A Hybrid Force-Motion Control Framework for Human-Like Affective Touch Using a Brush-Equipped Robot Arm

○ Rui ZHANG, Student Member, The University of Tokyo, zhang_rui@lelab.t.u-tokyo.ac.jp
Koki HONDA, Member, The University of Tokyo
Rui FUKUI, Member, The University of Tokyo

Affective touch plays an important role in human social-emotional interaction. However, existing robotic systems lack closed-loop force control and real-time skin surface estimation, potentially limiting their human-likeness. This study proposes a two-phase brush-skin contact model and a hybrid force-motion control framework for realizing human-like affective touch using a brush-equipped robot arm. The system employs a cascade control architecture where the inner loop solves a Quadratic Programming (QP) problem to regulate brush-skin distance, brush angle, and tangential velocity, while the outer loop controls the normal contact force through a spring-based force-motion bridge. A history-based geometry estimator enables real-time surface normal estimation under visual occlusion caused by the brush itself. Simulation results on unknown curved surfaces demonstrate that the framework successfully decouples the physically coupled geometry and force parameters, and achieves accurate tracking of all four parameters simultaneously.

Key Words: Human-Robot Interaction, Force-Motion Control, Affective Touch

1 Introduction

The human-likeness of robots is a crucial factor in human-robot interaction (HRI). Increasing the human-likeness of robots can enhance user trust and associated emotional response[1]. Large language models (LLMs) and speech synthesis now enable natural verbal communication, facial expression generation technologies enable robots to express emotions, while advances in motion planning allow humanoid robots to replicate human-like locomotion. However, compared to verbal interaction, visual expression, and movement in the physical world, the domain of human-like pleasant physical contact between robots and humans remains underexplored[2].

Pleasant physical contact between humans has diverse forms, such as hugging, holding hands and patting[3]. Among these, affective touch, a pleasant and friendly touch, plays an important role in human social-emotional interaction. Previous studies have found a positive correlation between pleasantness and the firing rate of C-tactile afferents (CTs), which are found in hairy skin areas like the forearm and face. CTs are activated by both direct human touch and soft brush strokes at slow velocities (1–10 cm/s) with sub-Newton contact force[4]. Brush-initiated affective touch has several advantages for HRI. For example, brush strokes can be more easily parameterized and standardized as experimental stimuli compared to direct human hand contact. Brushes also excel in controllability and safety owing to the compliance of their hair. Therefore, this study focuses on human-like robotic affective touch delivered by a brush-equipped manipulator stroking the human forearm.

Previous studies have successfully delivered pleasant affective touch through different robots and devices. However, participants can still distinguish between human-delivered and robot-delivered brush strokes, and robots received lower human-likeness scores compared to humans[2]. It remains unclear how to enable robots to realize human-like affective touch. In our pre-experiment, we observed that humans look at the forearm and dynamically adjust their force when using a brush to stroke others, and we hypothesize that the lack of closed-loop force control and real-time surface reconstruction leads to the low human-likeness scores. Although several devices[5] aimed at studying affective touch or realizing affective touch via robots have been developed, none of them have achieved closed-loop force control, which is considered crucial for affective touch. Methods for skin surface geometry reconstruction have been relatively coarse, typically relying on sampling a few points on the forearm. To fill this gap, this study proposes a new control framework, develops a control system, and verifies its feasibility in simulation. A comparison between existing methods and our system is listed in Table 1.

In general, this study aims to develop a hybrid force-motion control framework that simultaneously achieves closed-loop force control and real-time skin surface estimation for brush-based robotic touch, toward improving the human-likeness of robot-delivered affective touch.

2 Contact Modeling and Problem Formulation for Brush-Based Robotic Affective Touch

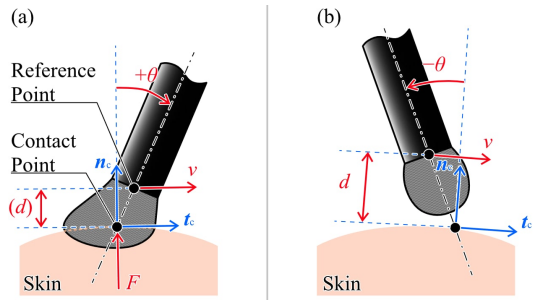

Fig.1 A two-phase contact model between brush and skin. (a) Contact phase (b) Non-contact phase

Due to the complexity of dynamics between brush and skin, we propose a simplified contact model (Fig. 1). The brush stroke is divided into two phases: the contact phase and the non-contact phase. In the contact phase, we propose a contact model that takes into consideration the angle of the brush and skin θ, the tangential sliding velocity v and the normal contact force F. The Reference Point (RP) is defined as the center point at the brush hair's root, the Contact Point (CP) is the intersection of the brush handle's axis and the skin. Contact normal vector $\boldsymbol{n}_c$ and contact tangential vector $\boldsymbol{t}_c$ are defined corresponding to the CP. In the non-contact phase, there is

Table 1 Comparison of different robotic touch systems

Method	Position Control	Posture Control	Force Control	Trajectory Following	Skin Geometry Estimation
RTS[5]	Limited	Limited	Pre-calibration	No	No
Cable Robot[5]	Yes	No	Open-loop	Yes	Pre-calibration
DMPs[2]	Yes	Implicit	No	Yes	Pre-calibration
Our System	Yes	Explicit	Real-time	Yes	Real-time

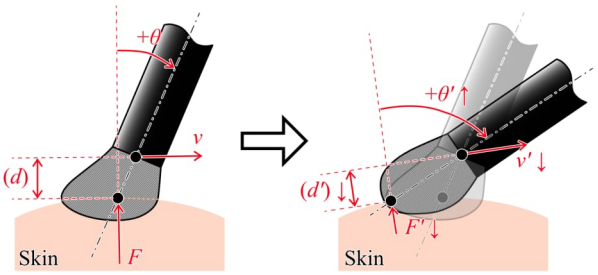

Fig.2 A diagram demonstrating the coupling problem of the parameters

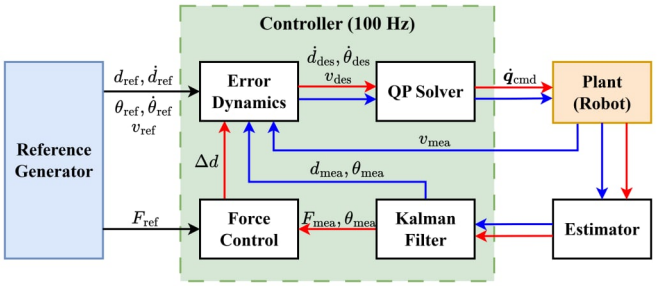

Fig.3 The control architecture diagram

no contact force; instead, we define the normal distance d between the brush and skin, thereby fully determining the current state of the brush. All parameters are defined in the parasagittal plane of forearm.

There are several challenges regarding such a control system. First, the parameters are coupled. As shown in Fig. 2, if the brush tilts around RP, θ changes, resulting in change of CP and $\boldsymbol{n}_c$, $\boldsymbol{t}_c$, and d, v, F change accordingly. Second, the parameters span different dimensions, specifically position, orientation, velocity, and force. Third, several safety constraints must be satisfied. For example, the joint velocity constraint and the planar constraint must be respected. Fourth, the geometry of the skin is unknown, and the participant's arm may move during the stroke. The CP is also occluded by the brush hair, making it difficult to obtain the current contact geometry.

In section 3, we will discuss the design of the control architecture and the implementation of the control system that addresses the challenges above.

3 Design of Cascade Force-Motion Control System

3.1 Approach Overview

For the first challenge, we designed a cascade controller that controls d, θ and v in the its inner loop and F in its outer loop, to decouple these parameters as possible. All parameters are controlled in closed-loop. The inner loop is modeled as a quadratic programming (QP) problem to make sure there will be always a solution. For the second challenge, we modeled the brush hair as a spring to bridge F and d, designed error dynamics to convert d, θ to $\dot{d}$ and $\dot{\theta}$, so that the QP problem can be constructed within velocity level. For the third challenge, we added constraints in QP problem to ensure the solution satisfies these constraints mathematically. For the fourth challenge, we introduce an eye-in-hand depth camera that is mounted beside the brush. We propose a history data-based matching algorithm that works as an estimator. The overall system architecture is shown in Fig. 3.

3.2 Control System Design

Hereafter, subscript $\cdot_{\text{ref}}$ denotes reference values (trajectory inputs), $\cdot_{\text{des}}$ denotes desired values computed as intermediate variables for the QP formulation, and $\cdot_{\text{mea}}$ denotes measured or estimated values obtained from sensors.

The control system is a cascade controller with a motion controller as inner loop (shown in Fig. 3 blue arrows) and a force controller as outer loop (shown in Fig. 3 red arrows). In non-contact phase, the force controller is not activated. The error dynamics of d and θ are given by

$$\dot{e}(t) = -ke(t), \quad e(t) = x - x_{\text{ref}} \tag{1}$$

where x is d or θ, k is the gain. To lessen phase lag and improve tracking performance, we add feed-forward term $\dot{d}_{\text{ref}}$ and $\dot{\theta}_{\text{ref}}$ and obtain

$$\begin{aligned} \dot{d}_{\text{des}} &= \dot{d}_{\text{ref}} + k_d(d_{\text{ref}} - d_{\text{mea}}) \\ \dot{\theta}_{\text{des}} &= \dot{\theta}_{\text{ref}} + k_\theta(\theta_{\text{ref}} - \theta_{\text{mea}}) \end{aligned} \tag{2}$$

where k_d and k_θ are gains. For v, we use proportional control for its error and add a feed-forward term. Since v is itself a velocity-level quantity, the feed-forward term is v_{ref} directly. With gain k_v, the control law is given by

$$v_{\text{des}} = v_{\text{ref}} + k_v(v_{\text{ref}} - v_{\text{mea}}) \tag{3}$$

To satisfy the constraints, the inner loop is modeled as a QP problem that calculates the optimal joint velocity $\dot{\boldsymbol{q}}$. From forward kinematics we know

$$\begin{bmatrix} \boldsymbol{v}_{\text{EE}} \\ \boldsymbol{\omega}_{\text{EE}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_v(\boldsymbol{q}) \\ \boldsymbol{J}_\omega(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} \tag{4}$$

where $\boldsymbol{v}_{\text{EE}}$, $\boldsymbol{\omega}_{\text{EE}}$ are the velocity and angular velocity of the end effector (EE), respectively. Define the state variable and the task Jacobian as

$$\boldsymbol{r} = \begin{bmatrix} \dot{d} \\ \dot{\theta} \\ v \end{bmatrix}, \quad \boldsymbol{J}_{\text{task}} = \begin{bmatrix} \boldsymbol{n}_c^\top \boldsymbol{J}_{v,\text{RP}}(\boldsymbol{q}) \\ -\boldsymbol{e}_x^\top \boldsymbol{J}_\omega(\boldsymbol{q}) \\ \boldsymbol{t}_c^\top \boldsymbol{J}_{v,\text{RP}}(\boldsymbol{q}) \end{bmatrix} \tag{5}$$

We want to obtain the optimal $\dot{\boldsymbol{q}}$ and set it as the current command $\dot{\boldsymbol{q}}_{\text{cmd}}$. The joint acceleration is also desired to be minimized. The function can be written as

$$\begin{aligned} \min_{\dot{\boldsymbol{q}}_{\text{cmd}}} \quad & \frac{1}{2}\dot{\boldsymbol{q}}_{\text{cmd}}^\top(\boldsymbol{J}_{\text{task}}^\top \boldsymbol{W}_{\text{task}} \boldsymbol{J}_{\text{task}} + \boldsymbol{\Lambda}^\top \boldsymbol{\Lambda})\dot{\boldsymbol{q}}_{\text{cmd}} \\ & - (\boldsymbol{r}_{\text{des}}^\top \boldsymbol{W}_{\text{task}} \boldsymbol{J}_{\text{task}} + \boldsymbol{\Lambda}^\top \boldsymbol{\Lambda}\dot{\boldsymbol{q}}_{\text{mea}})\dot{\boldsymbol{q}}_{\text{cmd}} \\ \text{s.t.} \quad & \boldsymbol{l} \le \boldsymbol{A}\dot{\boldsymbol{q}}_{\text{cmd}} \le \boldsymbol{u} \end{aligned} \tag{6}$$

where $\boldsymbol{W}_{\text{task}} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Lambda}$ are weight matrices. The constraint matrix, $\boldsymbol{A}$, includes two parts. First, the joint velocity must not exceed certain value, that is

$$\boldsymbol{l}_{\text{vel}} \leq \boldsymbol{I}\dot{\boldsymbol{q}}_{\text{cmd}} \leq \boldsymbol{u}_{\text{vel}} \tag{7}$$

where $\boldsymbol{l}_{\text{vel}}$ and $\boldsymbol{u}_{\text{vel}}$ are the lower and the upper limit vector of joint velocity, respectively. Second, the brush must be constrained to the parasagittal plane. This can be done by constraining RP to a certain plane and aligning an fixed axis in RP's local coordinate with one in the world coordinate. To constrain the RP to the parasagittal plane, we design a first-order error dynamics on its position along the plane-normal direction. Let x_0 denote the desired coordinate of the RP along that axis, and define the error $e = x - x_0$. Imposing $\dot{e} = -k_x e$ yields the desired velocity $\dot{x}_{\text{des}} = -k_x(x - x_0)$. This is then enforced as an inequality constraint with a small tolerance m:

$$\dot{x}_{\text{des}} - m \leq \boldsymbol{J}_x(\boldsymbol{q})\,\dot{\boldsymbol{q}} \leq \dot{x}_{\text{des}} + m \tag{8}$$

In addition, a fixed axis $\boldsymbol{a}$ in the RP's local frame (e.g., its y-axis) must be aligned with a designated world-frame axis (e.g., $\hat{\boldsymbol{x}}$). Since $\boldsymbol{a}$ is rigidly attached to the RP, its time derivative in the world frame satisfies $\dot{\boldsymbol{a}} = \boldsymbol{\omega} \times \boldsymbol{a} = -[\boldsymbol{a}]_\times\,\boldsymbol{\omega}$, where $\boldsymbol{\omega}$ is the angular velocity of the RP and $[\cdot]_\times$ denotes the skew-symmetric matrix. We constrain the projections of $\boldsymbol{a}$ onto the two orthogonal axes to be zero. For example, along the $\hat{\boldsymbol{y}}$ direction, we define the error $e_y = \hat{\boldsymbol{y}} \cdot \boldsymbol{a}$ and let $\dot{e}_y = -k_y\, e_y$. Expanding the time derivative gives the linear constraint:

$$\hat{\boldsymbol{y}}^\top [\boldsymbol{a}]_\times\, \boldsymbol{J}_\omega(\boldsymbol{q})\,\dot{\boldsymbol{q}} = k_y\,(\hat{\boldsymbol{y}}^\top \boldsymbol{a}) \tag{9}$$

The constraint along the $\hat{\boldsymbol{z}}$ direction is formulated analogously. Together, these two constraints drive $\boldsymbol{a}$ toward alignment with $\hat{\boldsymbol{x}}$ without singularity issues.

In contact phase, the outer loop, force controller is activated. We model the brush hair as a linear spring with stiffness k_b, so the normal contact force satisfies $F \approx k_b \cdot d_{\text{pen}}$, where d_{pen} is the penetration depth of the brush into the skin. Under this assumption, adjusting d directly modulates F, which bridges the force and motion domains. Since F is considered more important than d in this phase in regard to both pleasantness and safety, the force controller calculates the desired adjustment Δd and adds it to d_{ref}. To do this, define $e_F = F_{\text{ref}} - F_{\text{ema}}$ and construct a PID for e_F

$$-k_b \Delta d = K_p e_F + K_i \int e_F + K_d \dot{e}_F \tag{10}$$

where k_b is the stiffness coefficient of brush hair.

3.3 Contact Geometry Estimation

The contact point $\boldsymbol{p}_{\text{CP}}$, surface normal $\boldsymbol{n}_c$, and surface tangent $\boldsymbol{t}_c$ must be estimated in real time for closed-loop control. However, the CP is occluded by the brush hair, making direct observation difficult. We propose a two-stage estimation pipeline using depth camera point clouds, operating across three coordinate frames: the world frame, the brush handle frame, and the depth camera frame.

History-based surface point matching. The primary method maintains a temporal queue of surface points estimated from successive point cloud measurements. Since the depth camera is mounted on the front side of the brush, surface geometry ahead of the contact region can be measured in advance and stored before the brush arrives, enabling retrospective matching algorithm. At each control cycle, the point cloud is cropped in the camera frame, and the centroid and slope of the cropped region are computed on the brushing plane (the xz-plane of the brush handle frame). The result is transformed into the world frame and appended to the queue. To identify the current contact point, the queue is traversed from newest to oldest. For each stored surface point $\boldsymbol{p}_{\text{SP}}$, the direction vector is computed as $\boldsymbol{v}_{\text{dir}} = \boldsymbol{p}_{\text{SP}} - \boldsymbol{p}_{\text{RP}}$, where $\boldsymbol{p}_{\text{RP}}$ is the current RP position. The point is accepted as the current contact point if the dot product between the normalized $\boldsymbol{v}_{\text{dir}}$ and the brush z-axis exceeds a predefined threshold.

Fallback: line-fitting-based estimation. When the above method fails for a sustained number of consecutive cycles, the system falls back to a direct line-fitting approach. Each point in the raw cloud is transformed into the brush handle frame and projected onto the xz-plane. A line is fitted to the projected points to yield $\boldsymbol{p}_{\text{CP}}$, $\boldsymbol{n}_c$, and $\boldsymbol{t}_c$, which are then transformed into the world frame. The estimated normal and tangent are negated along the corresponding axes to align with the outward-pointing convention.

Kalman filtering. The raw estimates from either method are noisy and the depth camera publishes point clouds at around 60 Hz. To obtain temporally smooth results at around 100 Hz (which matches the controller's frequency), a standard discrete-time Kalman filter with a constant-velocity model is applied independently to each geometric quantity ($\boldsymbol{p}_{\text{CP}}$, $\boldsymbol{n}_c$, and $\boldsymbol{t}_c$). The six-dimensional state consists of the 3D position (or vector value) and its velocity; only the position is directly observed. The filtered estimates are then passed to the downstream controller.

4 Simulation Experiment for Decoupling Ability and Tracking Accuracy Evaluation

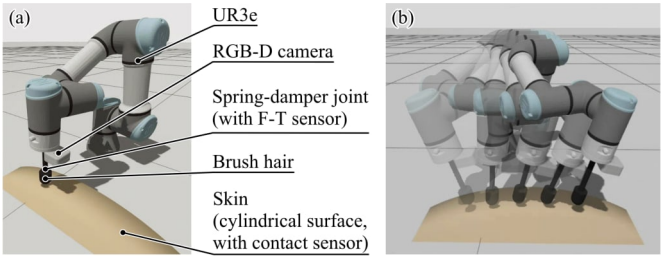

Fig.4 (a) Experiment setup in Gazebo (b) Snapshots of the experiment

4.1 Simulation Setup

The whole system is developed under ROS2 Jazzy framework, in Ubuntu 24.04 platform. The simulation experiment was conducted in Gazebo Harmonic environment. As shown in Fig. 4 (a), a UR3e robotic arm is used for brush manipulation. Our custom control node calculates the optimal joint velocity commands and send them to the robotic arm in simulator through `forward_command_controller` interface. The skin is modeled as a cylindrical surface. A contact sensor is attached to the cylinder to obtain the ground truth contact information during contact. The brush hair and brush handle is connected with spring-damper prismatic joint to simulate the compliance of real brush hair. A six-axis force-torque sensor is attached to this joint. An RGB-D camera is mounted besides the brush.

The parameters are set as follows: $k_d = 10.0$, $k_\theta = 8.0$, $k_v = 0.65$, $K_p = 10.0$, $K_i = 70.0$, $K_d = 0.16$, and $\boldsymbol{W}_{\text{task}} = \text{diag}(1.0, 5.0, 1.0)$.

4.2 Results: Decoupling Ability, Estimation Accuracy, and Tracking Performance

The experiment takes sinusoidal reference as input, records the sensor's output data and verifies 1) the decoupling ability; 2) the estimation accuracy and 3) the tracking accuracy. The

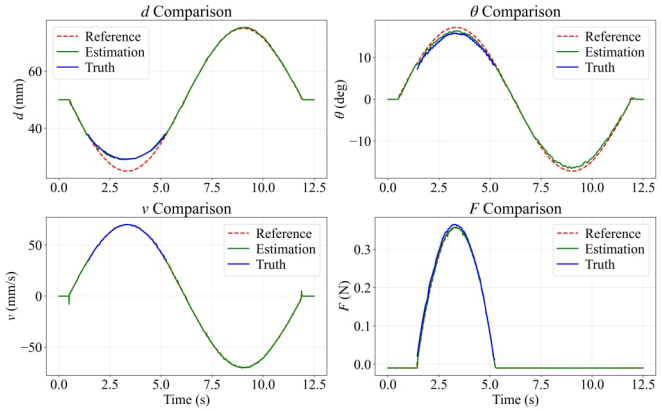

Fig.5 The comparison result between the reference, estimation and ground truth data

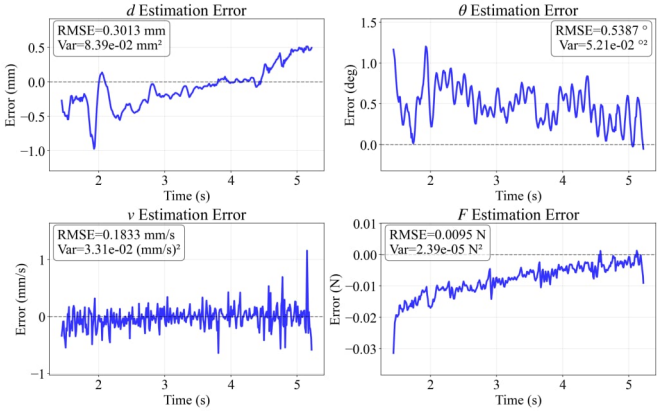

Fig.6 The estimator performance during contact phase

snapshots of the experiment is shown in Fig. 4 (b), the result of the experiment is shown in Fig. 5.

The decoupling ability. To evaluate the decoupling ability, d_{ref} is set such that its corresponding penetration depth under the spring model would produce a force different from F_{ref}, forcing the force controller to actively adjust d. As shown in Fig. 5, during the contact phase (approximately 1.5-5.5 s), the actual d deviates from d_{ref} as the force controller overrides the distance reference to track F_{ref}. Meanwhile, θ and v continue to follow their respective references with minimal disturbance, confirming that the cascade architecture successfully decouples force regulation from motion tracking.

The estimation accuracy. Since a contact sensor is attached to the skin surface, ground truth contact geometry is available during the contact phase. The estimation error, defined as the difference between the estimated and ground truth values ($\cdot_{\text{est}} - \cdot_{\text{truth}}$), is plotted in Fig. 6. The RMSE values are about 0.30 mm for d, 0.54° for θ, 0.18 mm/s for v and 0.0095 N for F. Notably, the force estimation error is well below the sub-Newton range and the velocity estimation error is negligible compared to the 1-10 cm/s operating range. These results confirm that the history-based estimator provides sufficient accuracy for closed-loop affective touch control.

The tracking accuracy. The tracking error, defined as the difference between the estimated value and the reference ($\cdot_{\text{est}} - \cdot_{\text{ref}}$), is plotted in Fig. 7. Note that during the contact phase d is governed by the force controller rather than d_{ref}, and during the non-contact phase F is not actively controlled; the corresponding segments are therefore omitted in the figure. In the non-contact phase, d is mostly tracked within ±0.5 mm and θ mostly within ±1° in steady state. The velocity v is tracked within ±2 mm/s except the peaks during transient

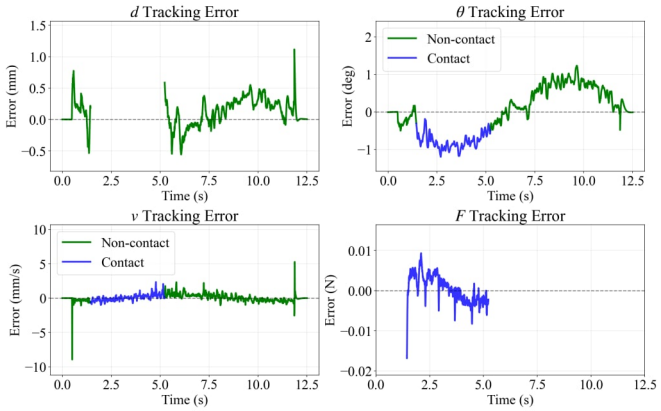

Fig.7 The tracking performance during the whole stroke

states. During the contact phase, the force tracking error remains within ±0.01 N after the initial transient, demonstrating that the controller can maintain sub-Newton-level force precision required for affective touch.

5 Conclusion

In this paper, we propose a two-phase brush-skin contact model that characterizes affective touch delivered by a brush-equipped robotic arm. In contact phase, the touch is modeled by the brush-skin angle θ, the tangential sliding velocity v, and the normal contact force F; in non-contact phase, the brush's state is determined by θ, v and the brush-skin normal distance d. A cascade control system with motion control as inner loop and force control as outer loop is developed to simultaneously regulates d, θ, v and F. In simulation, we demonstrated that these coupled parameters can be simultaneously tracked in real-time by robots on unknown surfaces.

The results show that 1) Our cascade and QP based force-motion control framework successfully resolves the physical coupling between geometry parameters and contact force; 2) History data based estimator is proven effective for estimating surface normals when contact point is occluded by the tool itself. Since our system achieves closed-loop force control and real-time surface estimation, both of which were absent in previous devices, it is expected to improve the human-likeness of robotic affective touch and achieve higher human-likeness scores in future user studies.

Future work includes real robot implementation, learning strategy from human demonstration and integration with user subjective evaluation.

References

[1] Cominelli, L., Feri, F., Garofalo, R. et al. Promises and trust in human–robot interaction. *Sci Rep*, 11, 9687, 2021.

[2] Tang, Yucheng, et al, "Evaluation of the Human-Like Robotic Touch: a user study." *In Proc. of IEEE International Conference on Rehabilitation Robotics*, pp. 1353-1360, 2025.

[3] M. Teyssier, G. Bailly, C. Pelachaud and E. Lecolinet, "Conveying Emotions Through Device-Initiated Touch," *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1477-1488, 2022.

[4] McGlone, Francis, Johan Wessberg, and Håkan Olausson. "Discriminative and affective touch: sensing and feeling." *Neuron*, vol. 82, no. 4, pp. 737-755, 2014.

[5] Schirmer, Annett, et al, "Stroking trajectory shapes velocity effects on pleasantness and other touch percepts." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 49, no. 1, pp. 71-86, 2023.