

Multifingered Grasp Planning Based on Gaussian Process Implicit Surface and its Partial Differentials

LIU, Hanzhong <<http://orcid.org/0000-0001-8175-029X>>, SAMPATH, Suhas Kadalagere <<http://orcid.org/0000-0003-1098-0535>>, WANG, Ning <<http://orcid.org/0000-0002-3264-1852>> and YANG, Chenguang <<http://orcid.org/0000-0001-5255-5559>>

Available from Sheffield Hallam University Research Archive (SHURA) at:

<https://shura.shu.ac.uk/35111/>

This document is the Accepted Version [AM]

Citation:

LIU, Hanzhong, SAMPATH, Suhas Kadalagere, WANG, Ning and YANG, Chenguang (2024). Multifingered Grasp Planning Based on Gaussian Process Implicit Surface and its Partial Differentials. IEEE/ASME Transactions on Mechatronics, 29 (5), 3522-3533. [Article]

Copyright and re-use policy

See <http://shura.shu.ac.uk/information.html>

Multi-fingered Grasp Planning Based on Gaussian Process Implicit Surface and its Partial Differentials

Hanzhong Liu, Suhas Kadalagere Sampath, Ning Wang[✉], *Member, IEEE*, Chenguang Yang[✉], *Fellow, IEEE*

Abstract—Grasp planning for irregularly shaped objects using multi-fingered robotic hands is challenging due to the high dimensionality of the search space and a lack of proper modeling methods for object geometry. To address these issues, we propose a grasp planning approach based on Gaussian process implicit surfaces (GPIS). To explore the object geometry and identify feasible contact positions and normals, our method introduces several moving points called attractors along with a dynamical system. The dynamical system constrains and guides the attractors with the partial differentials of the GPIS, which can be conveniently obtained through the linear expression of a Gaussian process (GP). The hand motion is also guided by the dynamical system. Additionally, an inverse kinematics (IK) method, which considers finger joint limits, is developed to simultaneously adjust the palm pose and finger joint angles for a feasible grasp. The performance of our method is demonstrated using various robotic hands and objects, and real robot experiments are conducted to validate the planned grasp’s effectiveness in reality. Experimental evaluation demonstrates that the method works for different robotic hands and objects of varying shapes, with a higher likelihood of generating grasps with better quality.

Index Terms—Grasp planning, Gaussian process implicit surfaces (GPIS), multi-fingered inverse kinematics.

I. INTRODUCTION

Over the past few years, research on robotic grasping has grown rapidly [1]. A variety of robotic hands with multiple fingers or high degrees of freedom have been developed to improve the dexterity of grasping [2], [3]. Despite the great progress in robotic hand design, grasp planning for everyday objects remains an open challenge due to the complexity of the object shapes and the hand kinematics, which leads to a high dimensionality of searching space for a feasible grasp.

Typically, a grasp can be classified as a power grasp and a precision grasp [4], [5]. The former aims to afford substantial stability for grasping, e.g., holding a heavy tool like a hammer or a wrench, and the latter provides increased dexterity, which mainly depends on the next-step manipulation tasks after grasping [6]. From the view of grasp planning, precision grasps can be considered a simplified version of power grasps, with fewer contact areas and kinematic constraints. Therefore, this work focuses on precision grasp planning, tackling the challenges posed by complicated shapes and hand kinematics. Besides, precision grasps show huge potential for

a wide application in dexterous manipulation, e.g., grasping small/flat objects or executing high-precision in-hand manipulation tasks [4], [7].

Grasp-planning methods can be broadly categorized as learning-based and analytical approaches [8], [9]. Automatic power grasps and precision grasps are planned using human demonstrations [10]–[13]. Neural networks have also been validated as efficient ways to combine the perception of the object pose and shape with grasp planning [14]–[18]. [19] proposes a high-DOF grasp planning method with good sample efficiency, dynamic adaptability, and cross-category generality. Frictions between the object and the fingers are also learned to perform reliable grasp planning and execution [20]. Su et al. [21] carry out compliant grasps with high accuracy by compensating the errors with the enhanced kinematic model and the sparse online Gaussian process. Li et al. [22] combine GPIS with the learning of robotic hand IK and encode the hand configurations with virtual frames. However, learning-based methods are confined to human demonstrations or artificially setting rules, which more or less depend on the analytical modeling of a grasp. Besides, learning-based approaches rely on a large amount of training data, which is not a trivial issue.

There have been plenty of analytical approaches for grasp planning [23], [24], which propose efficient ways to model the multi-rigid-body dynamics during a grasp and generate feasible finger postures by considering the hand-object contacts and frictions. However, these methods rely on manually setting the palm pose of the robotic hand or sampling a series of palm poses around the object, which is not a trivial step since it’s quite difficult for users to imagine the suitable palm pose before hand closing, and the sampling number for the palm poses is hard to determine. Many analytical approaches aim to find a specified number of grasping locations on the object’s surface [25]–[27]. Among most of these researches, the grasp quality is calculated with the grasp wrench space (GWS) [28], [29], which indicates all the feasible grasping wrenches produced by the contact areas. Nevertheless, these methods merely consider the contact forces or the object’s geometry, or simplify the robotic hand kinematics with some grasping representations, but the complete robotic hand kinematics is not taken into account. Moreover, these methods cannot specify the contact positions on the robotic hands, which hinders the later dexterous manipulation after grasping. Some grasping methods utilize compliance control to improve the robustness against uncertainties of the object’s shape and position [30], [31], but robotic hands with extremely sensitive force, torque or tactile sensors are needed.

Approaches combining learning-based and analytical meth-

Corresponding author: Chenguang Yang.

Hanzhong Liu is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China (email: hanzhong.liu@foxmail.com).

Suhas Kadalagere Sampath, Ning Wang, and Chenguang Yang are with Bristol Robotics Laboratory, the University of the West of England, Bristol BS16 1QY, UK (email: Suhas.kadalageresampath@uwe.ac.uk; Katie.Wang@brl.ac.uk; cyang@ieee.org).

ods take both their advantages and can achieve fast and accurate grasp planning [32]–[34]. However, many of these methods introduce constraints such as finger joint angles coupling, while others rely on well-behaved demonstrations or wide-covering grasping data-set.

Different from the above grasp planning studies that try to find the grasping locations on the object first and then integrate the robotic hand inverse kinematics, [35] proposes a multi-dimensional iterative surface fitting (MDISF) method, which integrates the robotic hand IK into the grasp planning process and solves it as an optimization problem. GPIS potentials are used to guide the fingers in a potential field [36], but the contact positions on the robotic hands are not specified either. Furthermore, it adopts the Gaussian covariance function, including parameters to be tuned according to the prior knowledge, e.g., the size of the robotic hand.

Similarly, we utilize the Gaussian process implicit surface and its partial differentials, and additionally, we introduce several intermediate points called attractors, which are driven by a dynamical system. Within the dynamical system, the attractors guide the IK of multi-fingered robotic hands with a moving palm, which also considers the finger joint limits.

The main contributions of our work are as follows:

- 1) We introduce several points called attractors into grasp planning. The attractors slide on the object’s surface and search for the feasible grasping locations, at which the normal vectors of the object’s surface are aligned to those of the fingertips. With the attractors as intermediation, the parameters of GPIS need not to be tuned according to the workspace size of the robotic hand, but only the size of the object.
- 2) A novel constraining and surface normals searching dynamical system for the attractors is designed, which utilizes the partial differentials of GPIS. During grasp planning, the dynamical system is driven by the positional and angular deviation between the fingertips and the attractors, which also intuitively reveals the deviation between the current grasp and a feasible grasp.

II. RELATED WORK

A. Object Surface Modeling for Grasping

Object geometry is one of the most important considerations in grasping. Different from many previous works which attempted to characterize the object geometry with the shape primitives [37], [38], many everyday objects are irregularly shaped. Among different modeling methods for object shapes, the implicit surface [39] is a suitable choice because it defines the surface with a continuous scalar-valued function over the domain \mathbb{R}^3 and allows an easy specification of the locations of points on the surface. The surface normals can also be specified with the partial differentials of the scalar-valued function. Williams et al. [40] propose the Gaussian process implicit surfaces (GPIS), which fits the object shape well and also provides a meaningful probabilistic interpretation. The partial differentials [41] of the Gaussian process are exploited to derive the surface normals [22], [42], which also inspires our work to constrain the aforementioned attractors on the object’s surface.

B. Multi-fingered Hand Inverse Kinematics

Inverse kinematics for robotic hands with multiple fingers is another interesting issue [43]–[45]. Because of the low DoFs of each robotic finger, IK solutions in explicit forms can be easily derived [43], [46], [47]. However, such methods ignore the 6 DoFs provided by the moving palm, which can be crucial in the scenario of grasping an object. Besides, the targeting position and orientation of each fingertip are specified in advance, which cannot satisfy our needs to search both the suitable contact locations on the object’s surface and the corresponding configurations of the palm and fingers. Inverse kinematics based on the inverse or transpose of the hand Jacobian [48], [49] provides iterative methods which converge to the desired fingertip poses, while the moving palm remains unconsidered. Qiu et al. [50] estimate the desired palm pose while reaching the object and the palm pose is further adjusted by the pose errors of the fingertips, but the adjustments for the palm and fingers are carried out alternately and predefined targeting poses of the fingertips are still needed. Similar to our previous hierarchical inverse kinematics (HIK) method [45] which makes online decisions for multi-fingered motions during a grasp, this work adopts the inverse-Jacobian based HIK, but in a global grasp planning scenario. Moreover, finger joint limits are well tackled by adjusting the null-space.

The rest of this paper is structured as follows. Section III explains the details of our method. Section IV provides the experimental results. Section V concludes the paper and discusses possible future work.

III. METHODOLOGY

A. Overview

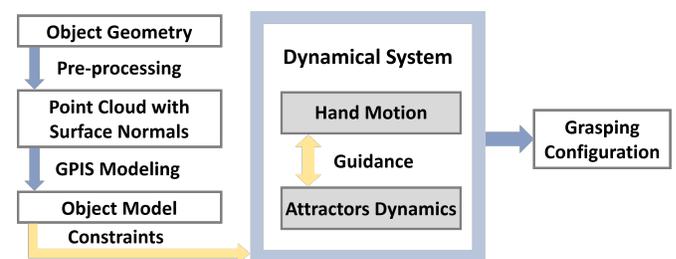


Fig. 1. Overview of the grasp planning algorithm.

As illustrated in Fig. 1, in our grasp planning algorithm, an accurate point cloud with surface normals is first sampled from the surface of the object whose geometry is known, and GPIS takes it to model the object shape (III-B). In III-D, we introduce several moving points called attractors along with a dynamical system, which is defined by the partial differentials of GPIS. The dynamical system constrains the attractors and guides them towards the position where the surface normal is opposite to the fingertip’s normal. The attractors also guide the fingertips to touch the object and form a grasp. The guided motion of the robotic hand is calculated with a task-oriented IK method described in Part III-C.

The grasp planning process iterates until each fingertip achieves feasible contact with the object, i.e., the position

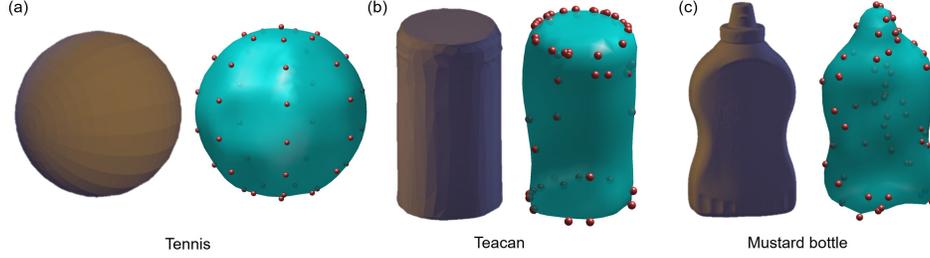


Fig. 2. The mesh models on the left show the original shapes of the objects and the cyan surfaces on the right are the corresponding objects' surfaces modelled by GPIS. The red spheres denote the point cloud sampled from the original object's surface.

of the attractor and the corresponding surface normal are overlapped with those of the fingertip. At last, a feasible grasp is planned and is denoted as follows:

$$\mathbf{G}^* = \{\mathbf{T}_{palm}^*, \mathbf{q}_1^*, \dots, \mathbf{q}_n^*\} \quad (1)$$

where \mathbf{T}_{palm}^* is the planned desired palm pose for the robotic hand and \mathbf{q}_i^* is the desired finger joint angles for the i^{th} finger. With the grasping configuration, downstream autonomous grasps can be executed with a given targeting object pose.

B. Object modeling

1) *Gaussian process implicit surfaces*: We model the object's surface along with the surface normal directions with GPIS [40], [42], which is trying to fit an implicit surface function:

$$g(\mathbf{x}) = \mathbf{y} \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^3$ denotes the given point in the 3-D space in which the targeting object is placed. Each point \mathbf{x} is expressed in the object frame whose origin locates at the center of the object's point cloud and the object frame is resized according to Eq. (9). The output $\mathbf{y} \in \mathbb{R}^4$ is defined as follows:

$$\mathbf{y} = [l \quad \mathbf{n}^T]^T \quad (3)$$

where $[\ast]^T$ denotes the transpose of a matrix or a vector and l is defined as a signed distance with the following meaning:

$$l \begin{cases} < 0, & \mathbf{x} \text{ is inside the object,} \\ = 0, & \mathbf{x} \text{ is on the object surface,} \\ > 0, & \mathbf{x} \text{ is outside the object.} \end{cases} \quad (4)$$

and \mathbf{n} is the unit normal vector of the equipotential surface at position \mathbf{x} . The equipotential surface is made up of points with the same value of l and happens to be the object's surface in the case when $l = 0$.

Given the training data including the inputs and outputs mentioned above, the function Eq. (2) can be fit by the Gaussian process. While a new testing position $\hat{\mathbf{x}}$ is given to predict its output $\hat{\mathbf{y}}$, the expectation and covariance of $\hat{\mathbf{y}}$ are:

$$\mathbf{E}(\hat{\mathbf{y}}) = \mathbf{K}_*[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{Y} \quad (5)$$

$$\text{cov}(\hat{\mathbf{y}}) = \mathbf{k}(\hat{\mathbf{x}}, \hat{\mathbf{x}}) - \mathbf{K}_*[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{K}_*^T \quad (6)$$

where $\mathbf{X} \in \mathbb{R}^{m \times 3}$ and $\mathbf{Y} \in \mathbb{R}^{4m \times 1}$ are the input and output of the training data-set with m samples and σ^2 denotes the variance of noise in the output. Symbols \mathbf{K}_* and $\mathbf{K}(\mathbf{X}, \mathbf{X})$

are in the typical formula of GP and their detail is included in Appendix A.

2) *Object modeling with point cloud*: After preprocessing, we have a point cloud of the object's surface, denoted as the red spheres in Fig. 2:

$$\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\} \quad (7)$$

and the surface normals of the point cloud \mathcal{P} :

$$\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_m\} \quad (8)$$

where \mathbf{p}_i is the position of the i^{th} point and \mathbf{n}_i is the surface normal at \mathbf{p}_i . To achieve both detailed modeling and less computation, we choose an appropriate size of the training data-set according to the modeling results in previous work [22], [42], i.e., 60 sampling points on the object's surface.

Given the point cloud, we first normalize the point cloud as the GPIS inputs:

$$\mathbf{X} = \left\{ \mathbf{x}_i \mid \mathbf{x}_i = \frac{\mathbf{p}_i - \bar{\mathbf{p}}}{\mathbf{p}_{max}} \right\} \quad (9)$$

in which \mathbf{p}_{max} is the Euclidean distance between the farthest pair of points and $\bar{\mathbf{p}}$ is the center of the point cloud.

The outputs are piled up with elements from Eq. (3):

$$\mathbf{Y} = [\mathbf{y}_1^T \quad \mathbf{y}_2^T \quad \dots \quad \mathbf{y}_m^T]^T \quad (10)$$

Additionally, we add one extra training sample inside the object at the center of point cloud:

$$\begin{cases} \mathbf{x}_{in} = [0 \quad 0 \quad 0]^T, \\ \mathbf{y}_{in} = [-1 \quad 0 \quad 0 \quad 0]^T \end{cases} \quad (11)$$

and 14 extra training samples outside the object, which are uniformly distributed on a sphere with a unit radius:

$$\begin{cases} \mathbf{x}_{out_i} = \mathbf{r}_i, \\ \mathbf{y}_{out_i} = [1 \quad \mathbf{r}_i^T]^T \end{cases} \quad (12)$$

where \mathbf{r}_i is the Cartesian position of the i^{th} point on the unit sphere's surface.

C. Multi-fingered Hand IK

1) *Hand Jacobian*: Since the robotic hand is mounted on the robot end-effector, the palm pose and the finger joint angles can be easily controlled. Here we take their derivatives as the controlled variable:

$$\mathbf{U} = [\mathbf{V}_{palm}^T \quad \dot{\mathbf{q}}_1^T \quad \dot{\mathbf{q}}_2^T \quad \dots \quad \dot{\mathbf{q}}_n^T]^T \quad (13)$$

where \mathbf{V}_{palm} is the palm twist and $\dot{\mathbf{q}}_i$ is the finger joint velocities of the i^{th} finger.

Since we expect to plan motions for all the fingertips, we have:

$$\mathbf{V} = [\mathbf{V}_1^T \quad \mathbf{V}_2^T \quad \cdots \quad \mathbf{V}_n^T]^T \quad (14)$$

where \mathbf{V}_i is the i^{th} fingertip's twist and a hand kinematic Jacobian \mathbf{J}_k (Appendix B) can describe the linearized relationship between the controlled variable and the fingertips' twists:

$$\mathbf{V} = \mathbf{J}_k \mathbf{U} \quad (15)$$

2) *Task-oriented inverse kinematics*: To formulate our iterative inverse kinematics with the robotic hand, we first define the task goal for the i^{th} fingertip, i.e., to achieve the desired position \mathbf{p}_i^* and fingertip normal \mathbf{n}_i^* . Together with the current position \mathbf{p}_i and fingertip normal \mathbf{n}_i , the task deviation is derived directly:

$$\Delta \mathbf{O}_i = [\mathbf{p}_i^{*T} - \mathbf{p}_i^T \quad 1 - \cos\psi_i]^T \quad (16)$$

where ψ_i denotes the angle between \mathbf{n}_i^* and \mathbf{n}_i and $1 - \cos\psi_i$ reflects the angular deviation with a range of between 0 and 2. In practice, the term $\cos\psi_i$ is substituted with the inner product $\mathbf{n}_i^T \mathbf{n}_i^*$, which is convenient for the derivative and therefore derives the bottom right part of the Jacobian in Eq. (52).

For all the fingertips, we have

$$\Delta \mathbf{O} = [\Delta \mathbf{O}_1^T \quad \Delta \mathbf{O}_2^T \quad \cdots \quad \Delta \mathbf{O}_n^T]^T \quad (17)$$

The linearized relationship between the fingertips' twists and the derivatives of the tasks can be expressed as another Jacobian \mathbf{J}_{task} (Appendix C):

$$\dot{\mathbf{O}} = \mathbf{J}_{task} \mathbf{V} \quad (18)$$

Substituting Eq. (15) into Eq. (18), we have

$$\dot{\mathbf{O}} = \mathbf{J} \mathbf{U} \quad (19)$$

where $\mathbf{J} = \mathbf{J}_{task} \mathbf{J}_k$, and now we can formulate our task-oriented IK method:

$$\mathbf{U} = \mathbf{J}^\dagger \Delta \mathbf{O} \quad (20)$$

where $[\ast]^\dagger$ denotes the pseudo-inverse of the matrix $[\ast]$.

In each planning step of the hand IK, the desired position and normal of each fingertip are given with those of the corresponding attractor, which is not static since it is simultaneously exploring the object's surface.

3) *Null-space adjustment for joint limits avoidance*: Most multi-fingered robotic hands are designed to have very limited finger joint ranges, which effectively avoids self collision and reduces the search space for planning a grasp. However, the limited joint ranges make it easier to fall into joint limits, which frequently traps the palm in the wrong pose and the fingers in the wrong postures and slows down the planning.

Therefore, we flexibly adjust the null-space to encourage the robotic hand to leave the joint limits during the planning. When a joint achieves its limits, we set the corresponding column of \mathbf{J} to 0. Besides, we add a secondary task goal to the task-oriented IK:

$$\mathbf{U} = \alpha \mathbf{J}^\dagger \Delta \mathbf{O} + \beta (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \mathbf{U}^* \quad (21)$$

where α determines the weight of the original task goal same as Eq. (20) and β determines the weight of the secondary task within the null-space. The secondary task goal is designed as follows:

$$\mathbf{U}^* = [\mathbf{V}_{palm}^{*T} \quad \dot{\mathbf{q}}_1^{*T} \quad \dot{\mathbf{q}}_2^{*T} \quad \cdots \quad \dot{\mathbf{q}}_n^{*T}]^T \quad (22)$$

in which $\dot{\mathbf{q}}_i^*$ is derived with the deviation between the current joint angles and the desired ones, i.e., $\mathbf{q}_i^* - \mathbf{q}_i$, and the desired finger joint angles are chosen with a pre-designed natural hand posture far away from joint limits. While achieving the limit, the joint is excluded from solving the primary task by setting the corresponding column of \mathbf{J} to 0. Meanwhile, the secondary task drives it towards the pre-designed angle and thus the joint can get rid of joint limits.

For the desired palm twist $\mathbf{V}_{palm}^* = [\mathbf{v}_{palm}^{*T} \quad \boldsymbol{\omega}_{palm}^{*T}]^T$ in the secondary task, we use the prior knowledge that in a well-planned grasp, the fingertips should be surrounding the object. Hence, the palm linear velocity is trying to move the average position of the fingertips towards the center of the point cloud:

$$\mathbf{v}_{palm}^* = \bar{\mathbf{p}} - \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i \quad (23)$$

where $\bar{\mathbf{p}}$ is the center of the object surface's point cloud and \mathbf{p}_i is the i^{th} fingertip position, and $\boldsymbol{\omega}_{palm}^*$ is set to zero.

D. Grasp Planning with Attractors

1) *Derivatives of Gaussian process*: According to our definition, the partial derivative of our GPIS output with respect to the input \mathbf{x} has the following form:

$$\frac{\partial \mathbf{E}}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{E}}{\partial \mathbf{x}} \right] \in \mathbb{R}^{4 \times 3} \quad (24)$$

where \mathbf{E} denotes the output of GPIS in Eq. (5).

Since the training data is constant, only \mathbf{K}_* in Eq. (5) will change along with \mathbf{x} . Thus we have

$$\frac{\partial \mathbf{E}}{\partial \mathbf{x}} = \frac{\partial \mathbf{K}_*}{\partial \mathbf{x}} [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{Y} \quad (25)$$

where we define a tensor $\frac{\partial \mathbf{K}_*}{\partial \mathbf{x}} \in \mathbb{R}^{4 \times 4m \times 3}$ to denote the partial derivative of the matrix \mathbf{K}_* , and for clarity, here we present the partial derivative with respect to the k^{th} ($k = 1, 2, 3$) dimension of \mathbf{x} , and denote it as $\frac{\partial \mathbf{K}_*}{\partial x^k} \in \mathbb{R}^{4 \times 4m}$:

$$\frac{\partial \mathbf{K}_*}{\partial x^k} = \frac{\partial}{\partial x^k} [\mathbf{k}(\mathbf{x}_a, \mathbf{x}_1) \quad \cdots \quad \mathbf{k}(\mathbf{x}_a, \mathbf{x}_m)]|_{\mathbf{x}_a=\mathbf{x}} \quad (26)$$

where $\frac{\partial}{\partial x^k} \mathbf{k}(\mathbf{x}_a, \mathbf{x}_b)|_{\mathbf{x}_a=\mathbf{x}} \in \mathbb{R}^{4 \times 4}$ is the partial derivative of the kernel with the incoming input x and the b^{th} training sample, and it can be expanded as:

$$\frac{\partial}{\partial x^k} \mathbf{k}(\mathbf{x}_a, \mathbf{x}_b)|_{\mathbf{x}_a=\mathbf{x}} = \frac{\partial}{\partial x^k} \begin{bmatrix} cov(l_a, l_b) & cov(l_a, \mathbf{n}_b) \\ cov(\mathbf{n}_a, l_b) & cov(\mathbf{n}_a, \mathbf{n}_b) \end{bmatrix} \Big|_{\mathbf{x}_a=\mathbf{x}} \quad (27)$$

The content of $\frac{\partial}{\partial x^k} cov(l_a, l_b)|_{\mathbf{x}_a=\mathbf{x}} \in \mathbb{R}$ is as follows:

$$\frac{\partial}{\partial x^k} cov(l_a, l_b) \Big|_{\mathbf{x}_a=\mathbf{x}} = 6(dist(\mathbf{x}, \mathbf{x}_b) - \psi) \Delta_b^k \quad (28)$$

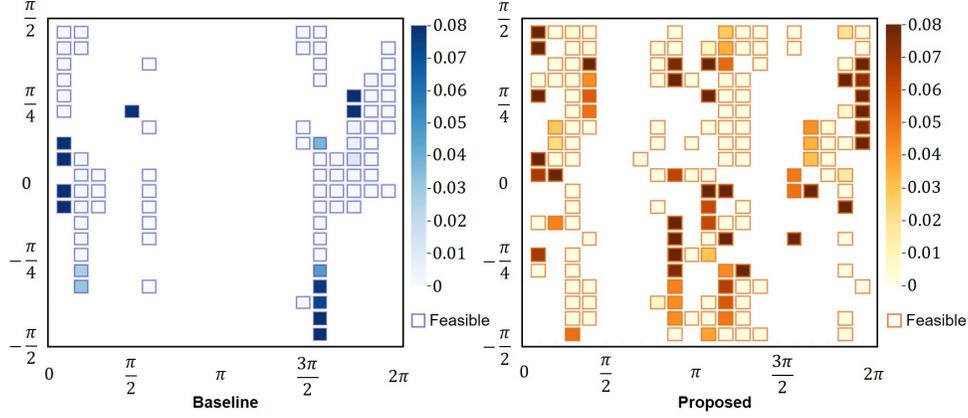


Fig. 3. Planning results on the teacup with different initial palm poses: The color shades indicate the grasp quality with a CoF of 0.6, and the boxes indicate that a feasible grasp is successfully planned with the given initial palm pose. The result shows that the baseline (joint-space-searching method) plans with a success rate of 20.5% (82 in 400) and that of the proposed method is 39.25%. The average grasp quality within the feasible grasps planned by the baseline method is 0.094578, and that of the proposed method is 0.272625.

where ψ denotes the maximum distance in the training data \mathbf{X} , same as that in Eq. (41), and $\Delta_b^k \in \mathbb{R}$ is the deviation between \mathbf{x} and \mathbf{x}_b in the k^{th} dimension:

$$\Delta_b^k = x^k - x_b^k \quad (29)$$

The entries of $\frac{\partial}{\partial x^k} cov(\mathbf{n}_a, l_b) \Big|_{\mathbf{x}_a=\mathbf{x}} \in \mathbb{R}^{3 \times 1}$ are as follows:

$$\left[\frac{\partial}{\partial x^k} cov(\mathbf{n}_a, l_b) \Big|_{\mathbf{x}_a=\mathbf{x}} \right]_i = 6(x^i - x_b^i) \frac{\Delta_b^k}{dist(\mathbf{x}, \mathbf{x}_b)} + 6 \frac{\partial x^i}{\partial x^k} (dist(\mathbf{x}, \mathbf{x}_b) - \psi) \quad (30)$$

The entries of $\frac{\partial}{\partial x^k} cov(\mathbf{n}_a, \mathbf{n}_b) \Big|_{\mathbf{x}_a=\mathbf{x}} \in \mathbb{R}^{3 \times 3}$ are

$$\left[\frac{\partial}{\partial x^k} cov(\mathbf{n}_a, \mathbf{n}_b) \Big|_{\mathbf{x}_a=\mathbf{x}} \right]_{i,j} = 6D_b^i D_b^j D_b^k - 6 \frac{\partial x^i}{\partial x^j} D_b^k - 6 \frac{\partial x^i}{\partial x^k} D_b^j - 6 \frac{\partial x^j}{\partial x^k} D_b^i \quad (31)$$

in which the term D_b^i is defined as follows:

$$D_b^i = \frac{x^i - x_b^i}{dist(\mathbf{x}, \mathbf{x}_b)} \quad (32)$$

and the terms D_b^j and D_b^k are defined similarly.

While $dist(\mathbf{x}, \mathbf{x}_b)$ is closed to 0, the limit of D_b^i does not exist, and the value of D_b^i will change between -1 and 1, depending on the relative position between \mathbf{x} and \mathbf{x}_b . Therefore, in our motion planning for the attractor, we simply set the value of D_b^i according to the approaching velocity \mathbf{v} of the attractor:

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}_b} D_b^i := \frac{v^i}{\|\mathbf{v}\|} \quad (33)$$

where $\|\mathbf{v}\|$ denotes the Euclidean length of the vector \mathbf{v} .

2) *Attractors dynamics*: For an n -fingered robotic hand, we introduce n attractors with a dynamical system to guide the fingertips to form a feasible grasp. Within the dynamical system, the attractors are maintained on the object's surface through the constraints in the surface's normal directions. At the same time, the dynamical system defines the tangential motion of the attractors, i.e., sliding towards the position where the surface normal is opposite to the fingertip's normal.

First of all, we set the initial positions of the attractors, which are shown in Fig. 4. To keep the figures clean, a 2-fingered example of the initialization is illustrated, but actually, the grasp planning method works on robotic hands with more than 3 fingers and we have validated the planning method with a 3-fingered DoraHand and a 4-fingered Allegro Hand. With an initial palm pose \mathbf{T}_{palm_0} and hand posture \mathbf{q}_0 , n lines are drawn from the fingertips to the center of the object, i.e., the center of the object's point cloud. Then the crossing points over the surface are taken as the initial positions.

Fig. 5 shows the formation of the attractors' velocities in the tangential subspace within the object's surface. With the partial derivatives of GPIS, we set the velocity of the i^{th} attractor in the tangential direction to

$$\mathbf{v}_{tan_i} = k_{tan} \left(\frac{\partial \mathbf{n}}{\partial \mathbf{p}} \right)^\dagger (-\mathbf{n}_{fin_i} - \mathbf{n}_{att_i}) \quad (34)$$

where k_{tan} is a small gain, \mathbf{n}_{fin_i} is the normal of the i^{th} fingertip and \mathbf{n}_{att_i} is the normal of the object's surface at the current position of the attractor. The term $\frac{\partial \mathbf{n}}{\partial \mathbf{p}}$ can be calculated with the partial differentials derived from Part III-D1:

$$\frac{\partial \mathbf{n}}{\partial \mathbf{p}} = \frac{\partial \mathbf{n}}{\mathbf{p}_{max} \partial \mathbf{x}} \quad (35)$$

where \mathbf{p}_{max} denotes the normalizing scale same as that in Eq. (9).

For the velocity in the normal direction, a simple feedback law is defined to keep the attractors within the surface:

$$\mathbf{v}_{nml_i} = -\frac{k_{nml}}{\mathbf{n}_{att_i}^T \frac{\partial \mathbf{l}}{\partial \mathbf{p}}} \quad (36)$$

where k_{nml} is a small gain, and $\frac{\partial l}{\partial \mathbf{p}}$ is also calculated with the partial differentials derived from Part III-D1:

$$\frac{\partial l}{\partial \mathbf{p}} = \frac{\partial l}{\mathbf{p}_{max} \partial \mathbf{x}} \quad (37)$$

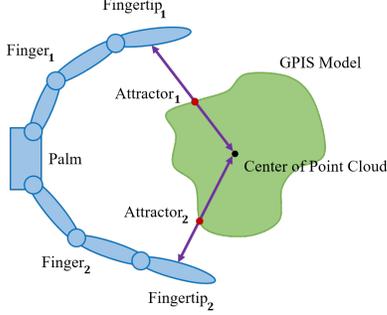


Fig. 4. Initialization of the attractors. To keep the figures clean, only 2 fingers are shown in Fig. 4, Fig. 5 and Fig. 6, but the proposed method can plan grasps with more than 3 fingers.

3) *Fingertips-attractors dynamical system*: In addition to the attractors' dynamics, the dynamical system also defines the motion of the robotic hand, which tries to guide the positions and normal directions of the fingertip towards those of the corresponding attractors.

With the task goals considering both the positions and normals, the task deviation in Eq. (16) can be determined. Fig. 6 shows the guidance provided by the attractors to the fingertips, with which the hand IK is integrated into the dynamical system.

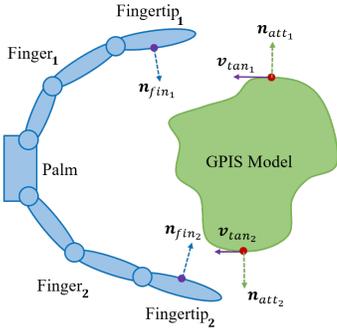


Fig. 5. The tangential velocity v_{tan_i} of the attractor is derived with the normal directions' deviation between the fingertip and the attractor (see Eq. (34)).

During grasp planning, an iterative process is performed. In each iteration, the attractors move a little step through Eq. (34) and Eq. (36). Then the robotic hand moves towards the attractors through the task-oriented IK. The iteration repeats until the positional deviation and the normal direction's deviation between the attractors and the fingertips are small enough. We limit the maximum number of iterations, and also break the iterative loop when the motions of the attractors and the robotic hand are too small, which is useful in cases when a local minimum is encountered.

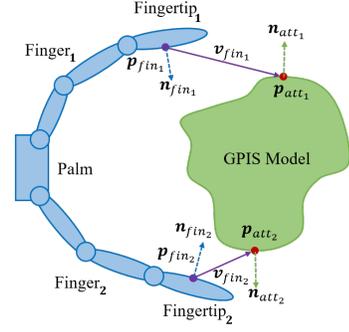


Fig. 6. In each grasp planning step, $\mathbf{p}_{att_i} - \mathbf{p}_{fin_i}$ provides the positional task deviation in Eq. (16), and the remaining part of task deviation is provided by the deviation between $-\mathbf{n}_{att_i}$ and \mathbf{n}_{fin_i} .

IV. EXPERIMENTS

We have tested our grasp planning method on some objects chosen from the YCB dataset and visualized the planned grasps with MuJoCo. The planning process and the grasp quality of the planned grasps are analyzed. Comparative experiments are conducted to investigate the advantages of our method over the baseline. We also carry out real-world experiments with some of the objects, where tiny objects' pose estimation errors and control errors may affect the grasping performance.

A. Grasp Planning Process Analysis

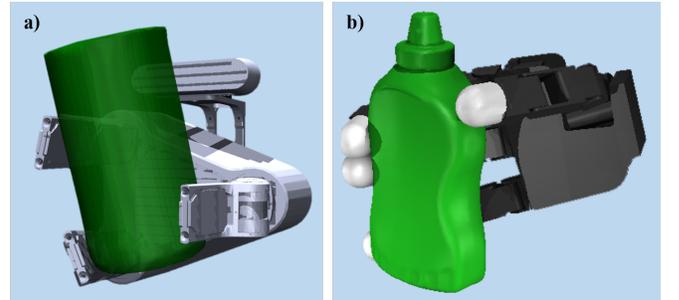


Fig. 7. a) A planned grasp for the teacan with the Dorahand; b) a planned grasp for the mustard bottle with the Allegro Hand.

Fig. 7 shows a planned grasp for the teacan with the Dorahand (Fig. 7 a), and a planned grasp for the mustard bottle with the Allegro Hand (Fig. 7 b). The modeling performance by GPIS is illustrated in Fig. 2. To explain the process of grasp planning more detailedly, we present the fingertip positions and finger joint angles within the iteration process of a single grasp planning trial. Fig. 8 shows the positions of the fingertips and the attractors, where the distances between the corresponding pairs of fingertips and attractors gradually decrease, and the attractors move within the neighbour space of the object's surface. Fig. 9 shows the finger joint angles during planning, which occasionally get into the joint limits while this situation doesn't deteriorate the solution of IK. During this situation the corresponding joint is temporarily stopped and taken place by other joints or the moving palm

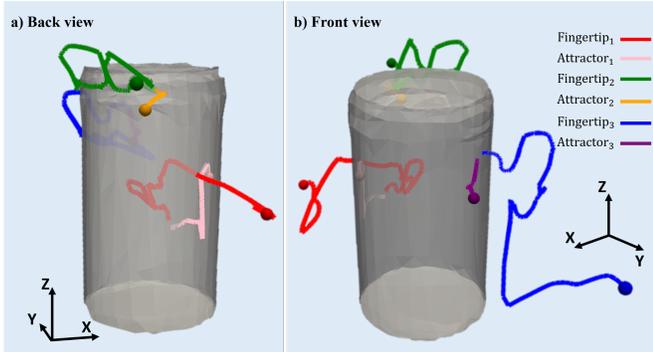


Fig. 8. The trajectories of the fingertips and attractors during grasp planning. The spheres indicate the initial positions of the fingertips or the attractors. The initial positions of the three fingertips (denoted as the red, green, and blue spheres) are initialized outside the object. After grasp planning, the positions of the fingertips overlap those of the corresponding attractors, e.g., the blue line (denoting the position of the first fingertip) intersects with the purple line (denoting the position of the first attractor).

TABLE I
PLANNING SUCCESS RATES WITH DIFFERENT ROBOTIC HANDS.

Object name	teacan	mustard bottle	banana
Dorahand	39.25%	41.5%	62.5%
Allegro Hand	20.25%	10.5%	32.75%

until the secondary task goal moves it away enough from the limiting values. Fig. 10 shows that the average deviation of the positions and normal directions decreases to a small value at the 1455th iteration step, after which the planning procedure is stopped and we obtain the feasible grasp configuration.

B. Grasp Planning on Different Objects and Robotic Hands

To investigate the feasibility of the proposed grasp planning method on different objects and robotic hands, we plan grasps for 3 objects of different shapes with both the Dorahand and the Allegro Hand. The Dorahand is a three-fingered robotic hand with 8 DoFs, and the Allegro Hand is a four-fingered hand with 16 DoFs. Grasp planning with both the Dorahand and Allegro Hand is conducted with plenty of initial palm poses, which are generated through the coordinate grid method with a spherical coordinate system. Fig. 11 (b) shows one of the initial palm poses. A spherical coordinate system, whose origin is the center of the object point cloud, is used in the coordinate grid method to sample initial palm poses for grasp planning. The azimuthal angles θ are evenly sampled from 0 to 2π (2π excluded) and the polar angles ϕ are evenly sampled from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ (both ends included). The radial distance r is fixed to the value with which the closing fingers can just reach the object and the palm poses are chosen to make the palm facing towards the object. Table I shows the planning success rates on different objects for both hands.

C. Planning Success Rates and Grasp Quality

The largest-minimum resisted wrench matrice [29] is widely used to measure the grasp quality, which finds the direction where the least disturbance wrench, i.e., force and torque,

TABLE II
PLANNING SUCCESS RATES ON DIFFERENT OBJECTS.

Object name	teacan	mustard bottle	banana
Baseline	20.5%	2%	48.25%
Proposed method	39.25%	41.5%	62.5%

TABLE III
AVERAGE GRASP QUALITY ON DIFFERENT OBJECTS WITH A COF OF 0.6.

Object name	teacan	mustard bottle	banana
Baseline	0.094578	0.004929	0.10446
Proposed	0.272625	0.201174	0.168635

is needed to break the grasp, and the value of the least disturbance wrench is taken as the grasp quality. Since our proposed method does not take this matrice into consideration directly, we investigate the grasp quality of the grasps planned by our method. We use the initial pose generating method in part IV-B and compare the qualities of the grasps planned by a joint-space-searching method, i.e., the baseline. The joint-space-searching method is similar to the auto-closing method provided by GraspIt! [51] but provides a wider variety in finger joint configurations, in which the fingers search in the whole joint space and the palm is fixed at the initial palm poses.

In our experiment, we choose the joint-space-searching method as the baseline and conduct comparative experiments on 3 objects. Fig. 3 shows that the proposed method successfully plans feasible grasps in a wider area, and Table II leads to the same conclusion. The planning success rates for some objects are low, especially for the 4-fingered Allegro Hand. The main reason is that the position's and the normal direction's deviation between the attractor and the fingertip doesn't converge to a value below the threshold, which may converge at last with more planning steps. While in some cases, the planning is trapped in a local minimum, which cannot be solved by increasing the planning steps. Table III shows that higher average grasp quality is achieved with the proposed method, even though it doesn't directly consider the grasp quality, but only encourages the fingertips to surround the object. However, the proposed grasp planning process is quite time-consuming, which takes 62.124 microseconds in average for each planning step and most of the planning ends up between 1300 and 1500 steps.

D. Grasping in Real World

The real-robot experiments are carried out on a 3-fingered Dorahand, mounted on a 6-DoF Elite EC-66 robot, as illustrated in Fig. 12. All the fingertips and finger pulps of the Dorahand are covered by planar tactile sensors, which provide real-time measurement of contact force in the normal direction.

Since small deviations may occur, e.g., modeling errors between the GPIS model and the actual object geometry, we integrate a simple touch-and-stop grasping controller, which first moves the palm to the pre-grasping pose calculated with the detected object pose and the planned grasping configuration. Then it will try to achieve the finger joint angles provided

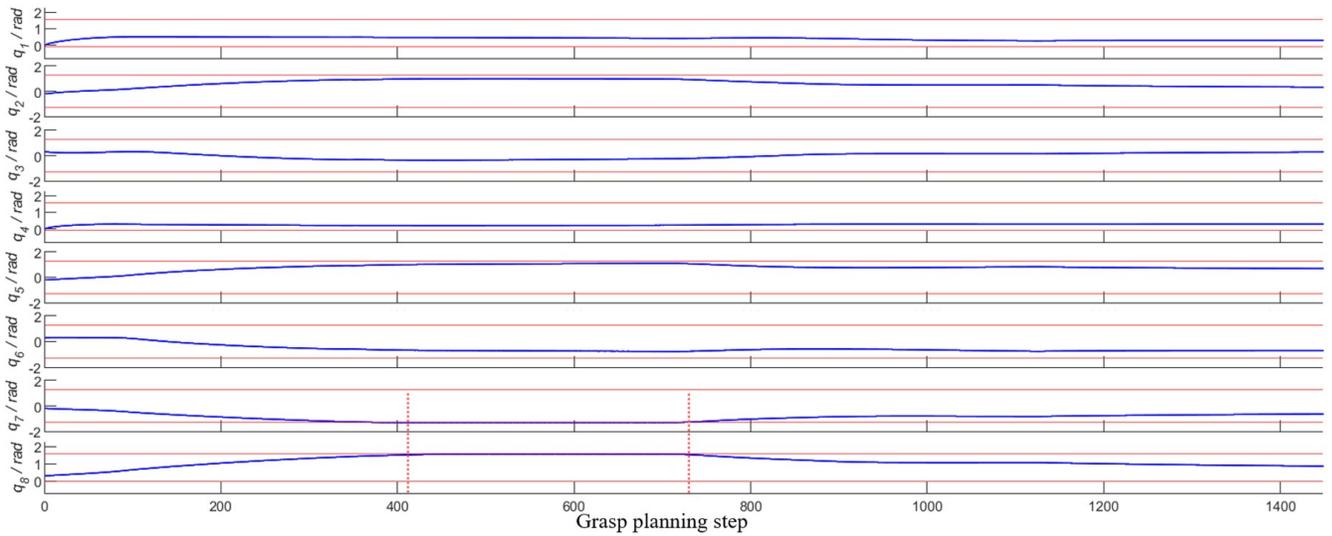


Fig. 9. Angles of the 8 finger joints during grasp planning: In each box, the blue line indicates the joint angle during the planning and the two horizontal red lines indicate the upper and lower bounds respectively. The two red vertical lines indicate the interval where the 7th and 8th joints are in joint limits. The joints are named with the same order as that in Fig. 11 (a).

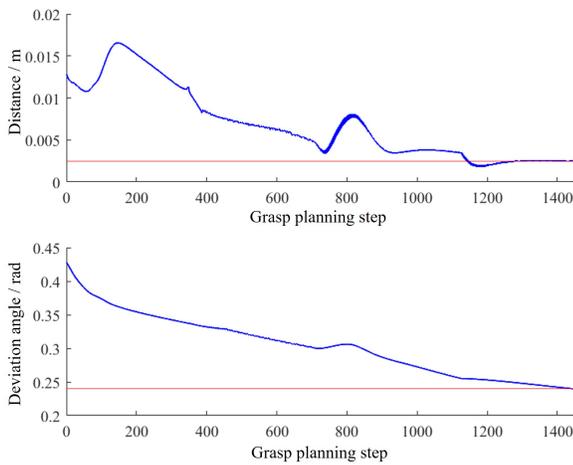


Fig. 10. The general position's and normal direction's deviation between the fingertips and the attractors: Top panel: the average distance between the fingertip and the corresponding attractor among 3 fingers. Bottom panel: the average deviation angle between the fingertip's normal and the surface normal at the location of the corresponding attractor.

by the grasping configuration until the tactile sensor detects contacts. If the fingertip does not touch the object, the angle of the endmost finger joint will be increased by 4 degrees at most. After all fingertips touch the object, the aforementioned joint angles are increased again by 4 degrees to exert enough grasping forces.

To provide an accurate pose of the targeting object, we designed a positioning plate, as shown in Fig. 13. With the four slots, the object can be precisely placed at the center of the positioning plate. Six marker points are located in a noncoplanar space, which can be reached by the end-effector of the robot arm and hence the position and orientation of the positioning plate and the object are calculated.

The grasping procedure of 3 of the evaluated objects is

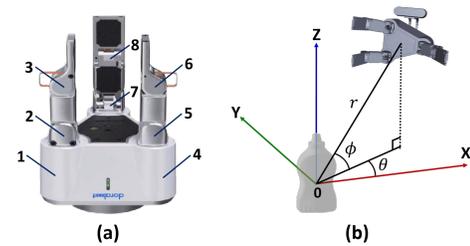


Fig. 11. (a) The joints of the DoraHand. (b) One of the initial palm poses generated through the coordinate grid method with a spherical coordinate (r, ϕ, θ) : r , ϕ , and θ denote the radial distance, polar angle, and azimuthal angle, respectively. The coordinate's origin O is set to be the center of the object surface's point cloud.

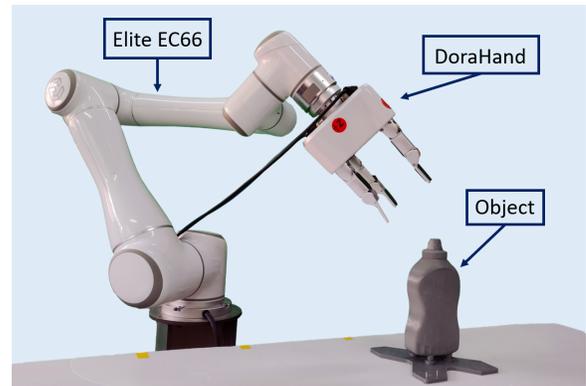


Fig. 12. Overview of the experiment platform.

shown in Fig. 14. The robotic hand first moves to the pre-grasping pose and then closes the fingers. It tries to lift the object once the grasp is established and all the chosen objects are successfully grasped except for the banana. The failure of the banana is mainly because of the unexpected change of the object pose during hand closing, which seriously deteriorates the grasp stability and may be improved by using compliance

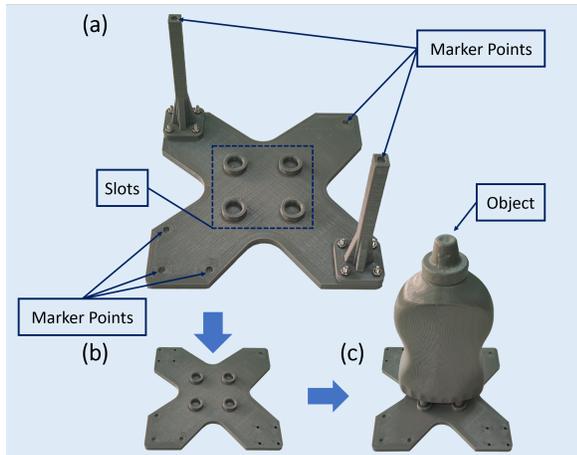


Fig. 13. A positioning plate is designed to accurately locate the targeting object. The objects are placed on the four slots. The two upholders, both with a marker point on the top, can be removed to avoid collision during grasping.

controllers with more sensitive tactile sensors.

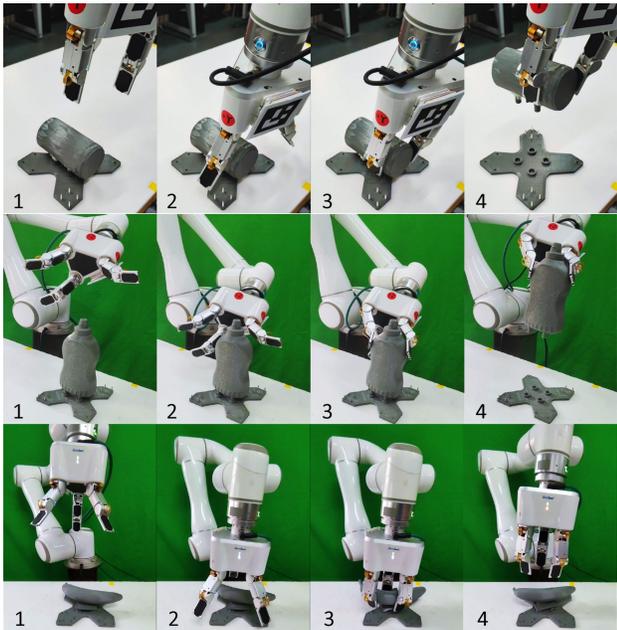


Fig. 14. Each row shows a set of grasping processes of the different objects, i.e., the teacan, mustard bottle and banana. The grasp of the banana fails because of the unexpected change of the object pose during hand closing.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a multi-fingered grasp planning method based on GPIS. We model the targeting object with GPIS and introduce some sliding points called attractors along with a dynamical system. The dynamical system utilizes the partial differentials of GPIS and constrains the attractors within the object's surface. The attractors are also guided by the dynamical system towards the positions where the surface normals are opposite to the fingertips' normals. The fingertips are guided by the dynamical system and an IK method for multi-fingered robotic hands is integrated. Experimental results

show our method works on objects of different shapes and different robotic hands and joint limits are well tackled. There is a higher probability for our method to plan a grasp with higher grasp quality. We also validated our grasp planning method with real robot experiments.

However, our method doesn't directly consider the grasp quality during planning, in terms of this issue, our current solution is to generate several initial palm poses for grasp planning. Our future work is to reasonably consider the grasp quality during the grasp planning process. Moreover, we hope to tackle the issue of uncertain object shape during the grasp planning. The geometrical uncertainty is expected to be quantified by the variance of GP and the fingertips would be encouraged to reach the positions with less uncertainty. The planning is expected to be accelerated by optimizing the code, e.g., pre-allocating memory for the matrices. It is also worth noting that the exploitation of the partial differentials is not limited to GPIS, all the GP-based object modeling methods can derive their partial differentials conveniently, and we are trying to utilize the gradient of other GP-based object modeling methods for grasp planning.

APPENDIX

A. Contents of the Gaussian Process Expression

$\mathbf{K}(\mathbf{X}, \mathbf{X})$ and \mathbf{K}_* are composed of the kernels with different inputs:

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} \mathbf{k}(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \mathbf{k}(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ \mathbf{k}(\mathbf{x}_m, \mathbf{x}_1) & \cdots & \mathbf{k}(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix} \quad (38)$$

$$\mathbf{K}_* = [\mathbf{k}(\hat{\mathbf{x}}, \mathbf{x}_1) \quad \cdots \quad \mathbf{k}(\hat{\mathbf{x}}, \mathbf{x}_m)] \quad (39)$$

where $\mathbf{k}(\mathbf{x}_a, \mathbf{x}_b)$ is the covariance function or so called kernel calculated with the a^{th} and b^{th} input of the training data, and we use the kernel from [42]:

$$\mathbf{k}(\mathbf{x}_a, \mathbf{x}_b) = \begin{bmatrix} \text{cov}(l_a, l_b) & \text{cov}(l_a, \mathbf{n}_b) \\ \text{cov}(\mathbf{n}_a, l_b) & \text{cov}(\mathbf{n}_a, \mathbf{n}_b) \end{bmatrix} \quad (40)$$

whose entries are defined as follows:

$$\text{cov}(l_a, l_b) = 2\|\mathbf{x}_a - \mathbf{x}_b\|^3 - 3\psi\|\mathbf{x}_a - \mathbf{x}_b\|^2 + \psi^3 \quad (41)$$

$$\text{cov}(l_a, \mathbf{n}_b) = \frac{\partial}{\partial \mathbf{x}_b} \text{cov}(l_a, l_b) \in \mathbb{R}^{1 \times 3} \quad (42)$$

$$\text{cov}(\mathbf{n}_a, l_b) = \frac{\partial^T}{\partial \mathbf{x}_a} \text{cov}(l_a, l_b) \in \mathbb{R}^{3 \times 1} \quad (43)$$

$$\text{cov}(\mathbf{n}_a, \mathbf{n}_b) = \frac{\partial^2}{\partial \mathbf{x}_a \partial \mathbf{x}_b} \text{cov}(l_a, l_b) \in \mathbb{R}^{3 \times 3} \quad (44)$$

where ψ is the Euclidean distance between the farthest pair of points in the training inputs \mathbf{X} .

The entries of Eq. (43) are as follows:

$$[\text{cov}(\mathbf{n}_a, l_b)]_i = 6(x_a^i - x_b^i) (\text{dist}(\mathbf{x}_a, \mathbf{x}_b) - \psi) \quad (45)$$

where $[\ast]_i$ denotes the i^{th} entry in the column vector $[\ast]$, $\text{dist}(\ast)$ denotes the Euclidean distance, and x^k denotes the k^{th} ($k = 1, 2, 3$) dimensional component of a 3-D point.

The entries of Eq. (42) can be determined with

$$\text{cov}(l_a, \mathbf{n}_b) = -\text{cov}(\mathbf{n}_a, l_b)^T \quad (46)$$

The entries of Eq. (44) are as follows:

$$\begin{aligned} & [\text{cov}(\mathbf{n}_a, \mathbf{n}_b)]_{i,j} \\ &= 6 \frac{\partial \mathbf{x}_b^i}{\partial \mathbf{x}_b^j} (\psi - \text{dist}(\mathbf{x}_a, \mathbf{x}_b)) - 6 \frac{(x_a^i - x_b^i)(x_a^j - x_b^j)}{\text{dist}(\mathbf{x}_a, \mathbf{x}_b)} \end{aligned} \quad (47)$$

where $[\ast]_{i,j}$ denotes the entry at the i^{th} row and the j^{th} column in the matrix $[\ast]$, and the partial differential is calculated as follows:

$$\frac{\partial \mathbf{x}_b^i}{\partial \mathbf{x}_b^j} = \begin{cases} 1, & \text{when } i = j, \\ 0, & \text{when } i \neq j. \end{cases} \quad (48)$$

B. Definition of the Hand Kinematic Jacobian

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{J}_{palm_1} & \mathbf{J}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{J}_{palm_2} & \mathbf{0} & \mathbf{J}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{J}_{palm_i} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_n \end{bmatrix} \quad (49)$$

in which \mathbf{J}_i is the i^{th} finger's Jacobian and \mathbf{J}_{palm_i} describes the contribution from palm twist to the twist of the i^{th} fingertip. According to our definition of twist $\mathbf{V} = [\mathbf{v}^T \boldsymbol{\omega}^T]^T$, where \mathbf{v} is the linear velocity and $\boldsymbol{\omega}$ is the angular velocity, \mathbf{J}_{palm_i} is written as

$$\mathbf{J}_{palm_i} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -[{}^{palm}\mathbf{p}_i]_{\times} \\ \mathbf{0} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (50)$$

where ${}^{palm}\mathbf{p}_i$ represents the position of the i^{th} fingertip from the origin of the palm frame, expressed in the static world frame, and $[{}^{palm}\mathbf{p}_i]_{\times}$ is its corresponding skew-symmetric matrix.

C. Definition of the Task-oriented Jacobian

$$\mathbf{J}_{task} = \begin{bmatrix} \mathbf{J}_{task_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{task_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_{task_n} \end{bmatrix} \quad (51)$$

where \mathbf{J}_{task_i} is written as

$$\mathbf{J}_{task_i} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & -\mathbf{n}_i^{*T} [{}^{\mathbf{n}_i}\mathbf{n}_i]_{\times} \end{bmatrix} \quad (52)$$

REFERENCES

- [1] Z. Tian, *et al.*, "Research on two-stage grasping planning method for multifingered dexterous hand," *Robotic Intelligence and Automation*, 2023.
- [2] M. Folgheraiter and G. Gini, "Blackfingers an artificial hand that copies human hand in structure, size, and function," *Proc. IEEE Humanoids*, p. 4, 2000.
- [3] ShadowRobot. (2005) Shadowrobot dexterous hand. [Online]. Available: <https://www.shadowrobot.com/products/dexterous-hand/>.
- [4] W. Yan, *et al.*, "Precision grasp planning for multi-finger hand to grasp unknown objects," *Robotica*, vol. 37, no. 8, pp. 1415–1437, 2019.
- [5] S. Qiu and M. R. Kermani, "Inverse kinematics of high dimensional robotic arm-hand systems for precision grasping," *J. Intell. Robot. Syst.*, vol. 101, no. 4, p. 70, 2021.
- [6] Y. Chen, *et al.*, "Cooperative distributed model predictive control for robot in-hand manipulation," *Robotic Intelligence and Automation*, vol. 43, no. 1, pp. 65–74, 2023.
- [7] Y. Fan, *et al.*, "Optimization model for planning precision grasps with multi-fingered hands," in *Proc. 2019 IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2019, pp. 1548–1554.
- [8] K. Kleebberger, *et al.*, "A survey on learning-based robotic grasping," *Curr. Robot. Rep.*, vol. 1, pp. 239–249, 2020.
- [9] R. Miao, *et al.*, "Long-term robot manipulation task planning with scene graph and semantic knowledge," *Robotic Intelligence and Automation*, vol. 43, no. 1, pp. 12–22, 2023.
- [10] F. Stulp, *et al.*, "Learning motion primitive goals for robust manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 325–331.
- [11] D. Zhang, *et al.*, "From teleoperation to autonomous robot-assisted microsurgery: A survey," *Machine Intelligence Research*, vol. 19, no. 4, pp. 288–306, 2022.
- [12] H. Dai, *et al.*, "A gripper-like exoskeleton design for robot grasping demonstration," in *Actuators*, vol. 12, no. 1. MDPI, 2023, p. 39.
- [13] Z. Lu, *et al.*, "Visual-tactile robot grasping based on human skill learning from demonstrations using a wearable parallel hand exoskeleton," *IEEE Robot. Autom. Lett.*, 2023.
- [14] K. Fang, *et al.*, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 202–216, 2020.
- [15] H.-S. Fang, *et al.*, "Grasnet-1billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.
- [16] S. Lin, *et al.*, "E2ek: End-to-end regression network based on keypoint for 6d pose estimation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6526–6533, 2022.
- [17] L. Wang, *et al.*, "Mvcontrast: Unsupervised pretraining for multi-view 3d object recognition," *Machine Intelligence Research*, pp. 1–12, 2023.
- [18] S. Lin, *et al.*, "Robot grasping based on object shape approximation and lightgbm," *Multimed. Tools. Appl.*, pp. 1–17, 2023.
- [19] Q. She, *et al.*, "Learning high-dof reaching-and-grasping via dynamic representation of gripper-object interaction," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–14, 2022.
- [20] Z. Liu and R. D. Howe, "Beyond coulomb: Stochastic friction models for practical grasping and manipulation," *IEEE Robot. Autom. Lett.*, 2023.
- [21] Y. Su, *et al.*, "Enhanced kinematic model for dexterous manipulation with an underactuated hand," in *Proc. 2013 IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2013, pp. 2493–2499.
- [22] M. Li, *et al.*, "Dexterous grasping under shape uncertainty," *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.
- [23] A. T. Miller, *et al.*, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1824–1829.
- [24] A. T. Miller and H. I. Christensen, "Implementation of multi-rigid-body dynamics within a robotic grasping simulator," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 2262–2268.
- [25] Y.-H. Liu, *et al.*, "A complete and efficient algorithm for searching 3-d form-closure grasps in the discrete domain," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 805–816, 2004.
- [26] W. Wan, *et al.*, "Planning grasps with suction cups and parallel grippers using superimposed segmentation of object meshes," *IEEE Trans. Robot.*, vol. 37, no. 1, pp. 166–184, 2020.
- [27] J. P. C. de Souza, *et al.*, "Robotic grasping: from wrench space heuristics to deep learning policies," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102176, 2021.
- [28] Y. Zheng, *et al.*, "A fast n-dimensional ray-shooting algorithm for grasping force optimization," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1300–1305.
- [29] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, no. 1, pp. 65–88, 2015.
- [30] Y. Su, *et al.*, "Robust grasping for an under-actuated anthropomorphic hand under object position uncertainty," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 719–725.
- [31] G. I. Boutselis, *et al.*, "Task specific robust grasping for multifingered robot hands," in *Proc. 2014 IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2014, pp. 858–863.

- [32] R. Pelosof, *et al.*, “An svm learning approach to robotic grasping,” in *Proc. 2004 IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4. IEEE, 2004, pp. 3512–3518.
- [33] U. Hillenbrand and M. A. Roa, “Transferring functional grasps through contact warping and local replanning,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2963–2970.
- [34] L.-H. Kong, *et al.*, “Dynamic movement primitives based robot skills learning,” *Machine Intelligence Research*, vol. 20, no. 3, pp. 396–407, 2023.
- [35] Y. Fan and M. Tomizuka, “Efficient grasp planning and execution with multifingered hands by surface fitting,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3995–4002, 2019.
- [36] S. Dragiev, *et al.*, “Gaussian process implicit surfaces for shape estimation and grasping,” in *Proc. 2011 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2011, pp. 2845–2850.
- [37] M. Nieuwenhuisen, *et al.*, “Shape-primitive based object recognition and grasping,” in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–5.
- [38] Y. Lin, *et al.*, “Using synthetic data and deep networks to recognize primitive shapes for object grasping,” in *Proc. 2020 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2020, pp. 10 494–10 501.
- [39] G. Turk and J. F. O’Brien, “Variational implicit surfaces,” Georgia Institute of Technology, Tech. Rep., 1999.
- [40] O. Williams and A. Fitzgibbon, “Gaussian process implicit surfaces,” in *Gaussian Processes in Practice*, 2006.
- [41] E. Solak, *et al.*, “Derivative observations in gaussian process models of dynamic systems,” *Adv. Neural Inf. Process. Syst.*, vol. 15, 2002.
- [42] N. Sommer, *et al.*, “Bimanual compliant tactile exploration for grasping unknown objects,” in *Proc. 2014 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2014, pp. 6400–6407.
- [43] Z. Gao, *et al.*, “Inverse kinematics and workspace analysis of the metamorphic hand,” *Proc. Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 229, no. 5, pp. 965–975, 2015.
- [44] S. Starke, *et al.*, “Evolutionary multi-objective inverse kinematics on highly articulated and humanoid robots,” in *Proc. 2017 IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2017, pp. 6959–6966.
- [45] H. Liu, *et al.*, “Multi-fingered tactile servoing for grasping adjustment under partial observation,” in *Proc. 2022 IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2022, pp. 7781–7788.
- [46] V. R. Garate, *et al.*, “On the common-mode and configuration-dependent stiffness control of multiple degrees of freedom hands,” in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 113–120.
- [47] M. S. Allahkaram, *et al.*, “Closed-form inverse kinematics equations of a robotic finger mechanism,” in *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*. IEEE, 2021, pp. 157–162.
- [48] C. Bensalah, *et al.*, “A new finger inverse kinematics method for an anthropomorphic hand,” in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 1314–1319.
- [49] S. Hong, *et al.*, “Realization of the fast real-time hierarchical inverse kinematics for dexterous full body manipulation,” in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2018, pp. 166–171.
- [50] S. Qiu and M. R. Kermani, “Arm-hand systems as hybrid parallel-serial systems: A novel inverse kinematics solution,” in *Proc. 2021 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2021, pp. 6401–6407.
- [51] A. T. Miller and P. K. Allen, “Graspit! a versatile simulator for robotic grasping,” *IEEE Robot. Autom. Mag.*, vol. 11, no. 4, pp. 110–122, 2004.



Hanzhong Liu received the B.Eng. degree in Automation and the M.Sc. degree in Control Science and Engineering from South China University of Technology, Guangzhou, China, in 2020 and in 2023. His research interests include robotics, human robot interaction, grasping, and machine learning.



Suhas Kadalagere Sampath received an M.Sc. degree in Robotics from the University of Bristol, Bristol, United Kingdom, in 2020. He is currently pursuing a Ph.D. in Robotics at the Bristol Robotics Laboratory (BRL), associated with the University of the West of England (UWE), Bristol, United Kingdom. His research interests include robotics, teleoperation, grasping, and manipulation.



Ning Wang (Member, IEEE) is a Senior Lecturer in Robotics at the Bristol Robotics Laboratory, University of the West of England, United Kingdom. She received the M.Phil. and Ph.D. degrees in electronics engineering from the Department of Electronics Engineering, The Chinese University of Hong Kong, Hong Kong, in 2007 and 2011, respectively. Ning has rich project experience, she has been key member of EU FP7 Project ROBOT-ERA, EU Regional Development Funded Project ASTUTE 2020 and industrial projects with UK companies. She has been awarded several awards including IET premium award for best paper 2022, best paper award of ICIRA'15, best student paper award nomination of ISCSLP'10, and award of merit of 2008 IEEE Signal Processing Postgraduate Forum, etc. Her research interests lie in signal processing, intelligent data analysis, human-robot interaction and autonomous driving.



Chenguang Yang (Fellow, IEEE) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xi'an, China, in 2005, the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010, and postdoctoral training in human robotics from the Imperial College London, London, U.K, from 2009 to 2010. He was awarded UK EPSRC UKRI Innovation Fellowship and individual EU Marie Curie International Incoming Fellowship. As the lead author, he won the IEEE Transactions on Robotics Best Paper Award (2012) and IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award (2022). He is the Corresponding Co-Chair of IEEE Technical Committee on Collaborative Automation for Flexible Manufacturing. His research interest lies in human robot interaction and intelligent system design.