a least-squares formulation because it yields a computationally tractable algorithm that does not need information about probability distribution functions we do not expect to have.

In this paper, detection and localization of a surface feature based on kinematics of a sensor (e.g., joint configurations of a robotic finger) is presented. A surface feature is defined by a distinct dynamical response of a sensor or finger when the surface feature is encountered. A macro geometrical change in

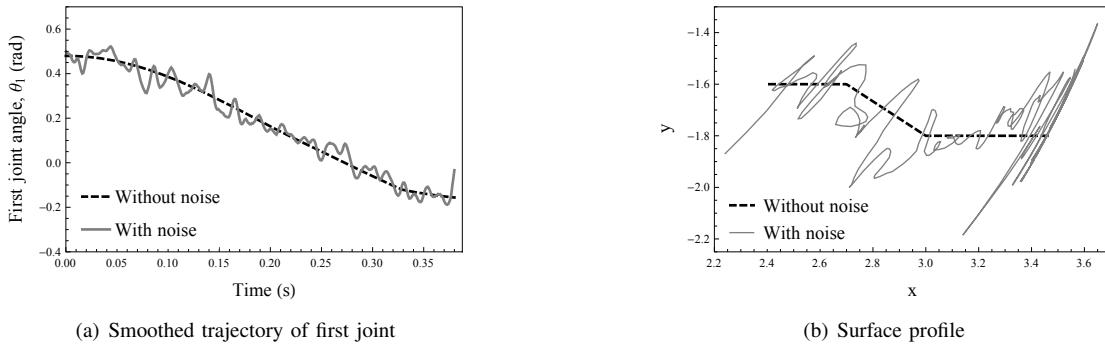


Fig. 2. Example of surface feature detection: Noisy measurement trajectories of a simple robotic finger with three revolute joints are recorded when tracing a surface with a slope. The dark dashed line is the actual surface while the dark grey line in (a) is the set of measurements and the light grey line in (b) is the (noisy) curvature. The surface profile is reconstructed using forward kinematics from the smoothed noisy joint trajectories. It is not trivial to compute a meaningful curvature plot from the surface profile in (b) to detect a surface feature.

the contour of a body or a micro change that forms different surface textures (Fig. 1) will result in a change in the dynamics of the sensor. In other words, a surface can be segmented into regions based on features (e.g., flat, curved, smooth, rough, etc.) found on the surface in which each feature corresponds to a transition between unique dynamical modes. By identifying the boundary of each dynamical mode, we can localize these regions and hence, the corresponding features.

Although the environment is usually in three dimensions, in this work, features are detected using a two-dimensional line model because when kinematic data are collected using a sensor, the path the sensor traced always forms a submanifold that is two-dimensional. Assuming that the path is generated by another algorithm (e.g., [2]), the algorithm presented in this paper analyzes the kinematic information gathered to gain insights on the path (i.e., surface profile) traced by the sensor. Specifically, the approach involves two components. First a relaxation of the hybrid optimization is performed to identify how many impulses and/or changes in dynamics are present and then a hybrid optimization is performed to localize the features. In Section III and IV this algorithm is also shown to perform well with the presence of measurement noise and model uncertainties in experiments.

Lastly, a comment on the impact of this work on sensor choices is called for. A tremendous amount of work in tactile sensing has, reasonably, focused on collocated tactile sensors—sensors that are at the point of contact, typically measuring forces as a function of time. The present work illustrates that this approach may not be necessary in cases where high-quality kinematic sensing is available and when object identification will focus on surface features. This relaxation of requirements is helpful in cases where collocated force sensors are not practical to implement or will not be physically robust. If such sensors are available, then the fusion of collocated tactile force data with kinematic data would be called for; this fusion problem is not discussed in this paper, but would be a natural next step in the work.

The rest of this paper is organized as follows: Section I-A presents recent works on tactile object identification and surface feature detection, and discusses contributions of this paper by comparing this work with other related works. Section I-B establishes the notation used in this paper. An

overview of the surface feature detection algorithm is given in Section II. This algorithm has two components, feature type estimation and feature localization, which are explained in Section II-A and Section II-B. The computational complexity of this algorithm is discussed in Section II-C. Section III shows results of implementing the algorithm in simulations when measurement data are noisy. Section IV discusses experimental results. Section V summarizes the findings of this work and discusses future research directions beyond this paper.

## A. Related Works & Contributions

Object identification based on surface feature detection is relatively mature in the context of robotic vision. However, vision-based identification possesses limitations that can be resolved by tactile object identification. As a result, exploratory procedures that benefit from both visual and touch feedbacks were developed [3], [4].

Robotic haptic exploration includes sensor design [5]–[7], control and exploration algorithms [2], [8]–[10], and data interpretation [11]–[18]. Two types of measurement data are usually available from robotic haptic exploration: geometric data (e.g., joint trajectories) [8], [9] and tactile sensor data (e.g., contact normal, forces, contact locations, etc.) [11]–[14], [18]. This work focuses on data interpretation using geometric data only.

In some works, tactile sensor data is used to detect surface features by treating the data as a type of “tactile image” so that image processing techniques could be exploited [12]–[14], [18]. In contrast to vision-based machine learning methods, Kikuuwe et al. [15] derived an impedance perception algorithm that estimates local surface properties such as normal direction, stiffness and friction coefficient of flat and convex cylindrical surfaces based on an end-effector’s positions and forces applied on the end-effector. More recently, Ibrayev et al. [11] derived a model matching method which can recognize closed-form surfaces and triangular meshes by searching a lookup table. Lastly, Okamura et al. [16] presented a curvature-based method to detect and recognize surface features using an estimate of local surface curvature. The curvature-based method is effective when the measurement trajectories are deterministic or when the noise level in the measurement

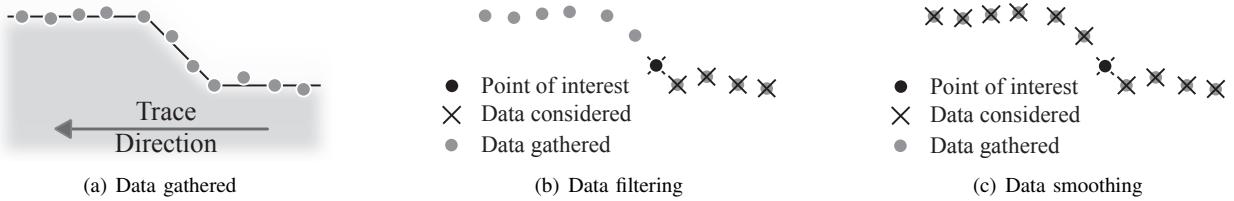


Fig. 3. Suppose that a robotic finger traces a surface and collects data as shown in (a). A data filtering-based method estimates whether a discontinuity in the surface is encountered at the current step (red dot in (b)) based on data gathered in previous steps. The difference between a feature and noise in data is unclear at this point. In (c), a clear picture of a slope is seen when data for the entire time window are considered in a smoothing-based method. Hence, a feature is considerably more salient when more data is included, so this paper focuses on detecting discontinuities over windows of data.

trajectories is relatively low. However, when the noise level is very high (Fig. 2(a)), it will not be trivial to compute the curvature from the constructed surface profile (Fig. 2(b)) to detect a surface feature.

The work in this paper is in contrast to all of the aforementioned research in that it directly uses a description of the dynamics of the sensor in the characterization of surface features. Simulation results in Section III show that the surface feature detection algorithm presented in this paper is able to localize a surface feature even when the noisy measurement trajectory shown in Fig. 2(a) is used.

For the purpose of feature detection, a data smoothing method is preferable to a data filtering method in handling noisy data. A data filtering method, such as multiple hypothesis testing [19], estimates the likelihood of a feature at a given time based on data gathered in previous steps. On the other hand, a data smoothing method estimates the likelihood that a feature was encountered based on data gathered for an entire time window including data encountered later in time. As illustrated in Fig. 3, because the algorithm in this paper considers data for the entire time window, it is less sensitive to data noise.

In addition to its noise tolerance, the algorithm presented here is computationally efficient and scalable. Numerous researchers [9], [20], [21] have applied Bayesian techniques to localize an object through touch interactions. Although Bayesian methods are known to produce a good estimate given enough computational resources, computational complexity of a Bayesian model usually grows exponentially in the amount of data considered. Consider a simple surface feature detection algorithm where two possible modes exist, “finger on feature” or “finger not on feature”. Using a nominal Bayesian model, the computational complexity is  $2^n$  where  $n$  depends on the number of data collected over a fixed time window. Suppose that the temporal data density is increased by a factor of 10 in order to gain a higher resolution of a surface feature for a more precise localization. The computational complexity of a Bayesian model grows to  $2^{10n}$ . (Methods such as the multi-scan Markov Chain Monte Carlo data association technique [22] reduce the computational load, but are still comparatively expensive to compute.) Lastly, the Bayesian approach generally assumes that uncertainty characteristics are known ahead of time, which as discussed earlier is undesirable in the context of feature identification.

The method presented here performs a least squares estimation [23] that has linear dependence on temporal data resolu-

tion. As a result, it benefits from increased temporal resolution of data without a superlinear increase in the computational cost.

### B. Notation

The trajectory,  $x(x_0, \tau_1, \tau_2, \dots, \tau_N, \delta_1, \delta_2, \dots, \delta_N, t)$ , is abbreviated as  $x(t)$ , and  $x_i(t)$  refers to a segment in  $x(t)$  when  $t \in [\tau_i, \tau_{i+1}]$ .

A derivative is written as  $Df(\cdot)$ . The derivative of a function  $f(\cdot)$  with respect to the  $n$ -th argument is written as  $D_n f(arg_1, arg_2, \dots)$ , and the derivative of a function  $f(\cdot)$  with respect to an argument named  $arg$  is written as  $D_{arg} f(\cdot)$ .

An operator  $M$  applied to  $U$  is written as  $M \circ U$ . A linear operator is written in matrix representation using square brackets such as  $M \circ V = [M]V$  and  $M \circ (V, U) = V^T [M]U$ .

## II. SURFACE FEATURE DETECTION ALGORITHM

We assume that a submanifold of a surface with features can have  $N$  segments separated by  $N - 1$  geometrical discontinuities. This submanifold will typically be the path the sensor traces during exploration. When a robotic finger is in contact with different surface segments, the dynamics of the finger vary because of the constraint imposed by the surface. Each surface segment corresponds to one dynamical mode of the finger’s dynamics. At a geometrical discontinuity, although the path is continuous, impulses are observed in the velocities (in the real physical system, these impulses will generally be slightly smoothed and form very high curvature regions where the amount of curvature will depend on material properties of the surface and sensor). These impulses act as markers for the locations of geometrical discontinuities which define a surface feature. By estimating the impulse times from the kinematic data, we find the joint configurations at these times and localize a feature using forward kinematics [24].

The algorithm presented in this paper uses an impulsive hybrid system optimization to localize the impulses and thus the features. However, this optimization requires the exact number of surface segments and their order. Hence, the feature detection algorithm has two components: (a) feature type estimation—is there a ledge, and (b) feature localization—where is the ledge. Using kinematic data gathered, the first component of the detection algorithm estimates the number of surface segments, their types and their sequence through a relaxation of the impulsive hybrid system optimization. The second component localizes the surface segments using

impulsive hybrid system optimization. The two components of this feature detection algorithm are explained in detail in Section II-A and Section II-B. The computational complexity of this algorithm is discussed in Section II-C.

### A. Feature Type Estimation

The first component of the surface feature detection algorithm is performing a relaxation of the impulsive hybrid system optimization to estimate the number of surface segments and their types and sequence based on the kinematic data gathered [25]. This step also provides the initial guess of impulse times for the impulsive hybrid system optimization during feature localization (Section II-B). The relaxation of the hybrid optimization is performed by solving a non-linear optimal control problem with the following cost function:

$$\begin{aligned} \arg \min_{x(s), u(s)} J(\cdot) &= \int_{t_0}^{t_f} l(x(s), u(s), s) ds \\ l(\cdot) &= \frac{1}{2}(x(s) - x_r(s))^T Q(x(s) - x_r(s)) \\ &\quad + \frac{1}{2}(u(s) - u_d(s))^T R(u(s) - u_d(s)) \end{aligned} \quad (1)$$

where  $x(\cdot)$  is the model trajectory which includes both position and velocity trajectories,  $x_r(\cdot)$  is the measurement trajectory consisting of kinematic data,  $u(\cdot)$  is the model control signal, and  $u_d(\cdot)$  is the desired control signal. The signal  $x(t)$  is subject to the following dynamic constraint:

$$\begin{aligned} \dot{x}(t) &= V(t) + \sum_{i=1}^{N-1} u_i(t) F_i(x(t), t) + (1 - \sum_{i=1}^{N-1} u_i) F_N(x(t), t) \\ V(t) &= [0_1, \dots, 0_M, v_1(t), \dots, v_M(t)]^T \end{aligned} \quad (2)$$

where  $F_i$  is the  $i$ -th dynamical mode of a robotic finger on the  $i$ -th type of surface,  $N$  is the total number of possible surface profile types, and  $M$  is the total number of states (trajectories) expected to have impulses.  $M$  is usually half of the total number of states because impulses only occur in the velocity trajectories when the path is continuous. Note that if one constrains  $u_i$  such that  $\sum_{i=1}^N u_i = 1$ ,  $u_i \in \{0, 1\}$ , and  $v_i$  to be delta functions, then optimizing this system over  $(u_i, v_i)$  is equivalent to the hybrid/impulsive optimization. We relax the optimization, allowing  $u_i$  and  $v_i$  to violate these constraints to make the computation a standard, smooth optimal control problem.

The control signal,  $u$ , in (1) includes both  $u_i$  and  $v_i$ . Information about “active” feature type and their sequence is captured by  $u_i$ . When  $u_k(t)$  is 1, the  $k$ -th feature type is “active” (i.e., the  $k$ -th feature is traced by the sensor at time  $t$ ). The other signals,  $u_i(t)$  where  $i \neq k$ , will be close to 0 because at anytime, only one feature can be “active”. Another set of control signals,  $v_j$ , capture the information about impulses. These  $v_j$  will remain 0 until an impulse occurs. Following this construction of the control signals, the desired control signals,  $u_d$ , in (1) are chosen to be 0 for all  $u_i$ . However, the  $u_d$  corresponding to  $F_k(\cdot)$  is 1 to make the  $k$ -th dynamical mode the default mode (typically the “flat” surface mode). The desired control signals for all  $v_i$  are 0 so that they remain 0 until an impulse occurs.

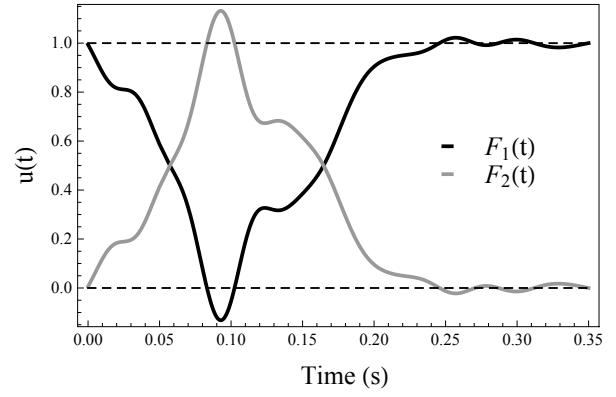


Fig. 4. The optimized control signals show that the dynamics of this system switches from the first dynamical mode to the second dynamical mode, and then back to the first dynamical mode. The switching between modes occurs near 0.06 s (when  $F_2$  becomes greater than  $F_1$ ) and 0.17 s when  $F_1$  becomes greater than  $F_2$  again.

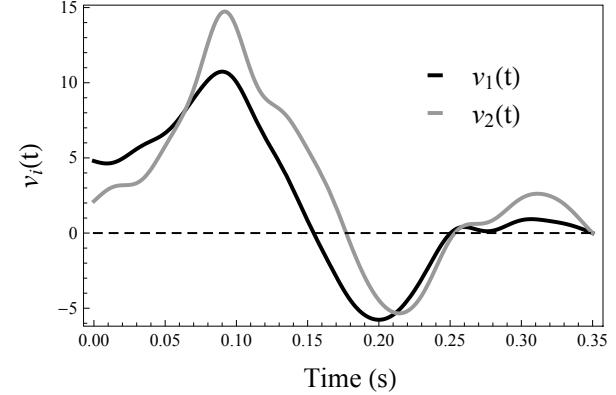


Fig. 5. The optimized control signals,  $v_1(t)$  and  $v_2(t)$ , show two impulses (spikes, as measured by having maxima that exceed a user-specified threshold of  $1/2$  the maximum value) near 0.08 s and 0.21 s.

The technical details of solving this optimal control problem are left to the paper by Caldwell et al. [25]. Figure 4 and 5 show an example of the optimized control signals. (The details of the parameters used in optimization to obtain these plots using experimental measurement data is presented in Section IV.)

Figure 4 shows the optimized control signal,  $u_1(t)$  and  $1 - u_1(t)$ , for surface type detection and sequencing. This plot shows that the dynamics of this system switches from the first dynamical mode to the second dynamical mode, and then back to the first dynamical mode. Figure 5 shows the optimized control signals,  $v_1(t)$  and  $v_2(t)$ , for impulses. In this figure, two impulses (spikes) are observed near 0.08 s and 0.21 s. These impulse times serve as initial guesses for the impulsive hybrid system optimization in feature localization, discussed next.

### B. Feature Localization

The rest of the surface feature detection algorithm involves using impulsive hybrid system optimization to localize a

feature detected in the former part of the algorithm. This section features the main mathematical contribution of this paper—impulsive hybrid system optimization. This optimization utilizes Newton's method [26] to perform least-squares estimation. An adjoint formulation is used to compute the first-order and second-order derivatives of the cost function. The results are stated below, and sketches of proofs are provided.

*1) Problem Definition:* Mathematically, the dynamical system of a robotic finger in contact with a surface is a nonlinear impulsive hybrid system of the form

$$\dot{x}(t) = f_i(x(t), t) \text{ for } \tau_i < t < \tau_{i+1} \quad (3)$$

where  $i \in \{1, 2, 3, \dots, N\}$  and  $x(\tau_i^+) = x_{i,0}$ . Each  $f_i(\cdot)$  is at least  $C^2$  in  $x$  and  $C^1$  in  $t$ . The number  $N$  is the total number of dynamical modes (surface segments). These dynamical modes are separated by  $N - 1$  distinct impulses for the total time horizon from  $\tau_1 = t_0$  to  $\tau_{N+1} = t_f$ . The  $i$ -th impulse time is  $\tau_i$ , and the impulse magnitude at  $\tau_i$  is  $\delta_i$ . At each impulse time,  $M$  out of a total of  $n$  states experience impulses whereby each impulse corresponds to a geometrical discontinuity on a surface. Hence,  $\delta_i$  is an  $M$ -length vector. Note that  $\delta_i$  is not a function of time,  $t$ , and  $\delta$  and  $\tau$  are independent of each other. The dynamical mode sequence and the number of impulses are known from the first part of the algorithm in Section II-A.

The goal is to simultaneously solve for the  $N - 1$  unknown impulse times and the  $M$  unknown impulse magnitudes,  $\delta_i$ , at each  $\tau_i$  by minimizing a cost function which is defined as

$$J(\cdot) = \int_{t_0}^{t_f} l(x(s), s) ds \quad (4)$$

where  $l(\cdot) = \frac{1}{2}(x(s) - x_r(s))^T(x(s) - x_r(s))$ . The model trajectory is represented by  $x(\cdot)$ , and the reference trajectory,  $x_r(\cdot)$ , is the smoothed continuous measurement signal.

*2) Optimization Method:* The optimization starts with first-order iterations (i.e., steepest descent) and then transitions to Newton's method (which converges quadratically [26]). At each iteration, we choose a descent direction  $z_k = -[H]^{-1}[D J(\cdot)]^T$ .  $H$  is a positive definite matrix and  $D J(\cdot)$  is the gradient defined as follows:

$$D J(\cdot) = (D_{\tau_1} J(\cdot), \dots, D_{\tau_i} J(\cdot), D_{\delta_1} J(\cdot), \dots, D_{\delta_i} J(\cdot)).$$

In steepest descent,  $H = I$  where  $I$  is the identity matrix. In Newton's method,  $H$  is the Hessian of the form

$$H = D^2 J(\cdot) = \begin{pmatrix} D_\tau^2 J(\cdot) & D_\tau D_\delta J(\cdot) \\ D_\delta D_\tau J(\cdot) & D_\delta^2 J(\cdot) \end{pmatrix}.$$

In many cases,  $D^2 J(\cdot)$  will not be positive definite. For these cases, a quasi-Newton's method [26] is implemented. The Hessian is decomposed into a matrix containing eigenvalues,  $\lambda$ , and a matrix with corresponding eigenvectors,  $P$ . The eigenvalues which are close to zero or negative are replaced with one. Then, the Hessian is reconstructed using the original matrix of eigenvectors,  $P$ , and the modified matrix of eigenvalues,  $\lambda^*$ , such that  $H = P \lambda^* P^{-1}$ . This modification results in using steepest descent in eigenvector directions with negative eigenvalues and Newton's method in eigenvector directions with positive eigenvalues.

After a descent direction is calculated, the Armijo line search algorithm is then performed to further reduce the step size as needed without changing the impulse times' order [27].

*3) Derivatives of Cost Function:* The first-order and second-order derivatives of the cost function,  $J(\cdot)$ , with respect to impulse times and impulse magnitudes are required to compute the gradient and the Hessian of  $J(\cdot)$ . The proofs are analogous to those in the work of Tarvers et al. [28], and a sketch of the proof is provided following each theorem statement. More detailed derivations can be found in the conference proceeding [29]. The adjoint formulation allows the derivation of both the first-order and second-order derivatives, and the integrations required are independent of the number of impulses being optimized over. This reduction in computational burden results in real-time or near real-time calculations for the example system in Section IV.

#### First-Order Derivatives of Cost Function:

*Lemma 1:* The first-order derivative of a cost function,  $J(\cdot)$ , with respect to an impulse time,  $\tau_i$ , is

$$\begin{aligned} D_{\tau_i} J(\cdot) &= \psi(t_f, \tau_i) \circ X^i + l(x(\tau_i^-), \tau_i^-) - l(x(\tau_i^+), \tau_i^+) \\ X^i &= (f_{i-1}(x(\tau_i^-), \tau_i^-) - f_i(x(\tau_i^+), \tau_i^+)) \end{aligned} \quad (5)$$

where the first-order adjoint operator,  $\psi(\cdot)$ , is found by integrating the following differential equation backwards along  $\tau$  from  $t_f$ :

$$\begin{aligned} \frac{\partial}{\partial \tau} \psi(t, \tau) &= -D_1 l(x(\tau), \tau) - \psi(t, \tau) \circ D_1 f(x(\tau), \tau) \\ \psi(t, t) &= 0. \end{aligned} \quad (6)$$

*Proof:* First, write (3) in integral form and differentiate the equation with respect to  $\tau_i$ . Then, rewrite the result in differential form:

$$\begin{aligned} \frac{\partial}{\partial t} D_{\tau_i} x(t) &= D_1 f_i(x(t), t) \circ D_{\tau_i} x(t) \\ D_{\tau_i} x(\tau_i) &= f_{i-1}(x(\tau_i^-), \tau_i^-) - f_i(x(\tau_i^+), \tau_i^+). \end{aligned} \quad (7)$$

When  $t \geq \tau_i$ , equation (7) can be represented as a state transition matrix operating on an initial condition:

$$D_{\tau_i} x(t) = \Phi(t, \tau_i) \circ D_{\tau_i} x(\tau_i) \quad (8)$$

where  $\Phi(\cdot)$  is the state transition matrix for the linearization. When  $t < \tau_i$ , changes in  $\tau_i$  will not affect  $x(t)$  because  $x(t)$  is in the past relative to  $\tau_i$ . Therefore,

$$\begin{aligned} D_{\tau_i} x(t) &= \begin{cases} 0, & t < \tau_i \\ \Phi(t, \tau_i) \circ X^i, & t \geq \tau_i \end{cases} \\ X^i &= (f_{i-1}(x(\tau_i^-), \tau_i^-) - f_i(x(\tau_i^+), \tau_i^+)). \end{aligned} \quad (9)$$

Next, take the derivative of (4) with respect to  $\tau_i$  and substitute (9) into the equation:

$$\begin{aligned} D_{\tau_i} J(\cdot) &= \int_{\tau_i}^{t_f} D_1 l(x(s), s) \circ D_{\tau_i} x(s) ds \\ &\quad + l(x(\tau_i^-), \tau_i^-) - l(x(\tau_i^+), \tau_i^+) \end{aligned} \quad (10a)$$

$$\begin{aligned} &= \int_{\tau_i}^{t_f} D_1 l(x(s), s) \circ \Phi(s, \tau_i) ds \circ X^i \\ &\quad + l(x(\tau_i^-), \tau_i^-) - l(x(\tau_i^+), \tau_i^+). \end{aligned} \quad (10b)$$

The last two terms in (10b) come from applying Leibniz rule. The integrand is integrated from  $\tau_i$  to  $t_f$  because  $D_{\tau_i}x(t) = 0$  up until  $t = \tau_i$  as stated in (9). Now, define

$$\psi(t, \tau) = \int_{\tau}^t D_1 l(x(s), s) \circ \Phi(s, \tau) ds. \quad (11)$$

Note that (11) is the integral representation of (6). Lastly, substitute (11) into (10b):

$$D_{\tau_i} J(\cdot) = \psi(t_f, t) \circ X^i + l(x(\tau_i^-), \tau_i^-) - l(x(\tau_i^+), \tau_i^+).$$

#### Second-Order Derivatives of Cost Function:

*Theorem 1:* The second-order derivative of a cost function,  $J(\cdot)$ , with respect to an impulse time,  $\tau_i$ , is

$$\begin{aligned} & D_{\tau_j} D_{\tau_i} J(\cdot) \\ &= D_1 l(x(\tau_i^-), \tau_i^-) \circ (D_{\tau_j} x(\tau_i^-) - D_{\tau_j} x_r(\tau_i^-) \delta_i^j) \\ &\quad - D_1 l(x(\tau_i^+), \tau_i^+) \circ (D_{\tau_j} x(\tau_i^+) - D_{\tau_j} x_r(\tau_i^+) \delta_i^j) \\ &\quad - D_1 l(x(\tau_i^+), \tau_i^+) \circ X^i \delta_i^j + \psi(t_f, \tau_i) \circ X^{i,j} \\ &\quad + \Omega(t_f, \tau_i) \circ (\Phi(\tau_i, \tau_j) \circ X^j, X^i) \end{aligned}$$

*Lemma 2:* The first-order derivative of a cost function,  $J(\cdot)$ , with respect to impulse magnitudes,  $\delta_i$ , is

$$D_{\delta_i} J(\cdot) = \psi(t_f, \tau_i) \circ \Delta^i$$

where  $\Delta^i$  has the form

$$\Delta^i = D_{\delta_i} x(\tau_i^+).$$

In general,  $D_{\delta_i} x(\tau_i^+)$  is the linearization of the transformation from the impulses' vector space to the  $x(\cdot)$  trajectories configuration space.

*Proof:* Follow similar steps to deriving  $D_{\tau_i} J(\cdot)$ . First, write (3) in integral form, and differentiate the equation with respect to  $\delta_i$  to obtain

$$D_{\delta_i} x(t) = D_{\delta_i} x(\tau_i^+) + \int_{\tau_i^+}^t D_1 f_i(x(s), s) \circ D_{\delta_i} x(s) ds. \quad (12)$$

Rewrite (12) in differential form:

$$\frac{\partial}{\partial t} D_{\delta_i} x(t) = D_1 f_i(x(t), t) \circ D_{\delta_i} x(t) \quad (13)$$

where the initial condition is  $D_{\delta_i} x(\tau_i^+)$ . When  $t \geq \tau_i$ , (13) can be represented as a state transition matrix operating on an initial condition:

$$D_{\delta_i} x(t) = \Phi(t, \tau_i) \circ D_{\delta_i} x(\tau_i^+).$$

When  $t < \tau_i$ , changes in  $\delta_i$  will not affect  $x(t)$  because  $x(t)$  is in the past relative to  $\delta_i$  at  $\tau_i$ . Hence,

$$\begin{aligned} D_{\delta_i} x(t) &= \begin{cases} 0, & t < \tau_i \\ \Phi(t, \tau_i) \circ \Delta^i, & t \geq \tau_i \end{cases} \\ \Delta^i &= D_{\delta_i} x(\tau_i^+). \end{aligned} \quad (14)$$

Next, take the derivative of (4) with respect to  $\delta_i$ , and substitute (11) and (14) into the resulting equation:

$$\begin{aligned} D_{\tau_i} J(\cdot) &= \int_{\tau_i}^{t_f} D_1 l(x(s), s) \circ D_{\delta_i} x(s) ds \\ &= \int_{\tau_i}^{t_f} D_1 l(x(s), s) \circ \Phi(s, \tau_i) ds \circ \Delta^i \\ &= \psi(t_f, \tau_i) \circ \Delta^i. \end{aligned} \quad (15)$$

At this point, impulse times and impulse magnitudes can be estimated using first-order optimization methods such as steepest descent. To implement a second-order method to obtain quadratic convergence, the second-order derivatives of the cost and the cross derivatives of the cost with respect to impulse times and impulse magnitudes are useful; this is discussed next. ■

where  $\delta_i^j$  is the Kronecker delta, and  $X^{i,j}$  is defined to be

$$X^{i,j} =$$

$$\begin{cases} D_1 f_i(x(\tau_i^+), \tau_i^+) \circ f_i(x(\tau_i^+), \tau_i^+) \\ + D_1 f_{i-1}(x(\tau_i^-), \tau_i^-) \circ f_{i-1}(x(\tau_i^-), \tau_i^-) \\ - 2D_1 f_i(x(\tau_i^+), \tau_i^+) \circ f_{i-1}(x(\tau_i^-), \tau_i^-) \\ + D_2 f_i(x(\tau_i^+), \tau_i^+) - D_2 f_{i-1}(x(\tau_i^-), \tau_i^-), & i = j \\ (D_1 f_{i-1}(x(\tau_i^-), \tau_i^-) - D_1 f_i(x(\tau_i^+), \tau_i^+)) \circ \\ \Phi(\tau_i, \tau_j) \circ X^j, & i > j \end{cases}$$

The second-order adjoint operator,  $\Omega(\cdot)$ , is found by integrating (16) backwards along  $\tau$  from  $t_f$ :

$$\begin{aligned} \frac{\partial}{\partial \tau} \Omega(t, \tau) &= -D_1^2 l(x(\tau), \tau) - \psi(t, \tau) \circ D_1^2 f_i(x(\tau), \tau) \\ &\quad - D_1 f_i(x(\tau), \tau)^T \circ \Omega(t, \tau) - \Omega(t, \tau) \circ D_1 f_i(x(\tau), \tau) \\ \Omega(t, t) &= 0. \end{aligned} \quad (16)$$

Note that in contrast to (9), the derivative of  $x(\tau_i)$  with respect to  $\tau_j$  is

$$D_{\tau_j} x(\tau_i) = \begin{cases} 0, & i < j \\ f(x(\tau_i), \tau_i), & i = j \\ \Phi(\tau_i, \tau_j) \circ X^j, & i > j \end{cases}$$

When  $i = j$ , the derivative of  $x(\tau_i)$  is taken with respect to its argument  $\tau_i$ , resulting in  $f(x(\tau_i), \tau_i)$ .

*Proof:* First, take the derivative of (9) with respect to  $\tau_j$  and apply the fundamental theorem of calculus. The derivation is similar to  $D_{\tau_i} x(\cdot)$ 's, and thus, only the result is stated:

$$\begin{aligned} & D_{\tau_j} D_{\tau_i} x(t) \\ &= \Phi(t, \tau_i) \circ X^{i,j} + \phi(t, \tau_i) \circ (\Phi(\tau_i, \tau_j) \circ X^j, X^i) \\ \phi(t, \tau) &= \int_{\tau_i}^t \Phi(t, s) D_1^2 f_i(x(s), s) \circ (\Phi(s, \tau), \Phi(s, \tau)) ds. \end{aligned} \quad (17)$$

Next, take the derivative of (10a) with respect to  $\tau_j$ , and substitute (9) and (17) into the resulting equation:

$$\begin{aligned} & D_{\tau_j} D_{\tau_i} J(\cdot) \\ &= D_1 l(x(\tau_i^-), \tau_i^-) \circ (D_{\tau_j} x(\tau_i^-) - D_{\tau_j} x_r(\tau_i^-) \delta_i^j) \\ &\quad - D_1 l(x(\tau_i^+), \tau_i^+) \circ (D_{\tau_j} x(\tau_i^+) - D_{\tau_j} x_r(\tau_i^+) \delta_i^j) \\ &\quad - D_1 l(x(\tau_i^+), \tau_i^+) \circ D_{\tau_j} x(\tau_i^+) \delta_i^j \\ &\quad + \int_{\tau_i^+}^{t_f} D_1 l(x(s), s) \circ D_{\tau_j} D_{\tau_i} x(s) ds \end{aligned}$$

$$\begin{aligned}
& + D_1^2 l(x(s), s) \circ (D_{\tau_j} x(s), D_{\tau_i} x(s)) ds \\
= & D_1 l(x(\tau_i^-), \tau_i^-) \circ (D_{\tau_j} x(\tau_i^-) - D_{\tau_j} x_r(\tau_i^-) \delta_i^j) \\
& - D_1 l(x(\tau_i^+), \tau_i^+) \circ (D_{\tau_j} x(\tau_i^+) - D_{\tau_j} x_r(\tau_i^+) \delta_i^j) \\
& - D_1 l(x(\tau_i^+), \tau_i^+) \circ X^i \delta_i^j + \psi(t_f, \tau_i) \circ X^{i,j} \\
& + \Omega(t_f, \tau_i) \circ (\Phi(\tau_i, \tau_j) \circ X^j, X^i) \quad (18)
\end{aligned}$$

where the second-order adjoint operator,  $\Omega(\cdot)$ , is defined as

$$\begin{aligned}
\Omega(t, \tau) = & \int_{\tau}^t D_1 l(x(s), s) \circ \phi(s, \tau) \\
& + D_1^2 l(x(s), s) \circ (\Phi(s, \tau), \Phi(s, \tau)) ds. \quad (19)
\end{aligned}$$

Note that (19) is the integral representation of (16). ■

*Theorem 2:* The second-order derivative of  $J(\cdot)$  with respect to an impulse magnitude is

$$D_{\delta_j} D_{\delta_i} J(\cdot) = \psi(t_f, \tau_i) \circ \Delta^{i,j} + \Omega(t_f, \tau_i) \circ (\Phi(\tau_i, \tau_j) \circ \Delta^j, \Delta^i)$$

where  $\Delta^{i,j}$  is defined as

$$\Delta^{i,j} = D_{\delta_j} D_{\delta_i} x(\tau_i^+).$$

Similar to  $D_{\delta_i} x(\tau^+)$ ,  $D_{\delta_j} D_{\delta_i} x(\tau_i^+)$  is nonzero when the impulses,  $\delta_i$ , are defined in a different configuration space from  $x(\cdot)$ .

*Proof:* The proof is omitted here because it follows similar steps in the proof of  $D_{\tau_j} D_{\tau_i} J(\cdot)$ . ■

*Theorem 3:* The cross derivative of  $J(\cdot)$  with respect to an impulse time and an impulse magnitude is

$$D_{\delta_j} D_{\tau_i} J(\cdot) = \begin{cases} \Omega(t_f, \tau_j) \circ (\Delta^j, \Phi(\tau_j, \tau_i) \circ X^i), & \tau_i < \tau_j \\ D_1 l(x(\tau_i^-), \tau_i^-) \circ D_{\delta_j} x(\tau_i^-) \\ -D_1 l(x(\tau_i^+), \tau_i^+) \circ D_{\delta_j} x(\tau_i^+) \\ +\psi(t_f, \tau_i) \circ \Delta X^{i,j} \\ +\Omega(t_f, \tau_i) \circ (\Phi(\tau_i, \tau_j) \circ \Delta^j, X^i), & \tau_i \geq \tau_j \end{cases}$$

where  $\Delta X^{i,j}$  is defined to be

$$\Delta X^{i,j} = \begin{cases} 0, & \tau_i < \tau_j \\ -D_1 f_i(x(\tau_i^+), \tau_i^+) \circ \Delta^j, & \tau_i = \tau_j \\ (D_1 f_{i-1}(x(\tau_i^-), \tau_i^-) \\ - D_1 f_i(x(\tau_i^+), \tau_i^+)) \circ \Phi(\tau_i, \tau_j) \circ \Delta^j, & \tau_i > \tau_j. \end{cases}$$

*Proof:* Proof follows the same steps as the proof of  $D_{\tau_j} D_{\tau_i} J(\cdot)$ , and it is omitted. ■

After computing these second-order derivatives for the Hessian, impulse times and impulse magnitudes can be estimated using a second-order optimization method such as Newton's method.

### C. Computational Complexity

As mentioned earlier, the algorithm presented in this paper has linear dependence on temporal data resolution. An increase in temporal data resolution is equivalent to taking a smaller time step (i.e., more integration steps) when computing integrations involved in the state and adjoint equations. As a result, the computational complexity of this optimization based algorithm only grows linearly with temporal data resolution. Hence, this surface feature detection algorithm is able to reap

benefits of increased temporal resolution of data without a superlinear increase in the computational cost.

The computational complexity of this algorithm is dominated by integrations required to solve for the gradient and the Hessian of the cost function. This algorithm utilizes an adjoint formulation which greatly reduces the total number of integrations required and hence, the computational cost. A dynamical system with  $n$  states (including both the position and velocity trajectories) and  $k$  impulses has  $\frac{n}{2}$  impulse magnitudes at each impulse time (i.e.,  $k \frac{n}{2}$  impulse magnitudes in total) because only the velocity trajectories experience impulses. Although the gradient has  $n \times k(1 + \frac{n}{2})$  entries, the first-order adjoint operator,  $\psi(\cdot)$ , that shows up in all entries of the gradient reduces the total number of integrations to  $n$  independent of the total number of impulses (features). Similarly, to compute the Hessian which has  $n \times (k(1 + \frac{n}{2}))^2$  entries, an additional  $2n^2$  integrations (independent of the total number of impulses) are required to compute the second-order adjoint operator,  $\Omega(\cdot)$ , and the state transition matrix,  $\Phi(\cdot)$ , that appear in all entries of the Hessian. Hence, for a system with  $n$  states, the total number of integrations required at each optimization step is  $n(2n + 1)$  independent of the total number of features. Accordingly, when the total time and the size of integration time step is fixed, this localization method has the benefit of zero increase in the computational complexity when the number of features are increased.

### III. SIMULATION WITH MEASUREMENT UNCERTAINTY

The feature localization component of the surface feature detection algorithm is implemented in simulations to characterize its performance in the presence of measurement noise. In simulations, a three-revolute-joint robotic finger was modeled to trace a surface shown in Section I in Fig. 6. This surface has three segments whereby each segment corresponds to a dynamical mode,  $f_i$ . The trajectory,  $x(t)$ , includes joint angles ( $\theta_1(t), \theta_2(t), \theta_3(t)$ ) and joint angular velocities ( $\omega_1(t), \omega_2(t), \omega_3(t)$ ). After the robotic finger traced the surface, the hybrid optimization was implemented to localize the surface segments based on kinematics of the finger assuming that the segment sequence is known. Note that all model

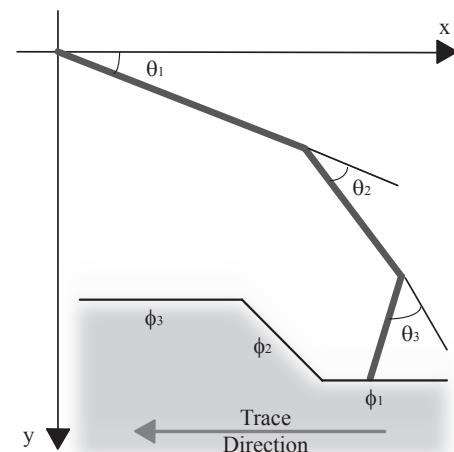


Fig. 6. Configuration of the robotic finger model used in simulation.

parameters involved including masses and lengths of the finger segments are chosen arbitrarily in the simulation. The ratio of the total finger length with respect to the slope height (in  $y$  direction) is 20:1.

Two types of measurement trajectories were considered: without noise and with noise. To generate a noisy trajectory, a measurement trajectory is simulated and sampled at 0.001 second intervals. Random noise sampled from a Gaussian distribution with mean,  $\mu = 0$ , and standard deviation,  $\sigma$ , is added at each time step. Then, the noisy trajectory is smoothed using a Gaussian filter (with standard deviation,  $\sigma_G = 5$  rad) before a spline is used to generate the continuous measurement trajectory required by the optimization formulation. Figure 7 shows the comparison of a measurement trajectory with noise (when  $\sigma = 0.1$  rad ( $5.73^\circ$ )), a measurement trajectory without noise and a smoothed measurement trajectory with noise for the first joint angle,  $\theta_1(t)$ .

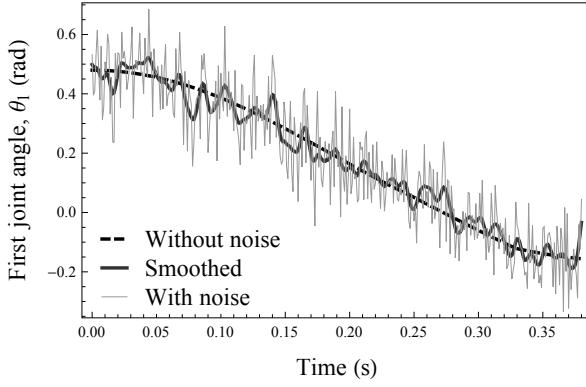


Fig. 7. Comparison of a measurement trajectory without noise, a measurement trajectory with noise ( $\sigma = 0.1$  rad ( $5.73^\circ$ ))), and a smoothed measurement trajectory with noise in the measurement of the first joint.

Both deterministic and noisy simulated measurement trajectories have impulse times,  $\tau = (0.247, 0.329)$ , impulse magnitudes  $\delta_1 = (0.249, -0.058, 4.074)$  (in the velocity) at the first impulse time, and impulse magnitudes  $\delta_2 = (-1.100, 0.102, 0.378)$  at second impulse time. The total time horizon is from  $\tau_1 = 0$  to  $\tau_4 = t_f = 0.380$  seconds. Both algorithms for measurement trajectories with noise and without noise are initialized to  $\tau = (0.270, 0.350)$ ,  $\delta_1 = (0.230, -0.030, 4.070)$ , and  $\delta_2 = (-0.900, 0.200, 0.410)$ . The optimization algorithm is terminated when the norm of the gradient,  $\|DJ(\cdot)\| < 10^{-5}$ .

*Convergence:* The optimization algorithm took sixteen iterations to converge when using measurement trajectories without noise. The convergence plots for the logarithm of the norm of the cost gradient (Fig. 8) show quadratic convergence when the Newton's method was applied after ten initial first-order iterations. Figure 8 also shows that varying the noise level did not drastically change the total number of iterations for convergence. The average number of iterations taken is seventeen when the standard deviation,  $\sigma$ , of the added noise is 0.1 rad ( $5.73^\circ$ ). This method converges even when the noise level in the measurement trajectories is high (Fig. 7), discussed next.

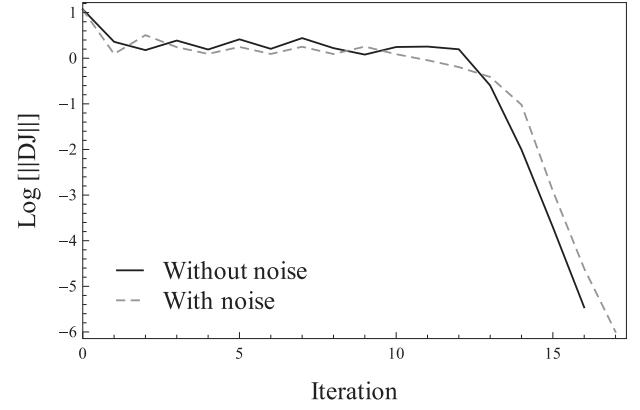


Fig. 8. Convergence plots for the logarithm of the norm of cost gradient of a finger model when the measurement trajectories are (a) without noise and (b) with noise. First ten iterations use steepest descent, and the remaining iterations use Newton's method.

*Measurement Noise:* The effect of measurement noise on the standard deviation of the estimates was investigated using Monte Carlo methods. For each noise level, thirty random simulations are computed. The noise level is quantified by the standard deviation of the noise added,  $\sigma$ . The largest  $\sigma$  considered is 0.1 rad ( $5.73^\circ$ ). A noise level above  $\sigma = 0.1$  rad results in frequent instability during optimization.

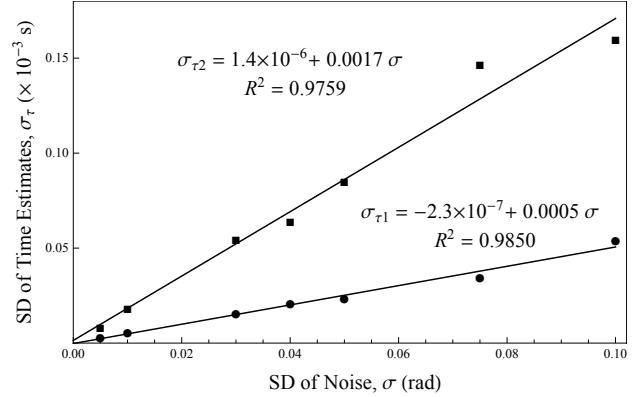


Fig. 9. Effect of the standard deviation (SD) of measurement noise on the SD of the estimated impulse times.

The optimization algorithm shows a linear increase in the standard deviation of estimates when the standard deviation of noise is increased (Fig. 9). Note that in Fig. 9, the trend lines go through the origin, and thus reflect that the estimates are unbiased. Also, note that the estimates at the second impulse time are more sensitive to the noise than the estimate at the first impulse time. This is expected because the derivatives in the gradient and the Hessian are calculated using backward integration. The estimate at the second impulse time includes fewer points, and thus are more prone to be affected by outliers in the noise. On the other hand, the estimate at the first impulse time includes more points, and hence the effect of outliers in the noise is reduced.

#### IV. EXPERIMENTAL VALIDATION

The proposed surface feature detection algorithm is implemented using experimental data, and the results are discussed in this section. The experiments used kinematic data from a PHANToM OMNI haptic device tracing surfaces. This device has six degrees of freedom. But, in the experiments, only a revolute joint at the base and a revolute joint on the arm are free allowing the arm to move in a plane. The unused joints are secured using strong adhesive tapes so that they are sufficiently rigid. As a result, this device only has two remaining degrees of freedom whereby  $x(t) = (\theta_1(t), \theta_2(t), \omega_1(t), \omega_2(t))$ . The configurations and geometrical parameters for the model of this device follows the work of Silva et al. [30]. The non-geometrical parameters such as damping coefficients of motors are determined using system identification.

In the experiments, the arm of the haptic device acts as the “finger”, and the stylus’ tip is the “fingertip” that traces a surface. Each joint is actuated based on a proportional control law. The surfaces considered in the experiments are shown in Fig. 14. The performance of this algorithm is primarily illustrated using the ledge as an example. The feature localization algorithm is also implemented on other surface examples including a bump, a hole and a flat surface with two different friction segments to show that the algorithm works with multiple surfaces.

##### A. Example: Ledge

The results of the feature detection and localization algorithm implemented for a ledge are presented in this section<sup>1</sup>. Note that although the algorithm assumes a dynamical model that is discontinuous (i.e., three distinct segments) and the kinematic data gathered are continuous (i.e., the boundaries of the three segments are less obvious), the algorithm is able to provide reasonable estimates on the locations of these boundaries.

1) *Feature Type Estimation:* The first component of the detection algorithm involves a relaxation of the impulsive hybrid system optimization using the following dynamics and optimization parameters,

$$\begin{aligned} f(t) &= [0, 0, u_2(t), u_3(t)]^T + u_1(t)F_1(t) + (1 - u_1(t))F_2(t), \\ Q &= 400 I, R = \text{diag}(500, 1, 1), \\ u_d(t) &= (1, 0, 0)^T \end{aligned}$$

where  $F_1$  is the dynamics of the stylus constrained to the flat surface and  $F_2$  is constrained to the slanted surface. The parameter values in  $Q$  and  $R$  are determined using simulated data to obtain clear optimized control signals before they are utilized in the optimization using the experimental data. The optimized trajectory for this example is shown in Fig. 10.

Figure 4 and 5 from Section II-A show the optimized control signals. Figure 4 shows that the dynamics of this system switches from the first dynamical mode to the second dynamical mode, and then back to the first dynamical mode.

<sup>1</sup>This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors. This includes a video of the experiments and readme file. This material is 5.2 MB in size

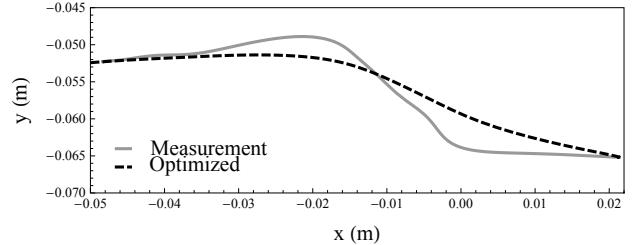


Fig. 10. Comparison of the optimized simulated trajectory and the measured trajectory.

This result is consistent with the experiment whereby the stylus moved from a flat surface to a slanted surface, and back to a flat surface. In Fig. 5, two impulses (spikes) are observed near 0.08 s and 0.21 s. These impulse times serve as initial guesses for the impulsive hybrid system optimization in feature localization.

2) *Feature Localization:* For the feature localization of the detection algorithm, trials were run to understand its performance against uncertainties in experiments.

*Model Uncertainty:* Experiments were conducted for a range of slope angles. The smallest slope angle is 11.5°. The largest slope angle is 53.1°. These angles are bounded above by the ability of the haptic device tracing across the surface. During the experiment, the stylus tip lost contact with the surface for a short amount of time right after it encountered the second corner. This momentarily contact lost was almost unnoticeable, and it was viewed as random noise in this analysis. Although the dynamical model of the haptic device is not precisely determined, this surface feature detection algorithm performs well. The algorithm converges for all angles considered, hence indicating that this algorithm can localize a reasonably wide range of ledges using kinematic data alone. As shown in Fig. 11, the optimized impulse times and impulse magnitudes produced a trajectory which closely matches the measured trajectory for a surface with a slope at 36.9° angle.

*Feature Size Sensitivity:* Using the dynamical model of the PHANToM OMNI haptic device, sensitivity of this algorithm with respect to the size of a feature is studied in simulation using Monte Carlo methods. Figure 12 shows the results of the simulated experiments where at each of 8 slope angles, 40 random trials were performed using a procedure similar to simulations in Section III. The Gaussian noise added has mean,  $\mu = 0$ , and standard deviation,  $\sigma = 0.005$  rad (0.29°). As slope angle decreases, the standard deviation of the time estimates increases superlinearly. This observation is expected because as the impulse magnitudes decrease with decreasing slope angle, the algorithm is more likely to mistake measurement noise as impulses.

*Partial Kinematic Information:* The feature localization algorithm was also implemented assuming that full information of the kinematics of a robotic finger is not available. In other words, the cost function of the least square estimation considers only some of the finger’s configurations instead of the full configurations. In this case, only the base angle of this

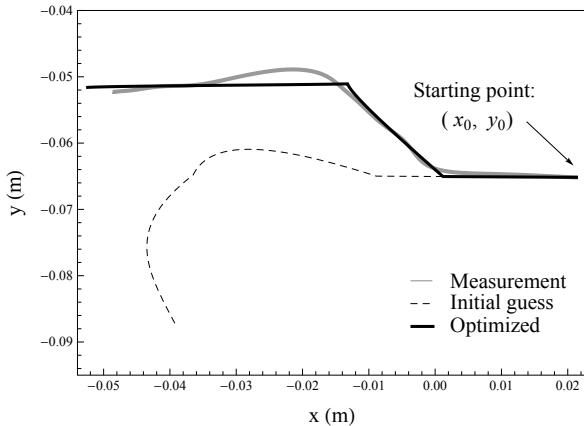


Fig. 11. Comparison of the stylus tip's trajectory computed from (a) measured joint configurations, (b) the initial guess of impulse times and impulse magnitudes:  $\tau = (0.1, 0.2)$ ,  $\delta_1 = (-1, -1)$ , and  $\delta_2 = (0, 1)$ , and (c) the optimized impulse times and impulse magnitudes:  $\tau = (0.08, 0.17)$ ,  $\delta_1 = (-1.26, -2.33)$ , and  $\delta_2 = (0.24, 0.83)$ .

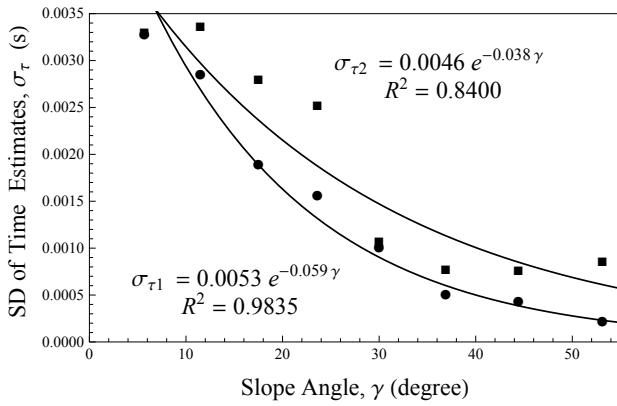


Fig. 12. Effect of slope angle on the standard deviation (SD) of the estimated impulse times.

device is assumed to be available and used as the measurement input to the algorithm. Figure 13 shows a comparison between the estimated trajectory from an optimization using only the measurement of the base joint and the estimated trajectory from an optimization using the measurement trajectories of all joints for a surface with slope angle of  $36.9^\circ$ . Using partial state information results in a less accurate feature localization as expected. Nevertheless, this result suggests that even when only the base angle of a sensor can be measured (e.g., whiskers [31]), a feature can be detected and localized.

#### B. Additional Examples

In addition to the ledge example, the localization method was also implemented on three other types of surface—a bump, a hole and a flat surface with two varying friction segments—as shown in Fig. 14. Ten trials of data were collected for each surface and the localization algorithm was implemented for each set of data. A summary of the mean and standard deviation of the estimates for each surface is presented in Table I and Table II. These results suggest that this algorithm is able to perform under a wide variety of

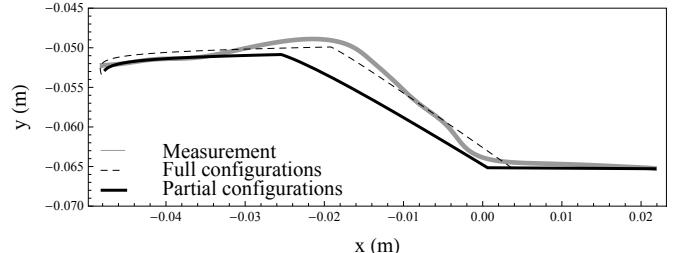


Fig. 13. Comparison of the stylus tip's trajectory computed from measured joint configurations (gray), the optimized impulse times and impulse magnitudes using measurement trajectories of all joints (dashed), and the optimized impulse times and impulse magnitudes using only measurement trajectory of the base joint (black).

surface profiles. Note that a change in surface texture has the same mathematical description as a change in surface shape, and hence the same algorithm applies in both situations. Note also that generally, impulse magnitudes have higher standard deviations than switching times. In practice, although locations of boundaries for each surface segment of a surface stay the same, impulse magnitudes vary with small changes in the set up such as small deviations in initial conditions, tracing replicas of the same surface and using different devices of the same model. Consistent with these practical scenarios, current results show that time estimates (which represent the locations of boundaries for each surface segment) are not as sensitive to deviations in the set up as the impulse magnitudes.

## V. CONCLUSION

An approach for detecting and localizing a surface feature based on kinematic sensing during haptic exploration is presented in this paper. The algorithm utilizes impulsive hybrid dynamical system optimization which performs a least squares estimation to localize a surface feature. The algorithm, which uses adjoint equations to compute first and second derivatives for the estimate, is computationally efficient and scalable. It is also shown to perform well in both simulation and experiment when measurement noise and model uncertainty are present. More broadly, this paper provides a preliminary framework to define the robotic sense of touch based on hybrid optimization techniques, without the use of collocated tactile sensors.

There are two substantial weaknesses involved in the approach discussed here. First, the use of the relaxation to estimate the number of impulses involves the choice of arbitrary parameters. Recent work in hybrid control suggests that the same basic approach can be taken without using a relaxation-based technique [32]; however, that approach does not currently allow for impulses and extending it to the optimizations in this present work is an area of future research. Second, the use of interpolation to generate smooth curves from measured data is potentially problematic because smoothing splines and other methods can introduce numerical artifacts. This is a consequence of representing the optimization problem in continuous time, where analyzing the smoothing problem is considerably easier. Although we anticipate that both of these weaknesses can be addressed in the future, they limit the applicability of these techniques to a fully automated system.

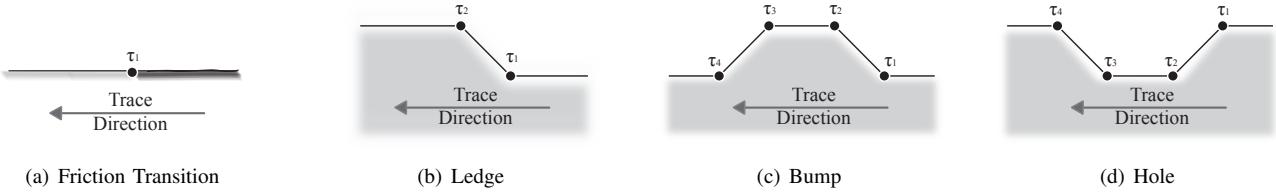


Fig. 14. Surfaces traced by the stylus of the PHANToM OMNI haptic device in experiments.

TABLE I  
MEAN AND STANDARD DEVIATION (SD) OF TIME ESTIMATES FOR A LEDGE, A BUMP, A HOLE, AND A FRICTION-CHANGING FLAT SURFACE.

	Friction <sup>1</sup>		Ledge		Bump		Hole	
	Mean (s)	SD (s)	Mean (s)	SD (s)	Mean (s)	SD (s)	Mean (s)	SD (s)
τ₁	0.11118	0.01879	0.08302	0.00089	0.15245	0.01491	0.12442	0.00984
τ₂	—	—	0.17345	0.00213	0.24749	0.01552	0.17336	0.00487
τ₃	—	—	—	—	0.33084	0.01754	0.19696	0.01203
τ₄	—	—	—	—	0.40501	0.02891	0.21945	0.00847

<sup>1</sup> Friction implies a flat surface with a change in coefficient in friction.

TABLE II  
MEAN AND STANDARD DEVIATION (SD) OF IMPULSE MAGNITUDE ESTIMATES FOR A LEDGE, A BUMP, A HOLE, AND A FRICTION-CHANGING FLAT SURFACE.

	Friction <sup>1</sup>		Ledge		Bump		Hole	
	Mean (rad)	SD (rad)	Mean (rad)	SD (rad)	Mean (rad)	SD (rad)	Mean (rad)	SD (rad)
δ <sub>1,1</sub>	0.13931	0.05466	-1.42034	0.15065	-0.80824	0.27309	-0.32228	0.24569
δ <sub>1,2</sub>	0.12632	0.14288	-2.36258	0.12621	-0.09322	0.34843	2.35943	0.48899
δ <sub>2,1</sub>	—	—	0.10905	0.12035	-0.19338	0.33353	-1.17752	0.46690
δ <sub>2,2</sub>	—	—	1.78068	0.68655	2.28606	0.79906	-0.88425	0.81428
δ <sub>3,1</sub>	—	—	—	—	0.82455	0.17740	-0.80558	0.67795
δ <sub>3,2</sub>	—	—	—	—	2.34246	0.74510	-1.15617	1.44109
δ <sub>4,1</sub>	—	—	—	—	-0.59041	0.27690	1.51281	0.35071
δ <sub>4,2</sub>	—	—	—	—	-0.43512	0.63792	0.57956	1.75373

<sup>1</sup> Friction implies a flat surface with a change in coefficient in friction.

<sup>2</sup>  $\delta_{i,j}$  means the impulse magnitude at the  $i$ -th switching time for joint angle  $j$ .

In addition, note that this algorithm analyzes features in a plane. One could extend this algorithm to understand three dimensional features using the information provided by the planar feature analysis. One simple example is to have the sensor trace across a surface multiple times in different directions to obtain boundaries of different feature regions on the surface. Extracting three dimensional features from two dimensional feature data is an area of future research.

#### ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under award IIS-1018167. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

#### REFERENCES

- R. Klatzky, S. J. Lederman, and V. A. Metzger, "Identifying objects by touch: an "expert system"," *Perception And Psychophysics*, vol. 37, pp. 299–302, 1985.
- L. M. Miller and T. D. Murphey, "Optimal contact decisions for ergodic exploration," in *IEEE Int. Conf. on Decision and Control (CDC)*, 2012.
- P. Allen, A. Miller, P. Oh, and B. Leibowitz, "Using tactile and visual sensing with a robotic hand," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, apr 1997, pp. 676–681.
- S. A. Stansfeld, "A robotic perceptual system utilizing passive vision and active touch," *International Journal of Robotics Research*, vol. 7, no. 6, pp. 138–161, 1988.
- A. Schultz, J. Solomon, M. Peshkin, and M. Hartmann, "Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, April 2005, pp. 2588 – 2593.
- K. Ioi, H. Maeda, and R. Sato, "Estimation of surface properties using a new robotic finger," in *IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 97 –102.
- P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401 –410, june 2011.
- A. Okamura and M. Cutkosky, "Feature-guided exploration with a robotic finger," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, 2001, pp. 589 – 596.
- R. Platt, F. Permenter, and J. Pfeiffer, "Using Bayesian filtering to localize flexible materials during manipulation," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 586 –598, 2011.
- P. Allen and P. Michelman, "Acquisition and interpretation of 3-D sensor data from touch," *IEEE Transactions on Robotics*, vol. 6, no. 4, pp. 397 –404, aug 1990.
- R. Ibrayev and Y.-B. Jia, "Recognition of curved surfaces from "one-dimensional" tactile data," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 3, pp. 613 –621, july 2012.
- Z. Pezzementi, E. Plaku, C. Reyda, and G. Hager, "Tactile-object recognition from appearance information," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 473 –487, 2011.
- A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009, pp. 243 –248.
- Z. Pezzementi, E. Jantho, L. Estrade, and G. Hager, "Characterization and simulation of tactile sensors," in *IEEE Haptics Symposium*, 2010, pp. 199 –205.
- R. Kikuwe and T. Yoshikawa, "Recognizing object surface properties using impedance perception," in *International Symposium on Micromechatronics and Human Science*, 2003, pp. 251 – 258.

- [16] A. M. Okamura and M. R. Cutkosky, "Feature detection for haptic exploration with robotic fingers," *International Journal of Robotics Research*, vol. 20, no. 12, pp. 925–938, 2001.
- [17] S. Decherchi, P. Gastaldo, R. Dahiya, M. Valle, and R. Zunino, "Tactile-data classification of contact materials using computational intelligence," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 635 –639, june 2011.
- [18] C. Muthukrishnan, D. Smith, D. Myers, J. Rebman, and A. Koivo, "Edge detection in tactile images," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 4, 1987, pp. 1500 – 1505.
- [19] T. J. Debus, P. E. Dupont, and R. D. Howe, "Contact state estimation using multiple model estimation and hidden Markov models," *International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 399–413, 2004.
- [20] S. Chhatpar and M. Branicky, "Localization for robotic assemblies with position uncertainty," in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 3, oct. 2003, pp. 2534 – 2540 vol.3.
- [21] A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, may 2006, pp. 707 –714.
- [22] S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo data association for multi-target tracking," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 481 –497, march 2009.
- [23] Y. Bar-shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 2001.
- [24] R. M. Murray, , Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC, 1994.
- [25] T. Caldwell and T. Murphrey, "Switching mode generation and optimal estimation with application to skid-steering," *Automatica*, vol. 47, no. 1, pp. 50 – 64, 2011.
- [26] C. Kelley, *Iterative Methods for Optimization*. Society for Industrial Mathematics, 1994.
- [27] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.
- [28] M. Travers, T. D. Murphrey, and L. Pao, "Impulsive data association with an unknown number of targets," in *Hybrid Systems: Computation and Control*, 2011, pp. 261–270.
- [29] Y. P. Leong and T. D. Murphrey, "Second order switching time and magnitude optimization for impulsive hybrid systems," in *American Controls Conf. (ACC)*, 2013.
- [30] A. Silva, O. Ramirez, V. Vega, and J. Oliver, "PHANToM OMNI haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conference*, sept. 2009, pp. 193 –198.
- [31] J. H. Solomon and M. J. Z. Hartmann, "Extracting object contours with the sweep of a robotic whisker using torque information," *International Journal of Robotics Research*, vol. 29, no. 9, pp. 1233–1245, 2010.
- [32] T. Caldwell and T. D. Murphrey, "Projection-based switching time optimization," in *American Controls Conf. (ACC)*, 2012.



**Yoke Peng Leong** received her B.S. and M.S. degrees in mechanical engineering from Northwestern University in 2012. While at Northwestern University, she was a member of the Neuroscience and Robotics Lab. Currently, she is a graduate student in control and dynamical systems at the California Institute of Technology.



**Todd Murphrey** received his undergraduate degree in mathematics from the University of Arizona and a Ph.D. in Control and Dynamical Systems from the California Institute of Technology. He was a postdoctoral scholar at Northwestern University for a year, after which he worked for the Aerospace Corporation in the Electro-Mechanical Control Department. He was an Assistant Professor of Electrical and Computer Engineering at the University of Colorado at Boulder from 2004 to 2008 and is now an Associate Professor of Mechanical Engineering and Physical Therapy and Human Movement Sciences at Northwestern University. He is the recipient of a National Science Foundation CAREER award.