One-Shot Demonstration for Slicing and Cutting Everyday Food Items

Yi Liu, Andreas Verleysen, and Francis wyffels

***Abstract*—Cutting everyday food items presents a significant challenge in robotics due to the multiple types of knife skills and the unpredictable mechanical behaviour of materials during manipulation. To address this, we propose a one-shot demonstration-based framework that integrates the imitation of both position and force trajectories of knife skills using dynamic movement primitives (DMPs). Our approach combines: (1) a compensation method to replicate human-like force trajectory, and (2) skill-specific constraints enabling online trajectory re-planning during cutting. We designed three knife skill demos for the robot and tested them on 14 unknown food items. The experiments are conducted to evaluate the effectiveness of the proposed force compensation and re-planning methods. The results demonstrate that our framework can successfully imitate various knife skills and cut previously unknown food items with high precision.**

***Index Terms*—Learning from demonstration, manipulation planning.**

I. Introduction

CHEFS employ a combination of visual and tactile feedback to analyze cuts, enabling them to differentiate in complex materials, like tender meats or juicy fruits. Replicating chef-like knife skills in robots, particularly when manipulating everyday objects like animal tissue, remains a challenge.

Knife skills include different cutting methods. Chopping/dicing are used to completely cut through objects [1], and slicing can modify the shape of objects (e.g., fish, eggplant) without fully cutting them [2]. Carving is used to alter the surface of objects [3] like bread or chicken legs. Different ingredients require different amounts of force [4]. We are developing a versatile robotic kitchen assistant capable of performing multiple knife skills across various ingredients. Robots can control knives in various ways. For surgery, teleoperation lets robots handle scalpels, reducing surgeon workload [5]. In daily tasks, robots rely on automation. Some methods simulate cutting virtually before real-world application [6], but material simulation can be challenging. Training ML models with real data improves performance [7], though collecting enough data is difficult [8], [9]. Demonstration-based methods reduce data needs but may lack versatility across objects [3]. While data-heavy, multi-sensor approaches enable precise cutting and replanning, their reliance on extensive data and hardware limits generalizability [10]. This motivates our development of a more adaptable, data-efficient method. We employ a one-shot demonstration to deploy multiple knife skills, as shown in Fig. 1, avoiding the need for numerous demonstrations. A force-compensated DMPs method then imitates the demonstrated trajectory, which is executed under constraints that can trigger re-planning for cutting unknown food items.

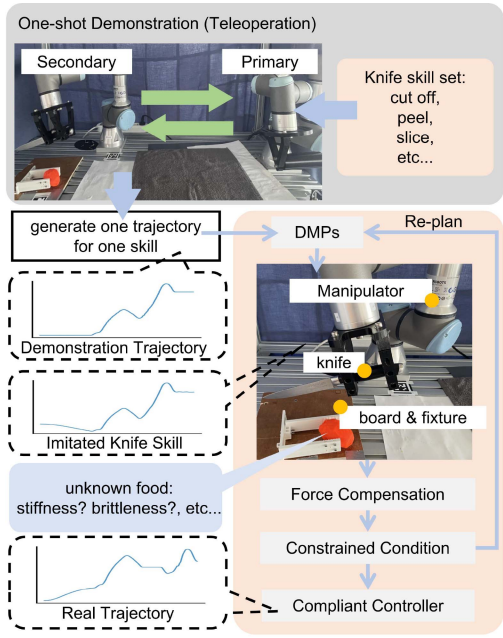


Fig. 1. Overall framework: a teleoperation system is utilized to generate the trajectories in the knife skill set, and the generated demonstration trajectories enable the robot to complete the cutting task through DMPs; the constrained condition is used to activate DMPs. Its environment includes a cutting board for fixing unknown objects and a manipulator for fixing the knife.

The contributions of this letter are summarized as follows:

1) Generate various knife skills with position and force trajectories using a one-shot demonstration method based on teleoperation;
2) A compensation method is proposed to generalize the DMPs model so that it can imitate the knife skills on unknown food items based on the demonstration trajectory;
3) Constraints designed based on the characteristics of the knife skills are used to activate DMPs during the

Received 2 April 2025; accepted 12 August 2025. Date of publication 4 September 2025; date of current version 11 September 2025. This article was recommended for publication by Associate Editor E. B?y?k and Editor A. Faust upon evaluation of the reviewers' comments. *(Corresponding author: Yi Liu.)*

The authors are with IDLab-AIRO, elis, Ghent University-imec, Ghent, Belgium (e-mail: yiyiliu.liu@ugent.be).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3606310, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3606310

cutting process, allowing the robot to trigger replanning in real-time;

4) The proposed method is deployed on a real knife-wielding robot, enabling it to cut unknown food items utilizing the learned knife skills.

II. Related Work

Knife skills must extend beyond a single technique, as real-world tasks demand diverse capabilities. A fundamental skill is direct cutting. Existing work includes compliant manipulation for vegetable cutting via reinforcement learning [11] and force/torque (F/T) sensor-based feedback control [1], [4]. However, these methods focus solely on cutting, limiting their applicability to tasks requiring multiple knife skills.

In some cases, slicing is required rather than simple cutting. Prior studies on meat [2] illustrate slicing without severing, while others use F/T sensors to avoid cores in foods [7]. In medical applications, knife skills include cutting epidermis or organs [12]; DMPs [13] have been used to reproduce demonstrated scalpel trajectories for tissue incisions [3] and teleoperated organ surgeries [14]. DMPs have also been adapted for slicing skills [10] using multimodal sensing (vibration, audio, force) to guide re-planning, though at the cost of excessive time on ineffective steps. These works highlight the diversity of knife skills and cutting paths, making the deployment of multiple skills in robots a key challenge we need to address.

Robotic knife manipulation demands different control methods for varying cutting paths and scenarios [15]. Teleoperation is a key technique for robot-assisted cutting [3], while shared control [16] reduces human effort by combining operator feedback with partial robot autonomy. In common cutting tasks, machine learning approaches include Long short-term memory models for approximating food-cutting dynamics with model predictive controllers (MPC) [17], and simulators for training reinforcement learning policies [6]. However, these methods often face challenges in real-world transfer or require extensive training data [8], [9], which is difficult to collect.

To simplify cutting strategy deployment, sensors embedded in knives provide feedback to guide the cutting path [18], while vision [19] or laser [20] systems assist surface cuts. Demonstration-based methods generate paths through repeated trials [3], [21], and bilateral teleoperation enhances human-like motion while minimizing operator forces [22], [23]. Although easy to deploy, these methods have been tested on limited object types and struggle to generalize to objects with varying physical properties, highlighting the need for adaptable cutting paths.

III. Methodology

In this section, we present our proposed framework, which consists of a demonstration system and a trajectory imitation approach based on position error.

A. One-Shot Demonstration

Our method captures human-like cutting paths through demonstrations. Using bilateral teleoperation (Fig. 1), we record force trajectories while isolating operator interference. We focus on local path segments, making system time delays negligible. The primary-secondary model structure is:

$$\mathbf{M}_m(\mathbf{p}_m)\ddot{\mathbf{p}}_m + \mathbf{C}_m(\mathbf{p}_m, \dot{\mathbf{p}}_m)\dot{\mathbf{p}}_m = \mathbf{f}_m - \mathbf{f}_h, \tag{1}$$

$$\mathbf{M}_s(\mathbf{p}_s)\ddot{\mathbf{p}}_s + \mathbf{C}_s(\mathbf{p}_s, \dot{\mathbf{p}}_s)\dot{\mathbf{p}}_s = \mathbf{f}_s - \mathbf{f}_e, \tag{2}$$

where $\mathbf{p}_m$, $\mathbf{p}_s \in \mathbb{R}^6$ represent the posture of the primary and secondary end-effectors in Euler angles with Cartesian position, $\mathbf{f}_m$, $\mathbf{f}_s \in \mathbb{R}^6$ represent controlled force outputted by the robot, $\mathbf{f}_h$, $\mathbf{f}_e \in \mathbb{R}^6$ are the external interaction forces from human and environment. $\mathbf{M}_m(\mathbf{p}_m)$, $\mathbf{M}_s(\mathbf{p}_s)$, $\mathbf{C}_m(\mathbf{p}_m, \dot{\mathbf{p}}_m)$, $\mathbf{C}_s(\mathbf{p}_s, \dot{\mathbf{p}}_s)$ are the inertial matrix and the Coriolis matrices, respectively. The matrix dimensions are 6 by 6. The matrix $\mathbf{Z}$ of this model is utilized to represent the system's stability by the Lyapunov method [24] with Equ 1 and Equ 2 as follows:

$$\begin{pmatrix} \mathbf{f}_m \\ -\mathbf{p}_s \end{pmatrix} = \mathbf{Z} \begin{pmatrix} \mathbf{p}_m \\ \mathbf{f}_s \end{pmatrix}, \tilde{\mathbf{Z}} = \begin{pmatrix} \mathbf{Z}_m & k_c \\ j_c & \frac{1}{\mathbf{Z}_s} \end{pmatrix}, \tag{3}$$

where $\mathbf{Z}_m$, $\mathbf{Z}_s$ represent the physical property of the system, j_c and k_c, related to $\mathbf{f}_h$ and $\mathbf{f}_e$, represent the scale of position and force respectively. Based on the above system, we can control the primary robot to let the secondary one cut food items to generate one parameterized demonstration trajectory $Y(p, f, t)$ for one knife skill, which includes the position $p(t)$ and the force $f(t)$ at time t.

B. Knife Skills Learning

Through III-A, we generate the demonstration trajectory that contains position $p_d(t)$ and force $f_d(t)$. For cutting unknown food items (with uncertain properties and positions), we develop a DMPs-based method that generalizes demonstrated trajectories across various ingredients.

1) Admittance Control: Since the robot with the obtained trajectories $p_d(t)$ and $f_d(t)$ needs to be tracked to enable the robot to control itself, the Eqn. 2 is rewritten as,

$$\mathbf{M}_d(\ddot{\mathbf{p}}_d - \ddot{\mathbf{p}}) + \mathbf{D}_d(\dot{\mathbf{p}}_d - \dot{\mathbf{p}}) + \mathbf{K}_d(\mathbf{p}_d - \mathbf{p}) = \mathbf{f}_d - \mathbf{f}_e, \tag{4}$$

where $\mathbf{p}_d$, $\dot{\mathbf{p}}_d$, $\ddot{\mathbf{p}}_d$ represent the end-effector's desired position, speed and acceleration. $\mathbf{f}_d$ denotes the desired force. The $\mathbf{M}_d$, $\mathbf{D}_d$, and $\mathbf{K}_d$ correspond to the inertia, damping, and stiffness characteristics. In particular, in the autonomous cutting part, we design the same admittance controller, where $\mathbf{f}_d$ is regarded as a virtual force compensation value (in III-B3).

2) Dynamic Movement Primitives: A general DMPs model is expressed as

$$\tau^2\ddot{y} = \alpha_y(\beta_y(g - y) - \tau\dot{y}) + u(s) + C_t, \tag{5}$$

where y is the demonstration trajectory, the $\alpha_y, \beta_y > 0$ are the coefficients, and $u(s)$ represents the nonlinear basic function superposition term, which enables the trajectory to converge from y to the goal g, and s represents the phase variable. The C_t is the constrained coupling term for obstacle avoidance [13]. We do not place obstacles on the cutting path ($C_t = 0$). To make Eqn. 5 converge to 0, the canonical system is expressed as $\tau\dot{s} = -\alpha_s s$, where $\alpha_s > 0$ is a coefficient. When $s = 0$, it means that the

system converges to the goal g. Therefore, the basic function $u(s)$ is defined as

$$u(s) = \frac{\sum_{i=1}^{N} \varphi_i(s)\omega_i s}{\sum_{i=1}^{N} \varphi_i(s)}(g - y_0), \tag{6}$$

where y_0 represents the initial state, ω_i represents the weights of the radial basis functions $\varphi_i(s) = \exp(-h_i(s - c_i)^2)$, the N represents the number of $\varphi_i(s)$, h_i and c_i are the widths and centers of $\varphi_i(s)$ respectively. To ensure that the shape of the entire demonstration trajectory can be learned, the centre of $\varphi_i(s)$ is evenly distributed to the timestamps of the demonstration trajectory. To ensure the convergence speed in the later stage, we set $h_i = N/c_i$. Then, the DMPs can be generalized in a new task to generate a similar shape of trajectory by changing the y_0 and goal g.

3) Compensation With Memory: We can control the robot based on the F/T sensor's feedback in III-B1, and imitate the demonstrated trajectory to obtain position trajectory $p_y(t)$ and force trajectory $f_y(t)$ from III-B2. When facing unknown food items to be cut, the robot can not determine the desired force of the food by $f_y(t)$ because the unknown food items have unknown physical properties. Therefore, we can only imitate the demonstrated trajectory $p_d(t)$ as $p_y(t)$ to cut. In this way, the F/T sensor's feedback $f(t)$ when cutting unknown food causes the shape of the original trajectory $p(t)$ to be changed. To ensure that the shape of $p(t)$ is close to the $p_y(t)$ one, we adjust the admittance controller by compensation.

Above all, the position error $e_p(t) = p_y(t) - p(t)$ — with the imitated position p_y and the current position p. — is used as the compensation condition. We record K historical data of the error for calculation. Inspired by the human forgetting curve [25], we set the remembered error $e_r(t)$ as $e_r(t) = \sum_{t=0}^{K} \exp(-t/S)e_p(t)$, where S indicates the relative memory strength. Based on previous research on manipulating food through force [26], the virtual force f_v obtained based on the error $e_r(t)$ is utilized as the compensation force, which can be expressed as a position-to-force mapping relationship by the momentum theorem and impulse theorem as follows:

$$f_v(t) = H(e_r)\frac{me_r(t)}{T^2}, \tag{7}$$

where m represents the mass of the knife, T represents the sampling period. However, when T is too small and $e_r(t)$ too large, $f_v(t)$ becomes unreasonably high. To address this, we introduce a heuristic constraint $H(e_r) = \exp(-e_r/h)$ to scale $f_v(t)$. Here, $f_v(t)$ serves as a position correction term for the admittance controller based on $e_r(t)$, but it cannot be directly added to $f_y(t)$ since $f_y(t)$ is not a reliable reference for unknown food. To ensure the generated force profile not only corrects for position errors but also mimics the shape and dynamics of the human demonstration, we design a blended compensation force, $\hat{f}_v(t)$. This force is a weighted average of two components: the reactive term $f_v(t)$ and a predictive, trend-following term. The trend-following term is designed to anticipate the next force value by following the slope of the demonstrated force trajectory, $f_y(t)$. We start with the finite difference approximation for the next force value $f'_v(t_n) \approx (f_v(t_n) - f_v(t_{n-2}))/2T$ with rearrangement gives $f_v(t_n) \approx f_v(t_{n-2}) + 2Tf'_v(t_n)$. To guide this prediction with the recorded demonstrations, we substitute the real-time derivative $f'_v(t_n)$ with the demonstrated derivative $f'_y(t_n)$. This yields a predictive term $f_v(t_{n-2}) + 2Tf'_y(t_n)$, which estimates what the force compensation should be if it were following the demonstrated trend. We combine these two components in a convex combination:

$$\hat{f}_v(t_n) = \alpha\left(f_v(t_{n-2}) + 2Tf'_y(t_n)\right) + (1 - \alpha)f_v(t_n), \tag{8}$$

where the weight $\alpha \in [0, 1]$ balances trust between following the demonstrated force trend and reacting to the immediate position error.

Lastly, we obtain the compensated force trajectory $\hat{f}_v(t)$, which adapts the robot to food items with unknown physical properties by adjusting the desired force $\mathbf{f}_d$ in III-B1. According to Eqn. 4, increasing $|\mathbf{f}_d - \mathbf{f}_e|$ accelerates the convergence of the position error $(\mathbf{p}_d - \mathbf{p})$, analogous to reducing $e_r(t)$. As $e_r(t)$ decreases, the influence of the virtual force $f_v(t)$ diminishes, and the desired force shape approaches that of $f_y(t)$. Fig. 2 compares the contribution of each factor during cutting. f_{y_1} is the demonstration trajectory from IV-B, while f'_y guides the force shape, varying significantly upon contact with food but being scaled down by $2T$ (e.g., -1 N to 1 N along the Z-axis). f_v serves as the dominant position-error correction term, compensating for f'_y with -4 N to 4 N. Overall, our method prioritizes f_v in $\hat{f}_v$, with f'_y playing a secondary, shape-guiding role.

4) Constraint-Based Re-Planning: Since the environment is dynamic, the model must adapt to changing target positions and initial states. The imitated trajectory $(p_y(t), f_y(t))$ in Eqn. 5 depends on the target g and initial state y_0, both varying over time. We employ DMPs to re-plan trajectories during execution, but running DMPs at every step is inefficient. Therefore, re-planning is activated only when constraints on g and y_0—defined by the food's state—are satisfied, as specified in **CONDITION 1** and **CONDITION 2**.

CONDITION 1: dynamic position of the food items. The position of the food affects the change of $p_y(t)$. The position of the chopping board for fixing food is easy to obtain, we use it as the goal $\mathbf{p}_g$. Based on the fixed given initial state $\mathbf{p}_{y_0}$ and the goal $\mathbf{p}_g$, we can use DMPs to re-plan the trajectory. The Eqn. 5 is rewritten in positional style as follows,

$$\tau^2\ddot{y} = \alpha_y(\beta_y(1_{d<\delta}\mathbf{p}_g + 1_{d\geq\delta}\mathbf{p}_c - y) - \tau\dot{y}) + u(s), \tag{9}$$

where $\mathbf{p}_g$ represents the target position previously recorded by DMPs, and $\mathbf{p}_c$ represents the current position of the food. The indicator function $1_{d<\delta}$ and $1_{d\geq\delta}$ activates p_g and p_c through distance $d = \|\mathbf{p}_g - \mathbf{p}_c\|_2$ and threshold δ. In particular, $1_{d<\delta}$ does not repeatedly activate $\mathbf{p}_g$, and $1_{d\geq\delta}$ refreshes $\mathbf{p}_g = \mathbf{p}_c$.

CONDITION 2: food thickness. To handle unknown food thickness, we split the trajectory into reaching and cutting phases. CONDITION 2 extends the reaching phase until contact is detected or stops it early upon contact. Fig. 3 shows DMPs performance with and without CONDITION 2. During reaching, $f_y(t) = 0$; it becomes non-zero upon contact, marking the start of cutting. When $\mathbf{f}_y > \epsilon$ for the first time, contact occurs and the time is marked as the segment node t_s, with ϵ as a threshold. If the F/T sensor detects $\mathbf{f}_e > \epsilon$ before (early touch) or after

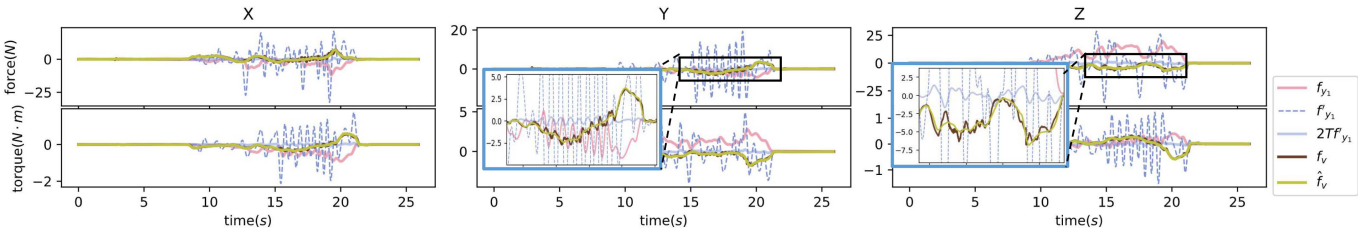


Fig. 2. Performance of using the compensation method was used to compare various factors (f_v vs. $\hat{f}_v$ vs. $2Tf'_{y_1}$). The y_1 is explained in Fig. 5.

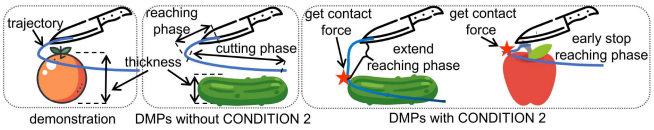


Fig. 3. Diagram of **CONDITION 2**. It demonstrates different phases and trajectory changes by using the proposed re-planning method. Left is the example demonstration. In the middle, we apply the DMPs based on the example demonstration without **CONDITION 2** on an object with a different thickness. On the right, we show how **CONDITION 2** allows to active re-planning to reach and cut the food.

(delayed touch) t_s, the trajectory y must be re-planned. The tool position error $e_k = p(t) - p_y(t_s)$ is added to the cutting phase to compensate:

$$\hat{p_y}(t) = p_y(t) + e_k, \tag{10}$$

where $p_y(t)$ is the original DMPs trajectory and $\hat{p_y}(t)$ is the compensated trajectory. For early touch, e_k is directly used; for delayed touch, a displacement Δp is added to p_d in Eqn. 4 to ensure timely contact.

Finally, the proposed algorithm flow is shown in Algorithm 1. Due to the changes caused by node t_s, the target g and the initial state y_0 are defined in two phases respectively. In the reaching phase, $\mathbf{p}_{y_0}$ remains unchanged, $\mathbf{p}_g = p_y(t_s)$. In the cutting phase, $\mathbf{p}_{y_0} = p(t)$, $p_g = \hat{p_y}$.

Algorithm 1: Constrained Re-planning of DMPs.

$Y(p, f, t) \leftarrow p_d(t), f_d(t)$ from one-shot demonstration
$y \leftarrow Y(p, f, t)$, generate $p_y(t), f_y(t)$ in Eqn. 5
$t \leftarrow 0$, $g_f \leftarrow false$
// g_f is the gate used to switch in CONDITION 2
while $true$ **do**
  update $p_y(t)$ by CONDITION 1 based on Eqn. 9
  if $t \leq t_s$ **then**
    update $p_y(t)$ from Eqn. 9
  else
    update $p_y(t)$ by CONDITION 2 based on Eqn. 10
  end if
  if $\mathbf{f}_e > \epsilon$ and $g_f = false$ **then**
    $g_f \leftarrow true$, $e_k \leftarrow p(t) - p_y(t_s)$, $\mathbf{p}_{y_0} \leftarrow p(t)$, $\mathbf{p}_g \leftarrow \hat{p_y}$
    update $p_y(t), f_y(t)$ from Eqn. 9
  end if
  $\mathbf{f}_d \leftarrow \hat{f}_v(t)$
  move robot based on Eqn. 4
end while

IV. Experiments and Results

We conducted experiments on teleoperation and knife skills to validate the proposed methodology. Teleoperation verified the feasibility of one-shot demonstrations, generating trajectories used in the knife skills experiments, which assessed the effectiveness of our method.

A. Experimental Setup

All experiments were performed using the UR3e robot (Fig. 1), with its built-in F/T sensor providing force feedback. The secondary end was enabled for autonomous control. AprilTag markers [27] were used for food posture estimation. Reference key parameters were provided and adjustable as detailed in the methodology. In III-A, the transfer matrix $\mathbf{Z}$ was positive definite, with $\mathbf{Z}_m = \mathbf{Z}_s$ as the identity matrix. Gains were set to $k_c = j_c = 0.6$ to avoid excessive feedback from the primary end. For the admittance controller III-B1, parameters were $\mathbf{M}_d = 1$, $\mathbf{K}_d = 5000$, and critically damped $\mathbf{D}_d = 2\sqrt{\mathbf{M}_d\mathbf{K}_d}$. In III-B2, DMPs used $N = 50$ radial basis functions for Eqn. 6, following [13]. In III-B3, memory retention was set with $S = h = 0.01$, sampling time $T = 0.02\,\mathrm{s}$, DMPs reactivation period $2T$, and trajectory weighting $\alpha = 0.3$. Thresholds in III-B4 were set close to zero, with $\delta = 0.01\,\mathrm{m}$ and $\epsilon = 0.5\,\mathrm{N}$.

B. Demonstration From Bilateral Teleoperation

We used the bilateral teleoperation system described in III-A to control the secondary robot for cutting food items and obtaining demonstration trajectories. Each demonstration trajectory contained position and force data. As shown in Fig. 4, the secondary robot's position $\mathbf{p}_s$ accurately tracked the primary's position $\mathbf{p}_m$, thanks to the local control design without time delay. For forces, the primary's force $\mathbf{f}_m$ included both human-applied and secondary feedback forces, whereas the secondary's force $\mathbf{f}_s$ reflected only the contact force with the environment. This force trajectory can be used to capture the cutting dynamics for imitation learning in IV-C.

To demonstrate the effectiveness of the proposed framework, we designed three knife skills: **cutting off** (y_1), which divides the food into two parts; **cutting in** (y_2), which slices into the food at an angle while keeping it mostly intact, e.g., cutting bread to add fillings; and **shearing** (y_3), which cuts along the surface

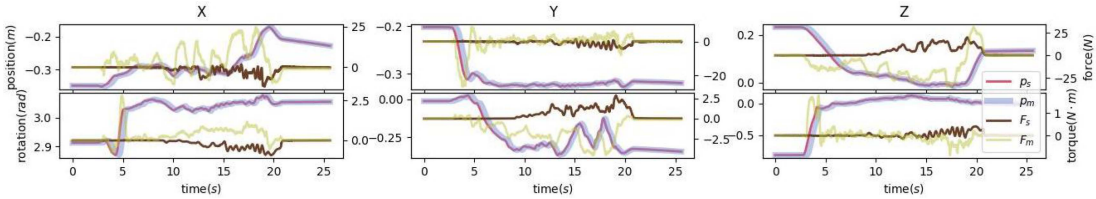


Fig. 4. Bilateral teleoperation system tracking results include the posture q_m of the master manipulator, the feedback f_m of the F/T sensor, the posture q_s of the slave and the feedback f_s of the F/T sensor.

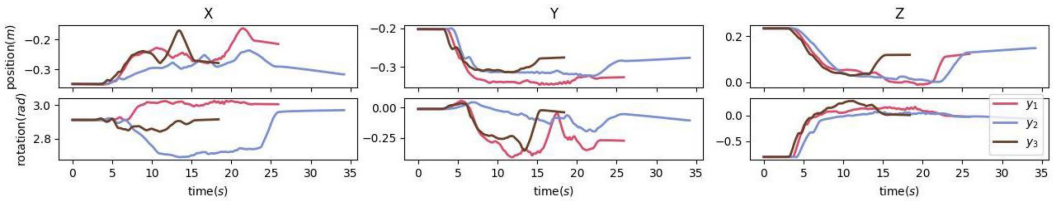


Fig. 5. Trajectory of the manipulator during the execution of knife skills, including three knife skills, each with varying execution times and motion trajectories.

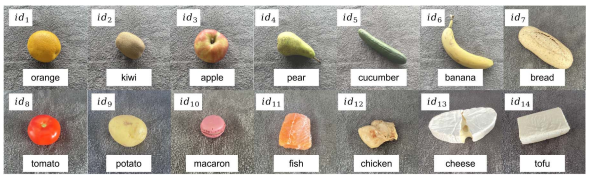


Fig. 6. Food items being cut. They were not familiar with the proposed framework, which included everyday items such as fruits, meats, and snacks.

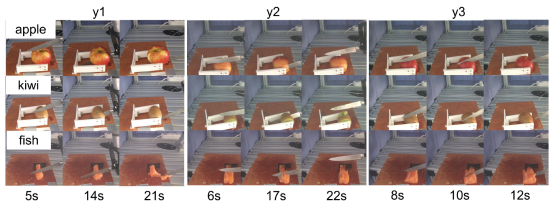


Fig. 7. Performance of the knife skills. We selected three types of food items with varying degrees of softness to demonstrate the process of cutting. The representation of all mentioned food items can be seen in the video.

without penetrating deeply, e.g., scoring dough for appearance. Demonstration trajectories were generated using foods with large surface tension; in our experiments, lemons were used as the cutting objects.

Finally, as shown in Fig. 5, the system generated one demonstration trajectory for each knife skill. The execution time for y_3 was the shortest, reflecting its shallow, surface-cutting nature, whereas y_1 (cut-off) required the deepest Z-axis penetration to fully cut through the food. To ensure a clean cut, the trajectory's final depth was set to match the cutting board height measured via an AprilTag. The X/Y axes represent the knife's cutting angle. Fluctuations in X-axis position, with stable Y-axis position, indicate cutting along the X direction, while Y-axis rotation fluctuations correspond to downward penetration. Notably, y_2, which cuts obliquely into the food, showed pronounced changes in X-axis rotation and Y-axis position. All skills exhibited minimal Z-axis rotation, consistent with the absence of churning actions.

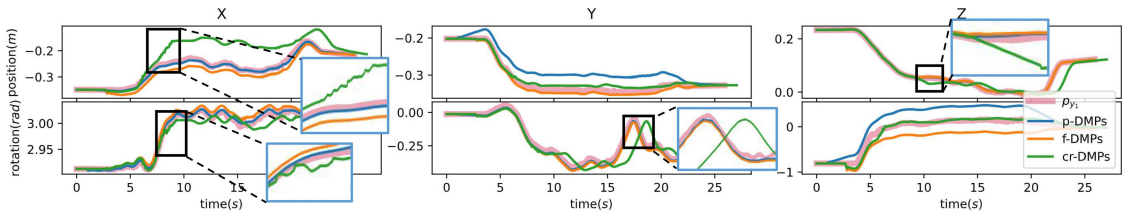
C. Knife Skills Imitation

In this section, we presented the results of the framework proposed in III-B, including the use of various knife skills and cutting of unknown food items.

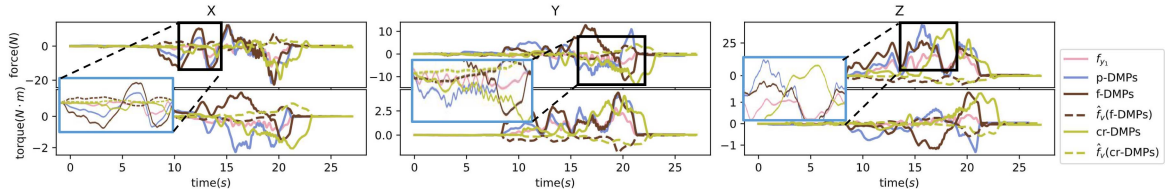
First, we selected 14 everyday food items shown in Fig. 6 for testing. To focus on learning knife skills, none of the food contained hard pits, and all physical properties were unknown. The food positions were randomized within the manipulator's reachable workspace, with $\pm 10\,\mathrm{cm}$ in translation and $\pm 20^\circ$ in rotation. Each item was initially placed in a natural posture on the chopping board and secured with a fixture, as shown in Fig. 1, to prevent displacement during cutting.

Then, we selected one knife skill (y_1) from IV-B for demonstration. The choice of knife skill was specified manually, as the manipulator did not autonomously determine it. The methods compared in the experiment included: imitating the demonstration trajectory using pure DMPs (`p-DMPs`), DMPs with force compensation (`f-DMPs`), and the complete proposed framework (`cr-DMPs`). Since the food positions were randomized, the initial imitation trajectories in `p-DMPs` and `f-DMPs` required re-planning to generate reasonable trajectories. Fig. 7 presents three representative food items (apple, kiwi, fish) and the corresponding knife skills (y_1–y_3) executed with `cr-DMPs`, with time nodes indicating object states. To highlight performance differences, Fig. 8 shows a comparison for one food item (apple) using knife skill y_1 across all three methods. In Fig. 8(a), trajectories for `p-DMPs` and `f-DMPs` were based on the initial food position, whereas the trajectory of `cr-DMPs` adapted in real-time due to the re-planning method. Fig. 8(b) depicts force trajectories, where $\hat{f}_v$ represents the force compensation values in `f-DMPs` and `cr-DMPs`.

Finally, we tested the food items in Fig. 6 using `p-DMPs`, `f-DMPs`, and `cr-DMPs`, with each method performing five independent trials per item. A test was considered successful



(a) Each of the three mentioned methods was used to apply the same knife skill (y_1) to the same food item (apple). The results showed different shapes of the knife skill on the posture. Zoomed-in details highlight the differences between the various methods.



(b) The F/T sensor's feedback of using the same knife skill on different models. The zoomed-in details highlight the differences between the various methods. The method `p-DMPs` did not include the force compensation $\hat{f}_v$.

Fig. 8. Comparison of the effects of the three methods (`p-DMPs` vs. `f-DMPs` vs. `cr-DMPs`). Subfigure (a) shows the position information of the three methods on the same knife skill, while subfigure (b) compares their force information.

TABLE I
MEAN COSINE DISTANCE e_f BETWEEN DEMONSTRATION FORCE f_y AND FEEDBACK FORCE f_e ON Z-AXIS, AND THE REFERENCE HARDNESS w

		id_1	id_2	id_3	id_4	id_5	id_6	id_7	id_8	id_9	id_{10}	id_{11}	id_{12}	id_{13}	id_{14}
w (unit:N)		18.70	13.45	20.32	17.72	17.03	17.56	18.30	14.53	14.94	19.21	10.16	11.17	6.76	5.23
`p-DMPs`	y_1	0.815	0.756	0.860	0.813	0.841	0.813	0.832	0.831	0.834	0.735	0.791	0.801	0.759	0.817
	y_2	0.804	0.839	0.853	0.868	0.814	0.800	0.828	0.743	0.797	0.755	0.764	0.735	0.840	0.811
	y_3	0.787	0.823	0.785	0.837	0.851	0.798	0.774	0.763	0.791	0.721	0.753	0.807	0.788	0.819
`f-DMPs`	y_1	0.932	0.907	0.927	0.916	0.906	0.891	0.889	0.936	0.929	0.921	0.907	0.910	0.899	0.909
	y_2	0.903	0.934	0.913	0.909	0.918	0.910	0.915	0.889	0.901	0.928	0.907	0.896	0.913	0.901
	y_3	0.908	0.910	0.893	0.914	0.891	0.922	0.931	0.901	0.904	0.859	0.867	0.864	0.903	0.916
`cr-DMPs`	y_1	0.912	0.924	0.933	0.936	0.923	0.904	0.914	0.906	0.909	0.910	0.901	0.924	0.936	0.918
	y_2	0.925	0.911	0.906	0.941	0.915	0.929	0.902	0.932	0.922	0.920	0.930	0.917	0.933	0.912
	y_3	0.942	0.915	0.895	0.899	0.929	0.921	0.908	0.930	0.916	0.893	0.916	0.927	0.931	0.929

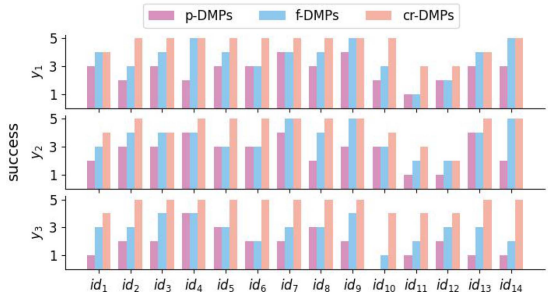


Fig. 9. Test results for all the food items mentioned in Fig. 6. The horizontal axis shows the identification of different food, the vertical axis shows the number of successful tests, and the different lines show the test results of different knife skills. We tested each food item five times using each method.

if the cutting requirements were satisfied; otherwise, it was marked as a failure. Fig. 9 shows the success rates of all methods, with each set of bars representing a specific trajectory. Table I presents the mean cosine distance e_f of the Z-axis force across the five trials. Here, e_f indicates how closely the feedback force $\mathbf{f}_e$ matched the demonstration force $\mathbf{f}_y$, with values closer to 1 representing better alignment. Quantifying the diverse physical properties of the food items—such as hardness, surface tension, density, and brittleness—is challenging, as these depend on factors including growth location, season, and size. For the cutting task, we introduced a reference measure of food hardness w, defined as the force required to vertically press the food by 5 mm using the same tool (a circular flat surface with a diameter of 1 cm).

D. Discussion

In IV-B, demonstration trajectories were generated via bilateral teleoperation, and the system's stability and transparency were analyzed. The proposed method achieved sufficient tracking performance to meet the framework's requirements, allowing the extraction of both positional and pure force trajectories.

1) Force Compensation: In Fig. 8(b), `p-DMPs` lacked force compensation, and thus $\hat{f}_v$ was absent ($\alpha = 0$). While the X-axis force profiles were similar between `p-DMPs` and `f-DMPs`, their Z-axis behaviors differed markedly: `f-DMPs` closely followed the demonstrated cutting force, whereas `p-DMPs` produced excessively large, unsustainable forces. The `cr-DMPs` exhibited force compensation effects comparable to `f-DMPs`. As shown in Fig. 9, `cr-DMPs` achieved superior test results ($\alpha = 0.3$), adapting effectively to food items with varying physical

properties. Due to the softness and fibrous structure of meat, all methods occasionally failed in manipulating meat (id_{11}, id_{12}). For fruits (id_1–id_6), both cr-DMPs and f-DMPs outperformed p-DMPs across all knife skills by deploying force compensation to adapt to different food stiffness. Similarly, f-DMPs performed better than p-DMPs on foods with hard (id_{10}) and soft (id_{14}) surfaces. Additionally, we presented the reference hardness and corresponding force compensation performance in Table I. Considering the reference hardness w, we observed that cutting performance could be similar even when w differed (e.g., id_4 vs. id_{14}). In particular, for surface-cutting motions (e.g., p-DMPs with y_3), foods with high w (id_1, id_3 vs. id_2, id_4) were difficult to cut because y_3 involved only a single pressing and sliding action. The reference hardness w served as a one-sided quantitative indicator; other physical factors also affected performance. For instance, among foods with similar high w (id_3 vs. id_{10}), the more fragile food (id_{10}) was harder to cut; among foods with similar low w (id_{13} vs. id_{14}), the food with higher viscosity (id_{13}) was harder to cut. Regarding the cosine similarity e_f, the temporal offset of cr-DMPs was corrected based on the Z-axis position to align with other trajectories, and analysis focused on the Z-axis force where variations were largest. Overall, p-DMPs exhibited low e_f, while cr-DMPs achieved high e_f across all food items and knife skills, reflecting the absence of force compensation and re-planning in p-DMPs. For hard items such as apple (id_3), p-DMPs reached e_f close to 0.9, indicating minimal compensation was needed. For soft items like cheese (id_{13}), p-DMPs showed significantly lower e_f than other methods, highlighting the importance of force compensation. For meat (id_{11}, id_{12}), all methods had low success rates across knife skills; correspondingly, e_f values were reduced, indicating that not all skills are suitable for foods with complex internal structures. Comparing p-DMPs and cr-DMPs, we conclude that imitating the force trajectory shape improves the success rate of knife skills applied to diverse food items.

Finally, the amplitude of force compensation influenced the manipulator's cutting speed. Using knife skill y_1 as an example, we report the average Z-axis cutting speeds for selected food items. For the hard item (id_3), p-DMPs achieved 0.941 cm/s, f-DMPs 1.357 cm/s, and cr-DMPs 1.418 cm/s. The faster speeds of f-DMPs and cr-DMPs were due to force compensation, which generated larger virtual forces. Additionally, re-planning allowed cr-DMPs to cut slightly faster than f-DMPs, depending on the magnitude of food position changes. For the soft and thin item (id_{11}), the cutting speeds were lower: p-DMPs 0.807 cm/s, f-DMPs 1.106 cm/s, and cr-DMPs 1.291 cm/s. This reduction is attributed to the lower force required for soft foods, which decreases both the necessary compensation force and the resulting cutting speed. As with hard foods, the use of force compensation and re-planning increased the cutting speed relative to p-DMPs.

2) Re-Planning With DMPs: As shown in Figs. 5 and 7, the proposed framework enabled the robot to reproduce knife skills on unknown food items while following similar trajectories. The food items varied in stiffness: the apple was hard, the kiwi was soft, and the fish was deformable. Each knife skill corresponded to a distinct cutting path. Since the properties of the food were unknown a priori, the re-planning method described in III-B4 dynamically adjusted the DMPs in real time, modifying the knife posture at appropriate time points to match the specific food.

As shown in Fig. 8(a), between 5–8 seconds, cr-DMPs exhibited greater X-axis position and rotation fluctuations than p-DMPs and f-DMPs, attributable to the changing cutting board posture during food transfer. The other methods maintained their original trajectories as they lacked replanning constraints. During 8–10 seconds, cr-DMPs demonstrated a controlled Z-axis descent to probe undetected food items, while p-DMPs and f-DMPs followed their predefined cutting paths. This exploration phase in cr-DMPs introduced a trajectory delay, visible in the Y-axis rotation peak at 15–20 seconds.

Then, knife skills y_1 and y_2 had similar trajectory, which results in y_2 appearing when testing y_1 or vice versa. Therefore, as shown in Fig. 9, for y_1 and y_2, all methods had the similar test result. For y_3, which required surface exploration before shallow slicing, p-DMPs and f-DMPs showed significantly lower success rates than for y_1/y_2. Without re-planning, these methods followed the original trajectory, unlike cr-DMPs which adapted effectively. Thanks to the re-planning policy, the cr-DMPs had the best result on y_3, and similar to f-DMPs, because of the force compensation policy deployment, it could distinguish y_1 and y_2 by force feedback, so it has a better success rate in these two knife skills than p-DMPs.

Finally, as shown in Table I, f-DMPs and cr-DMPs performed similarly due to their deployed force compensation mechanisms. The key distinction emerged in skill y_3, where food thickness significantly impacted results. When cutting thinner food items (id_{11} and id_{12}), it can be seen from e_f and success rate that f-DMPs could not complete the task well. The re-planning mechanism in cr-DMPs effectively compensated for these dimensional differences, reducing phase errors in the force trajectory and maintaining stable performance across varying food thicknesses.

We used the results of other similar letters [17] as baselines and compared them with our method to highlight key differences. In [17], a model-based (MPC) approach was used to learn the cutting policy, whereas our method is model-free. Their cutting duration was faster, between 1 and 5 seconds, while ours took 10 to 20 seconds—slower due to the need to ensure the availability of the derivative force signal f'_y. Regarding data requirements, the model in [17] was trained on 34 cutting trials over 6 objects, with 15 additional trials used for validation on the same object categories. In comparison, our approach only requires a single demonstration trajectory after parameter adjustment for all objects (20 objects). In summary, compared to [17], our method achieves a model-free cutting policy with a much simpler hardware setup and significantly lower data requirements, offering improved practicality and ease of deployment.

In conclusion, through the bilateral teleoperation system, we were able to generate demonstration trajectories that enabled the robot to imitate both position and force trajectory shapes. The compensation allowed the robot to apply forces closer to human levels, enhancing performance beyond simple tracking. The proposed re-planning constraints enabled the robot to dynamically adjust its trajectory, allowing for flexible adaptation to unknown environments.

V. Conclusion and Future Work

This work presents a novel framework for daily robotic manipulation of unknown food items, emulating human cutting techniques and transferring learned knife skills to unknown foods. A bilateral teleoperation system provides one-shot demonstration trajectories, capturing both the manipulator's position and force data from the slave device. These demonstrations are then reproduced by `cr-DMPs`, which incorporate force compensation to mimic the demonstrated force trajectories and enforce constraints for re-planning generated paths. `cr-DMPs` enables smooth, high-frequency action sequences without substantially increasing inference time, making the approach practical for real-world robotic applications. Overall, the proposed framework offers a gentle, demonstration-based solution for manipulating unknown food items, effectively learning from limited data while preserving high-quality motion, and represents a promising direction for future autonomous robotic systems.

However, there are several limitations to the proposed approach, which are the focus of ongoing and future work. First, knife techniques vary widely, and the strategies for each technique differ, making it beneficial to classify these techniques to better imitate the corresponding trajectories. Second, while a marker is used on the board to determine the food's posture, it cannot assess its thickness. Moreover, more complex knife skills can be attempted, such as removing potato sprouts, which require the assistance of precise visual detection methods. Lastly, completing a cut in 10 to 20 seconds does not meet the efficiency needs of actual life, and improving the cutting efficiency of knife skills is a challenge. In the future, variable thresholds can be added to the switching of each phase in replanning to make the switching smoother; A dual-arm framework could be developed, replacing the fixture with additional manipulators to hold the food items autonomously.

References

[1] I. Mitsioni, Y. Karayiannidis, J. A. Stork, and D. Kragic, "Data-driven model predictive control for the contact-rich task of food cutting," in *Proc. IEEE-RAS 19th Int. Conf. Humanoid Robots*, 2019, pp. 244–250.

[2] W. Xu et al., "Robotization and intelligent digital systems in the meat cutting industry: From the perspectives of robotic cutting, perception, and digital development," *Trends Food Sci. Technol.*, vol. 135, pp. 234–251, 2023.

[3] A. Straižys, M. Burke, and S. Ramamoorthy, "Learning robotic cutting from demonstration: Non-holonomic dmps using the udwadia-kalaba method," in *Proc. 2023 IEEE Int. Conf. Robot. Automat.*, 2023, pp. 5034–5040.

[4] X. Mu, Y. Xue, and Y.-B. Jia, "Robotic cutting: Mechanics and control of knife motion," in *Proc. 2019 Int. Conf. Robot. Automat.*, 2019, pp. 3066–3072.

[5] G. Gonzalez et al., "DESERTS: DElay-tolerant semi-autonomous robot teleoperation for surgery," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 12693–12700.

[6] C. C. Beltran-Hernandez, N. Erbetti, and M. Hamaya, "SliceIt!: Simulation-based reinforcement learning for compliant robotic food slicing," in *Proc. ICRA2024 Workshop Cooking Robotics: Percep. Motion Plan.*, 2024.

[7] Z. Xu et al., "RoboNinja: Learning an adaptive cutting policy for multi-material objects," in *Proc. Robotics: Sci. Syst.*, 2023.

[8] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Proc. Conf. Robot Learn.*, 2024.

[9] A. O'Neill et al., "Open X-embodiment: Robotic learning datasets and RT-X models : Open X-embodiment collaboration0," in *Proc. 2024 IEEE Int. Conf. Robot. Automat.*, 2024, pp. 6892–6903.

[10] K. Zhang, M. Sharma, M. Veloso, and O. Kroemer, "Leveraging multimodal haptic sensory data for robust cutting," in *Proc. 2019 IEEE-RAS 19th Int. Conf. Humanoid Robots*, 2019, pp. 409–416.

[11] A. Padalkar., M. Nieuwenhuisen., S. Schneider., and D. Schulz., "Learning to close the gap: Combining task frame formalism and reinforcement learning for compliant vegetable cutting," in *Proc. 17th Int. Conf. Informat. Control, Autom. Robot. - ICINCO*, INSTICC, 2020, pp. 221–231.

[12] O. M. Omisore, S. Han, J. Xiong, H. Li, Z. Li, and L. Wang, "A review on flexible robotic systems for minimally invasive surgery," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 1, pp. 631–644, Jan. 2022.

[13] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[14] Y.-H. Hwang, S.-R. Kang, S.-W. Cha, and S.-B. Choi, "A robot-assisted cutting surgery of human-like tissues using a haptic master operated by magnetorheological clutches and brakes," *Smart Mater. Structures*, vol. 28, no. 6, 2019, Art. no. 0 65016.

[15] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *Int. J. Robot. Res.*, vol. 37, no. 7, pp. 688–716, 2018.

[16] R. Rahal, F. Abi-Farraj, P. R. Giordano, and C. Pacchierotti, "Haptic shared-control methods for robotic cutting under nonholonomic constraints," in *Proc. 2019 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 8151–8157.

[17] I. Mitsioni, Y. Karayiannidis, and D. Kragic, "Modelling and learning dynamics for robotic food-cutting," in *Proc. IEEE 17th Int. Conf. Automat. Sci. Eng.*, 2021, pp. 1194–1200.

[18] A. Mason, D. Romanov, L. E. Cordova-Lopez, and O. Korostynska, "Smart knife: Integrated intelligence for robotic meat cutting," *IEEE Sensors J.*, vol. 22, no. 21, pp. 20475–20483, Oct. 2022.

[19] L. Han, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Vision-based cutting control of deformable objects with surface tracking," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 4, pp. 2016–2026, Apr. 2021.

[20] Y. Liu, C. Guo, and M. J. Er, "Robotic 3D laser-guided approach for efficient cutting of Porcine belly," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 2963–2972, May 2022.

[21] R. Wu and A. Billard, "Learning from demonstration and interactive control of variable-impedance to cut soft tissues," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 2740–2751, May 2022.

[22] J.-K. Lee and J.-H. Ryu, "Learning robotic rotational manipulation skill from bilateral teleoperation," in *Proc. 19th Int. Conf. Ubiquitous Robots*, 2022, pp. 318–325.

[23] K. Liang et al., "A robot learning from demonstration method based on neural network and teleoperation," *Arabian J. Sci. Eng.*, vol. 49, no. 2, pp. 1659–1672, 2024.

[24] A. M. Lyapunov, "The general problem of the stability of motion," *Int. J. Control*, vol. 55, no. 3, pp. 531–534, 1992.

[25] D. E. Trahan and G. J. Larrabee, "Effect of normal aging on rate of forgetting," *Neuropsychol.*, vol. 6, no. 2, p. 115, 1992.

[26] X. Mu, Y. Xue, and Y.-B. Jia, "Dexterous robotic cutting based on fracture mechanics and force control," *IEEE Trans. Automat. Sci. Eng.*, vol. 21, no. 4, pp. 5198–5215, Apr. 2024.

[27] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proc. 2016 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4193–4198.