Autonomous Robots (2020) 44:773–790
https://doi.org/10.1007/s10514-019-09896-7

Adaptive force and velocity control based on intrinsic contact sensing during surface exploration of dynamic objects

Teng Sun[1] · Hongbin Liu[1]

Received: 2 July 2019 / Accepted: 19 December 2019 / Published online: 21 January 2020

Abstract
Haptic exploration is a process of using haptic feedback to interact and perceive an unknown object. It is an essential approach to understand the physical and geometrical properties of the object. While numerous research has been carried out for haptic exploration on static objects, haptic exploration on objects with dynamic movements has not been reported. It is due to the significant challenges to achieve robust force and velocity control when the object is nonstationary. In this work, a novel adaptive force and velocity control algorithm based on intrinsic contact sensing (ICS) for haptic surface exploration of dynamic objects is presented. A fuzzy-logic control framework making use of the information obtained from ICS has been developed. To validate the proposed control algorithm, extensive surface exploration experiments have been carried out on objects with different surface properties, geometries, stiffness, and concave or convex patterns. The validation results demonstrate the high accuracy and robustness of the proposed algorithm using different experimental platforms.

Keywords Haptic · Autonomous control · Fuzzy logic

1 Introduction

Haptic exploration of the surface is a fundamental tool for humans to understand the surface properties of unknown objects with only tactile sensing (Lederman and Klatzky 1993, 2009), especially in reduced visibility scenarios like underwater, smoky disaster environment and inside tissues during surgery. It has many useful applications in areas like robot surgery (Amirabdollahian et al. 2017), shape recovery (Rosales et al. 2018), heritage recording (Jamil et al. 2018), dexterous robot hand (Kappassov et al. 2015). Unlike computer vision which has been progressing tremendously in recent years, haptic sensing and interaction remain as technically challenging due to the relatively limited development of contact sensing hardware and intelligent control algorithms (Tegin and Wikander 2005).

Due to the direct interaction with the object, significant hurdles in control and sensing are presented for haptic exploration (Okamura and Cutkosky 2001). With the emergence of new force/tactile sensing hardware and data processing methods, several new haptic exploration algorithms on unknown objects and unstructured environment have been developed. Lepora et al. have been working on the development of the optical tactile sensors with bio-mimetic morphologies (Ward-Cherrier et al. 2018), and they have used thus sensors for robust contour following with deep learning method (Lepora et al. 2019). Yin et al. (2018) developed a bio-inspired tactile sensor skin for measuring dynamic shear force and vibration. Fishel and Loeb (2012) and Xu et al. (2013) proposed a Bayesian exploration method for identification of textures where they choose the exploratory movements that would yield the most useful information. Hellman et al. (2018) presented a reinforcement learning method for contour following with haptic feedback. Yang and Lepora (2017) tried to combine vision for object exploration for edge estimation. Sommer and Billard (2016) presented a Multi-contact haptic exploration method manly for grasping tasks. Su et al. (2012) controlled the movement of the robot arm in the contact normal direction with haptic feedback to

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-019-09896-7) contains supplementary material, which is available to authorized users.

✉ Hongbin Liu
hongbin.liu@kcl.ac.uk

Teng Sun
teng.sun@kcl.ac.uk

[1] Haptic Mechatronics and Medical Robotics (HaMMeR) Lab, King's College London, London, UK

characterize the compliance of the pressed object. Similarly, Sornkarn and Nanayakkara (2017) used a soft robotic probe to measure object stiffness with haptic control.

Konstantinova et al. (2014) presented that humans use different force or velocity control strategies to detect abnormality inside soft objects, and the performance of the perception can be enhanced by controlling the exploration speed. And Okamura (2000) showed that it is important to maintain a proper normal force level during sliding exploration procedures to guarantee the force sensing accuracy, considering the practical limitation of the sensor hardware. Furthermore, if the explore velocity can be controlled, more information of the explored object such as surface viscosity (Liu et al. 2012), break-away ratio (Song et al. 2014) can be revealed. These all showed the paramount importance of the good control of contact forces and scanning velocity during haptic exploration.

However, to date, to achieve the force and explore velocity control during haptic exploration is still an open challenge, and an effective method for doing so has not been found in literature. In most of the previous works, the exploration procedure has been carried out by scanning over a relatively flat surface of the object or by pressing. Some researchers considered force control during 1 DoF compressing motion (Su et al. 2012; Sornkarn and Nanayakkara 2017). A few studies have been done on the sliding exploration but only limited to static objects with simple geometry (like flat surface), avoiding the problem of feedback force and velocity control (Fishel and Loeb 2012; Hellman et al. 2018). Similar to the work proposed in this study, Lepora et al. (2017) has proposed an exploratory tactile servoing method where the next exploration position was decided with a Bayesian perceptual framework, but the force was not actively controlled in their study and the objects were static. Li et al. (2013) introduced a tactile servoing control framework adapting a PID controller, which can be used to realize a specific tactile interaction pattern. Yang et al. (2013) presented a learning algorithm for simultaneous force control and haptic identification, where they used complicated models instead of real force sensor, and the algorithm was only tested in simulation. Okamura and Cutkosky (2001) used a 2 DoFs robot finger to detect features with haptic exploration, where they presented that the force was kept low; however, no detailed results were shown. Therefore, to the best of the surveyed knowledge, the robust and accurate normal force and explore velocity control when the object is dynamically moving during surface exploration has not been reported.

To overcome the challenges, continuous study has been carried out on enhancing haptic sensing capability to realise adaptive contact force and velocity control. Adapting ICS with 6-axis force-torque measurement (Bicchi et al. 1993), a real-time contact sensing algorithm has been developed to provide instantaneous contact location, normal force and tangential force (Liu et al. 2012, 2015). With the precise contact information, an efficient algorithm has been designed based on normal force conditions to control the robot finger to explore the surface while maintaining the contact normal force (Back et al. 2015). However, there are three major limitations in the previous algorithms to prevent it from being applicable to dynamic objects. First, previous control method is prone to induce large force control error when the surface has large curvature changes or uneven. Second, the controller has a fixed set of parameters and cannot be adapted to different objects to achieve similar good performance. Third, the velocity of the finger sliding is not controllable.

In this study, the following contributions have been made: (1) A robust and adaptive normal force and explore (scanning) velocity control algorithm was developed based on haptic information during surface exploration. It was evaluated using a self-fabricated contact sensing finger on non-stationary objects. Objects with variant surface physical and geometry properties were tested and results show that the normal force control method can achieve an average mean square error (MSE) less than 0.006 and standard deviation (SD) less than 0.07 N. Also, the explore velocity can be controlled with an average Δv of 0.48 mm/s and SD of 2.07 mm/s. (2) A fuzzy-logic control framework was created that can increase the flexibility of the control algorithm, especially when the object is non-stationary and its properties are unstructured. The control rules of the fuzzy method can be changed adaptively corresponding to different objects being explored by shifting the membership boundaries. (3) The control algorithm can be applied to devices with different kinematic configurations that have different workspace and force ranges. The algorithm was validated successfully with a 6 DoFs robot arm (with a contact sensing tip mounted) using objects with different stiffness and geometries. (4) Even though the type of contact was point contact in the tests, it also works for surface contact (using tactile array sensor) if contact locations can be obtained in real-time.

2 Experimental platform

Three platforms have been created to validate the proposed algorithm, which are denoted as platform-a, b and c. The designed experimental platform-a consists of two main parts: one is the self-fabricated contact sensing finger, and the other is the Universal Robot (UR)-3 robot arm. The contact sensing finger consists of two joints, one fingertip and the base connecting the UR-3 robot arm, as shown in Fig. 1. Each joint consists of a DC motor (Maxon DC motor RE10 with encoder) and a bevel-gear box (reduction: 1:1). The Maxon control panel (POS2 24/2) is used to control the position or velocity of the motor. The maximal frequency of the position or velocity control is 1 kHz, and such high frequency enables

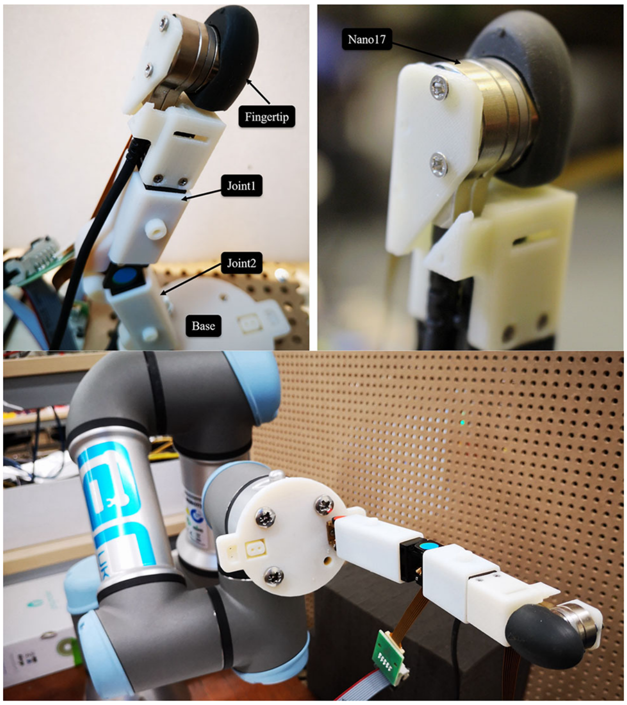


Fig. 1 The contacting sensing finger (platform-a), which consists of 2 joints and a fingertip, is shown in the top pictures. The joints consist of 3D printed links and metal connection parts, and each has a Maxon motor and bevel gearbox inside. An ATI Nano17 6-axis force-torque sensor is embedded inside the fingertip and covered with a polyurethane rubber layer. It has a 3D printed base that can be mounted on the UR-3 robot arm (bottom picture) as platform-b. The robot arm can rotate to any desired angle to allow the finger to explore more areas

the high-level (trajectory generation and decision making) control to be accurate and rapid. For motor control, a multi-thread method is used to eliminate the asynchronization when controlling multiple motors, which ensures the accuracy of the position control and is beneficial for the control of more joints or fingers. The high precision of the motor control panel and the self-locking mechanism on the joint (the baffles at 90 degrees) ensure the repeatability of the exploration, making the online multi-time haptic exploration possible. Experimental tests were also carried out to record the position deviation of the joints after several explorations and input the error into the control system as the compensation to reduce the error caused by mechanical fault and friction between the components.

For fabricating the fingertip, the inner core was firstly 3D-printed with plastic and then embedded into a layer of 2 mm rubber. The Poly 74-30 liquid rubber from Polytek^{TM} was used to construct the polyurethane rubber layer, its hardness is 30A Shore, and young 's Modulus is 100psi. The whole fingertip (3D printed core and rubber layer) has very high hardness and is close to rigid body; therefore there was no deformation within the range of the forces used in experiments, and no effect of hysteresis. The study of fingertip materials that are softer and more flexible will be future work. Inside the fingertip, an ATI Nano17 force-torque sensor (resolution: 1/160 N) was installed. It is a six-axis sensor providing force and torque data along its x, y, z-axis, and the data can be updated at a frequency of 7200 Hz with a DAQ data acquisition box.

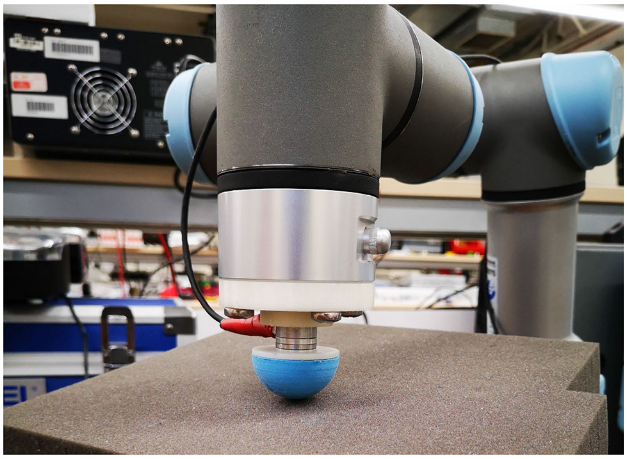


Fig. 2 The haptic exploration platform-c involving a UR-3 robot arm and a contact sensing tip. The tip has a Nano17 force-torque sensor and a self-fabricated cap made of silicon. The tip is mounted to the end-effector of the arm using a 3D printed connection

The software was developed with ROS (robot operating system, www.ros.org) using an Ubuntu PC (CPU: i7-6700 @3.4GHz and 16G RAM). The PCIe-6320 data acquisition card (up to 16 kHz) is used to obtain sensor data, and then ROS programs are used to communicate with the card to get and process the data. In practical, the tactile sensing toolbox (TST) proposed in Ciotti et al. (2019) was used for sensor data processing and contact location estimation. Including the other modules like the control and calculation, the loop frequency of the ROS is 100 Hz, meaning it updates the sensor data and finish all the necessary computations every 1/100 s. The finger is installed on a 3D printed base, which can be mounted on the UR-3 robot arm. This finger and robot arm combined platform is denoted as the platform-b. The base can be rotated to any angle through the robot arm to achieve the exploration on more surfaces, as shown in Fig. 1 bottom. For the next steps, more fingers will be mounted on the base to realize the larger area's exploration and achieve more actions other than surface exploration. These together will help to lay a solid hardware foundation for the active exploration and dexterous manipulation system.

Moreover, to evaluate the proposed algorithm with larger working space and force ranges, the UR-3 robot arm was used for haptic surface exploration as platform-c. The Nano17 sensor was covered with a silicone cover and fixed onto the robot arm end-effector to provide real-time force-torque data for the control algorithm, shown in Fig. 2. Since the workspace of the finger is limited and the force generated by it is relatively small, the robot arm with more flexibilities helps to validate the proposed algorithm in larger workspace and force control range. On the contrary, the contacting sensing finger can realize finer force and velocity control due to the high frequency and accuracy of the motors, which is a good tool to verify the robustness of the control algorithm in the low target range. More will be detailed in the experiment section.

3 Control algorithm

With the experimental platform, an adaptive real-time normal force and explore velocity control algorithm has been developed. A fuzzy-logic framework using the haptic information was designed for haptic exploration on objects with dynamic movements. In this section, the overall control process will be shown first, and then the detailed introduction of the main components constituting the control algorithm will be presented.

3.1 Overall control process

The overall control process of the proposed algorithm is shown in Fig. 3 and Tables 1, 2 show the definition of the variables. The control algorithm is a feedback control system, and the input of it has two parts: force and torque values from the Nano17 force-torque sensor, each with three components ($[f_x, f_y, f_z)]^T$ and $[m_x, m_y, m_z]^T$), and the position information from the photoelectric encoders inside motors (E1, E2). The control process starts when the fingertip is contacting the object, and to determine that state, a threshold function is used, i.e. it is determined that the finger is contacting the object when the force obtained from the force sensor is larger than a pre-decided value ϑ_n. With this method, at the beginning, the finger is controlled to move close to the surface of the object, and the contact state is confirmed when $sqrt(f_x^2 + f_y^2 + f_z^2) > \vartheta_n$. In practice, this action is similar to humans' attempt to feel the object with only touch. With the force and torque values obtained from the force-torque sensor, the contact point position in fingertip frame $\mathbf{P}_c$ can be obtained through methods presented in Liu et al. (2015) and Ciotti et al. (2019). With (E1, E2) from motor encoders, the contact location in the base frame $\mathbf{P}_{now}$ is obtained through forward kinematics.

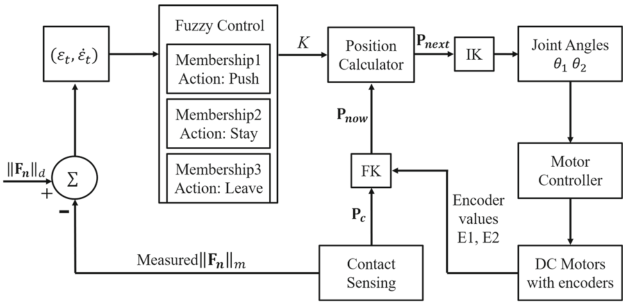


Fig. 3 The overall flow of the control algorithm, which is a closed-loop control system. The output K is obtained after the inputs $(\varepsilon_t, \dot{\varepsilon}_t)$ have been processed, and the inputs are updated with the motor movement. The overall loop rate is 100 Hz using ROS

Table 1 Parameter notation for control system

Name	Definition
ε_t	Force error between measured and target, N
$\dot{\varepsilon}_t$	Change rate of error, N/s
$\|\mathbf{F}_N\|_d$	Desired normal force, abs value, N
$\|\mathbf{F}_N\|_m$	Measured normal force, abs value, N
K	Fuzzy control output, a coefficient
$\mathbf{P}_{now}$	Coordinates of current position in base frame
$\mathbf{P}_c$	Coordinates of contact point in fingertip frame
$\mathbf{P}_{next}$	Coordinates of next position in base frame

Table 2 Parameter calculations

Notation	Calculation
Frames	Base frame $\{O\}(X_O, Y_O)$, fingertip frame $\{E\}(Y_E, Z_E)$
Sum of joint angles	$\theta_{12} = \theta_1 + \theta_2$
Angle of contact point in fingertip frame	$\theta_c = arctan2(z_e, y_e)$
Angle of the normal force in base frame	$\theta_N = \theta_{12} - \theta_c$
Correcting force	$\mathbf{F}'_N = abs(K(\mathbf{F}_N - \mathbf{F}_d))$
Virtual tangential force	$\mathbf{F}'_T = K_P \mathbf{F}_T, (K_P = -1)$
Angle between correcting and guiding force	$\beta' = arctan2(-\|\mathbf{F}'_T\|, K\|\mathbf{F}'_N\|/\|K\|)$
Angle of guiding force in base frame	$\alpha = \theta_N + \beta'$
Next position to reach	$\mathbf{P}_{next} = \mathbf{P}_{now} + S_v \cdot [cos\alpha, sin\alpha]^T$

With $\mathbf{P}_{now}$ obtained, the next is to decide the new location on the surface for the fingertip to reach while keeping the normal force as desired. The resultant normal force and tangential force are obtained simultaneously with the contact position through the TST. The control target $\mathbf{F}_d$ is then subtracted from the measured normal force to get the error ε_t and ε_t is divided by the cycling time to get the changing rate of the error: $\dot{\varepsilon}_t = \varepsilon_t/dt$. And $(\varepsilon_t, \dot{\varepsilon}_t)$ are the input of the fuzzy control method. The output of the fuzzy control method is a coefficient K, which is used to generate a correcting force, helping the finger maintain the normal force. In detail, each K is used to provide a correcting force vector that leads the finger to move either far away or closer to the object surface along the normal direction of the object. Then a virtual tangential force $\mathbf{F}'_T$ is obtained by taking the opposite direction of the measure friction force $\mathbf{F}_T$, to make the finger to move forward along the surface of the object. These two forces form a total force vector that can be used to decide the next explore location. As shown in Table 2, K is first used to decide the magnitude of the correction force ($\mathbf{F}'_N$), and the angle α of the total force in frame {O} is calculated after getting the required angles (θ_N and β'). After that, a S_v parameter is used to determine the size of the increment for the next location ($\mathbf{P}_{next}$), whose unit is mm. It can be obtained through experiments or controlled with closed-loop control.

With $\mathbf{P}_{next}$ been determined, the control signals for motors are generated through inverse kinematics of the platform. The motors are then controlled by motor controllers to rotate to desired positions. The force changes induced by the movement of the object or local geometric changes will increase the force error, and the control algorithm will make adjustments accordingly as quickly as possible within its capability. This control loop keeps cycling until the finger is not contacting the object or any one of the finger joints has reached the predefined limit. After that, the haptic surface exploration on an area of the contacting object is completed subsequently.

3.2 Contact location estimation

In the control algorithm, there are two main coordinate frames, the base {O} and ellipsoid fingertip frame {E}, and the control method is based in the base frame. The control goal is to determine the next exploration location and maintain the magnitude of the normal force simultaneously. To realize this goal, the current contact point in the base frame $\mathbf{P}_{now}$ needs to be calculated firstly. Let the coordinates of the contact point to be $\mathbf{P}_c^E = [x_e, y_e, z_e]^T$ in frame {E}, and $\mathbf{P}_{now} = [x_o, y_o, z_o]^T$ in frame {O}.

It should be noted that the fingertip material used in the experimental platform is very stiff and the effect of hysteresis can be ignored. And to verify this, experiments were carried to evaluate the force-deformation relationship when loading and releasing forces on the fingertip. As a result, the force reached 5 N when the deformation was less than 0.5 mm; thus, it was reasonable to ignore the deformation of the fingertip in this study. Therefore, the equilibrium equation in Liu et al. (2015) has been simplified to (1), where $a = b = 15$, $c = 9.5$ are the dimensions of the ellipsoid fingertip, S is the function for the ellipsoid, and k is a scaling factor. As future work, softer fingertips will be fabricated to estimate the surface haptic exploration on objects with different stiffness, which will be of great help in the research of flexible and soft manipulators.

$$g(\mathbf{x}) = \begin{bmatrix} k\frac{\partial S}{\partial x_e} - f_y z_e + f_z y_e - m_x \\ k\frac{\partial S}{\partial y_e} - f_z x_e + f_x z_e - m_y \\ k\frac{\partial S}{\partial z_e} - f_y z_e + f_y x_e - m_z \\ \frac{x_e^2}{a^2} + \frac{y_e^2}{b^2} + \frac{z_e^2}{c^2} - 1 \end{bmatrix} = 0 \tag{1}$$

The contact information $\mathbf{x} = [\mathbf{P}_c^E, k]^T$ can be obtained by identifying the roots of $g(\mathbf{x}) = 0$. The Levenberg–Marquardt (LM) method, which is a gradient-based iterative method for non-linear parameter estimation, was used to get the solution iteratively. After $\mathbf{P}_c^E$ has been acquired, with joint angles measured by motor encoders, $\mathbf{P}_{now}$ can be obtained through forward kinematics:

$$\mathbf{P}_{now} = \mathbf{P}_c^E \cdot [\mathbf{T}_1\mathbf{T}_2\mathbf{T}_3] \cdot \mathbf{R}_1 \tag{2}$$

$$\mathbf{T}_1 = \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & cos\theta_1 l_1 \\ sin\theta_1 & cos\theta_1 & 0 & sin\theta_1 l_1 + l_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\mathbf{T}_2 = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & cos\theta_2 l_2 \\ sin\theta_2 & cos\theta_2 & 0 & sin\theta_2 l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{T}_3 = \begin{bmatrix} 1 & 0 & 0 & y_e \\ 0 & 1 & 0 & -(l_3 + z_e) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$\mathbf{R}_1 = [0, 0, 0, 1]^T \tag{6}$$

3.3 Next position determination

With the obtained current contact position in the base frame, the next position was derived considering the contact normal force. The parameters required for the next position decision were calculated geometrically, and for easier view, a schematic diagram shown in Fig. 4 was drew. Table 2 shows the calculation of the relevant variables. Two key points have been considered when deciding the next position to explore: the first one is to maintain the magnitude of the normal force within the desired range, the other one is to control the finger to move forward following the surface of the object. For the first point, a virtual normal force was generated based on $(\varepsilon_t, \dot{\varepsilon}_t)$, which can guide the finger to move closer or away from the normal direction of the object, and the normal force will increase if closer to, and vice versa. This virtual normal force is called the "*correcting force*", and denoted as $\mathbf{F}'_N$. K was obtained through the fuzzy control method, which will be

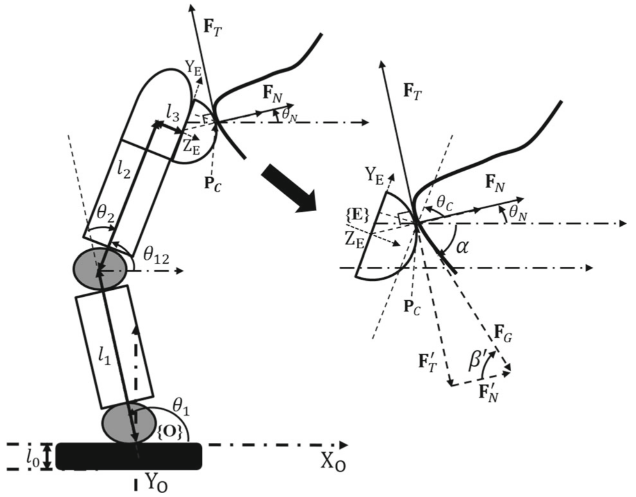


Fig. 4 A schematic diagram showing the calculation of related angles when deciding the next explore location. The definition of the notations can be found in Table 2. The black curve contacting the contact sensing finger is a sketch of the object outline. The right-hand side is a magnified view of the contact area details. This figure only shows the situation where the correcting force $\mathbf{F}'_N$ has the same direction with the real normal force, which means the real force is smaller than the target force, and the finger needs the push towards the object to increase the force

detailed in the next section. In practice, this decision-making strategy is similar to the rule human beings use when they explore the surface of the object. Humans normally push further when they want to increase the pressure and leave away to reduce it.

For the second point, a virtual tangential force $\mathbf{F}'_T$ was introduced proportional to the tangential force on the fingertip, but in the opposite direction. This force can ensure that the finger moves forward along the surface of the object. Combined with $\mathbf{F}'_N$, a total force $\mathbf{F}_G$ was obtained. The force $\mathbf{F}_G$ can guide the finger to move along the surface of the object and maintain the normal force, which is called the "*guiding force*". And based on this guiding force, the direction of the next position $\mathbf{P}_{next}$ in base frame was calculated. Table 2 shows the detailed calculation of related angles. First, the angle θ_N of the normal force in frame {O} was calculated based on the angle of contact point in frame {E}, then the angle β' between $\mathbf{F}'_N$ and $\mathbf{F}_G$ was obtained after obtaining the forces, finally the angle α of $\mathbf{F}_G$ in frame {O} was calculated.

Next, S_v was introduced to determine the incremental size of the next exploration position compared to the current position. As mentioned above, it can determine not only the velocity of exploration but also the control accuracy and sensitivity to external disturbances induced by geometry changes, surface roughness and the movement of objects. Practically, when the object moves, the force error with the target force changes accordingly, the control method makes adjustments to eliminate the error as quickly as possible, which means the response time is affected by S_v. For example, if the object moves away from the finger, the error will become larger and $\mathbf{P}_{next}$ will be closer to the object to increase the force.

After $\mathbf{P}_{next}$ has been obtained, the motor control signals were obtained through inverse kinematics. By rotating the motors to the required positions, the finger will move to explore the next location. Repeating this action until the finger finished contacting the object surface, the surface exploration was completed while maintaining the local normal force in the whole exploration process. As shown above, providing the geometric configuration of the tip, forward and inverse kinematics of the platform, the control variables can be obtained, regardless of the number of DoF of the platform. Which indicates that the control algorithm is not limited to this two joints finger, it can also be applied to more complicated device like robotic arm or soft robots. However, the parameters for the fuzzy control method and rotation matrix for different devices are slightly different due to their specific moving resolution, exploration direction and force ranges. Therefore, for different devices, re-calibration is necessary to achieve the best performance, which in theory needs to be done only once. Practically, the calibration of the fuzzy control method can be done automatically with optimized methods by miniaturizing the error between the requested and real haptic exploration data, which could be explored more in future work.

3.4 Fuzzy control method

Fuzzy control method has been studied extensively in robot systems recently (Aviles et al. 2016). da Fonseca et al. (2017a) and da Fonseca et al. (2017b) used the fuzzy control method to realize grasp and exploration tasks with a three-fingered robotic hand and have achieved good results. It is a method that mimics the intuitive decision-making process of humans. Humans can solve complex problems using imprecise information such as common sense, expert knowledge, or prior experience which does not require precise models (Takagi and Sugeno 1993). Such information can be represented by linguistic rules, like in If-Then format (Tanaka and Wang 2004). And fuzzy logic is the theory of fuzzy sets that can represent mathematically the human spirit for approximation reasoning based on imprecise information, which is also a major advantage over other classic control methods that usually require very precise modelling or parameter fine-tuning. When the external environment is complex, or the current state is unknown, it will be difficult to obtain an accurate model due to the unclear state and influence factors. In such scenarios, the fuzzy control method can better satisfy the control requirements (de Silva 1995). For example, when a person touches a soft or brittle object, they will choose to use less pressure to avoid breaking the object, and when they encounter a bump or obstacle, they will leave the surface of the object for further exploration correspondingly. These decisions are not precisely made but intuited by learnt knowledge or former experience.

Conventional control techniques such as PID control, nonlinear feedback control, etc. can provide good stability, robustness to uncertainties and disturbances when the underlying assumptions are satisfied and values of the controller parameters are known. However, these control algorithms are inflexible and cannot generally handle vague or changing situations. Fuzzy control method is a particular type of intelligent control, which has a knowledge-base within the controller, and the control actions are generated through an inference mechanism. It can handle vague or incomplete information and the knowledge itself can evolve through learning (de Silva 1995). Experiments using classic control method—PID control have been carried out, and the results show that for a fixed object and a certain area on the object, the normal force can be controlled at a relatively fine level during exploration with fixed P, I and D values. However, for another object or a different area on the surface, the P, I and D values needed to be fine-tuned before they can be used to get similar performance. As in this study, the finger was controlled to explore objects with dynamic movement at random areas, the fuzzy control method which has more flexibilities is more suitable. Figure 5 shows the normal force result of using the fuzzy method and PID method for the rough plate (no. 25, shown in Sect. 4). The parameters were obtained for both methods using the light bulb as the learning object, and the rough plate was used for the testing object. As a result, the SD and MSE of the resultant normal force (target normal force is 0.6 N) when using the fuzzy method are 0.08 N and 0.0067 N, while for the PID method, they are 0.11 N and 0.013 N. The results indicate the better flexibility of the fuzzy method when dealing with dynamic and unknown object. Also, by changing the boundary of the memberships, the control method can be adapted to different objects. Therefore, in this study, the fuzzy control method was selected.

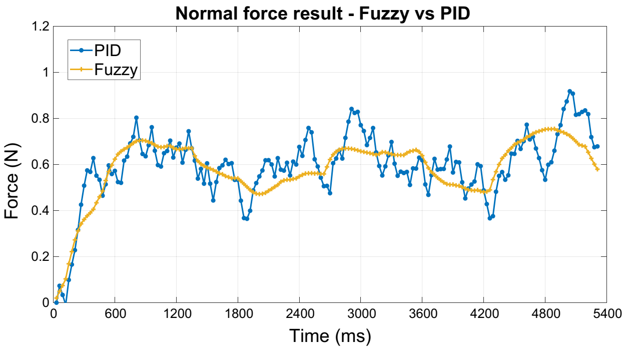


Fig. 5 The figure shows the normal force result during the exploration of the rough plate (no. 25) using the fuzzy and PID method. The parameters were obtained for both methods using the light bulb as the training object, and the rough plate was used as the testing object. It can be seen that the vibration of the normal force when using the PID method is much larger than that when using the fuzzy method. The SD and MSE of the normal force are 0.08 N and 0.0067 N when using the fuzzy control method, and 0.11 N and 0.013 N when using the PID method. The results indicate the better flexibility of the fuzzy method

A fuzzy control system mainly consists of 4 components: *Fuzzifiers*: It maps the real-valued input into fuzzy sets characterized by membership functions. *Rule Base*: It is a knowledge base consisting of linguistic rules in If-Then format. It can be obtained through experiments or experiences. *Fuzzy Inference Engine*: Using the If-Then rules in the knowledge base, it performs reasoning by producing a fuzzy output according to the fuzzy input given by the fuzzifier. *De-fuzzifiers*: It converts the fuzzy output given by the fuzzy inference engine to produce a real-valued output. To create the fuzzy control framework of the control algorithm, these four components were designed.

The inputs of the system are the force error between the measured and target value and its changing rate: $(\varepsilon_t, \dot{\varepsilon}_t)$. The input will be converted into fuzzy sets represented by membership functions, and the continuous fuzzy set was used in this work, which is denoted as:

$$F_s = \frac{\int \mu_F(x)}{x} \tag{7}$$

where x is indicating the input data. The boundaries are defined as the region of the universe of which $0 < \mu_F(x) < 1$. There are common membership functions such as Triangular function, Gaussian function, Bell function, etc. Different types of membership functions were tested, and the Gaussian function has been selected, which showed the best performance. The fuzzy output is also an inferred membership function, and by moving the boundaries of output membership functions, the coverage of different memberships can be changed, and the final control inclination following predefined logic rules is modified adaptively. For example, after one exploration, if the average normal force error is too large, the boundaries can be shifted rightward to decrease the coverage of the closing action. In this way, if the controller attempts to increase the normal force, the magnitude will be smaller. Figures 6, 7 and 8 show the fuzzy control frame of using the Gaussian function for the input and output memberships. The number of output memberships is decided by the knowledge base, and they are shown in different line types.

For creating the knowledge base, a 9-rule IF-THEN (IF premise THEN conclusion) rule base was formed shown in Table 3. In which “Leave” means controlling the finger to leave away from the surface of the object, “Close” means moving closer to the object and “Stay” means maintaining the distance from the object in the contact normal direction. And “small”, “medium” and “large” mean the level of the actions.

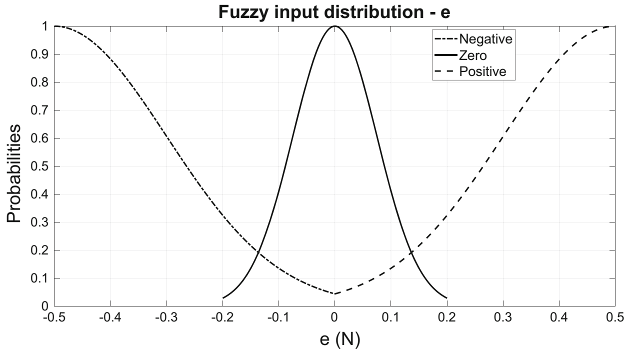


Fig. 6 The figure is showing the Gaussian distributions of the fuzzy input (ε_t). The line types present the different memberships. ε_t has the range of (− 0.5, 0.5)N, and it has three components: the negative, zero and positive with mean value of − 0.5, 0, and 0.5 respectively

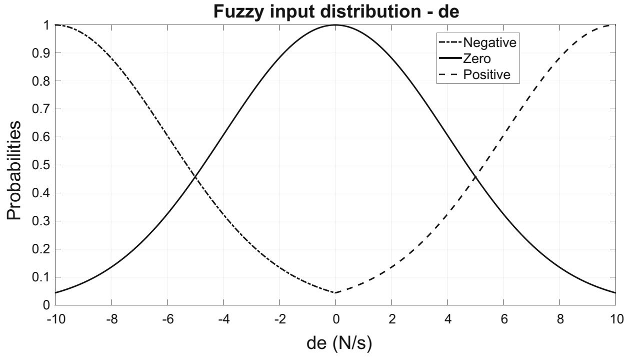


Fig. 7 The figure is showing the Gaussian distributions of the fuzzy input ($\dot{\varepsilon}_t$). The line types present the different memberships. $\dot{\varepsilon}_t$ has the range of (− 10, 10)N, and it has three components: the negative, zero and positive with mean value of − 10, 0, and 10 respectively

Both (ε_t, $\dot{\varepsilon}_t$) have values of positive, zero and negative, the combination of these situations forms 9 (3x3) rules. The rules can be adjusted based on the experiment results, for example, when ε_t is positive and $\dot{\varepsilon}_t$ is zero, the conclusion can either be "leave medium" if ε_t has larger effect or "leave small" if the effect of ε_t is not significant. As mentioned before, fuzzy logic represents mathematically the experienced knowledge for approximation reasoning; it requires the learnt knowledge or experience to decide the rules. The bulb object was chosen to get the fuzzy rules and decide the range of output membership functions. For the fuzzy inference engine, the Mamdani (max-min) inference method was selected, which can present a fuzzy output in membership function according to the fuzzy input. The centroid method was selected (also known as the centre of gravity (CoG)), which provided us with satisfied output values. The CoG for continuous form can be calculated using:

$$K = z^* = \frac{\int \mu_F(z) z d_z}{\int \mu_F(z) d_z} \qquad (8)$$

where $\mu_F(z)$ is the membership function for the output and $K = z^*$ is the output value.

With the input being mapped into membership functions, the output K used for deciding the next explore location was obtained with the de-fuzzifier, following the rules in the knowledge base. Thus, the fuzzy control method has been finalized. For the software implementation of the fuzzy control method, the fuzzylite library (https://fuzzylite.com/) has been used.

3.5 Explore velocity control

It has been found out that the S_v is the parameter that affects the explore velocity and study on how S_v can be used to control the explore velocity has been carried out. For each exploration, S_v can either be fixed or controlled with closed-loop control. In the test, when S_v was fixed, the average

Fig. 8 The figure is showing the Gaussian distributions of the fuzzy output K. The line types present the different memberships. As shown in the figure, K has the range of (− 1,1), and it is following 9 rules involving 7 actions: leave large, leave medium, leave small, stay, close small, close medium and close large, with μ_K of − 1, − 0.7, − 0.3, 0, 0.25, 0.6 and 1 respectively

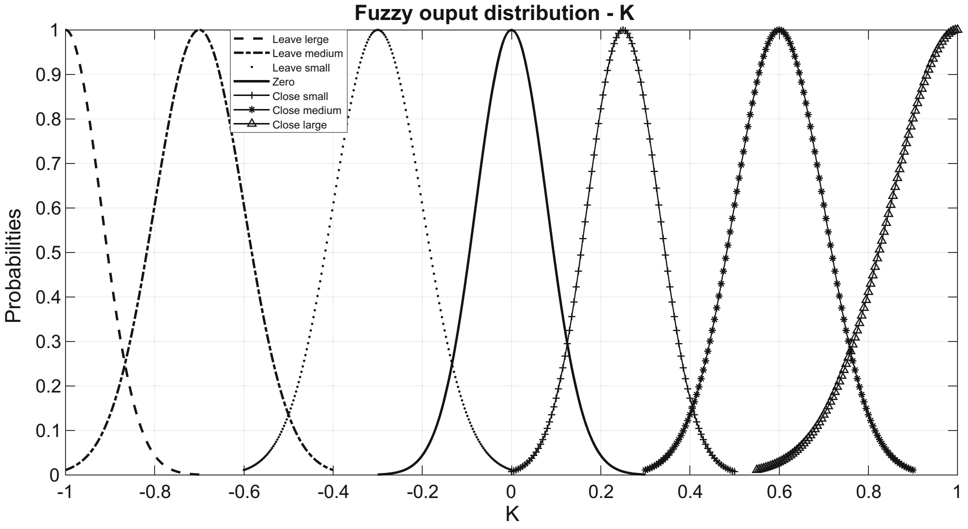


Table 3 Fuzzy knowledge base

	IF-THEN Linguistic rules		
ε_t	Positive (0 to 0.5)	Zero (− 0.2 to 0.2)	Negative (− 0.5 to 0)
$\dot{\varepsilon}_t$			
Positive (0 to 10)	Leave large	Leave medium	Close small
Zero (− 10 to 10)	Leave medium	Stay	Close medium
Negative (− 10 to 0)	Leave small	Close medium	Close large

explore velocity turned out to be similar for different objects, but the standard deviation was relatively high. In this work, a PID controller for the velocity control was created. And for creating the controller, the real-time velocity was calculated with $v_m = l_m/dt$, where l_m is the length the finger moved in the base frame in one loop and dt is the time used for one loop. After filtering the v_m with a moving average filter (the filter takes 10 points as the input), the error with the desired v_d was fed into the PID controller to generate the S_v value for the next position decision. A new S_v will change the explore velocity in the next movement. Algorithm 1 shows the detail of the control process.

Algorithm 1 Velocity control algorithm

```
Initialise variables, $l_m = 0, count = 1, I = 0, S_v = 0.45mm$,
  decide $k_p, k_i, k_d$, input target velocity $v_d$, $\varepsilon_v = v_d$
while joint angles $\theta_1, \theta_2 < \theta_{limit}$ do
    if $count == 1$ then
        Get current time t and contact position $\mathbf{P}_t$
        Action: Haptic exploration with initial $S_v$
        $count \leftarrow count + 1$
    Get new contact position $\mathbf{P}'_t$ and time $t'$
    Compute the distance the finger travelled: $l_m \leftarrow dis(\mathbf{P}_t, \mathbf{P}'_t)$
    Compute the velocity $v_m \leftarrow l_m/(t' - t)$
    Smooth $v_m$ with moving average filter
    New error $\varepsilon'_v \leftarrow v_d - v_m$, $I \leftarrow I + \varepsilon'_v * dt$
    Compute the new $S'_v$ with PID controller:
    $S'_v \leftarrow k_p * \varepsilon'_v + k_i * I + k_d * (\varepsilon'_v - \varepsilon_v)/dt$
    Store current error: $\varepsilon_v \leftarrow \varepsilon'_v$, time $t \leftarrow t'$
    Store current position $\mathbf{P}_t \leftarrow \mathbf{P}'_t$
    Action: Haptic exploration with $S'_v$
return 0                              ▷ Return 0 if finished successfully
```

When S_v was changed with the PID controller, the standard deviation of the velocity became smaller, thus, the velocity control for surface exploration has been achieved. The velocity control can help to get velocity related properties such as viscous friction coefficient f_v which can be obtained by measuring the viscous friction $\mathbf{F}_v$ and the coefficient is calculated using $f_v = \mathbf{F}_v/v$. With the explore velocity been controlled, the instantaneous power generated by the friction during the contact and sliding is also controllable, which is helpful in manufacturing procedures like grinding. Moreover, it can be beneficial to the control of explore efficiency during haptic perception, facilitating the development of an active exploration algorithm where velocity is important for deciding the time cost.

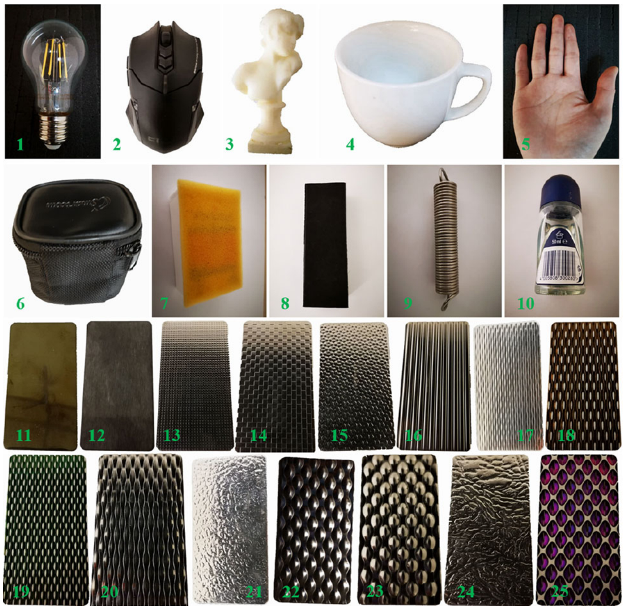


Fig. 9 The objects used in the haptic exploration experiments. From top to bottom, left to right is: bulb, mouse, 3D printed statue, cup, human finger, soft box, sponge (softer), black sponge, metal spring, glass bottle; and plates made with different materials and patterns: mirror stainless, satin stainless, BA stainless (canvas), BA stainless (squares), BA stainless (13SD), BA stainless (cambridge), BA stainless (11PS), bronze stainless (pippin), green stainless (poppy), BA stainless (5WL), BA stainless (2WL), BA stainless (7GM), BA stainless (6WL), BA stainless (9EH), red stainless (pearl)

4 Haptic surface exploration experiments

With different platforms, various haptic surface exploration experiments were carried out to evaluate the performance of the developed control algorithm. In this section, the one-time exploration with the contact sensing finger (platform-a) will be introduced first, followed by the adaptive parameter tuning for the multi-time explorations using platform-b, then the haptic exploration with the robot arm (platform-c) will be presented, and the velocity control experiments using the platform-a will be shown at last.

Table 4 Force control results of haptic exploration

	Resultant normal force standard deviation (N) and MSE									
No.	F_d-0.2 N		F_d-0.4 N		F_d-0.5 N		F_d-0.6 N		F_d-0.7 N	
1	0.047	0.0038	0.045	0.035	0.075	0.0077	0.09	0.0083	0.091	0.0085
2	0.054	0.0036	0.056	0.037	0.059	0.006	0.076	0.0067	0.083	0.008
3	0.052	0.0033	0.07	0.006	0.076	0.0074	0.083	0.0084	0.12	0.013
4	0.048	0.0026	0.05	0.0033	0.065	0.0067	0.075	0.0063	0.09	0.0083
5	0.063	0.0048	0.07	0.0055	0.086	0.008	0.1	0.01	0.13	0.012
6	0.051	0.003	0.062	0.006	0.072	0.0065	0.076	0.0066	0.09	0.01
7	0.04	0.0033	0.052	0.005	0.064	0.0059	0.085	0.0082	0.09	0.0085
8	0.04	0.003	0.045	0.004	0.075	0.0065	0.078	0.0075	0.093	0.0094
9	0.047	0.0032	0.053	0.003	0.077	0.007	0.086	0.0088	0.1	0.01
10	0.045	0.0031	0.053	0.003	0.064	0.0079	0.089	0.0085	0.095	0.009
11	0.039	0.0027	0.05	0.0039	0.068	0.0052	0.071	0.006	0.076	0.0076
12	0.04	0.0027	0.05	0.004	0.066	0.0052	0.069	0.0077	0.079	0.0073
13	0.04	0.003	0.049	0.0032	0.066	0.0062	0.069	0.0066	0.09	0.0082
14	0.053	0.0037	0.057	0.0036	0.066	0.0047	0.071	0.0048	0.08	0.0078
15	0.043	0.003	0.056	0.0039	0.07	0.0059	0.072	0.007	0.093	0.0083
16	0.038	0.0031	0.056	0.0039	0.073	0.006	0.079	0.0067	0.095	0.0085
17	0.042	0.0038	0.052	0.0035	0.079	0.0066	0.068	0.0071	0.09	0.0082
18	0.029	0.0032	0.052	0.0033	0.067	0.006	0.073	0.0063	0.086	0.0086
19	0.038	0.0033	0.048	0.0038	0.067	0.0052	0.076	0.007	0.09	0.0085
20	0.035	0.003	0.055	0.0046	0.079	0.0076	0.081	0.0075	0.09	0.009
21	0.037	0.0027	0.054	0.0035	0.071	0.0057	0.075	0.007	0.09	0.0084
22	0.054	0.0034	0.057	0.0042	0.068	0.008	0.078	0.0088	0.1	0.0093
23	0.056	0.0037	0.059	0.003	0.07	0.006	0.073	0.0085	0.093	0.01
24	0.049	0.0038	0.058	0.0046	0.082	0.008	0.089	0.0096	0.1	0.0096
25	0.055	0.0036	0.062	0.0045	0.085	0.008	0.095	0.0085	0.12	0.0099
Avg.	0.045	0.0036	0.055	0.0039	0.072	0.0064	0.076	0.0072	0.093	0.0089

4.1 Single time haptic surface exploration

For carrying out the experiments with the contact sensing finger (platform-a), 10 objects with different surface properties (friction, roughness), geometry and stiffness, and 15 plates made of different materials with different surface patterns were selected, shown in Fig. 9. It should be noted that, due to the limited power of the motor, the maximum force the finger can generate is less than 1.1 N, thus, the controllable force was constrained to 0.2–0.7 N in the tests. The target forces used in the experiments were shown in Table 4. For each object, the haptic exploration started after the contact was confirmed and ended until the finger finished contacting the object. The S_v was chosen from 0.35, 0.45, and 0.55 mm randomly for each exploration. During each exploration, the normal force and velocity at each contact point were stored in vectors. And after the exploration, the mean value μ_f and SD σ_f of the force vector were calculated, and the MSE was obtained by comparing the resultant normal force with the target force. The μ_v and σ_v of the velocity vector were also calculated. Algorithm 2 summarizes the experiment protocol.

To provide the object with dynamic movement, two ways have been used, the first was to use hand to hold the object which would induce slight movement during the haptic exploration, and the other was to move the object with random displacement through a created device. For the second approach, the object was mounted on the linear guide shown in Fig. 10, which can generate a random displacement of 2–4 mm during the exploration. Please refer to the video of "dynamic_object_exploration (MOESM1).mp4" for more details.

The normal force control results of one-time exploration on the light bulb, black sponge, rough plate (no. 25), smooth plate (no. 12) and 3D printed statue hold by hand are shown in Fig. 11. It can be seen that, ignoring the rising period of the normal force, the performance of the control is satisfied for the bulb, black sponge and smooth plate, which in total achieved an average standard deviation of 0.044 N.

Algorithm 2 Object haptic surface exploration

```
Initialise vector $\mathbf{V_f}$, $\mathbf{V_v}$, $\mathbf{V_d}$ to store the resultant normal force, velocity and target force
Initialise variables, $pointer = 1, sum = 0$, input target force $F_d$ and put it into the vector $\mathbf{V_d} \leftarrow F_d$
Initialise the parameters for fuzzy control method (FCM)
if contact normal force $F_N > \vartheta_f$ then
    while joint angles $\theta_1, \theta_2 < \theta_{limit}$ do
        Get current contact normal force and store it in vector $\mathbf{V_f}(pointer) \leftarrow F_n$
        Compute next moving position $\mathbf{P}_{next}$ with FCM
        Get the new angles to move next: $\theta_1, \theta_2 \leftarrow IK(\mathbf{P}_{next})$
        Action: The Motors rotate to the new positions
         Record the time used $dt$ and compute the distance $L$ travelled
         Compute the explore velocity $v \leftarrow L/dt$
         Store the velocity in vector $\mathbf{V_v}(pointer) \leftarrow v$
         Update $pointer \leftarrow pointer + 1$
    Smooth the data with filter, $(\mathbf{V_f}, \mathbf{V_v}) \leftarrow filter(\mathbf{V_f}, \mathbf{V_v})$, to get rid of the erroneous data
    Compute $\mu_f \leftarrow Mean(\mathbf{V_f}), \sigma_f \leftarrow SD(\mathbf{V_f})$
    Compute $\mu_v \leftarrow Mean(\mathbf{V_v}), \sigma_v \leftarrow SD(\mathbf{V_v})$
    for each element $i$ in $\mathbf{V_f}$ and $\mathbf{V_d}$ do
        $sum \leftarrow sum + (\mathbf{V_f}(i) - \mathbf{V_d}(i))^2$
    $MSE \leftarrow sum/pointer$
return 0                              ▷ Return 0 if finished successfully
```

However, for the statue and rough plate, the vibrations of the control are larger (SD = 0.075 N) due to the complex geometry and large concave areas on the surface. In this work, for getting the final results of the haptic exploration experiments, only the data when the force became stable were used, which means that the force data in the rising period were omitted. In practical, the object was held by hand and not fixed during the haptic exploration, which added dynamic movement to the object. There are two notable facts during the experiments. First, the initial contact position was random and different for each test, and second, as the hand was shaking when holding the object, the object was moving slightly (1–3 mm) during the exploration. These all increased the difficulties for the control algorithm and in return can show the robustness of the algorithm. Please refer to videos of “Force_control_01 (MOESM4), Force_control_02 (MOESM5), surface_exploration_statue (MOESM6).mp4” for more details.

4.2 Adaptive parameter tuning

After the one-time exploration experiments were carried out, the finger (platform-a) was mounted onto the end-effector of the robot arm, forming the platform-b. The robot arm end-effector can rotate to any degree after each exploration to help the finger explore more areas on the surface of the object. In the experiments, the arm was controlled to rotate every 30 degrees after one exploration, and the exploration was carried out again, in total, the arm rotated 60 degrees and returned to the starting position. The resultant haptic information of all the exploration procedures were recorded.

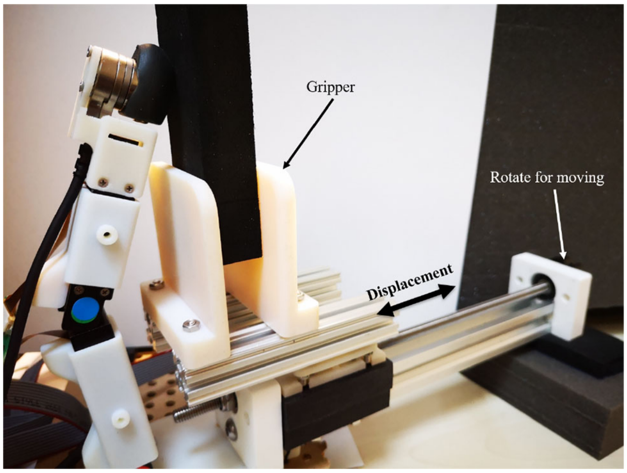


Fig. 10 The device used to generate a random displacement for the objects. It consists of a linear guide and a 3D printed gripper. The linear guide is controlled with a gear that can be rotated manually. The linear guide moves 1 mm after rotating the gear by one circle. The 3D printed gripper can be adjusted on the metal rails for different objects

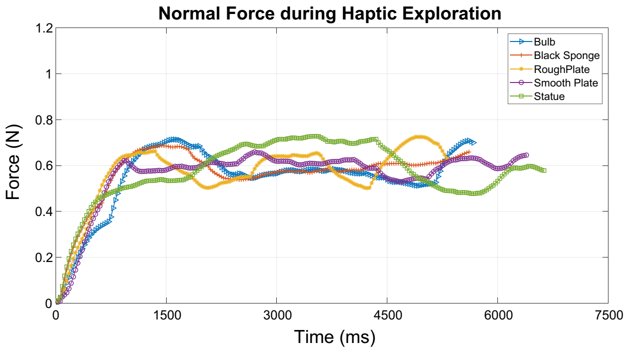


Fig. 11 Normal force results of the one-time haptic exploration on the surface of the light bulb, black sponge, rough plate, smooth plate and 3D-printed statue. The force control target was 0.6 N and the S_v value was kept at 0.45 mm. As can be seen, the rough plate and 3D-printed statue have the largest vibrations. This is because the 3D-printed statue has the most complex geometries and the rough plate has large concave patterns, both of them make it difficult to adjust the force during the exploration. And for the light bulb, black sponge and smooth plate, regardless of their complex geometry, texture, and stiffness, the force control results are satisfied, showing the robustness of the algorithm. However, as can be seen, the rising time of the algorithm is sluggish, which could be improved with a faster control loop

The adaptive parameter tuning method was developed for multi-time explorations. After each exploration, if the average force value μ_f was too much smaller than the target and the MSE was large, the control algorithm will shift the boundaries of the output fuzzy memberships to the right to make the “close” action to have more coverage of the whole output distribution. This can be done by increasing the mean values (μ_{close}) of the Gaussian distribution of the “close” membership functions shown in Fig. 8. Through this way, the finger will push more during the next exploration for the error correction. Meanwhile, it is required to guarantee that the changing of the control parameters will not induce too much noise to the haptic exploration by monitoring the standard deviation σ_f. Similarly, if μ_f was too much larger than F_d and MSE was large, the boundaries will be shifted leftwards to make the "leave" action to play a more important role during the exploration. Also, it has been learnt that the K value is a notable parameter for the adjustment of the normal force during the exploration, because a larger K will increase the movement of the finger, which permits the force control to become more dramatic. Thus, a variable for the K has been introduced: $K = \alpha_k * K$, to change its scale. For different devices, the α_k needs to be calibrated to achieve the best performance. For instance, for objects with larger curvature changes, a larger α_k can help to adjust the normal force more quickly. In this study, for the validation experiments, α_k was set to 1 for all the objects, the adaptive selection of α_k will be developed in future work. Algorithm 3 summarizes the adaptive control algorithm.

Algorithm 3 Adaptive control algorithm

```
Define $\Delta f$ as the magnitude of changing of the membership functions, $\Delta f = 0.1$
After one exploration, get $\mu_f \leftarrow mean(\mathbf{V_f}), \sigma_f \leftarrow SD(\mathbf{V_f})$ and $MSE$
while $abs(F_d - \mu_f) > \vartheta_1$ and $MSE > \vartheta_e$ and $\sigma_f < \vartheta_\sigma$ do
    Increase the $\mu_{close}$ of "Close" membership functions, which means the output distributions of "Close" move rightwards: $\mu_{close} \leftarrow \mu_{close} + \Delta f$
    Action: Haptic exploration again with new FCM
    Get new $\mu_f, \sigma_f$ and $MSE$
while $abs(F_d - \mu_f) < \vartheta_2$ and $MSE > \vartheta_e$ and $\sigma_f < \vartheta_\sigma$ do
    Decrease the $\mu_{close}$ of "Close" membership functions, which means the output distributions of "Close" move leftwards: $\mu_{close} \leftarrow \mu_{close} - \Delta f$
    Action: Haptic exploration again with new FCM
    Get new $\mu_f, \sigma_f$ and $MSE$
return 0                                    ▷ Return 0 if finished successfully
```

For each experimental setup, 9 times of exploration were carried out on random areas (3 times with the arm at 0, 30, and 60 degrees) of the objects with random S_v values chosen from 0.35 mm, 0.45 mm, and 0.55 mm. 5 target forces have been set, and the total number of surface explorations was 1125 in the force control experiments. After each exploration, the mean value μ_f and σ_f of the force vector were calculated, and the MSE was obtained. Also, the algorithm adjusted its fuzzy control method parameters to achieve an overall optimised performance. After 9 times of exploration, the average of the σ_f and MSE were obtained.

4.3 Contact finger exploration results and discussion

After all the haptic surface exploration experiments have been carried out, the overall results of the force control were obtained. The detailed results can be found in Table 4, and from the results, the following points were concluded.

1. When the target force is increasing, the MSE and σ_f become larger. It could be due to the reason that larger errors require more correcting force to adjust, and there are chances that the finger is unable to correct such errors quickly enough with constrained power.
2. Objects with more complex geometries tend to have larger MSE and σ_f, such as the statue and the bulb. Objects with changing stiffness also have relatively higher MSE and σ_f, like the human finger. The reason could be that the complex geometries and changing stiffness of the object surface will induce dramatic force changes during the haptic exploration, and the force errors can not be eliminated completely and quickly with the control algorithm, as a consequence, the error accumulates and the MSE and SD become larger than the situation when exploring the smoother objects
3. When the target force is getting larger, the hand holding the objects is more unstable. The stronger handshaking induces more instabilities to the system, especially when the force is 0.6 N or larger, making the noise (σ_f) larger. Also, for objects with complex geometries, this effect is greater, while for flat plates, this effect is smaller.
4. In Fig. 9, the order of the plates was made by looking at the surface texture and patterns rather than the real roughness and surface pattern complexity level. For plate 1, even though it has the smoothest surface by vision, due to the material property of the fingertip, during the exploration, the contact of the fingertip and surface of the plate can generate viscous friction that makes the control of the force harder, especially when the force is greater than 0.4 N. The 15th plate (no. 25) has large and deep concave areas, and plate13 (no. 23) has large and high convex areas, these patterns will induce sudden force changes during the exploration, making the MSE and σ_f larger.
5. For soft objects that can deform with force applied, when the force is large, the deformation of the object makes the contact location estimation algorithm become inaccurate, thus, forces larger than 0.7 N were not used in the experiments, especially for the yellow soft sponge and soft black box. The metal spring has gaps between threads that can generate frequent drops of force values during the exploration, making the control harder.
6. When the target force is small (like 0.2 N and 0.4 N), the MSE and σ_f of different objects are similar for the same target force, however, when the force is larger, the differences of the MSE and σ_f of different objects get larger. Results of the black rubber, plate1 (no. 11), statue, plate15 (no. 25), human finger and yellow soft sponge representing different types of objects were used to plot figures (shown in Figs. 12, 13).

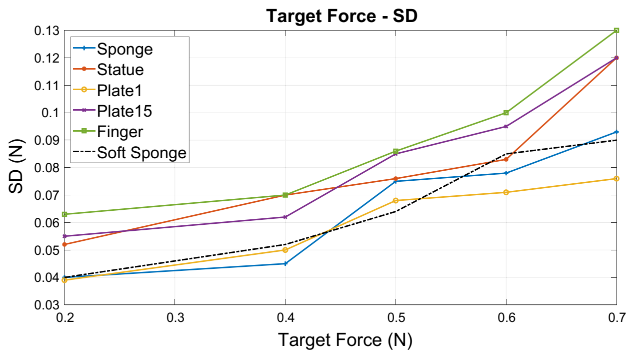


Fig. 12 The SD results for different objects with different target forces. The black sponge and plate1 are objects with flat and fine surface that are easier for the force control, and their SDs are smaller than other objects. The 3D printed statue is the object with the most complex geometry, and plate15 has large concave areas, these features all hinder the control performance, and the SDs for these two objects are larger. The human finger has different stiffness and changing geometry in different locations, thereby the SD of which is the largest. The SD of the yellow soft sponge is small at the beginning but gets larger when the force is increasing due to the fact that when the force increases, the finger penetrates the soft layer of the sponge and touches the plastic surface underneath (Color figure online)

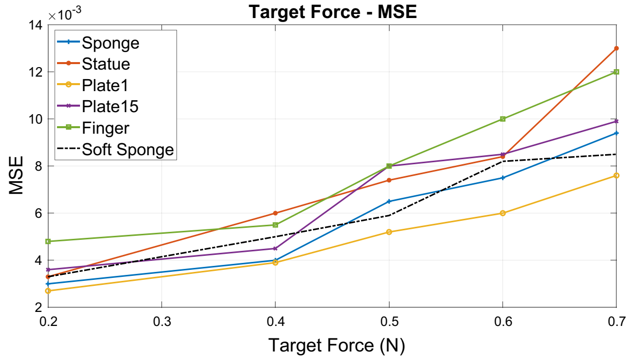


Fig. 13 The MSE results for different objects with different target forces. The black sponge and plate1 have smaller MSE. The 3D printed statue and plate15 with more complex geometries and patterns have larger MSE. The human finger has the largest MSE because of its changing stiffness and geometry. MSE of the yellow soft sponge is changing due to the same reason explained in Fig. 12 (Color figure online)

From the results, the limitations of the platform-a and control system have been learnt. First, the force it can generate is limited, making it difficult to correct the error quickly enough, especially when the target force is more massive. This can be improved by using more powerful motors, but at the same time, the size of the platform will vary since powerful motors usually require more space. Second, the high-level control is working at 100 Hz currently, which may not be fast enough to adapt to quick force changes induced by complex geometry or stiffness changes. The relative low working frequency is due to the time consumption of the software modules like the angle calculation and the fuzzy control method. This can be improved by using higher spec PCs or simplifying the software implementation. Despite these limitations, it can be seen that the average MSE is 0.006 and σ_f is 0.07 N. Also, from Fig. 14, it can be seen that the MSE and σ_f for all objects are constrained within a controllable range, which can show the robustness of the control algorithm for different objects with various properties and dynamic movement. Moreover, videos of the real-time experiments show that the platform-a works well with subtle motions following the surface of the dynamically moving object with complex outline, when the target force is small.

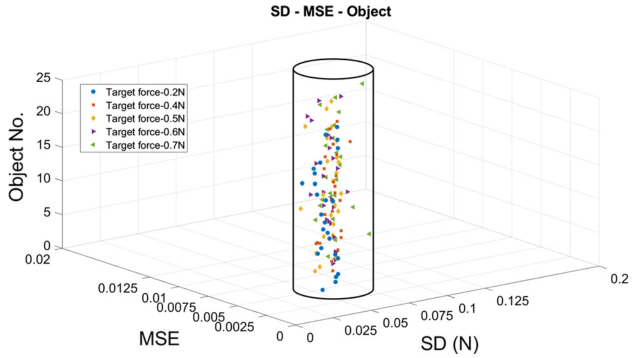


Fig. 14 The points in the figure are generated with the SD value, MSE value and the object number as the x, y and z coordinate. It can be seen that the resultant control performance of the haptic exploration on different objects are similar within a small range shown by a transparent cylinder, which reveals the robustness of the proposed algorithm

4.4 Haptic exploration with robot arm UR-3

Since the workspace of the finger is limited with an average travelling length of 60 mm, the robot arm UR-3 was used to verify the control method with larger workspace and force ranges. A contacting sensing tip was mounted onto the robot arm end-effector, forming the platform-c. The study of controlling the robot arm and finger simultaneously to realize a synergic control is also envisioned, which can be useful for more complex haptic exploration. The robot arm can reach the object at almost any point on the surface, and the exploration is not constrained in any specific plane, but for one-time exploration, the controlled movement is within two axes. The robot arm can also realize a sequence of movements, covering more surfaces of the object. Figure 15 (left) shows the designed experimental setup. The Nano 17 sensor was mounted on the robot arm with a 3D printed connection part and covered with a hemisphere cap made of silicone rubber (Ecoflex-50, Shore A 50). The stiffness of the rubber cap is large enough that there is no hysteresis effect during the experiments. A device that can generate a random displacement within 5–12 mm in the vertical direction was used to add movement to the holding object. The corresponding kinematic and fuzzy control parameters of the control algorithm were modified for controlling the robot arm platform

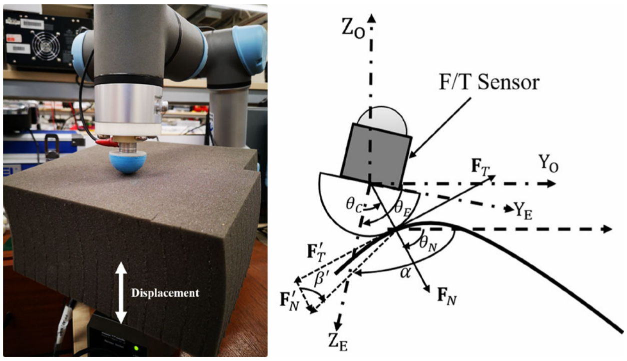


Fig. 15 Left: The haptic exploration platform-c using the UR3 robot arm and a contact sensing tip. The object is mounted on a moveable device that can move 5–12 mm vertically. Right: The schematic drawing of how the necessary angles have been calculated for deciding the next moving location of the tip. Same with Fig. 4, this only shows the situation where the correcting force $\mathbf{F}'_N$ has the same direction of the real normal force

(platform-c). Similarly, the required angles can be calculated with the equations below:

$$\theta_E = 2 * \pi - \theta_{j2} - \theta_{j3} + 1.5\pi - \theta_{j4} + 2\pi - \theta_{j5} \tag{9}$$

$$\theta_C = atan2(y_e, z_e) \tag{10}$$

$$\theta_N = \theta_E + \theta_C \tag{11}$$

$$\beta' = atan2(\|\mathbf{F}'_T\|, \|\mathbf{F}'_N\|) \tag{12}$$

$$\alpha = \beta' + \theta_N \tag{13}$$

And the next position to reach is:

$$\mathbf{P}_{next} = \mathbf{P}_{now} + S_v \cdot [cos\alpha, sin\alpha]^T \tag{14}$$

where $\theta_{j2}, \theta_{j3}, \theta_{j4}, \theta_{j5}$ are the current joint angles of the robot, which can be obtained through the driver in ROS.

With the proposed control algorithm, the robot arm was controlled to explore the surface of the object while keeping the normal force as desired. Since the robot arm has more DoFs and can work in larger target force range (more powerful), it can explore the object in a much wider workspace with feedbacks from the contact sensing tip, which helps to realize larger area's haptic exploration and make more exploratory procedures (EPs) available (Lederman and Klatzky 1993). More EPs will allow the haptic perception system to obtain more information about the object, which will be beneficial in future research on the active haptic control and perception. Here, the haptic surface exploration is the main focus, and more EPs will be implemented in the next study.

However, the control of the robot arm has limitations due to the slow frequency of the software. The control loop is limited to 10 Hz in this study, which may hinder the response speed when adjusting the force or velocity. Moreover, compared to the contact sensing finger, the smallest S_v it can achieve accurately is no smaller than 0.8 mm, and the force it generates is much larger than the finger per movement. These all present that the robot arm is suitable for larger force control scenarios.

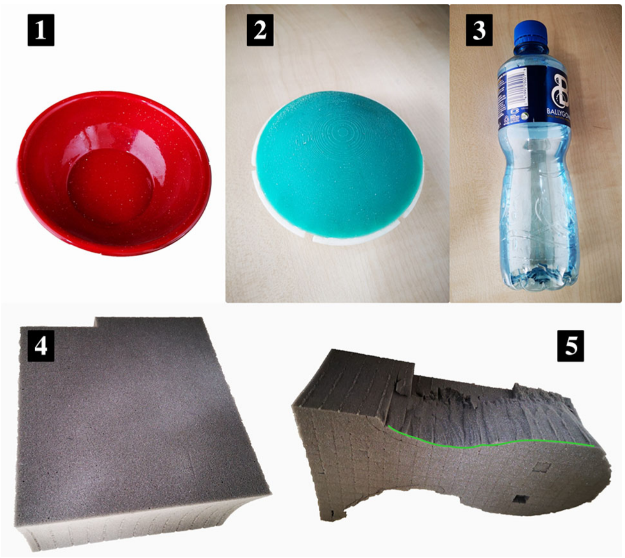


Fig. 16 The objects used in haptic exploration experiments with the robot arm platform-c. From left to right: (1) metal bowl, (2) sphere like hat made by silicon, (3) plastic bottle filled with water (4) flat sponge, (5) sponge with curved shape, the explore trajectory is drew with green colour (Color figure online)

5 objects shown in Fig. 16 were used to evaluate the performance of the control algorithm. There are four soft objects, all can deform when force applied, and 3 of them have complex outlines. There is one metal object with a curved shape. For different objects, different target forces were set, this is because with too much force applied, the deformation of the soft object will cause inaccurate sensor readings, while for the solid object, the force changed by the smallest displacement is more than 1 N. Thus, the target forces were set to 1 N, 1.25 N and 1.5 N for the soft objects, and 1.5 N, 2 N, 2.5 N for the metal object. The S_v size was chosen from 0.8 mm, 1 mm and 1.5 mm. For each object, 10 times of exploration with the adaptive control method were carried out. The video of "surface_exploration_ur3-1 (MOESM7).mp4" shows the haptic surface exploration on the sponge with curved shape. The objects were mounted on the moveable platform with 5–12 mm random vertical displacement.

Table 5 shows the results of the haptic exploration experiments using the platform-c. From the results, it can be seen that the average MSE and σ_f are larger than those obtained with the contact sensing finger (platform-a) due to larger target forces. As mentioned before, when the target force gets larger, the MSE and σ_f also increase. It has been learnt that, in some situations, even though the algorithm output the largest correcting movement, it was still not enough to correct the force in time. Thus, it becomes necessary for the

Table 5 Force control results with robot arm

	Resultant normal force standard deviation (N) and MSE					
Object no.	Target normal force-1 N/1.5 N		Target normal force-1.25 N/2 N		Target normal force-1.5 N/2.5 N	
1	0.21	0.03	0.31	0.035	0.38	0.042
2	0.15	0.013	0.17	0.019	0.19	0.024
3	0.14	0.016	0.18	0.02	0.21	0.027
4	0.13	0.02	0.15	0.02	0.16	0.03
5	0.12	0.013	0.1	0.015	0.18	0.025
Avg.	0.15	0.0184	0.182	0.0218	0.224	0.0296

Table 6 Velocity control results

Open loop resultant explore velocity mean and SD							PID control resultant explore velocity mean and SD							
S_v value (mm)	F_d-0.2 N		F_d-0.4 N		F_d-0.6 N		v_d (mm/s)	F_d-0.2 N		F_d-0.4 N		F_d-0.6 N		Δv
0.15	1.92	2.11	1.86	2.31	1.78	2.18	2	1.78	1.41	1.75	1.54	1.81	1.45	0.22
0.25	3.16	2.59	3.16	2.68	2.95	2.5	3	2.69	1.73	2.65	1.78	2.72	1.67	0.31
0.35	3.72	2.68	3.8	2.8	3.68	2.64	4	3.62	1.78	3.58	1.86	3.65	1.76	0.38
0.45	4.69	2.86	4.55	2.85	4.67	2.84	5	4.56	1.91	4.51	1.9	4.59	1.89	0.45
0.55	5.52	3.02	5.55	3.12	5.3	2.9	6	5.51	2.01	5.46	2.08	5.54	1.93	0.50
0.65	6.99	3.15	6.89	3.2	6.43	2.96	7	6.48	2.1	6.42	2.13	6.4	2.1	0.57
0.75	7.53	3.24	7.22	3.34	7.19	2.9	8	7.43	2.22	7.37	2.43	7.46	2.23	0.58
0.85	8.38	3.32	8.02	3.54	8.23	3.09	9	8.35	2.41	8.41	2.56	8.41	2.56	0.61
0.95	9.03	3.5	9.04	3.66	8.95	3.41	10	9.26	2.8	9.27	2.84	9.34	2.75	0.71

algorithm to know the relationship of the indentation and force changes for different objects, through which way, the more accurate S_v can be decided for adjusting the force error. As the next step, the robot arm can be controlled to push the object first to characterize the indentation and force changing relation of the object, and then the best S_v can be derived. However, the overall result of this experimental validation is stratified, confirming the robustness of the control algorithm. Moreover, as can be seen from videos of the real experiment (refer to videos of ”dynamic_object_exploration_ur3 (MOESM2), dynamic_object_exploration_ur3-with_back ground (MOESM3).mp4”), the arm can adjust the position quickly following the movement of the object.

4.5 Velocity control results

To verify the performance of the velocity control method, haptic exploration experiments with changing S_v values using the contact sensing finger (platform-a) were carried out. The target force was set to 0.2 N, 0.4 N and 0.6 N in experiments, and the resultant mean value μ_v and σ_v were obtained with different S_v values. Three objects: the black sponge, the light bulb, and plate-15 (no. 25) were selected. First, a fixed S_v value was used during each haptic exploration to check velocity changes and compute the σ_v, then the PID control method was applied to adjust the S_v value during the exploration to check the change of σ_v. The target velocity for the PID control was set from 2 to 10 mm/s. The mean velocity was calculated by $v_m = L_e/T_e$, where L_e is the total explored distance and T_e is the overall time used for the exploration process. For each object, 3 times exploration were carried out for each fixed S_v value and 3 times exploration for each target velocity using the PID control.

Table 6 shows the results, and the values are averages of the three objects. It can be seen from the results, when S_v is fixed, the mean explore velocities are similar regardless of the target force. In addition, for different objects, the velocity results are also similar. For example, when S_v is 0.45 mm and F_d is 0.4 N, the mean velocities for the three objects are 4.63 mm/s, 4.59 mm/s and 4.43 mm/s respectively. Therefore, with different S_v values, the overall exploration velocity can be controlled during each haptic exploration on various objects, as shown in Fig. 17. This is because when S_v is used for deciding the next moving location, the increment along the tangential direction of the contact surface on the object decides the moving velocity when the used time for every loop is the same (almost). The videos of “velocity_control-01 (MOESM8), velocity_control-02 (MOESM9).mp4” show the examples of the open-loop velocity control. However, the σ_v is relatively high with an average value of 2.94 mm/s

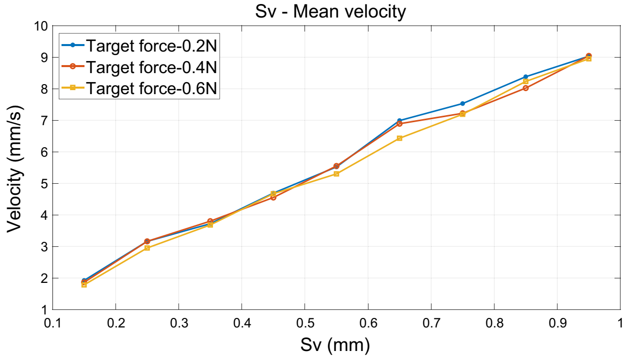


Fig. 17 The figure is showing the changing of the mean velocity of one exploration with relation to the S_v changes using the open-loop control. The velocity is obtained by getting the average of all sample data. It can be seen that as S_v increases, the velocity increases and the relation of these two values is close to a linear correlation regardless of the control target force. These curves show that S_v can be used to control the overall velocity of the exploration procedure

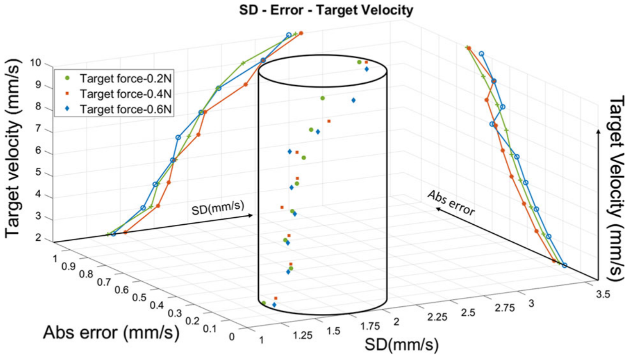


Fig. 18 The points in this figure are generated with the σ_v, Δv and v_d as the x, y and z coordinate. Δv is obtained by $abs(v_d - v_m)$. The curves on the left side are representing the relation between SD and v_d at different F_d, and curves on the right side are representing the relation between Δv and v_d at different F_d. It can be seen from the figure, the points are constrained in a cylinder with a larger radius compared to the force control results, showing the relative larger SD of the velocity control result. However, it also shows that the control method can be used to control the explore velocity with low absolute errors

when S_v is fixed. When using the PID controller, the σ_v of the velocity is around 2/3 of that when S_v is fixed. And an average absolute error Δv of 0.48 mm/s has been achieved, showing the good performance of the feedback velocity control.

With the feedback control, the performance of explore velocity control is not excellent as the σ_v is still not low enough with an average value of 2.07 mm/s. As shown in Fig. 18, the σ_v and Δv for each v_d are constrained in a transparent cylinder that is larger than the one in Fig. 14. It could be caused by two main reasons. The first is the unstructured surface of the object, which adds many difficulties to the control process, and the other is the limited frequency and relative long response time of the control software. Considering the limitations, the velocity control can be improved in two ways: (1) Increase the frequency of the overall control loop, which can help to update the error more rapidly and the platform can be controlled to take correction actions more quickly to eliminate the error. In this way, the overall control quality could be improved, especially when there are sudden surface curvature changes on the object. (2) Modify the mechanical design of the finger joints to allow smaller S_v value for deciding the next location. As shown in Table 6, a smaller S_v enables the system to respond to larger and quicker changes more smoothly. This works when the finger is controlled with very slow velocities and small target force. However, in this study, the position control of the joints becomes inaccurate when S_v is less than 0.1 mm due to the mechanical limitation of the motors and joints.

5 Conclusion

As the conclusion, with the proposed control method, the contact normal force and explore velocity during haptic surface exploration on objects with dynamic movement can be controlled adaptively, regardless of their surface and geometry properties. There are three points worthy of further research to improve the performance of the control method. First, during the exploration, if the force change generated by the movement of the device in the normal force direction is not enough to correct the error, the adjustment will be delayed; and if the force change is too large, it will generate overshoots, inducing more fluctuations to the system. Thus, understanding the indentation and force changes relation is essential to make the control more precise. For unknown objects, this relation can be obtained first as a pre-information and fed back to the system to adjust the related parameters. Also, this can help us understand the stiffness of the object, presenting more information about the contacting object. Second, it is important to understand the best control force range of each device, since the force ranges generated by different devices are varied. By selecting or designing different devices or platforms, different forces can be controlled precisely. Third, because the S_v ranges, what each device can achieve are different, the control resolution (force change per movement) is also varied for each device. To understand the resolution of the force control for each specific device is also important to achieve more robust and precise control.

As the extensions of this work, we will focus on the active haptic exploration, perception and soft contact. This proposed control method can be applied to any situations where contact happens, and the contact force is critical to the contacting object. Also, if the normal force and velocity can be controlled, it is helpful to implement different manipulation tasks that require different contact forces and explore velocities. This will lead to an active exploration method that can change exploration strategies (different force or velocity) actively depending on the utility function for more efficient object recognition or manipulation tasks. In addition, we will test fingertips made of softer material that can deform during contact, which will be helpful when contacting soft objects like tissues inside the organs or fragile daily objects. Moreover, more fingers will be created to form a manipulator, combing the robot arm, the whole platform can be used to carry out dexterous manipulations with controllable force and velocity. As reported in Okamura and Cutkosky (2001), with controllable haptic explorations, dexterous manipulation can be realised with grasping, holding and other hand controls. Finally, by combing the haptic feedback with the surgery devices, it will be beneficial in robotic surgery to help the surgeon to monitor and control the amount of force applied to minimise the risk of tissue damage (Amirabdollahian et al. 2017).

Acknowledgements This work is supported by the EPSRC (No. EP/N020421/1) and NSFC (No. 51520105006). And we would like to thank Dr. Hak Keung Lam for his kind suggestions on fuzzy control method.

References

Amirabdollahian, F., Livatino, S., Vahedi, B., Gudipati, R., Sheen, P., Gawrie-Mohan, S., et al. (2017). Prevalence of haptic feedback in robot-mediated surgery: A systematic review of literature. *Journal of Robotic Surgery*, *12*, 11–25.

Aviles, A. I., Alsaleh, S. M., Montseny, E., Sobrevilla, P., & Casals, A. (2016). A deep-neuro-fuzzy approach for estimating the interaction forces in robotic surgery. In *IEEE international conference on fuzzy systems (FUZZ-IEEE)* (pp. 1113–1119).

Back, J., Bimbo, J., Noh, Y., Seneviratne, L., Althoefer, K., & Liu, H. (2015). Control a contact sensing finger for surface haptic exploration. In *IEEE international conference on robotics and automation (ICRA)* (pp. 2736–2741).

Bicchi, A., Salisbury, J. K., & Brock, D. L. (1993). Contact sensing from force measurements. *The International Journal of Robotics Research*, *12*, 249–262.

Ciotti, S., Sun, T., Battaglia, E., Bicchi, A., Liu, H., & Bianchi, M. (2019). Soft tactile sensing: Retrieving force, torque and contact point information from deformable surfaces. In: *2019 IEEE international conference on robotics and automation, ICRA*.

da Fonseca, V. P., de Oliveira, T. E. A., Eyre, K., & Petriu, E. M. (2017b). Stable grasping and object reorientation with a three-fingered robotic hand. In *2017 IEEE international symposium on robotics and intelligent sensors (IRIS)* (pp. 311–317).

da Fonseca, V. P., Kucherhan, D. J., de Oliveira, T. E. A., Zhi, D., & Petriu, E. M. (2017a). Fuzzy controlled object manipulation using a three-fingered robotic hand. In *2017 Annual IEEE international systems conference (SysCon)* (pp. 1–6).

de Silva, C. W. (1995). Intelligent control: fuzzy logic applications. https://academic.microsoft.com/paper/598128971

Fishel, J. A., & Loeb, G. E. (2012). Bayesian exploration for intelligent identification of textures. *Frontiers in Neurorobotics*, *6*, 4.

Hellman, R. B., Tekin, C., van der Schaar, M., & Santos, V. J. (2018). Functional contour-following via haptic perception and reinforcement learning. *IEEE Transactions on Haptics*, *11*, 61–72.

Jamil, M. H., Annor, P. S., Sharfman, J., Parthesius, R., Garachon, I., & Eid, M. (2018). The role of haptics in digital archaeology and heritage recording processes—IEEE conference publication. In *IEEE international symposium on haptic, audio and visual environments and games (HAVE)*.

Kappassov, Z., Corrales, J. A., & Perdereau, V. (2015). Tactile sensing in dexterous robot hands—Review. *Robotics and Autonomous Systems*, *74*, 195–220.

Konstantinova, J., Li, M., Mehra, G., Dasgupta, P., Althoefer, K., & Nanayakkara, T. (2014). Behavioral characteristics of manual palpation to localize hard nodules in soft tissues. *IEEE Transactions on Biomedical Engineering*, *61*, 1651–1659.

Lederman, S. J., & Klatzky, R. L. (1993). Extracting object properties through haptic exploration. *Acta Psychologica*, *84*, 29–40.

Lederman, S. J., & Klatzky, R. L. (2009). Haptic perception: A tutorial. *Attention, Perception and Psychophysics*, *71*, 1439–1459.

Lepora, N. F., Aquilina, K., & Cramphorn, L. (2017). Exploratory tactile servoing with active touch. *IEEE Robotics and Automation Letters*, *2*, 1156–1163.

Lepora, N. F., Church, A., de Kerckhove, C., Hadsell, R., & Lloyd, J. (2019). From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, *4*, 2101–2107.

Li, Q., Schürmann, C., Haschke, R., & Ritter, H. J. (2013). A control framework for tactile servoing. In *Robotics: Science and systems 2013*, Vol. 9, https://academic.microsoft.com/paper/102778925.

Liu, H., Song, X., Bimbo, J., Seneviratne, L., & Althoefer, K. (2012). Surface material recognition through haptic exploration using an intelligent contact sensing finger. In *2012 IEEE/RSJ international conference on intelligent robots and systems*.

Liu, H., Nguyen, K. C., Perdereau, V., Bimbo, J., Back, J., Godden, M., et al. (2015). Finger contact sensing and the application in dexterous hand manipulation. *Autonomous Robots*, *39*, 25–41.

Okamura, A. M. (2000). *Haptic exploration of objects with rolling and sliding*. Ph.D. thesis, Stanford University, Department of Mechanical Engineering.

Okamura, A. M., & Cutkosky, M. R. (2001). Feature detection for haptic exploration with robotic fingers. *The International Journal of Robotics Research*, *20*, 925–938.

Okamura, A. M., & Cutkosky, M. R. (2001). Feature detection for haptic exploration with robotic fingers. *The International Journal of Robotics Research*, *20*, 925–938.

Rosales, C., Spinelli, F., Gabiccini, M., Zito, C., & Wyatt, J. L. (2018). Gpatlasrrt: A local tactile exploration planner for recovering the shape of novel objects. *International Journal of Humanoid Robotics*, *15*, 1850014.

Sommer, N., & Billard, A. (2016). Multi-contact haptic exploration and grasping with tactile sensors. *Robotics and Autonomous Systems*, *85*, 48–61.

Song, X., Liu, H., Althoefer, K., Nanayakkara, T., & Seneviratne, L. D. (2014). Efficient break-away friction ratio and slip prediction based on haptic surface exploration. *IEEE Transactions on Robotics*, *30*, 203–219.

Sornkarn, N., & Nanayakkara, T. (2017). Can a soft robotic probe use stiffness control like a human finger to improve efficacy of haptic perception? *IEEE Transactions on Haptics*, *10*, 183–195.

Su, Z., Fishel, J. A., Yamamoto, T., & Loeb, G. E. (2012). Use of tactile feedback to control exploratory movements to characterize object compliance. *Frontiers in Neurorobotics*, *6*, 7.

Takagi, T., & Sugeno, M. (1993). Fuzzy identification of systems and its applications to modeling and control. In *Readings in fuzzy sets for intelligent systems* (pp. 387–403).

Tanaka, K., & Wang, H. O. (2004). *Fuzzy control systems design and analysis: A linear matrix inequality approach*. New York: Wiley.

Tegin, J., & Wikander, J. (2005). Tactile sensing in intelligent robotic manipulation—A review. *Industrial Robot: An International Journal*, *32*, 64–70.

Ward-Cherrier, B., Pestell, N., Cramphorn, L., Winstone, B., Giannaccini, M. E., Rossiter, J., et al. (2018). The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft Robotics*, *5*, 216–227.

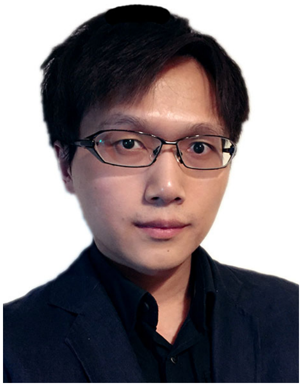
Xu, D., Loeb, G. E., & Fishel, J. A. (2013). Tactile identification of objects using Bayesian exploration. In *IEEE international conference on robotics and automation (ICRA)*.

Yang, C., & Lepora, N. F. (2017) Object exploration using vision and active touch. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Yang, C., Li, Z., & Burdet, E. (2013). Human like learning algorithm for simultaneous force control and haptic identification. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp 710–715).

Yin, J., Aspinall, P., Santos, V. J., & Posner, J. D. (2018). Measuring dynamic shear force and vibration with a bioinspired tactile sensor skin. *IEEE Sensors Journal*, *18*, 3544–3553.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Teng Sun received the B.Sc. degree in mechanical engineering and automation from Beijing Institute of Technology, Beijing, China, in 2011, and worked as an autonomous driving engineer in Nissan Advanced Technology Centre, Atsugi, Japan from 2011 to 2014. He is currently working toward the Ph.D. degree in robotics in the Haptic Mechatronics and Medical Robotics (HaMMeR) Laboratory, Centre for Robotics Research, Department of Informatics, King's College London, London, UK. His current research interests include contact location estimation based on intrinsic contact sensing, object recognition based on haptic information through Bayesian classifier, active perception and decision making with information gain and utility method.

Hongbin Liu is an Associate Professor (Senior Lecturer) in the school of biomedical engineering and imaging sciences, King's College London (KCL) where he is directing the Haptic Mechatronics and Medical Robotics Laboratory (HaMMeR). His lab designs robots with advanced haptic perception and interaction capabilities to address unmet needs in medicine, enabling safer and more effective diagnosis and treatment. Dr. Liu is a member of the IEEE, and a Technical Committee Member of IEEE EMBS BioRobotics. He has published over 100 peer-reviewed publications at top international robotic journals and conferences and is inventor for 4 patents. His research has been funded by EPSRC, Innovate UK, NHS Trust and EU Commissions, with strong industrial collaborations such as IBM, Ericsson. More information can be found on our online website: http://kclhammerlab.com/.