Hybrid Force Motion Control with Estimated Surface Normal for Manufacturing Applications

Ehsan Nasiri and Long Wang

Abstract— **This paper proposes a hybrid force-motion framework that utilizes real-time surface normal updates. The surface normal is estimated via a novel method that leverages force sensing measurements and velocity commands to compensate the friction bias.**

This approach is critical for robust execution of precision force-controlled tasks in manufacturing, such as thermoplastic tape replacement that traces surfaces or paths on a workpiece subject to uncertainties deviated from the model.

We formulated the proposed method and implemented the framework in *ROS2* environment. The approach was validated using kinematic simulations and a hardware platform. Specifically, we demonstrated the approach on a 7-DoF manipulator equipped with a force/torque sensor at the end-effector.

I. INTRODUCTION

The utilization of robot force control across various sectors such as manufacturing, healthcare, and more is on the rise. On the other hand, force control tasks on a workpiece represent a complex challenge, requiring the robot to dynamically and autonomously adjust its path. Such adaptation is essential to manage the uncertainties encountered in the workpiece during the manufacturing process [1], [2]. In numerous scenarios, robots are required to trace a contour or path while exerting a force perpendicular to the surface. This capability is particularly relevant in processes such as grinding [3], thermoplastic tape replacement [4] and injection molding [5], [6].

Many researchers have presented hybrid methods for the control of position and force in the manufacturing process, for instance [7]–[9]. Solanes et al. use the hybrid force-motion control of robots for surface polishing, employing a task priority method [10]. In [11], the authors proposed an adaptive hybrid force-motion control method to guarantee asymptotic tracking of desired motion-force trajectories in an unknown environment. The application of a Kalman filter-based hybrid force-motion control to estimate the actual contact point has been addressed in [12]. The development of an adaptive fuzzy control-based hybrid framework, aimed at improving the manipulator's performance in uncertain environments, has been presented in [13]. One of our envisioned future applications is thermoplastic fiber layup for 3D surfaces of workpieces with complex geometry.

Precise hybrid motion-force control is greatly sought after in robotic manufacturing processes. However, such precision is constrained and affected by environmental factors

E. Nasiri and L. Wang are with the Department of Mechanical Engineering, Stevens Institute of Technology, New Jersey, NJ, 07030, USA, e-mail: long.wang@stevens.edu
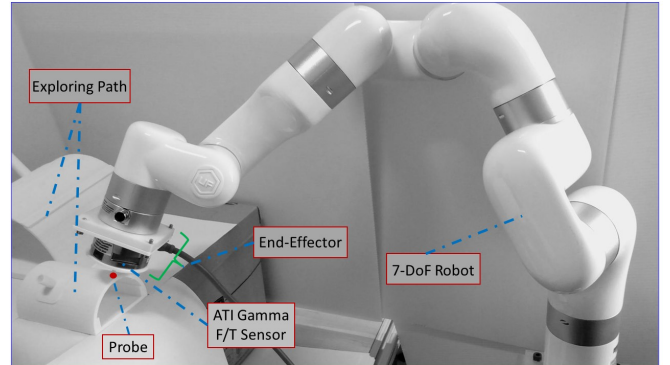
Fig. 1: The Robotic System Framework: A 7-DoF Manipulator Equipped with an F/T Sensor and Probe at the End-Effector, Including Workpieces.

like friction. Thus, friction compensation becomes essential for the effectiveness of high-precision motion-force control systems. Although there is a broad spectrum of methods for friction compensation in robotics, as documented in the literature, references such as [14]–[20], [27]–[29], indicate that diverse friction models, essential for this purpose, have been explored (these methods can broadly be categorized into two main types: static models and dynamic models [23]–[26]). Yet, there are very few studies specifically addressing friction compensation in hybrid force-motion control. The challenges associated with friction compensation at joint-level for hybrid force-motion control, have been the subject of research over the past years, as in [21], [22].

When a robot manipulator's end-effector comes into contact with an environment, there is a dynamic interaction between the manipulator and the environment. This interaction produces reaction forces that need to be managed effectively at the contact point. Given the complexities inherent in advanced industrial processes, it is crucial to investigate surface normal estimation. Therefore, accurately estimating this norm, especially considering the friction on contact surfaces, is essential for precise manufacturing. Failing to do so can lead to undesired outcomes, potentially causing the task to fail. So far, only a few studies have concentrated on motion-force control that involves estimating the surface normal during the trajectory. Some do not use real-time sensory feedback as in [30] while others, such as [31], do not account for friction in their surface normal estimation.

In this study, we employ a 7-DoF manipulator equipped with an F/T sensor and a probe at the end-effector. This setup is used to execute tasks on various workpieces us-

TABLE I: Nomenclature for Kinematics

Symbol	Description
ee	End-effector.
$\mathbf{x}_{ee}$	Probe's center position.
$\boldsymbol{J}_{ee}$	Jacobian of the end-effector (ee).
$\boldsymbol{e}_{ee}$	Probe (end-effector) error.
$\boldsymbol{K}_{ee}$	Proportional 3×3 diagonal gain.
$\boldsymbol{K}_{adm}$	Admittance gain.
$\boldsymbol{K}_m, \boldsymbol{K}_f$	Diagonal 3×3 gains in hybrid force-motion control command.
$\boldsymbol{K}_{m\perp}, \boldsymbol{K}_{f\perp}$	Projected positional and derivative, diagonal gains in hybrid force-motion control command.
$\boldsymbol{N}, \boldsymbol{T}$	Desired force and motion control directions.
$\hat{\boldsymbol{n}}_{surf}, \boldsymbol{n}_{ee}$	Estimation of surface normal and end-effector directions.
γ	The angle between the estimated surface normal and the estimated probe directions.
μ	The Coulomb coefficient of friction.
$\boldsymbol{f}_{des}$	The reference desired force.
$\boldsymbol{f}_\tau$	The surface friction force.
$\boldsymbol{f}_s$	Force sensor data.
$\boldsymbol{f}_\perp$	The projected surface friction force onto "null space" of robot velocity.
$\boldsymbol{f}_v$	The projected surface friction onto robot velocity.
$\boldsymbol{\Omega}_m, \boldsymbol{\Omega}_f$	Motion and force projection matrices.
ρ	A criterion that could be chosen to maximize an objective function g, within the scope of resolving the robot's redundancy.
$\vec{\boldsymbol{\Phi}}$	The configuration space velocity.

ing the proposed hybrid force-motion control strategy. The robot utilizes a novel algorithm primarily to estimate the direction of the surface normal in real-time, as detailed in this study, while also determining the Coulomb coefficient of the surface. This method leverages force sensor readings to compensate for surface friction, thereby facilitating the estimation of the surface normal. The contributions of this work include:

- A novel hybrid force-motion control framework is introduced, featuring a newly developed methodology to estimate surface normal. This method computes the surface normal by constructing a surface friction model. It does so by utilizing the force sensor feedback and estimating the surface Coulomb coefficient.
- Validations were carried out in simulations and experiments, on a 7-DoF robot manipulator equipped with an F/T sensor.

II. MODELING OF THE ROBOTIC SYSTEM

A. Kinematic Modeling

Throughout this paper, we adopt the nomenclature in TableI. Our robotic system framework, as shown in Fig.1, is developed within the *ROS2* environment. The 7-DoF robotic system manipulates a force-sensing probe attached to the end-effector, comprising an F/T sensor.

The velocity of the robot's end-effector, denoted as $\dot{\mathbf{x}}_{ee}$, is given by,

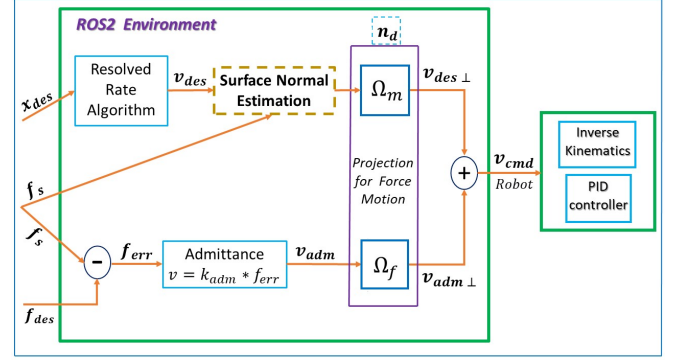$$\dot{\mathbf{x}}_{ee} = \mathbf{J}_{ee}(\boldsymbol{\Phi})\,\dot{\boldsymbol{\Phi}} \tag{1}$$

Fig. 2: Hybrid Force-Motion Controller

Where, $\mathbf{J}_{ee}$ represents the manipulator's Jacobian matrix which maps joint velocities to the end-effector's velocity. The general solution that satisfies (1), is as follow:

$$\dot{\boldsymbol{\Phi}}_{cmd} = \mathbf{J}_{ee}^{\dagger}\dot{\mathbf{x}}_{ee} + \left(\mathbf{I} - \mathbf{J}_{ee}^{\dagger}\mathbf{J}_{ee}\right)\boldsymbol{\rho} \tag{2}$$

Here, we substitute the end-effector velocity $\dot{\mathbf{x}}_{ee}$, as follows,

$$\dot{\mathbf{x}}_{ee} = \mathbf{K}_{ee}\mathbf{e}_{ee}, \quad \mathbf{K}_{ee} \in \mathbb{R}^{3\times 3} \tag{3}$$

$$\mathbf{e}_{ee} = \mathbf{x}_{des} - \mathbf{x}_{ee} \tag{4}$$

ρ is a residue vector that can be selected to resolve the redundancy of the robot. In (2) the error, denoted as $\mathbf{e}_{ee}$, is defined as the distance between the probe (ee) and the targeted contact surface.

Considering the position of the contact point $\mathbf{x}_{cnt}$, it is defined as follows:

$$\mathbf{x}_{cnt} = \mathbf{x}_{ee} + \mathbf{n}_{surf}\,d, \quad d \in \mathbb{R} \tag{5}$$

An estimation form of equation (5) can be given by,

$$\hat{\mathbf{x}}_{cnt} = \mathbf{x}_{ee} + \hat{\mathbf{n}}_{surf}\,d \tag{6}$$

Here, d signifies a scalar offset. We managed to maintain an offset distance between the probes and the goal, for instance, a 5mm offset between the center point of the probe and the desired contact point along the z-axis. Then, (4) can be rewritten,

$$\mathbf{e}_{ee} = \mathbf{x}_{cnt,des} - \hat{\mathbf{x}}_{cnt} \tag{7}$$

Taking into account equations (4) to (7) and considering the offset, the error is expressed as follows:

$$\mathbf{e}_{ee} = (\mathbf{x}_{cnt,des} - \hat{\mathbf{x}}_{cnt}) - \hat{\mathbf{n}}_{surf}\,d \tag{8}$$

In the subsequent sections, we introduce a method for real-time estimation of the surface normal, denoted as $\hat{\mathbf{n}}_{surf}$. This method facilitates the redefinition of the end-effector error, as outlined in (8), thereby transitioning to a more robust approach. Ultimately, the simulation and experimental validations section will present a in-depth comparative evaluation, focusing on the accuracy of the proposed method. This will be illustrated through detailed plots and figures.

B. Hybrid Force-Motion Control

To facilitate hybrid force-motion, a dedicated node was developed within the ROS2 environment. This integration is schematically depicted in Fig.2. The mechanism for motion-force projection, derived by inspiration from the research conducted by [32] and [33].

$$\mathbf{\Omega}_f = \mathbf{N}(\mathbf{N}^{\mathrm{T}}\mathbf{N})^{-1}\mathbf{N}^{\mathrm{T}} = \mathbf{I} - \mathbf{\Omega}_m \quad (9)$$

$$\mathbf{\Omega}_m = \mathbf{T}(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}} = \mathbf{I} - \mathbf{\Omega}_f \quad (10)$$

In the context of manipulation, a velocity command, denoted by $\mathbf{v}_{\mathrm{des}}$, is formulated using the resolved rate algorithm, as described by Whitney [34]. The velocity command $\mathbf{v}_{\mathrm{des}}$, derived from the resolved rate control block, is subsequently input into the surface normal estimation block. Additionally, this block receives input from force sensor measurements. These measurements are crucial in estimating both the surface normal and the surface friction coefficient. The process of this estimation follows the methodology detailed in Algorithm1.

In addition, a separate velocity command, denoted as $\mathbf{v}_{\mathrm{adm}}$, is generated based on the force error, $\mathbf{f}_{\mathrm{err}}$, and an admittance gain, $\mathbf{K}_{\mathrm{adm}}$.

$$\mathbf{v}_{\mathrm{adm}} = \mathbf{K}_{\mathrm{adm}}\,\mathbf{f}_{\mathrm{err}}, \quad \mathbf{K}_{\mathrm{adm}} \in \mathbb{R}^{3\times3} \quad (11)$$

$$\mathbf{f}_{\mathrm{err}} = \mathbf{f}_{\mathrm{des}} - \mathbf{f}_s \quad (12)$$

As illustrated in Fig.2, the velocity vectors $\mathbf{v}_{\mathrm{des}}$ and $\mathbf{v}_{\mathrm{adm}}$ undergo projection onto their respective motion-force planes. These projections are facilitated by the force-motion projection matrices, as defined in (9),(10). This step results in the respective projected command velocities $\mathbf{v}_{\mathrm{des}\perp}$ and $\mathbf{v}_{\mathrm{adm}\perp}$. Following this, the combined velocity command $\mathbf{v}_{\mathrm{cmd}}$ is dispatched to the robot controller. This is executed through a ROS2 node, which publishes the requisite joint velocities.

It should be noticed that in (9), $\mathbf{N}$ is defined to be the desired force control direction (denoted by $\mathbf{n}_d$),

$$\mathbf{N} = \mathbf{n}_d = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^{\mathrm{T}} \quad (13)$$

The robot's inverse kinematics, along with the necessary PID controller for guiding the manipulator in hybrid force-motion control, are implemented in a manner similar to that described in (2).

According to Fig.2, the resulting hybrid force-motion control command in the configuration space can be expressed as follows:

$$\dot{\mathbf{\Phi}}_{\mathrm{cmd}} = \dot{\mathbf{\Phi}}_m + \dot{\mathbf{\Phi}}_f \quad (14)$$

Here, $\dot{\mathbf{\Phi}}_{\mathrm{cmd}}$ represents the resultant integrated motion and force velocity control commands.

$$\dot{\mathbf{\Phi}}_m = \mathbf{J}_{\mathrm{ee}}^{\dagger}(\dot{\mathbf{x}}_{\mathrm{des},m} + \boldsymbol{\xi}_m) \quad (15)$$

$$\dot{\mathbf{\Phi}}_f = \mathbf{J}_{\mathrm{ee}}^{\dagger}(\dot{\mathbf{x}}_{\mathrm{des},f} + \boldsymbol{\xi}_f) \quad (16)$$

$\dot{\mathbf{x}}_{\mathrm{des}}$ and $\boldsymbol{\xi}$, in (15),(16) are detailed as follows, referring to Fig.2:

$$\boldsymbol{\xi}_m = \mathbf{K}_{m\perp}\,\mathbf{e}_m \quad (17)$$

$$\dot{\mathbf{x}}_{\mathrm{des},m} = \mathbf{v}_{\mathrm{des}\perp} \quad (18)$$

$$\boldsymbol{\xi}_f = \mathbf{K}_{f\perp}\,\mathbf{e}_f \quad (19)$$

$$\dot{\mathbf{x}}_{\mathrm{des},f} = \mathbf{v}_{\mathrm{adm}\perp} \quad (20)$$

The projected gain $\mathbf{K}_{m\perp}$ is given by,

$$\mathbf{K}_{m\perp} = \mathbf{K}_m\,(\mathbf{I} - \mathbf{\Omega}_f), \quad \mathbf{K}_m \in \mathbb{R}^{3\times3} \quad (21)$$

Similar to (21), for the force velocity command (16), the gain is given as follows:

$$\mathbf{K}_{f\perp} = \mathbf{K}_f\,\mathbf{\Omega}_f, \quad \mathbf{K}_f \in \mathbb{R}^{3\times3} \quad (22)$$

Using equations (21) and (22), the motion-force errors, which are expressed as follows, can be projected onto the motion-force directions, respectively:

$$\mathbf{e}_m, \mathbf{e}_f = (\mathbf{x}_{\mathrm{des}} - \mathbf{x}_{\mathrm{ee}}) - \mathbf{d}_h, \quad \mathbf{d}_h \in \mathbb{R}^{3\times1} \quad (23)$$

$\mathbf{d}_h$ represent the offsets in the motion-force directions.

The term $\mathbf{v}_{\mathrm{cmd}}$, as illustrated in Fig.2, represents the integrated command velocity derived from the decoupled inputs of the force-motion control components. This velocity is defined as follows:

$$\mathbf{v}_{\mathrm{cmd}} = \mathbf{v}_{\mathrm{adm}\perp} + \mathbf{v}_{\mathrm{des}\perp} \quad (24)$$

Similar to (8), the error is computed at each step along the trajectory. Contrary to (8), subsequent error computations incorporate force sensor feedback. This feedback is utilized to estimate the surface normal, $\hat{\mathbf{n}}_{\mathrm{surf}}$, and to derive the projection matrices accordingly, thereby compensating for the surface friction bias. Following the above explanations, the configuration space velocity command for the hybrid force-motion control can be expressed as follows:

$$\dot{\mathbf{\Phi}}_{\mathrm{cmd}} = \mathbf{J}_{\mathrm{ee}}^{\dagger}(\mathbf{v}_{\mathrm{cmd}} + \boldsymbol{\xi}_h) + \left(\mathbf{I} - \mathbf{J}_{\mathrm{ee}}^{\dagger}\mathbf{J}_{\mathrm{ee}}\right)\boldsymbol{\rho} \quad (25)$$

Here, $\boldsymbol{\xi}_h$ is the precision control component of the hybrid force-motion control command and is defined as follows:

$$\boldsymbol{\xi}_h = \boldsymbol{\xi}_m + \boldsymbol{\xi}_f, \quad \boldsymbol{\xi}_h \in \mathbb{R}^{3\times1} \quad (26)$$

Substituting (17) and (19) into (26),

$$\boldsymbol{\xi}_h = \mathbf{K}_{m\perp}\mathbf{e}_m + \mathbf{K}_{f\perp}\mathbf{e}_f \quad (27)$$

The equation (25) can be implemented to direct the robot to follow a desired velocity, denoted as $\mathbf{v}_{\mathrm{cmd}}$, for the end-effector *ee*. The precision of this trajectory in motion-force directions is regulated by the gains, as defined in (27).

C. Estimation of Surface Normal Using Online-Estimated Friction

The novel method for surface normal estimation presented in this paper is a key component of the proposed hybrid force-motion framework. It dynamically updates the surface normal estimation in real-time, initially by constructing a surface friction model. This method primarily utilizes force sensing measurements and velocity commands to effectively compensate for friction bias. In doing so, it ensures that the force controller can utilize the most current estimates of the surface normal as well as the Coulomb coefficient for enhanced control accuracy. The algorithm adjusts the surface normal force, $\hat{\mathbf{f}}_n$, during the trajectory, as schematically shown in Fig.3.

Algorithm.1, shows the surface normal estimation method. In this approach, a feed-forward Coulomb friction force, denoted as $\mathbf{f}_\tau$, is computed. This computation relies on a projection force $\mathbf{f}_\perp$ and $\bar{\mu}$, where the latter represents a weighted moving average estimate of the friction coefficient. The force $\mathbf{f}_\perp$ is defined as the projection of the sensed force $\mathbf{f}_s$, onto the null space of the robot's velocity, which is represented by $\hat{\mathbf{v}}$. Consequently, the corrected surface normal force $\hat{\mathbf{f}}_n$, is derived by subtracting $\mathbf{f}_\tau$ from $\mathbf{f}_s$. The algorithm is executed at a frequency of 1 kHz.

In addition, at each time step, particularly at the kth step, an updated measurement μ_k is calculated using the two orthogonal components, $\mathbf{f}_\perp$ and $\mathbf{f}_v$. This measurement is subsequently integrated into the weighted moving average filter. The estimated surface normal direction is finally determined by normalizing $\hat{\mathbf{f}}_n$.
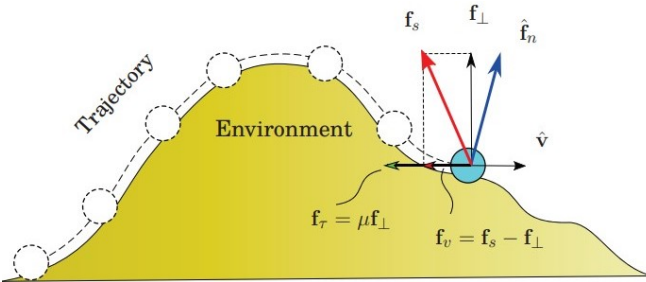
Fig. 3: Estimation of Surface Normal Force Compensating for Environmental Friction.

Following the above descriptions, along the trajectory, the F/T sensor reads the force data. An estimated velocity $\hat{\mathbf{v}}$, and the friction coefficient $\bar{\mu}$, are then considered, based on the data from the previous step. Subsequently, the force sensor reading $\mathbf{f}_s$ is projected onto the null space of $\hat{\mathbf{v}}$ using the following projection matrix.

$$\mathbf{\Omega_v} = \hat{\mathbf{v}}(\hat{\mathbf{v}}^{\mathrm{T}}\hat{\mathbf{v}})^{-1}\hat{\mathbf{v}}^{\mathrm{T}} \tag{28}$$

According to Fig.3, projection of sensed force $\mathbf{f}_s$ onto robot velocity is given by,

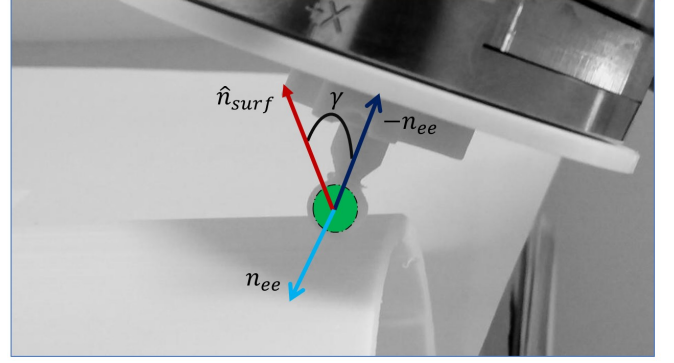$$\mathbf{f}_v = \mathbf{f}_s - \mathbf{f}_\perp \tag{29}$$

Fig. 4: Optimizing Probe Orientation: Utilizing Estimated Surface Normal and End-Effector Directions.

It is understood that $\mathbf{f}_v$ represents the projection of the sensed force onto the direction of the robot's velocity.

$$\mathbf{f}_v = \mathbf{\Omega_v}\mathbf{f}_s \tag{30}$$

Thus, the force sensor projection onto the null space of the robot's velocity can be given as follows:

$$\mathbf{f}_\perp = (\mathbf{I} - \mathbf{\Omega_v})\mathbf{f}_s \tag{31}$$

Then, a weighted moving average $\bar{\mu}$, is calculated as follows:

$$\bar{\mu} = \frac{1}{m}\sum_{i=1}^{m} w_i \hat{\mu}_{k-i} \tag{32}$$

Here, w_i represents a weight within the set of moving average weights:

$$\mathbf{w} = [w_1, \ldots, w_m]^{\mathrm{T}} \tag{33}$$

Given (31),(32) and understanding that the general surface friction is defined as follows:

$$\mathbf{f}_\tau = \mu\mathbf{f}_\perp \tag{34}$$

We can accordingly recompute the surface friction:

$$\mathbf{f}_\tau = -\bar{\mu}\|\mathbf{f}_\perp\|\frac{\hat{\mathbf{v}}}{\|\hat{\mathbf{v}}\|} \tag{35}$$

Referring to Fig.3, the surface normal force can be computed by subtracting the force sensor reading from the computed friction force,

$$\hat{\mathbf{f}}_n = \mathbf{f}_s - \mathbf{f}_\tau \tag{36}$$

By normalizing this estimated surface normal force, we can estimate the direction of the surface normal at each step of the robot's trajectory.

$$\hat{\mathbf{n}}_{\mathrm{surf}} = \frac{\hat{\mathbf{f}}_n}{\|\hat{\mathbf{f}}_n\|} \tag{37}$$

Furthermore, the surface friction coefficient, as detailed in (35) , is continuously updated at each step of the trajectory. This process begins with the projection of the force sensor reading $\mathbf{f}_s$ onto the direction of the velocity $\hat{\mathbf{v}}$, as described in (30). Subsequently, by utilizing the projection of the force

Algorithm 1 : The Proposed Method for Enhanced Surface Normal Estimation with Friction Compensation

Given:	**"Description"**
$\mathbf{f}_s \in \mathbb{R}^3$	Force sensor reading
$\hat{\mathbf{v}} \in \mathbb{R}^3$	Estimated velocity based on previous trajectory
$\hat{\mathbf{M}}_a = [\hat{\mu}_{k-1}, \ldots, \hat{\mu}_{k-m}]^{\mathrm{T}}$	Previous m estimations of friction coefficient
$\mathbf{w} = [w_1, \ldots, w_m]^{\mathrm{T}}$	Moving average weights
Compute "$\hat{\mathbf{f}}_n$" :	
Step 1: **if** $\|\hat{\mathbf{v}}\| > v_\epsilon$ **then**	If the robot is moving, v_ϵ - velocity threshold
Step 2: $\bar{\mu} = \frac{1}{m} \sum_{i=1}^{m} w_i \hat{\mu}_{k-i}$	Weighted moving average on $\hat{\mathbf{M}}_a$
Step 3: $\mathbf{\Omega}_{\mathbf{v}} = \hat{\mathbf{v}}(\hat{\mathbf{v}}^{\mathrm{T}}\hat{\mathbf{v}})^{-1}\hat{\mathbf{v}}^{\mathrm{T}}, \ \mathbf{f}_\perp = (\mathbf{I} - \mathbf{\Omega}_{\mathbf{v}})\mathbf{f}_s$	Projection onto "null space" of $\hat{\mathbf{v}}$
Step 4: $\mathbf{f}_\tau = -\bar{\mu}\|\mathbf{f}_\perp\|\frac{\hat{\mathbf{v}}}{\|\hat{\mathbf{v}}\|}$	Compute surface friction
Step 5: $\hat{\mathbf{f}}_n = \mathbf{f}_s - \mathbf{f}_\tau$	Obtain "surface normal force" by subtracting friction from force sensor reading
Step 6: **else** $\hat{\mathbf{f}}_n = \mathbf{f}_s$	
Step 7: **end if**	
Update "μ_k" :	
Step 8: **if** $\|\hat{\mathbf{v}}\| > v_\epsilon$ **then**	If the robot is moving, v_ϵ - velocity threshold
Step 9: $\mathbf{f}_v = \mathbf{\Omega}_{\mathbf{v}}\mathbf{f}_s$	Projection onto $\hat{\mathbf{v}}$ direction
Step 10: $\mu_k = \frac{\|\mathbf{f}_v\|}{\|\mathbf{f}_\perp\|}$	Update current friction coefficient estimation
Step 11: **else** $\mu_k = \mu_{k-1}$	
Step 12: **end if**	
Output: "$\hat{\mathbf{f}}_n, \ \boldsymbol{\mu}_k$"	

sensor reading data onto both the direction of $\hat{\mathbf{v}}$ and its null space, the friction coefficient is calculated as follows:

$$\mu_k = \frac{\|\mathbf{f}_v\|}{\|\mathbf{f}_\perp\|} \tag{38}$$

D. Optimizing Probe Orientation

In (25), $\boldsymbol{\rho}$ can be chosen to maximize the objective function $\mathbf{g}(\Phi)$. This function aims to optimize the alignment of the end-effector, *ee*, with the estimated surface normal $\hat{\mathbf{n}}_{\mathrm{surf}}$, by minimizing the angle $\boldsymbol{\gamma}$, as depicted in Fig.4. The formulation of this optimization problem is as follows:

$$\boldsymbol{\rho} = \alpha \, \nabla_\Phi g, \quad \alpha > 0, \, \alpha \in \mathbb{R} \tag{39}$$

$$g(\Phi) = \cos\gamma = \hat{\mathbf{n}}_{\mathrm{surf}}^{\mathrm{T}}(-\mathbf{n}_{\mathrm{ee}}) \tag{40}$$

Here, α is a positive scalar, and the following relationship is then established:

$$\begin{aligned} \underset{g}{\text{maximize}} \quad & \hat{\mathbf{n}}_{\mathrm{surf}}^{\mathrm{T}}(-\mathbf{n}_{\mathrm{ee}}) \\ \text{subject to} \quad & \mathbf{v} = \mathbf{v}_{\mathrm{cmd}} \end{aligned} \tag{41}$$

Here, $\hat{\mathbf{n}}_{\mathrm{surf}}$ denotes the estimated direction of the surface normal. This is derived using the algorithm proposed in
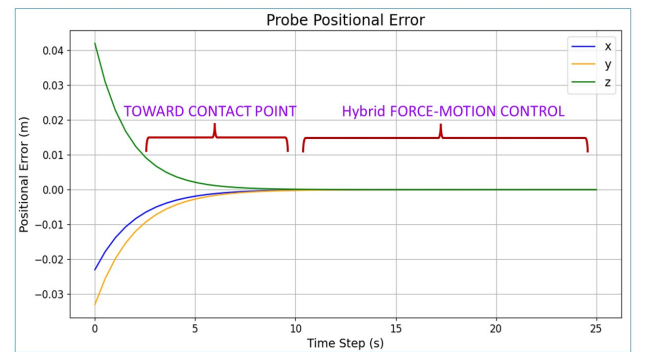

Fig. 5: Tracking Hybrid Motion-Force Errors of the Probe.

the previous section. $\mathbf{n}_{\mathrm{ee}}$, on the other hand, represents the direction of the end-effector *ee*, and can be derived from the robot's forward kinematics. To explain how equation (39) optimizes the probe orientation, it is necessary to define $\nabla_\Phi g$. To do so, we take the derivative of the objective
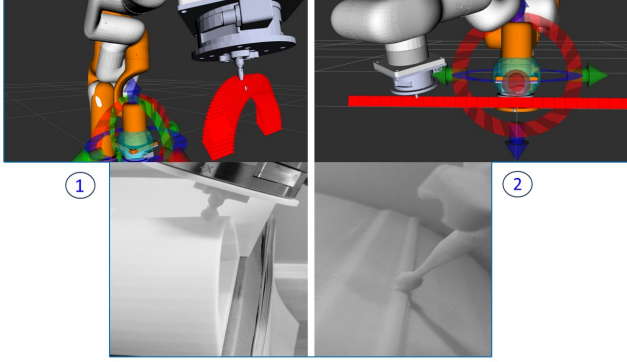
Fig. 6: ROS2 Simulation and Experimentation: 1) Estimating Surface Normal on a Dome-Shaped Workpiece, 2) Estimation of Surface Normal Along a Linear Path on a Workpiece.

function g as follows:

$$\dot{g} = \underbrace{-\dot{\hat{\mathbf{n}}}_{\text{surf}}^{\text{T}}\mathbf{n}_{\text{ee}}}_{\longrightarrow 0} - \hat{\mathbf{n}}_{\text{surf}}^{\text{T}}\dot{\mathbf{n}}_{\text{ee}} \tag{42}$$

$$Thus, \quad \dot{g} = -\hat{\mathbf{n}}_{\text{surf}}^{\text{T}}\dot{\mathbf{n}}_{\text{ee}} \tag{43}$$

Here, $\dot{\mathbf{n}}_{\text{ee}}$ represents the velocity of the end-effector, and it can also be given by,

$$\dot{\mathbf{n}}_{\text{ee}} = \boldsymbol{\omega} \times \mathbf{n}_{\text{ee}}, \quad \boldsymbol{\omega} \in \mathbb{R}^{3\times 1} \tag{44}$$

Here, $\boldsymbol{\omega}$ represents the angular velocity of the *ee*. By substituting (44) into (43) and utilizing the properties of the vector cross product:

$$\dot{g} = -\hat{\mathbf{n}}_{\text{surf}}^{\text{T}} \underbrace{\left([\mathbf{n}_{\text{ee}}]_{\times}^{\text{T}} \boldsymbol{\omega}\right)}_{\triangleq \dot{\mathbf{n}}_{\text{ee}}} \tag{45}$$

Where, $[\mathbf{n}_{\text{ee}}]_{\times}$ represents the Skew-symmetric matrix operator:

$$[\mathbf{n}_{\text{ee}}]_{\times} = \begin{bmatrix} 0 & -\mathbf{n}_{\text{ee},3} & \mathbf{n}_{\text{ee},2} \\ \mathbf{n}_{\text{ee},3} & 0 & -\mathbf{n}_{\text{ee},1} \\ -\mathbf{n}_{\text{ee},2} & \mathbf{n}_{\text{ee},1} & 0 \end{bmatrix} \tag{46}$$

Knowing that the angular velocity of (*ee*), $\boldsymbol{\omega}$, is given by,

$$\boldsymbol{\omega} = \mathbf{J}_{\text{ee,w}}\dot{\Phi} \tag{47}$$

Here, $\mathbf{J}_{\text{ee,w}}$ represents the angular velocity component of the robot's end-effector Jacobian. Equation (45) can be rewritten:

$$\dot{g} \triangleq (\nabla_{\Phi}g)^{\text{T}}\dot{\Phi} = \underbrace{-\hat{\mathbf{n}}_{\text{surf}}^{\text{T}}[\mathbf{n}_{\text{ee}}]_{\times}^{\text{T}}\mathbf{J}_{\text{ee},\omega}}_{\triangleq (\nabla_{\Phi}g)^{\text{T}}}\dot{\Phi} \tag{48}$$

Considering (39),(40) and (48), the definition of $\nabla_{\Phi}g$ is as follows:

$$\nabla_{\Phi}g = -\left(\mathbf{J}_{\text{ee},\omega}\right)^{\text{T}}[\mathbf{n}_{\text{ee}}]_{\times}\,\hat{\mathbf{n}}_{\text{surf}} \tag{49}$$

III. SIMULATION AND EXPERIMENTAL VALIDATION

Building upon the explanations provided in Section II, the robot moves towards the targeted contact point by utilizing a resolved rate algorithm. Once the robot makes contact with the surface at the designated contact point, the hybrid force-motion controller initiates operation, using the command

specified in (25). The following gains, desired force, and offset have been selected for the controller:

$$\mathbf{K}_m, \mathbf{K}_f = \text{diag}([10, 10, 10]), \tag{50}$$

$$\mathbf{K}_{\text{adm}} = \text{diag}([0.1, 0.1, 0.1]) \tag{51}$$

$$\mathbf{f}_{\text{des}} = \begin{bmatrix} 0 & 0 & -2 \end{bmatrix}^{\text{T}} \tag{52}$$

$$\mathbf{d}_h = \begin{bmatrix} 0 & 0 & 0.05 \end{bmatrix}^{\text{T}} \tag{53}$$

Fig.5 illustrates an example of error convergence for the end-effector, as detailed in (23), following a linear trajectory as depicted in Fig.6-2. This indicates that the probe accurately reached the contact point on the workpiece. It then begins following the trajectory, using the hybrid force-motion command from (25), starting around time step 10 and continuing to the end. This process updates the surface normal in accordance with the method outlined in Algorithm1.

To assess the accuracy of the proposed approach, a 7-DoF manipulator equipped with a probe at its end-effector is considered. An F/T sensor is integrated at the wrist of the manipulator, as illustrated in Fig1. During the testing phase, which included both simulations and experiments, a diverse range of trajectories on workpieces were assessed. These included linear, sinusoidal, and curved paths, as well as paths resembling a semi-circular arc(dome), among others. The goal was to evaluate the precision of the surface normal estimation method within the hybrid force-motion control framework. This evaluation focused on two scenarios: one where the surface normal direction remains constant, and another where it varies, such as when following paths along an arc-shaped contour.

Fig.6 showcases some of these path trajectories, illustrated both in a ROS2 simulation environment and in their corresponding experimental setups. The trajectories are categorized as follows:

- A trajectory where the direction of the surface normal remains unchanged.
- A trajectory where the direction of the surface normal may change.

In the first image of Fig.7, the trajectory of the robot's end-effector (ee) for the hybrid force-motion task on the dome-shaped workpiece is in Y-Z plane. Throughout the path trajectory, as indicated by the blue dashed line, the estimated surface normals are parallel to the X-axis. Conversely, in the second image of Fig.7, the robot traverses over a dome-shaped object. Here, the estimated surface normals have different directions along the path.

The surface normal is calculated and continuously updated based on feedback from the force sensor, which is leveraged for surface friction bias compensation, as shown in (37). Therefore, it's essential to evaluate the accuracy of this method, especially in tracking positional-force errors on surfaces of differently shaped objects. This evaluation will ascertain whether the proposed normal estimation method enables more accurate tracing compared to methods with non-updated normals.
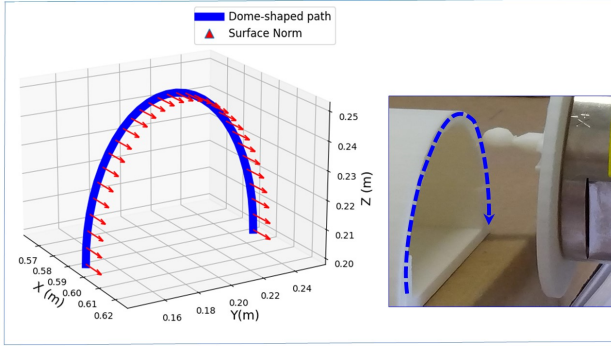
(a) Surface normal in the X-axis direction remains unchanged throughout the trajectory
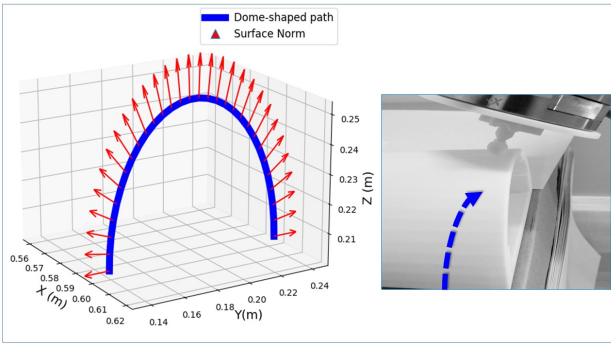
(b) Variation in surface normal direction during the trajectory

Fig. 7: Surface Normal Plots for Various Trajectories: (a) Surface Normal with a Fixed Direction, and (b) Surface Normal that Changes Throughout the Trajectory.

In Fig.8, the red plots represent the probe's (*ee*) hybrid force-motion trajectory over variously shaped surfaces, conducted without using the algorithm proposed in this paper. Conversely, the green plots illustrate the results obtained using the proposed surface normal estimation algorithm, as detailed in Algorithm1. It is evident that the positional tracking error on the surfaces is reduced with the proposed method across all differently shaped surfaces. The improved accuracy in these plots is particularly noticeable on curved shapes, especially at the peaks or troughs. In the dome-shaped and sinusoidal-shaped paths, the differences between the target and actual paths are 0.3mm at minimum and 1mm at maximum, respectively. The error becomes smaller in paths with an unchanged surface normal, as shown in Fig.7a and 8a-2, reducing to approximately 0.3mm and 0.8mm, respectively. The average accuracy increase achieved by the proposed method is approximately 5%, compared to the method without surface norm updates. In addition, in both Fig.8a and Fig.8b, the end-effector (*ee*) controlled force, followed precisely with the desired force, as detailed in (52).

IV. CONCLUSION

In this paper, we introduce a novel method for real-time surface normal estimation. It leverages force sensor readings and velocity commands, compensating for surface friction within a hybrid force-motion control framework.
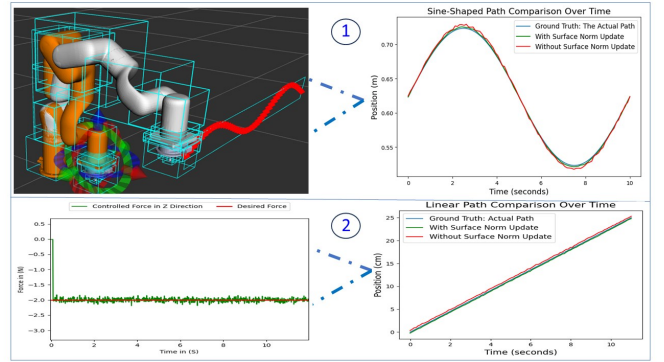
(a) Hybrid force-motion control, with and without surface normal estimation method on two different paths: 1) A sine-shaped path, and 2) A linear path with it's force control plot
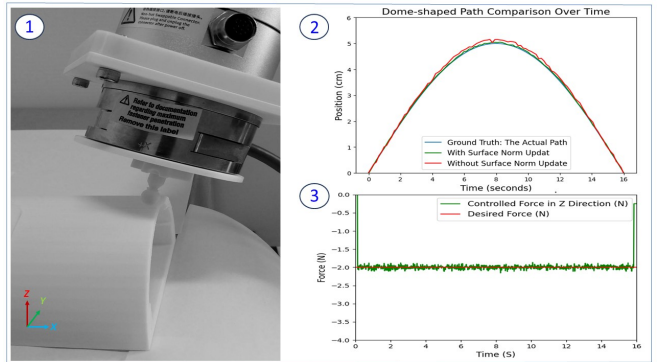
(b) Tracking error for a dome-shaped workpiece: 1- Probe Path following on the workpiece in experimental Setup, 2- Force control error, with and without surface normal update for the dome-shaped object, and 3- Force control plot.

Fig. 8: Plots of Position and Force Tracking Errors for: a) A Sine-Shaped and a Linear Paths, and b) A Dome-Shaped Path.

The proposed algorithm is implemented in a *ROS2* environment and has been tested through both simulation and experiment on multiple workpieces with variously shaped surfaces. These include a line, a sine-shaped path, a dome-shaped, among others. The current implementation delivers satisfactory performance, as illustrated in the accompanying figures and plots. Utilizing the estimated surface normal method, the position-force errors tracking of the explored paths — especially on surfaces where the normal changes direction along the path — aligns very closely with the actual path. The proposed method has increased positional tracking accuracy by up to 5%, compared to previous methods where the surface normal is not updated during the trajectory.

REFERENCES

[1] A. Stolt, M. Linderoth, A. Robertsson and R. Johansson, "Force controlled robotic assembly without a force sensor," 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 2012, pp. 1538-1543, doi: 10.1109/ICRA.2012.6224837.

[2] Y. H. Yin, Y. Xu, Z. H. Jiang and Q. R. Wang, "Tracking and Understanding Unknown Surface With High Speed by Force Sensing and Control for Robot," in IEEE Sensors Journal, vol. 12, no. 9, pp. 2910-2916, Sept. 2012, doi: 10.1109/JSEN.2012.2205098.

[3] Thomessen, Trygve and Lien, Terje. (2000). Robot control system for safe and rapid programming of grinding applications. Industrial Robot: An International Journal. 27. 437-444. 10.1108/01439910010355766.

[4] M. Ahrens, V. Mallick and K. Parfrey, "Robotic based thermoplastic fibre placement process," Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Leuven, Belgium, 1998, pp. 1148-1153 vol.2, doi: 10.1109/ROBOT.1998.677247.

[5] Y. Ohba et al., "Sensorless Force Control for Injection Molding Machine Using Reaction Torque Observer Considering Torsion Phenomenon," in IEEE Transactions on Industrial Electronics, vol. 56, no. 8, pp. 2955-2960, Aug. 2009, doi: 10.1109/TIE.2009.2024444.

[6] H. B. Huang, D. Sun, J. K. Mills and S. H. Cheng, "Robotic Cell Injection System With Position and Force Control: Toward Automatic Batch Biomanipulation," in IEEE Transactions on Robotics, vol. 25, no. 3, pp. 727-737, June 2009, doi: 10.1109/TRO.2009.2017109.

[7] G. Ferretti, G. Magnani and P. Rocco, "Toward the implementation of hybrid position/force control in industrial robots," in IEEE Transactions on Robotics and Automation, vol. 13, no. 6, pp. 838-845, Dec. 1997, doi: 10.1109/70.650162.

[8] C. Natale, B. Siciliano and L. Villani, "Robust hybrid force/position control with experiments on an industrial robot," 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399), Atlanta, GA, USA, 1999, pp. 956-960, doi: 10.1109/AIM.1999.803301.

[9] T. Yoshikawa and A. Sudou, "Dynamic hybrid position/force control of robot manipulators-on-line estimation of unknown constraint," in IEEE Transactions on Robotics and Automation, vol. 9, no. 2, pp. 220-226, April 1993, doi: 10.1109/70.238286.

[10] Solanes, J. E., Gracia, L., Muñoz-Benavent, P., Valls Miro, J., Perez-Vidal, C., and Tornero, J. (November 28, 2018). "Robust Hybrid Position-Force Control for Robotic Surface Polishing." ASME. J. Manuf. Sci. Eng. January 2019; 141(1): 011013. https://doi.org/10.1115/1.4041836

[11] M. Namvar and F. Aghili, "Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments," in IEEE Transactions on Robotics, vol. 21, no. 4, pp. 678-694, Aug. 2005, doi: 10.1109/TRO.2004.842346.

[12] G. Xia, C. Li, Q. Zhu and X. Xie, "Hybrid force/position control of industrial robotic manipulator based on Kalman filter," 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 2016, pp. 2070-2075, doi: 10.1109/ICMA.2016.7558885.

[13] Ziling Wang, Lai Zou, Xiaojie Su, Guoyue Luo, Rui Li, and Yun Huang. 2021. Hybrid force/position control in workspace of robotic manipulator in uncertain environments based on adaptive fuzzy control. Robot. Auton. Syst. 145, C (Nov 2021). https://doi.org/10.1016/j.robot.2021.103870

[14] B. Bona and M. Indri, "Friction compensation in robotics: An overview,"in Proc. 44th IEEE Conf. Decision Control Eur. Control Conf., Seville, Spain, Dec. 15–15, 2005, pp. 4360–4367.

[15] P. E. Dupont, "Avoiding stick-slip through PD control," IEEE Trans. Autom. Control, vol. 39, no. 5, pp. 1094–1097, May 1994.

[16] H. Olsson, K. J. Astrm, C. Canudas de Wit, M. Gfvert, and P. Lischinsky, "Friction models and friction compensation," Eur. J. Control, vol. 4, no. 3, pp. 176–195, 1998.

[17] K. M. Misovec and A. M. Annaswamy, "Friction compensation using adaptive nonlinear control with persistent excitation," Int. J. Control, vol. 72, no. 5, pp. 457–479, 1999.

[18] M. Kermani, M. Wong, R. Patel, M. Moallem, and M. Ostojic, "Friction compensation in low and high-reversal-velocity manipulators," in Proc. IEEE Int. Conf. Robotics Automation, New Orleans, LA, USA, Apr. 26–May 1, 2004, pp. 4320–4325.

[19] C. Aguilar-Avelar and J. Moreno-Valenzuela, "New feedback linearization-based control for arm trajectory tracking of the Furuta Pendulum," IEEE/ASME Trans. Mechatronics, vol. 21, no. 2, pp. 638–648, Apr. 2016.

[20] S. Huang, W. Liang and K. K. Tan, "Intelligent Friction Compensation: A Review," in IEEE/ASME Transactions on Mechatronics, vol. 24, no. 4, pp. 1763-1774, Aug. 2019, doi: 10.1109/TMECH.2019.2916665.

[21] F. Jatta, G. Legnani and A. Visioli, "Friction compensation in hybrid force/velocity control of industrial manipulators," in IEEE Transactions on Industrial Electronics, vol. 53, no. 2, pp. 604-613, April 2006, doi: 10.1109/TIE.2006.870682.

[22] H. Cao, X. Chen, Y. He and X. Zhao, "Dynamic Adaptive Hybrid Impedance Control for Dynamic Contact Force Tracking in Uncertain Environments," in IEEE Access, vol. 7, pp. 83162-83174, 2019, doi: 10.1109/ACCESS.2019.2924696.

[23] D. P. Hess and A. Soom, "Friction at a lubricated line contact operating at oscillating sliding velocities," Trans. ASME J. Tribology, vol. 112, no. 1, pp. 147–152, 1990.

[24] C. Canudas DeWit, C. H. Olsson, K. J. Astrm, and P. Lischinsky, "A new model for control of systems with friction," IEEE Trans. Autom. Control, vol. 40, no. 3, pp. 419–425, Mar. 1995.

[25] J. Swevers, F. Al-Bender, C. Ganseman, and T. Prajogo, "An integrated friction model structure with improved presliding behaviour for accurate friction compensation," IEEE Trans. Autom. Control, vol. 45, no. 4, pp. 675–686, Apr. 2000.

[26] V. Lampaert, J. Swevers, and F. Al-Bender, "Modification of the Leuven integrated friction model structure," IEEE Trans. Autom. Control, vol. 47, no. 4, pp. 683–687, Apr. 2002.

[27] S. L. Chen, K. K. Tan, and S. Huang, "Friction modeling and compensation of servo-mechanical system with dual-relay feedback approach," IEEE Trans. Control Syst. Technol., vol. 17, no. 6, pp. 1295–1305, Nov.2009.

[28] S. W. Lee and J. H. Kim, "Robust adaptive stick-slip friction compensation," IEEE Trans. Ind. Electron., vol. 42, no. 5, pp. 474–479, Oct. 1995.

[29] X. Yang et al., "Continuous Friction Feedforward Sliding Mode Controller for a TriMule Hybrid Robot," in IEEE/ASME Transactions on Mechatronics, vol. 23, no. 4, pp. 1673-1683, Aug. 2018, doi: 10.1109/TMECH.2018.2853764.

[30] Y. Qian, J. Yuan, S. Bao and L. Gao, "Sensorless Hybrid Normal-Force Controller With Surface Prediction," 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 2019, pp. 83-88, doi: 10.1109/ROBIO49542.2019.8961532.

[31] Li, Jian, Guan, Yisheng, Chen, Haowen, Wang, Bing, Zhang, Tao, Hong, Jie and Wang, Danwei. (2022). Real-time normal contact force control for robotic surface processing of workpieces without a priori geometric model. The International Journal of Advanced Manufacturing Technology. 119. 10.1007/s00170-021-07497-2.

[32] Khatib, O., 1987, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," IEEE J. Rob. Autom., RA-3(1), pp. 43–53.

[33] Featherstone, R., Thiebaut, S., and Khatib, O., 1999, "A General Contact Model for Dynamically-Decoupled Force/Motion Control," IEEE International Conference on Robotics and Automation (ICRA), Detroit, MI, May 10–15, pp. 3281–3286.

[34] Whitney, D. E., 1969, "Resolved Motion Rate Control of Manipulators and Human Prostheses," IEEE Trans. Man-Mach. Syst., 10(2), pp. 47–53.