

## Article

# Human–Robot Skill Transferring and Inverse Velocity Admittance Control for Soft Tissue Cutting Tasks

Kaidong Liu <sup>1,2</sup> , Bin Xie <sup>1,2,\*</sup>, Zhouyang Chen <sup>1,2</sup>, Zhenhao Luo <sup>1,2</sup>, Shan Jiang <sup>1,3</sup> and Zhen Gao <sup>1,3</sup>

<sup>1</sup> College of Engineering, China Agricultural University, Beijing 100083, China; liukd@cau.edu.cn (K.L.); chzy@cau.edu.cn (Z.C.); zhenhaoluo@cau.edu.cn (Z.L.); BS20233070701@cau.edu.cn (S.J.); B20203070561@cau.edu.cn (Z.G.)

<sup>2</sup> State Key Laboratory of Intelligent Agricultural Power Equipment, Beijing 100083, China

<sup>3</sup> Key Laboratory of Agricultural Equipment for Conservation Tillage, Ministry of Agricultural and Rural Affairs, Beijing 100083, China

\* Correspondence: xb0306@cau.edu.cn; Tel.: +86-136-6103-6672

**Abstract:** Robotic meat cutting is increasingly in demand in meat industries due to safety issues, labor shortages, and inefficiencies. This paper proposes a multi-demonstration human–robot skill transfer framework to address the flexible and generalized cutting of sheep hindquarters with complex 3D anatomy structures by imitating humans. To improve the generalization with meat sizes and demonstrations and extract target cutting behaviors, multi-demonstrations of cutting are encoded into low-dimension latent space through principal components analysis (PCA), Gaussian mixture model (GMM), and Gaussian mixture regression (GMR). To improve the robotic cutting flexibility and the cutting behavior reproducing accuracy, this study combines a modified dynamic movement primitive (DMP) high-level behavior generator with the low-level joints admittance control (AC) through real-time inverse velocity (IV) kinematics solving and constructs the IVAC-DMP control module. The experimental results show that the maximum residual meat thickness in the sheep hindquarter cutting of sample 1 is 3.1 mm, and sample 2 is 3.8 mm. The residual rates of samples 1 and 2 are 5.6% and 4.8%. Both meet the requirements for sheep hindquarter separation. The proposed framework is advantageous for harvesting high-value meat products and providing a reference technique for robot skill learning in interaction tasks.



**Citation:** Liu, K.; Xie, B.; Chen, Z.; Luo, Z.; Jiang, S.; Gao, Z. Human–Robot Skill Transferring and Inverse Velocity Admittance Control for Soft Tissue Cutting Tasks. *Agriculture* **2024**, *14*, 394. <https://doi.org/10.3390/agriculture14030394>

Academic Editor: José Lima

Received: 18 December 2023

Revised: 25 February 2024

Accepted: 27 February 2024

Published: 29 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** sheep hindquarters; robotic meat cutting; learning from demonstration; admittance control; Gaussian mixture regression

## 1. Introduction

Meat cutting is a crucial component of the meat industry's value chain. This process involves breaking down a carcass into primal parts and precisely harvesting meat cuts with varying values [1,2]. Skilled workers with a profound understanding of animal anatomy are vital to guaranteeing the final cuts' consistency and quality and determining their market prices. However, the traditional manual cutting methods have several issues, including biological variation, training, sanitation concerns, and harsh working conditions [3]. With the recent labor shortages, supply inefficiencies caused by COVID-19, and other limitations, the meat industry has seen a considerable increase in the need for automation. Robotic solutions offer a promising approach to address these challenges by enabling human-like cutting operations, reducing labor costs, improving food safety, and promoting a sustainable meat supply [4].

Robotic cutting is achieved through three main methods [5], i.e., primal sawing, 2D-space, and 3D-space cutting. The primal sawing method is efficient in primal cutting and middle pieces separation [6]. However, the simple sawing actions are insufficient for harvesting meat cuts from forequarters, hindquarters, rib cages, and carcass flanks. The 2D-space methods have been primarily adopted for precisely cutting primal pieces [7,8],

such as carcass flanks or rib cages, utilizing deep learning and multi-DoF robot systems. The 2D-space methods are highly limited in cutting meat with more complex 3D anatomy structures, e.g., sheep hindquarters. The motion planning and control of robot cutting for sheep hindquarter separation is difficult due to several factors, i.e., cutting scenario variability, cutting close to the bone, variable bone size, and complex anatomic structures [9]. Compliant cutting in a 3D space can precisely separate meat tissue from bones or different meat tissues from each other. Xie et al. [10] proposed an admittance control method to cut along the ischial bone with comparable efficiency and accuracy to humans for sheep hindquarter cutting. However, the cutting path was constrained on several planes around the bone, which limited the flexibility of the cutting. Meat and Livestock Australia (MLA) proposed a force-controlled robot for sheep hindquarter cutting [11]. The robot equipped a blade with an optimized shape to lower the cutting forces. However, the generalization of the cutting motions is not addressed. Furthermore, a robotic prototype was proposed to cut the shoulders of the sheep forequarter [12] through an oscillating cutting motion and force control. It is a problem to generalize the oscillating cutting pattern between the forequarters and hindquarters. In addition, Nabil et al. [13] utilized a finite element model to accurately predict the deformation of the meat tissue during cutting and control the robot's motion. Although the study allows for precise tracking of the fascial tissues between the meat tissues of a beef hind leg, the bones were not considered. More work is needed to achieve flexible and generalized motion planning while coupling with the force control.

Robots have been trained to execute flexible motions in interaction tasks based on the human–robot skill transfer (HRST) framework [14]. The framework involves a high-level nonlinear dynamical system such as a dynamic movement primitive (DMP) and a low-level compliant controller [15]. The nonlinear dynamical system combines a set of radial basis functions (RBFs) that mimic the muscle unit behaviors in humans to generate flexible motion trajectories [16]. Meanwhile, the low-level controller is an impedance controller (IC) or admittance controller (AC) for the robot joints, which handles the interaction between the robot and objects [17,18]. The DMP model has been coupled with feedback terms related to the interaction force to form a Cartesian space AC [19,20] in grinding, handling, and robot collaboration tasks. However, the method was highly limited by inverse kinematics solving (IKS), since the desired trajectories were modulated online. In the primary research [21], we utilized the DMP model to generate the desired movement in sheep hindquarter separation and the inverse velocity (IV) algorithm for online IKS of the robot with AC. However, the low-level joint control has no feedback to the DMP model, which limits the flexibility of the method. The studies mentioned above consider joint space control and high-level planning separately, so the robotic cutting flexibility and the cutting behavior reproducing accuracy are highly limited. More studies combining low-level joint AC and the high-level DMP behavior system by real-time IV kinematics solving are in demand to improve the integration, consistency, and efficiency of the collaboration between the two systems.

Moreover, The studies mentioned above are single-demonstrated, whose generalization is a crucial issue due to the cutting objects' variable types, locations, and dimensions. The generalization can be addressed through multi-demonstration HRST. Gams et al. [22] proposed the compliant movement primitive (CMP) model to learn the interactive forces and motion trajectories and utilized the Gaussian process regression (GPR) model to encode multi-demonstrations. Low motion tracking error and compliance control were achieved. In addition, Wu and Billard [23] combined the Stable Estimator of Dynamical Systems and Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) to learn a compliant cutting behavior model from multiple demonstrations. The model was then integrated into the IC of the robot for cutting single-layered skin-like soft tissue and responding to human interactions. The studies mentioned above are implemented under an IC framework. An expensive lightweight backdrivable robot arm is needed to produce joint torques precisely. In the task of cutting hindquarters, HRST through multi-demonstration on a robot with AC and real-time IV remains an open problem.

To address the efficient collaboration between human-like behavior generation and joint-space compliant control and achieve flexible and generalized hindquarter cutting, this paper proposes a multi-demonstration HRST framework. To improve the generalization with meat sizes and demonstrations and extract target cutting behaviors, multi demonstrations of cutting are encoded into a low-dimension latent space through principal components analysis (PCA), Gaussian mixture model (GMM), and Gaussian mixture regression (GMR). To improve the robotic cutting flexibility and the cutting behavior reproducing accuracy, this study combines a modified dynamic movement primitive (DMP) high-level behavior generator with the low-level joints admittance control (AC) through real-time inverse velocity (IV) kinematics solving and constructs the IVAC-DMP control module. The proposed framework is evaluated in four cutting scenarios. The proposed method has the potential to reduce the cutting force, minimize meat loss, and increase meat yield, which is advantageous for harvesting high-value meat products. Additionally, it can serve as a reference technique for flexible and generalized robot skill learning in interaction tasks.

The rest of the paper is structured as follows. The proposed framework is presented in Section 2. Section 2.2 presents the data acquisition and preprocessing. The multi-demonstration human–robot skill transfer with GMM-GMR is then explained in Section 2.3. Moreover, Section 2.4 demonstrates the IVAC of robot joints. The evaluation and the discussion are presented in Section 3. The conclusion is provided in Section 4.

## 2. Materials and Methods

This section provides a detailed description of the proposed framework for imitating and controlling the cutting of soft tissue by robots.

### 2.1. System Overview

#### 2.1.1. Cutting Behavior Representation

The cutting behavior is represented by  $\mathbf{h}_d(t)$ ,

$$\begin{aligned}\mathbf{H}_d &= [\mathbf{h}_d(t_1), \mathbf{h}_d(t_2), \dots, \mathbf{h}_d(t_M)], \\ \mathbf{F}_d &= [\mathbf{f}_d(t_1), \mathbf{f}_d(t_2), \dots, \mathbf{f}_d(t_M)], \\ \mathbf{T}_d &= [t_1, t_2, \dots, t_M], \\ \mathbf{h}_d(t) &= [\mathbf{x}_d(t), \mathbf{q}_d(t)]^T, \\ \mathbf{x}_d(t) &= [x_{d,x}(t), x_{d,y}(t), x_{d,z}(t)]^T, \\ \mathbf{q}_d(t) &= [q_{d,w}(t), q_{d,x}(t), q_{d,y}(t), q_{d,z}(t)]^T, \\ \mathbf{f}_d(t) &= [f_{d,x}(t), f_{d,y}(t), f_{d,z}(t), \tau_{d,x}(t), \tau_{d,y}(t), \tau_{d,z}(t)]^T,\end{aligned}\tag{1}$$

where  $t$  is the time, and  $\mathbf{H}_d$  is the collected time series of cutting behavior.  $M$  is the number of time points in the time series.  $\mathbf{F}_d$  is the collected time series of cutting force.  $\mathbf{T}_d$  is the time points series. The position of the cutting tool in the world coordinate system is represented by  $\mathbf{x}_d(t)$ , which has  $x_{d,x}(t)$ ,  $x_{d,y}(t)$ ,  $x_{d,z}(t)$  as coordinates for the  $x$ ,  $y$ ,  $z$  axes, respectively. The quaternion of the cutting tool orientation in the world coordinate system is represented by  $\mathbf{q}_d(t)$ , which consists of  $q_{d,w}(t)$  as the scalar part and  $q_{d,x}(t)$ ,  $q_{d,y}(t)$ ,  $q_{d,z}(t)$  as the imaginary part. The cutting force/torque measured by the F/T sensor is represented by  $\mathbf{f}_d(t)$ , where  $f_{d,x}(t)$ ,  $f_{d,y}(t)$ ,  $f_{d,z}(t)$  are the decoupled measured forces of  $x$ ,  $y$ ,  $z$  while  $\tau_{d,x}(t)$ ,  $\tau_{d,y}(t)$ ,  $\tau_{d,z}(t)$  are the decoupled measured torques of the  $x$ ,  $y$ ,  $z$  axes, respectively.

#### 2.1.2. Process of Training and Control

The process of imitation learning and control is divided into three stages, as shown in Figure 1. By giving multiple groups of human behavior time-series samples in the same cutting scenario, we train a GMM in the probability distribution of human behavior at each time point. Then, we generate the target cutting behavior through GMR with the trained GMM. The target cutting behavior is transformed into a DMP system in this work

to generate the target cutting behavior (position, orientation, and feedforward cutting force) of the robot end effector in the base coordinate system. Finally, we input the target cutting behavior to the IVAC module to control the joint angles of the robot during cutting to achieve human-like flexible cutting.

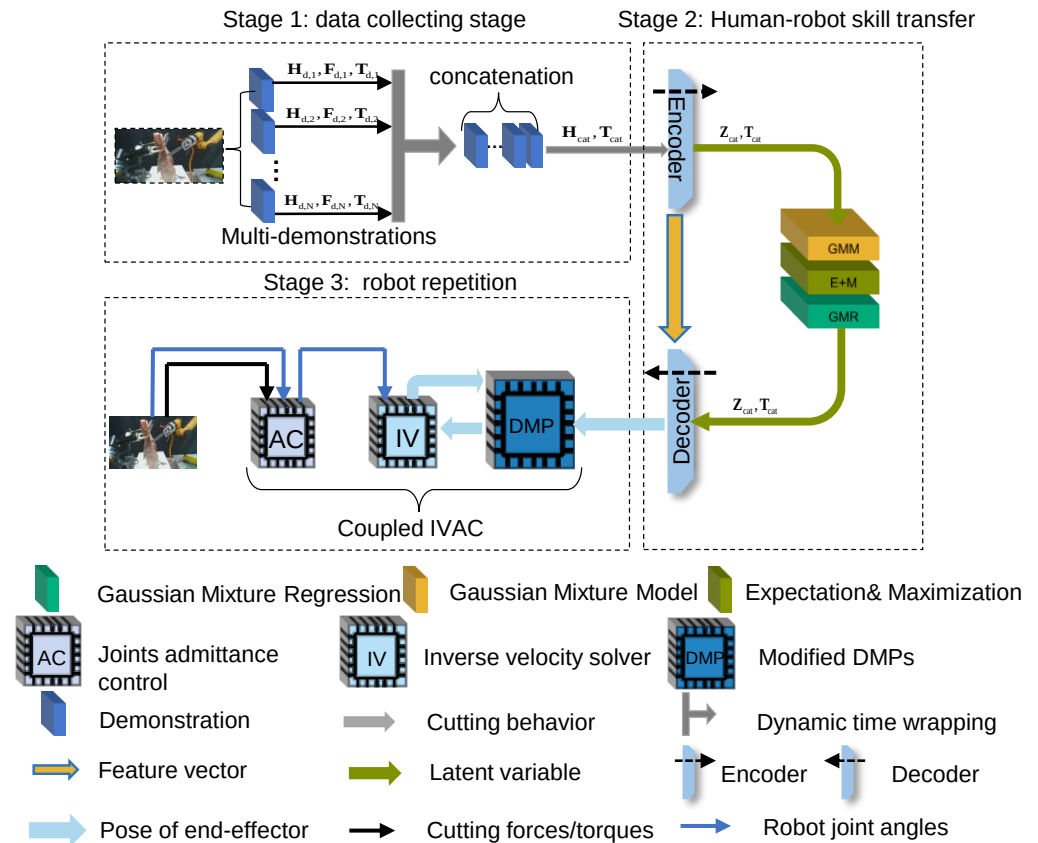


**Figure 1.** Diagram of the human-robot skill transfer and control framework.

### 2.1.3. Experiment Platform

As is shown in Figure 2, the experiment platform includes a loading device, a cutting tool, a visual motion capture system, and a multi-degree-of-freedom robot arm. The cutting tool is constructed by connecting a meat separation blade with a six-axis F/T sensor using a metal frame. The ArUco augmented reality marker [24] is fixed to the left side of the cutting tool, and two intel RealSense D435i cameras track its pose to form a visual motion capture system. The D435i cameras detect the ArUco marker at a frequency of 30 Hz. They verify the marker's encoding ID and utilize the corner detection tool in OpenCV to determine the marker's position and orientation in the camera's coordinate system. This process enables the camera to track the marker accurately. The marker's ID is 123. This unique ID distinguishes and verifies other QR codes within view. The marker is 60 mm × 60 mm, ensuring clear visibility from a preset distance. The marker is a finely machined anodized aluminum sheet with distinct corners and low reflection. Meanwhile, the F/T sensor measures the decoupled forces and torques in the x, y, and z axes.

The F/T sensor has flanges at the front and rear; the front flange is connected to the metal frame of the cutting tool, and the rear flange can be connected with the flange of the robot end or with a 3D-printed grip. When the rear flange connects with the grip, the demonstrator holds the cutting tool by the grip to cut. Otherwise, the robot arm manipulates the cutting tool to cut.

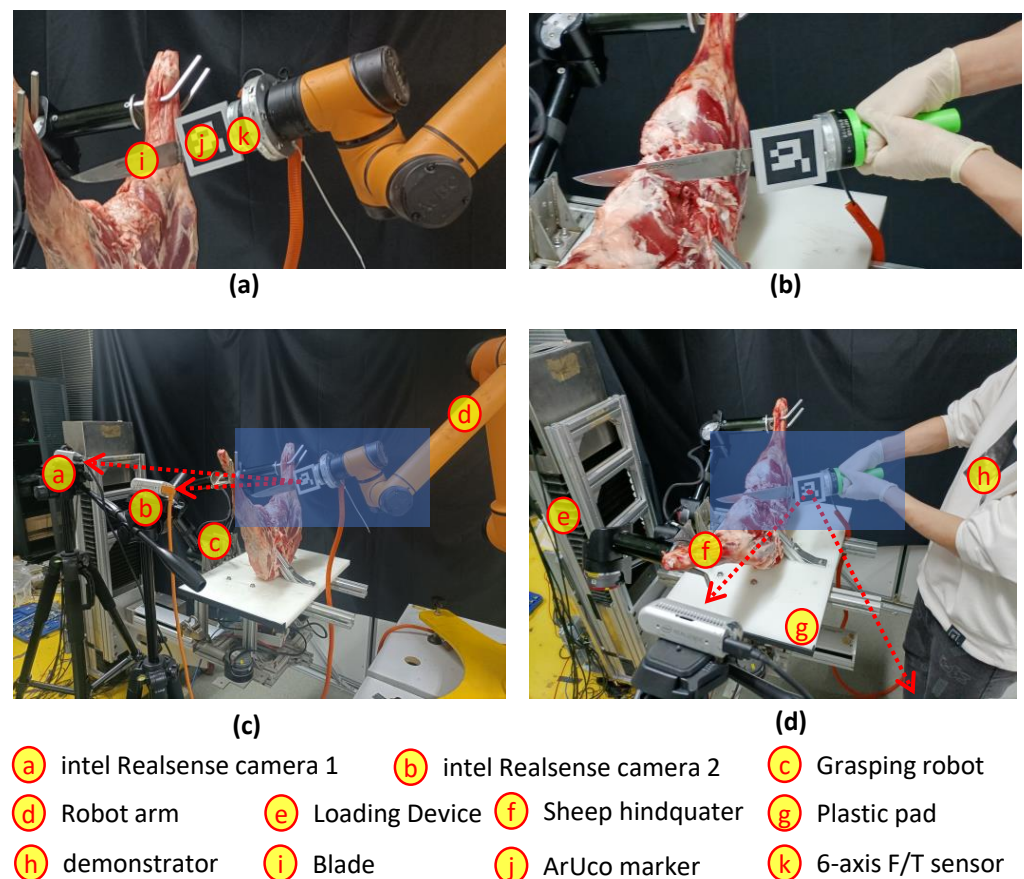


**Figure 2.** Experiment Platform. (a) the cutting tool is manipulated by the robot, (b) the cutting tool is manipulated manually, (c) the set of robotic cutting, (d) the set of demonstration. The cutting tool is manipulated manually first to sample multiple demonstrations. Then, the robot is trained and reproduces the same cutting behavior.

## 2.2. Data Acquisition and Preprocessing

### 2.2.1. Data Acquisition

This study collects the cutting behavior and force data in four demonstrating scenarios to construct the datasets. The experiment was conducted in the College of Engineering, China Agricultural University with indoor white light and at a room temperature of 17 °C. The experiment materials include the sheep meat from the local (Beijing) meat plants and foam blocks. The raw data are calibrated, aligned, and then utilized to train the GMM-GMR model. To ensure the robot learns the exact human behaviors, this study only uses the actual demonstrations from real humans, and no data augmentation methods are employed. The datasets include 9570 data points (2348 from scenario 1, 884 from scenario 2, 1358 from scenario 3, and 4980 from scenario 4) from 38 demonstration samples (4 from scenario 1, 2 from scenario 2, 2 from scenario 3, and 30 from scenario 4). Scenario 1 is cutting the medial hind-leg muscles attached with the ischium ventral; 2 is cutting the lateral hind-leg muscles attached with the ischium dorsal; 3 is cutting the lateral hind-leg muscles attached with the ilium dorsal; 4 is to segment a foam block pasted on a rigid surface.

### 2.2.2. Data Calibration and Filtering

The raw data are calibrated to obtain the transform matrices between the robot base coordinate system  $W$ , the robot end mounting flange center coordinate system  $E$ , the F/T sensor coordinate system  $S$ , the marker coordinate system  $T$ , the Camera 1 coordinate system  $C_1$ , and the Camera 2 coordinate system  $C_2$ .  $T_{WE}$  represents the transform matrix of  $W$  and  $E$ .  $T_{ES}$  represents the transform matrix of  $E$  and  $S$ .  $T_{C_1T}$  represents the transform

matrix of  $C_1$  and  $T$ .  $T_{C_2T}$  represents the transform matrix of  $C_2$  and  $T$ .  $T_{WC_1}$  represents the transform matrix of  $W$  and  $C_1$ .  $T_{WC_2}$  represents the transform matrix of  $W$  and  $C_2$ .  $T_{TE}$  represents the transform matrix of  $T$  and  $E$ . The data calibration process includes seven steps:

1.  $T_{WE}$  is the homogeneous matrix representation of  $E$ 's pose in  $W$ . The pose is read from the low-level embedded controller of the robot. Since the end mounting flange and the F/T sensor are connected by a mechanical structure,  $T_{ES}$  can be pre-obtained.
2.  $T_{C_1T}$  is the homogeneous matrix representation of  $T$ 's pose in  $C_1$ , and  $T_{C_2T}$  is the homogeneous matrix representation of  $T$ 's pose in  $C_2$ .  $T_{C_1T}$  and  $T_{C_2T}$  are provided by the ArUco marker detection algorithm. Furthermore,  $T_{WE}$ ,  $T_{C_1T}$ , and  $T_{C_2T}$ ,  $T_{WC_1}$ ,  $T_{WC_2}$ ,  $T_{TE_1}$ , and  $T_{TE_2}$  are obtained through hand-eye calibration.
3. We convert the measured pose of  $T$  on  $C_1$  and  $C_2$  to the pose of  $E$  on  $W$  with Equation (2) using the calibrated  $T_{WC_1}$ ,  $T_{WC_2}$ ,  $T_{TE_1}$ , and  $T_{TE_2}$ ,

$$\begin{aligned} T_{WE_1} &= T_{WC_1} T_{C_1T} T_{TE_1}, \\ T_{WE_2} &= T_{WC_2} T_{C_2T} T_{TE_2}. \end{aligned} \quad (2)$$

4. We rewrite the homogeneous matrices  $T_{WE_1}$  and  $T_{WE_2}$  into pose vectors and obtain the position  $\mathbf{x}_d$  and quaternion  $\mathbf{q}_d$  of the cutting tool in the world coordinate system by Kalman filtering the pose vectors from the two camera sources.
5. We convert the cutting forces measured on the coordinate system  $S$  to the coordinate system  $E$  by (3) using the calibrated  $T_{SE}$ ,

$$\mathbf{f}_e = [\mathbf{Ad}_{T_{SE}}]^T \mathbf{f}_s, \quad (3)$$

where  $\mathbf{f}_e$  denotes the cutting force on  $E$ ,  $\mathbf{f}_s$  denotes the cutting force on  $S$ , and  $[\mathbf{Ad}_{T_{SE}}]$  denotes the adjoint representation of  $T_{SE}$ .

6. To ensure accurate cutting force measurements, we use a gravity-compensated calibration method [25] to eliminate the impact of the cutting tool's self-weight on the collected data. The resulting cutting force is denoted as  $\mathbf{f}_d$ . The components of this force,  $f_{d,x}$ ,  $f_{d,y}$ , and  $f_{d,z}$ , represent different directions.  $f_{d,x}$  points towards the side of the blade,  $f_{d,z}$  points towards the tip, and  $f_{d,y}$  points towards the edge, all within the blade plane.
7. Dynamic time warping (DTW) is utilized to align the sample time-series length in the same scenario in a Python environment.

### 2.2.3. Latent Space Representation of Dataset

In order to improve learning efficiency, this study utilizes the encoding of the high-dimension time series in the dataset into the one-dimension latent variable series  $\mathbf{Z}_{\text{cat}}$ . First, the samples in the dataset are concatenated, as shown in Equation (4),

$$\begin{aligned} \mathbf{H}_{\text{cat}} &= [\mathbf{H}_{d,1}, \mathbf{H}_{d,2}, \dots, \mathbf{H}_{d,L}], \\ \mathbf{T}_{\text{cat}} &= [\mathbf{T}_{d,1}, \mathbf{T}_{d,2}, \dots, \mathbf{T}_{d,L}], \end{aligned} \quad (4)$$

where  $\mathbf{H}_{\text{cat}}$  denotes the concatenated series of cutting behavior.  $\mathbf{T}_{\text{cat}}$  denotes the concatenated series of time points.  $L$  denotes the number of samples.  $\mathbf{H}_{d,1}, \mathbf{H}_{d,2}, \dots, \mathbf{H}_{d,L}$  are the samples. Second, we perform a principal components analysis (PCA) on  $\mathbf{H}_{\text{cat}}$ . The PCA extracts the correlation and importance information among different axes of the cutting tool in the latent variables. The feature vector with the largest eigenvalue obtained by PCA is as follows,

$$\mathbf{W}_{\text{pca}} = [w_{\text{pca},1}, w_{\text{pca},2}, \dots, w_{\text{pca},6}], \quad (5)$$

where  $\mathbf{W}_{\text{pca}}$  denotes the feature vector with elements  $w_{\text{pca},1}, w_{\text{pca},2}, \dots, w_{\text{pca},6}$ . Then,  $\mathbf{H}_{\text{cat}}$  is transformed into its latent variable representation using Equation (6),

$$\mathbf{Z}_{\text{cat}} = \mathbf{W}_{\text{pca}} \mathbf{H}_{\text{cat}}, \quad (6)$$

where  $\mathbf{Z}_{\text{cat}}$  denotes the latent variable series with a length of  $L \times M$ .

### 2.3. Learning of Cutting Behaviors with GMM and GMR

#### 2.3.1. Learning from Multi-demonstrations of Cutting Behaviors with GMM

The GMM is a statistical model for representing a given population as several Gaussian-distributed subpopulations. Therefore, it is suitable for learning cutting behavior with multi-demonstrations from the same scenario [26]. In this study, the GMM is trained on the dataset  $\{\mathbf{Z}_{\text{cat}}, \mathbf{T}_{\text{cat}}\}$ , with the state vector consisting of the cutting behavior latent variable  $z_{\text{cat}}$  and time  $t_{\text{cat}}$ , as shown in Equation (7),

$$p(\xi) \sim \sum_{k=1}^K \pi_k \mathcal{N}_k(\mu_k, \Sigma_k), \text{ with } \sum_{k=1}^K \pi_k = 1, \quad (7)$$

$$\xi = \begin{bmatrix} t_{\text{cat}} \\ z_{\text{cat}} \end{bmatrix}, \mu_k = \begin{bmatrix} \mu_{t,k} \\ \mu_{z,k} \end{bmatrix}, \Sigma_k = \begin{bmatrix} \Sigma_{t,k} & \Sigma_{tz,k} \\ \Sigma_{zt,k} & \Sigma_{z,k} \end{bmatrix},$$

where  $p(\xi)$  is the probability distribution of state vectors, and  $p(\xi)$  is the mixture of  $K$  multivariate Gaussians.  $\mathcal{N}_k(\mu_k, \Sigma_k)$  represents the  $k$ th Gaussian with a center  $\mu_k$ , a covariance matrix  $\Sigma_k$ , and a weight  $\pi_k$ .

The values of  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  are obtained by training on input data through the expectation maximization (EM) method. The EM algorithm iteratively improves the similarity between the GMM and the realistic distribution of data samples to achieve maximum likelihood estimation of the sample distribution. The iterative optimization process involves initialization, E-step, and M-step:

1. The mixture model consists of 10 Gaussians. To reduce the sensitivity of the EM algorithm to the selection of the initial values, we use k-means to initialize the centers and covariance matrices of the Gaussians. After initializing, we train the GMM with the dataset of the four cutting scenarios that were collected;
2. The E-step calculates the intermediate variable  $\gamma_{m,k}$  using current  $\mu_k$  and  $\Sigma_k$ , as is shown in Equation (8),

$$\gamma_{m,k} = \frac{\pi_k \mathcal{N}_k(\xi_m | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k \mathcal{N}_k(\xi_m | \mu_k, \Sigma_k)}. \quad (8)$$

where  $\gamma_{m,k}$  denotes the condition probability of  $\xi_m$  generating from the  $k$ th Gaussian;

3. Then, the M-step updates  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  using  $\gamma_{m,k}$  calculated in the E-step, as is shown in Equation (9),

$$\begin{aligned} \pi_k^{\text{new}} &= \frac{\sum_{m=1}^M \gamma_{m,k}}{M}, \\ \mu_k^{\text{new}} &= \frac{\sum_{m=1}^M \gamma_{m,k} \xi_m}{\sum_{m=1}^M \gamma_{m,k}}, \\ \Sigma_k^{\text{new}} &= \frac{\sum_{m=1}^M \gamma_{m,k} (\xi_m - \mu_k)(\xi_m - \mu_k)^T}{\sum_{m=1}^M \gamma_{m,k}}, \end{aligned} \quad (9)$$

where  $\pi_k^{\text{new}}$  denotes the weight after updating.  $\mu_k^{\text{new}}$  denotes the center after updating.  $\Sigma_k^{\text{new}}$  denotes the covariance matrix after updating.

#### 2.3.2. Robotic Cutting Behavior Generation with GMR

As is shown in Figure 3, this study employs GMR to generate the target latent space trajectory  $\mathbf{Z}_{\text{ref}}$  for the cutting behaviors. In Figure 3a–d, the green-colored dots represent the points in the datasets, and thick red lines represent regression trajectories generated

by the GMR. The ellipses represent the Gaussians in the GMM. The radius and angle of every ellipse represent a Gaussian covariance matrix  $\Sigma_k$ , and the crosses represent the mean values of the Gaussians. Figure 3e–h show the corresponding cutting paths. The latent variable trajectories are almost averaged out by the GMM-GMR model in the four scenarios. Significant differences are found in the GMM and GMR trajectories learned in the four scenarios, since the cutting paths vary.

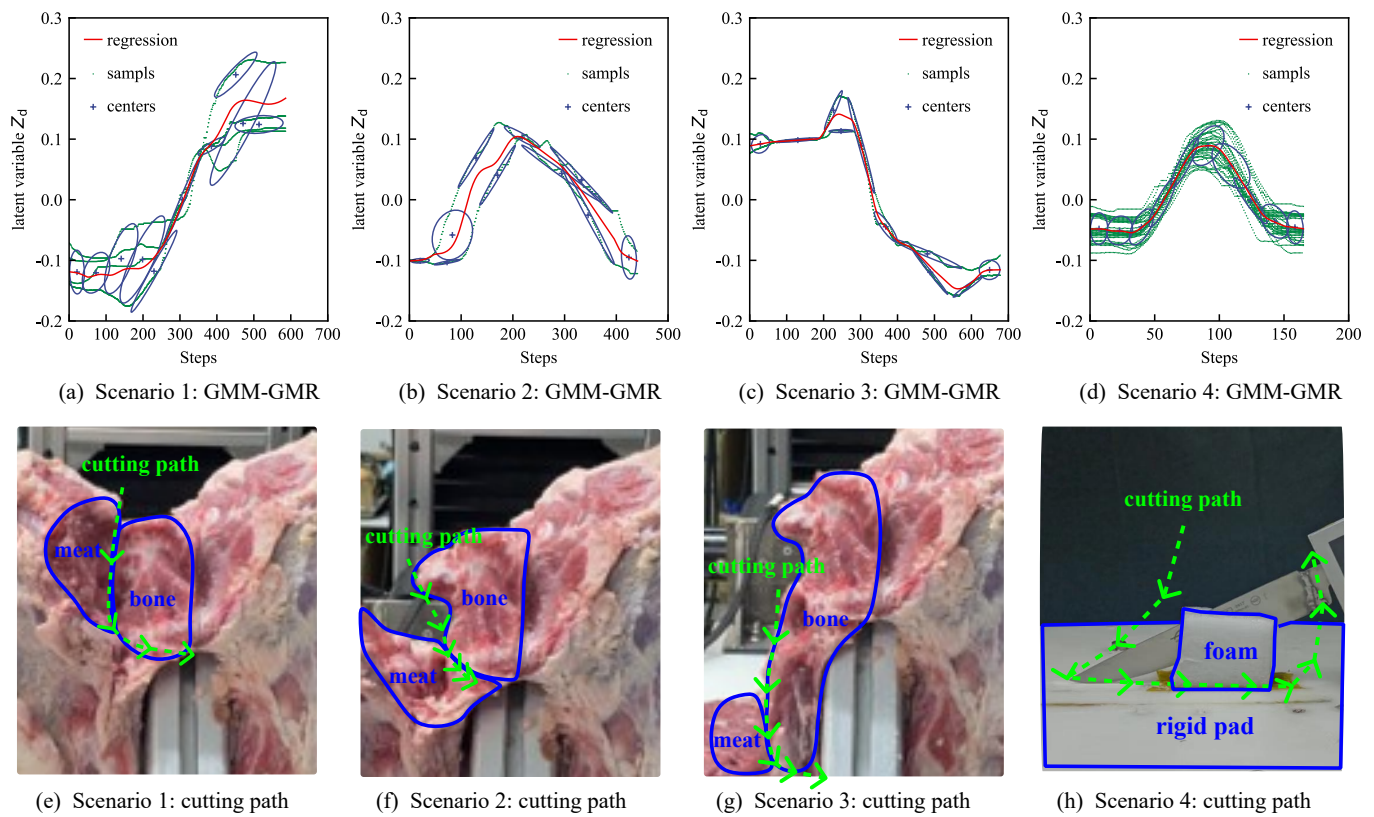


**Figure 3.** GMM-GMR models and generated latent variable trajectories in 4 scenarios. (a) the learned Gaussian components of GMM and the trajectory generated by GMR in Scenario 1, (b) the learned Gaussian components of GMM and the trajectory generated by GMR in Scenario 2, (c) the learned Gaussian components of GMM and the trajectory generated by GMR in Scenario 3, (d) the learned Gaussian components of GMM and the trajectory generated by GMR in Scenario 4, (e) the medial hind-leg muscles cutting path in Scenario 1, (f) the lateral hind-leg muscles cutting path in Scenario 2, (g) the lateral hind-leg muscles cutting path in Scenario 3, (h) the foam cutting path in Scenario 4.

The regression function is calculated using the following equation:

$$z_{\text{ref}}(t) = \hat{\mu}_z(t) = \sum_{k=1}^K w_k(t) \hat{\mu}_{z,k}(t), \quad (10)$$

with,

$$\begin{aligned} \hat{\mu}_{z,k}(t) &= \mu_{z,k} + \Sigma_{z,k} \Sigma_{t,k}^{-1} (t - \mu_{t,k}), \\ w_k(t) &= \frac{\pi_k \mathcal{N}(t | \mu_{t,k}, \Sigma_{t,k})}{\sum_{k=1}^K \pi_k \mathcal{N}(t | \mu_{t,k}, \Sigma_{t,k})}, \end{aligned} \quad (11)$$

where  $z_{\text{ref}}(t)$  denotes the regression value of the latent variable at time  $t$ .  $\hat{\mu}_{z,k}(t)$  denotes the regression value of  $\mu_{z,k}$  at time  $t$ , and  $w_k$  denotes the weight.  $\mu_{z,k}$ ,  $\Sigma_{t,k}$ , and  $\Sigma_{z,k}$  are parameters defined in Equation (7). In this study, the values of the latent variables were

calculated using Formula (10) at each time point, and the reference cutting behavior time series  $\mathbf{Z}_{\text{ref}}$  in latent space was obtained,

$$\mathbf{Z}_{\text{ref}} = [z_{\text{ref}}(t_1), z_{\text{ref}}(t_2), \dots, z_{\text{ref}}(t_M)], \quad (12)$$

### 2.3.3. Post-Processing

The latent variable series of cutting behavior obtained by GMR were post-processed to generate the reference cutting behavior time series. First, by decoding the sequence of latent variables in (12), the reference cutting behavior (positions, quaternions, and forces) time series is generated,

$$\begin{aligned} \mathbf{H}_{\text{ref}} &= [h_{\text{ref}}(t_1), h_{\text{ref}}(t_2), \dots, h_{\text{ref}}(t_M)], \\ \mathbf{H}_{\text{ref}} &= \mathbf{W}_{\text{pca}}^T \cdot \mathbf{Z}_{\text{ref}}, \\ h_{\text{ref}}(t_i) &= \mathbf{W}_{\text{pca}}^T \cdot z_{\text{ref}}(t_i), \text{ with } i = 1, \dots, M, \end{aligned} \quad (13)$$

where the symbol  $\mathbf{H}_{\text{ref}}$  refers to the time series of reference cutting behaviors. Each element  $h_{\text{ref}}(t_i)$  is a cutting behavior vector in the form of Equation (1) at time  $t_i$ . The symbol  $\mathbf{W}_{\text{pca}}$  refers to the eigenvector in Equation (5). The symbol  $z_{\text{ref}}(t_i)$  is a latent variable at time  $t_i$ . The time series has a length of  $M$ .

According to Equation (1), the time series  $\mathbf{H}_{\text{ref}}$  of reference cutting behaviors is divided into  $\mathbf{X}_{\text{ref}}$ ,  $\mathbf{Q}_{\text{ref}}$ , and  $\mathbf{F}_{\text{ref}}$ ,

$$\begin{aligned} \mathbf{X}_{\text{ref}} &= [\mathbf{x}_{\text{ref}}(t_1), \mathbf{x}_{\text{ref}}(t_2), \dots, \mathbf{x}_{\text{ref}}(t_M)], \\ \mathbf{Q}_{\text{ref}} &= [\mathbf{q}_{\text{ref}}(t_1), \mathbf{q}_{\text{ref}}(t_2), \dots, \mathbf{q}_{\text{ref}}(t_M)], \\ \mathbf{F}_{\text{ref}} &= [\mathbf{f}_{\text{ref}}(t_1), \mathbf{f}_{\text{ref}}(t_2), \dots, \mathbf{f}_{\text{ref}}(t_M)], \end{aligned} \quad (14)$$

where  $\mathbf{X}_{\text{ref}}$  denotes the reference tool position sequence,  $\mathbf{Q}_{\text{ref}}$  denotes the reference tool orientation sequence, and  $\mathbf{F}_{\text{ref}}$  denotes the reference cutting force sequence.

Furthermore, the first- and second-order derivatives of the reference positions and quaternions are calculated,

$$\begin{aligned} \mathbf{V}_{\text{ref}} &= [\mathbf{v}_{\text{ref}}(t_1), \mathbf{v}_{\text{ref}}(t_2), \dots, \mathbf{v}_{\text{ref}}(t_M)], \\ \dot{\mathbf{V}}_{\text{ref}} &= [\dot{\mathbf{v}}_{\text{ref}}(t_1), \dot{\mathbf{v}}_{\text{ref}}(t_2), \dots, \dot{\mathbf{v}}_{\text{ref}}(t_M)], \\ \boldsymbol{\Omega}_{\text{ref}} &= [\boldsymbol{\omega}_{\text{ref}}(t_1), \boldsymbol{\omega}_{\text{ref}}(t_2), \dots, \boldsymbol{\omega}_{\text{ref}}(t_M)], \\ \dot{\boldsymbol{\Omega}}_{\text{ref}} &= [\dot{\boldsymbol{\omega}}_{\text{ref}}(t_1), \dot{\boldsymbol{\omega}}_{\text{ref}}(t_2), \dots, \dot{\boldsymbol{\omega}}_{\text{ref}}(t_M)], \end{aligned} \quad (15)$$

with

$$\begin{aligned} \mathbf{v}_{\text{ref}}(t_i) &= \frac{\mathbf{x}_{\text{ref}}(t_i) - \mathbf{x}_{\text{ref}}(t_{i-1})}{t_i - t_{i-1}}, \quad i > 1, \\ \dot{\mathbf{v}}_{\text{ref}}(t_i) &= \frac{\mathbf{v}_{\text{ref}}(t_i) - \mathbf{v}_{\text{ref}}(t_{i-1})}{t_i - t_{i-1}}, \quad i > 1, \\ \boldsymbol{\omega}_{\text{ref}}(t_i) &= \frac{\mathbf{d}_q(\mathbf{q}_{\text{ref}}(t_{i-1}), \mathbf{q}_{\text{ref}}(t_i))}{t_i - t_{i-1}}, \quad i > 1, \\ \dot{\boldsymbol{\omega}}_{\text{ref}}(t_i) &= \frac{\boldsymbol{\omega}_{\text{ref}}(t_i) - \boldsymbol{\omega}_{\text{ref}}(t_{i-1})}{t_i - t_{i-1}}, \quad i > 1, \end{aligned} \quad (16)$$

where  $t_i$  denotes the time with  $t_1 = 0$ ,  $\mathbf{v}_{\text{ref}}(t_1) = 0$  m/s,  $\dot{\mathbf{v}}_{\text{ref}}(t_1) = 0$  m/s<sup>2</sup>,  $\boldsymbol{\omega}_{\text{ref}}(t_1) = 0$  rad/s, and  $\dot{\boldsymbol{\omega}}_{\text{ref}}(t_1) = 0$  rad/s<sup>2</sup>. The unit quaternion representation is adopted to avoid the orientation singularity.  $\mathbf{d}_q(\mathbf{q}_{\text{ref}}(t_{i-1}), \mathbf{q}_{\text{ref}}(t_i))$  denotes the orientation offset, which maps the quaternion offset into angular velocity space for derivation.

## 2.4. Admittance Control in Joint Space Coupling with DMPs

### 2.4.1. Dynamic Motion Primitives of Target Cutting Behaviors

(1) Activation Function: Like original DMPs, we convert time variables into phase space activation signals on  $[0, 1)$  through an activation function. We take the Sigmoid system (17) with upper and lower limits as the activation function

$$\tau \dot{s} = \alpha_s s(1 - s), \quad (17)$$

where  $s$  is the phase space activation signal, and  $\alpha_s = 2 \log(0.01/0.99)$ .  $s$  starts at 0.99 and decreases to 0.01 at  $t = \tau$  to simulate neural signal activation and attenuation.

(2) Modified DMPs model: first, considering the transformation relationship between quaternions and angular velocities, we extend the model proposed by Hoffmann et al. [27] to the quaternion space and build the following improved DMP model (18) for reference cutting behavior,

$$\tau^2 \begin{bmatrix} \mathbf{a}_d \\ \alpha_d \end{bmatrix} = \begin{bmatrix} \mathbf{K}_p \\ \mathbf{K}_q \end{bmatrix} \begin{bmatrix} s(\mathbf{w}_p^T \mathbf{g} - \mathbf{x} + \mathbf{x}_0) + (1 - s)(\mathbf{x}_g - \mathbf{x}) \\ s(\mathbf{w}_q^T \mathbf{g} - \mathbf{d}_q(\mathbf{q}, \mathbf{q}_0)) + (1 - s)\mathbf{d}_q(\mathbf{q}_g, \mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{D}_p & \mathbf{D}_q \end{bmatrix} \begin{bmatrix} \mathbf{v}_d \\ \omega_d \end{bmatrix}, \quad (18)$$

with

$$\tau \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\eta} * \mathbf{q}/2 \end{bmatrix}, \quad \tau \begin{bmatrix} \mathbf{a}_d \\ \alpha_d \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{v}}_d \\ \dot{\omega}_d \end{bmatrix} \quad (19)$$

where  $\mathbf{K}_p, \mathbf{K}_q, \mathbf{D}_p, \mathbf{D}_q$  are constant coefficients.  $\tau$  is the time scaling factor.  $\mathbf{x}$  denotes the end-effector position,  $\mathbf{x}_0$  the initialing position, and  $\mathbf{x}_g$  the final position.  $\mathbf{q}$  denotes the quaternion of the end effector's orientation,  $\mathbf{q}_0$  the initialing quaternion, and  $\mathbf{q}_g$  the final quaternion.  $\mathbf{v}$  denotes the velocity of the end effector and  $\omega$  the angular velocity.  $\mathbf{a}_d$  denotes the acceleration of the end effector and  $\alpha_d$  the angular acceleration.  $\boldsymbol{\eta}$  is a quaternion with a scalar part of 0 and an imaginary part of  $\omega$ .  $\mathbf{d}_q : \mathbb{R}^4 \mapsto \text{so}(3)$  is the logarithmic mapping of the quaternion deviation to angular velocity space, as defined in [28].  $\mathbf{g}$  denotes the radial basis vector, and  $\mathbf{w}_p$  and  $\mathbf{w}_q$  are the parameter matrices of  $N \times 3$ . The  $n$ th element of  $\mathbf{g}$  is defined as follows

$$[\mathbf{g}]_n = \frac{\phi_n(s)}{\sum_{n=1}^N \phi_n(s)}, \quad (20)$$

with,

$$\phi_n(s) = \exp(-h_n(s - c_n)^2), \quad (21)$$

where  $\phi_n(s)$  denotes the  $n$ th Gaussian radial basis function,  $h_n$  the width of this basis, and  $c_n$  the center of this basis.

The Equation (18) defines a virtual critical damping system with attraction points  $[\mathbf{x}_g, \mathbf{q}_g]^T$  and  $[\mathbf{w}_p^T \mathbf{g}, \mathbf{w}_q^T \mathbf{g}]^T$ .  $[\mathbf{x}_g, \mathbf{q}_g]^T$  converges the system towards the final position and  $[\mathbf{w}_p^T \mathbf{g}, \mathbf{w}_q^T \mathbf{g}]^T$  adjusts the shape of the motion trajectory. The importance of  $[\mathbf{x}_g, \mathbf{q}_g]^T$  and  $[\mathbf{w}_p^T \mathbf{g}, \mathbf{w}_q^T \mathbf{g}]^T$  to the system is weighted by  $1 - s$  and  $s$ . When  $t$  is small, the shape of the generated trajectory is more important, and as  $s$  trends toward 0, convergence to the final position  $[\mathbf{x}_g, \mathbf{q}_g]^T$  becomes more important.

(3) Learning Parameters:  $\mathbf{Y}_p$  and  $\mathbf{Y}_q$  are the targets derived from the demonstration trajectory time series,

$$\begin{aligned} \mathbf{Y}_p &= [\mathbf{y}_p(t_1), \mathbf{y}_p(t_2), \dots, \mathbf{y}_p(t_M)], \\ \mathbf{Y}_q &= [\mathbf{y}_q(t_1), \mathbf{y}_q(t_2), \dots, \mathbf{y}_q(t_M)], \end{aligned} \quad (22)$$

with

$$\begin{aligned} \mathbf{y}_p(t) &= -s[\mathbf{x}_{\text{ref}}(t) - \mathbf{x}_{\text{ref}}(t_1)] + (1-s)[\mathbf{x}_{\text{ref}}(t_M) - \mathbf{x}_{\text{ref}}(t) - \frac{\mathbf{D}_p \mathbf{v}_{\text{ref}}(t)}{\mathbf{K}_p} - \frac{\mathbf{a}_{\text{ref}}(t)}{\mathbf{K}_p}], \\ \mathbf{y}_q(t) &= -s\mathbf{d}(\mathbf{q}_{\text{ref}}(t), \mathbf{q}_{\text{ref}}(t_1)) + (1-s)\mathbf{d}(\mathbf{q}_{\text{ref}}(t_M), \mathbf{q}_{\text{ref}}(t)) - \frac{\mathbf{D}_q \boldsymbol{\omega}_{\text{ref}}(t)}{\mathbf{K}_q} - \frac{\boldsymbol{\alpha}_{\text{ref}}(t)}{\mathbf{K}_q}. \end{aligned} \quad (23)$$

The reference trajectory and feedforward force jointly describe the cutting behavior. The parameter matrices of the nonlinear terms  $\mathbf{w}_p$  and  $\mathbf{w}_q$  are the representations of the cutting behavior in the parameter space.

$\mathbf{w}_p$  and  $\mathbf{w}_q$  are learned from the demonstration data by solving the optimization problems (24),

$$\begin{aligned} \min \quad & \sum_{i=1}^M \|\mathbf{w}_p^T \mathbf{g}(s_i) s_i - \mathbf{y}_p(t_i)\|, \\ \min \quad & \sum_{i=1}^M \|\mathbf{w}_q^T \mathbf{g}(s_i) s_i - \mathbf{y}_q(t_i)\|, \end{aligned} \quad (24)$$

where, at each time step,  $i$ ,  $s_i$  represents the activation signal, and  $\mathbf{g}(s_i)$  is the corresponding radial basis vector. The parameter matrices to be solved are denoted as  $\mathbf{w}_p$  and  $\mathbf{w}_q$ . We utilized the locally weighted linear regression (LWR) method to solve Equation (24) in this study.

#### 2.4.2. Robot Joints Admittance Control

In this study, we implement the proposed algorithm on an Aubo-i7 robot arm (Shanghai, China). The arm has six joints. The controller interacts with the low-level embedded motion controller of the robot to set and read the joints' position. The interaction is through robot operating system (ROS) software architecture and TCP-CAN transparent transmission at 50 Hz. Regarding the robot body as rigid, we design the following admittance control law (25) to the joints' motion for safe and effective cutting.

$$\mathbf{K}_j(\boldsymbol{\theta}_m - \boldsymbol{\theta}_r) - \mathbf{D}_j(\dot{\boldsymbol{\theta}}_m - \dot{\boldsymbol{\theta}}_r) = \Delta \boldsymbol{\tau}, \quad (25)$$

where  $\mathbf{k}_j$  is the joint stiffness coefficient matrix and  $\mathbf{d}_j$  the joint damping coefficient matrix.  $\boldsymbol{\theta}_m$  is the command positions of the joints.  $\delta \boldsymbol{\tau}$  is the joints' torque offsets. We conduct the following transformation (26) to transform the cutting force offsets in the coordinate system of the end effector into the joints' torque offsets

$$\Delta \boldsymbol{\tau} = \mathbf{J}^T(\mathbf{F}_{fb} - \mathbf{F}_{ff}), \quad (26)$$

where,  $\mathbf{F}_{ff}$  is the feedforward force,  $\mathbf{F}_{fb}$  is the measured feedback force.  $\mathbf{K}_j$  and  $\mathbf{D}_j$  are calculated by Equations (27) and (28)

$$\mathbf{K}_j = \mathbf{J}^T \boldsymbol{\Gamma} \mathbf{K} \mathbf{J}, \quad (27)$$

$$\mathbf{D}_j = \mathbf{J}^T \boldsymbol{\Gamma}^{\frac{1}{2}} \mathbf{D} \mathbf{J}, \quad (28)$$

$$\text{where } \mathbf{D} = 2\sqrt{\mathbf{K}\mathbf{M}}, \mathbf{M} = \begin{bmatrix} m_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & m_6 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} k_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & k_6 \end{bmatrix}, \text{ and } \boldsymbol{\Gamma} = \begin{bmatrix} \gamma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma_6 \end{bmatrix}.$$

$\mathbf{k}$  is the stiffness coefficient matrix on the end effector coordinates, and  $\mathbf{d}$  is the damping coefficient matrix on the end effector coordinates.  $\boldsymbol{\Gamma}$  is the stiffness adjustment factor matrix, and its diagonal elements  $\gamma_1, \dots, \gamma_6$  are the stiffness adjustment factors in the direction of the corresponding axis of the end effector. The initial values of  $\gamma_1, \dots, \gamma_6$  are all 1. The stiffness and damping of the admittance control in different directions can be set separately by changing  $\gamma_1, \dots, \gamma_6$ .

The admittance control converts the end effector force offsets into the angular velocity of the robot's joints. Based on the Equations (25)–(28), we propose the following update law (29) and (30) of the velocities and angles of the joints

$$\dot{\theta}_m \leftarrow \dot{\theta}_r + (\mathbf{J}^T \mathbf{D} \mathbf{J})^{-1} [\mathbf{J}^T (\mathbf{F}_{fb} - \mathbf{F}_{ff}) - \mathbf{J}^T \mathbf{K} \mathbf{J} (\theta_m - \theta_r)], \quad (29)$$

$$\theta_m \leftarrow \theta_m + \dot{\theta}_m dt, \quad (30)$$

After that, the joints' angles are sent to the low-level embedded motion controller of the robot for execution. This paper does not include a discussion of the low-level embedded motion controller.

#### 2.4.3. Inverse Velocity Motion Control

The robot updates the reference pose and feedforward force each time step during cutting. First, it updates  $\dot{\mathbf{v}}_d$  and  $\dot{\omega}_d$  using Formula (18). Moreover,  $\mathbf{v}_d$  and  $\omega_d$  are updated via numerical integrals (31)

$$\begin{bmatrix} \mathbf{v}_d \\ \omega_d \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{v}_d \\ \omega_d \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{v}}_d \\ \dot{\omega}_d \end{bmatrix} \Delta t, \quad (31)$$

Then, the reference position  $\mathbf{x}_d$  and the reference pose  $\mathbf{q}_d$  are updated

$$\mathbf{x}_d \leftarrow \mathbf{x} + \frac{\mathbf{v}_d \Delta t}{\tau} \quad (32)$$

$$\mathbf{q}_d \leftarrow \exp_q\left(\frac{\omega_d \Delta t}{2\tau}\right) * \mathbf{q}. \quad (33)$$

To achieve continuous tracking of the end effector's motion and avoid joint angle and velocity limits, we need to solve for the inverse kinematics [29–31] and obtain the command joint angles  $\theta_r$ , as described in Equation (25). In detail, we use the instantaneous kinetic energy of the robot joints as the cost function. The constraints we take into account include the first-order forward kinematics of the robot and the physical limitations of the joint angles and velocities. The joint velocities are then obtained by solving the following optimization problem

$$\min_{\theta_r} \quad \dot{\theta}_r^T \mathbf{M} \dot{\theta}_r, \quad (34)$$

$$\text{s.t.} \quad \mathbf{J} \dot{\theta}_r = [\mathbf{v}_r, \omega_r]^T, \quad (35)$$

$$\theta_r^{\min} \leq \theta_r \leq \theta_r^{\max}, \quad (36)$$

$$\dot{\theta}_r^{\min} \leq \dot{\theta}_r \leq \dot{\theta}_r^{\max}. \quad (37)$$

where  $\mathbf{M}$  is the inertial matrix. Equation (34) is the instantaneous kinetic energy of the joint to be optimized. Equation (35) is the forward kinematic constraint. Equation (36) is the constraint on the physical position of the joint angle. Equation (37) is the physical velocity constraint of the joint. The solution to the above optimization problem is

$$\dot{\theta}_{opt} = \mathbf{J}^\dagger [\mathbf{v}_d, \omega_d]^T. \quad (38)$$

In Equation (38), the pseudo-inverse  $\mathbf{J}^\dagger$  of the joint Jacobi  $\mathbf{J}$  is calculated as

$$\mathbf{J}^\dagger = \mathbf{M}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T)^{-1}. \quad (39)$$

In this study, we have introduced compensation in Formula (38) to eliminate the cumulative error of numerical calculation. It ensures that the robot tracks the motions generated by the modulated DMPs. The final calculation Formula (40) of  $\dot{\theta}_r$  is as follows:

$$\dot{\theta}_r = \dot{\theta}_{opt} + \mathbf{J}^\dagger \log_6(\mathbf{X}_r^{-1} \mathbf{X}_d), \quad (40)$$

where  $\mathbf{X}_r$  is the homogeneous matrix of the end effector pose at  $\theta_r$  derived from the forward kinematics of the robot.  $\mathbf{X}_d$  is the homogeneous matrix of the expected end effector pose, which corresponds to  $\mathbf{x}_d, \mathbf{q}_d$ .  $\log_6 : \mathbf{SE}(3) \mapsto \mathbb{R}^6$  is the logarithmic mapping of the robot's homogeneous matrix to the twist in motion.

### 3. Results and Discussion

#### 3.1. Human–Robot Skill Transfer

We train the proposed method on the datasets mentioned in Section 2.2 to demonstrate its effectiveness and applicability in learning and generating soft tissue cutting behaviors.

It can be observed from Figure 4 that the PCA eigenvector heat maps vary remarkably in scenarios 1, 2, 3, and 4. This suggests that the latent space variables in Equation (6) are able to effectively identify the cutting behaviors of four scenarios, although we use PCA to calculate one-dimensional latent space variables for simplicity.

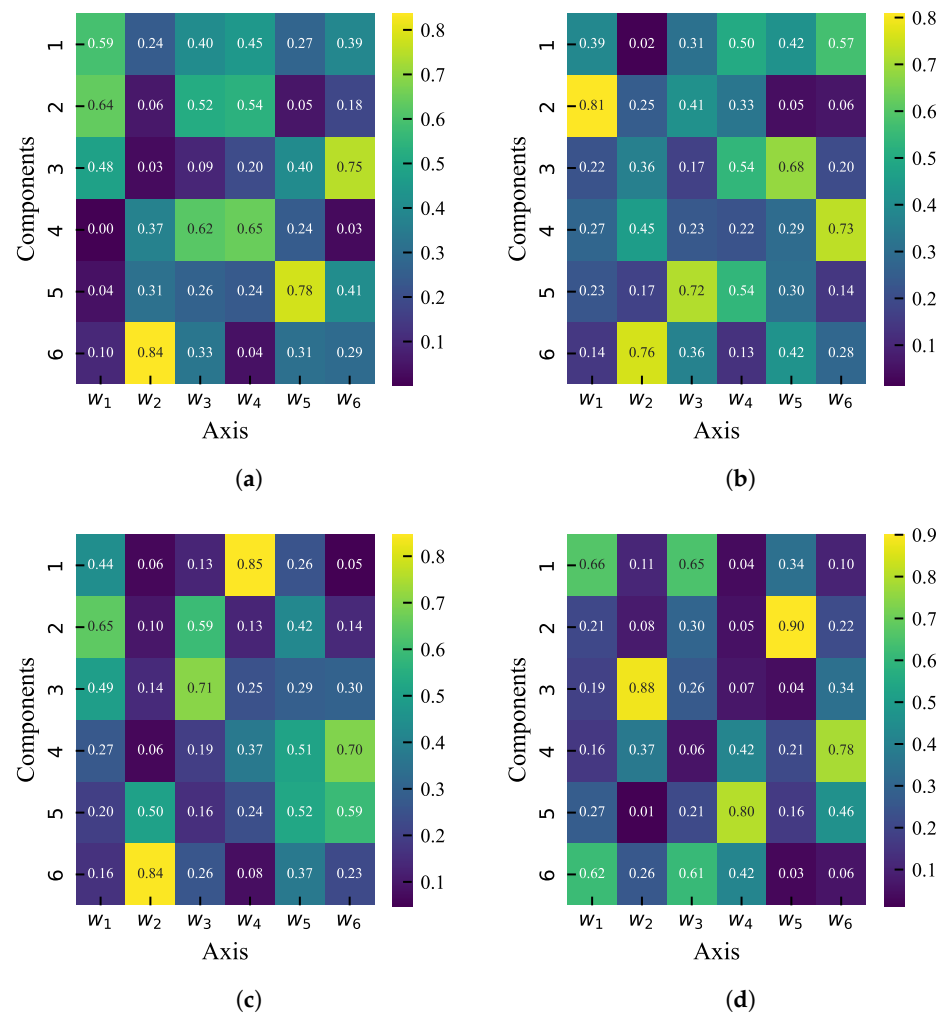


**Figure 4.** PCA Feature Matrix: (a) heat map of PCA eigenvectors in scenario 1, (b) heat map of PCA eigenvectors in scenario 2, (c) heat map of PCA eigenvectors in scenario 3, (d) heat map of PCA eigenvectors in scenario 4.

Table 1 shows the explanation rates of GMM, axis weights of PCA, and pose reconstruction errors of GMR in four scenarios. We find in Table 1 that Scenario 1 has the smallest difference in axis weights, indicating a more complex contact situation. In this scenario, there are both contacts between the side of the cutting blade and the ischial surface and

contact between the cutting blade and the ischial surface. Moreover, it can be observed from Table 1 that  $w_2$  is small among the four scenarios. This indicates that when the side of the cutting knife is in contact with the rigid medium, the large elastic deformation may reduce the measured value of the cutting force in this direction. From the perspective of the interpretation rate, on the one hand, the interpretation rates of Scenarios 4 and 2 are 74.71% and 80.96%, indicating that in these two scenarios, the GMM-GMR module in this work can effectively represent the cutting behavior in the latent space. On the other hand, the explanation rates of scenarios 1 and 3 are nearly 50.00%, indicating that in these two scenarios, the effectiveness of latent space variables in representing cutting behavior can be further improved.

In addition, in Scenarios 2, 3, and 4, the position reconstruction error is less than 0.05 m, the position reconstruction error of Scenario 3 is close to 0.05 m, and in the four scenarios, the attitude reconstruction error is less than 0.05 rad. Considering the jitter and operating accuracy during human demonstrations, this suggests that the latent space GMM-GMR module can effectively generate the target cutting behaviors for the robot in all the four scenarios.

**Table 1.** Results of latent variable encoding and generation

| Scenario | Samples | $w_1$   | $w_2$    | $w_3$   | $w_4$    | $w_5$   | $w_6$    | $PE_{avg}$ (m) | $OE_{avg}$ (rad) | Ratio (%) |
|----------|---------|---------|----------|---------|----------|---------|----------|----------------|------------------|-----------|
| 1        | 4       | 0.3864  | −0.01621 | −0.3109 | −0.4989  | 0.4202  | −0.5730  | 0.056          | 0.032            | 55.50     |
| 2        | 2       | 0.4390  | −0.06339 | 0.1329  | −0.8468  | 0.2571  | −0.04962 | 0.038          | 0.025            | 80.96     |
| 3        | 2       | −0.6637 | −0.1099  | 0.6496  | −0.04189 | −0.3370 | −0.1010  | 0.030          | 0.040            | 48.80     |
| 4        | 30      | −0.5927 | −0.2432  | −0.4050 | −0.4467  | −0.2707 | 0.3908   | 0.047          | 0.038            | 74.71     |

$w_1, w_2, w_3, w_4, w_5$ , and  $w_6$  denote the elements of the PCA eigenvector.

### 3.2. Foam Cutting Test

We carry out a foam-cutting experiment to verify the reliability and effectiveness of the proposed framework in adapting to the position changes of the rigid medium under soft tissue. The robot uses the learned parameters and fixed stiffness coefficient to perform one cutting each at the loading platform heights of +30 mm, +20 mm, +10 mm, 0 mm, −10 mm, −20 mm, and −30 mm. The to-be-cut soft tissue (a foam block) is pasted on the surface of the rigid medium, which is a plastic pad used for food processing. We fix the plastic pad on the loading robot platform, and the loading robot platform can adjust the height of the plastic pad by itself. The size of the foam block used is 60 × 50 × 50 mm, and the height of the plastic backing plate at the beginning of the test is 0 mm. As is shown in the Figure 5, the cutting process can be divided into four phases, i.e., entering, cutting, releasing, and returning.

As shown in Figure 6, in each test, the robot successfully separates the foam. The first phase is 0–5.0 s, the second phase is 5.0–10.5 s, the third phase is 10.5–15.0 s, and the fourth phase is 15.0–20.0 s. In the first phase, the main contact direction is the Z-axis. The direction of  $F_x$  is negative, the direction of  $F_y$  is positive, and the direction of  $F_z$  is negative. On the one hand, the changes in  $F_x$  and  $F_y$  are small, falling from 0 N to approximately −5 N and rising from 0 N to approximately 5 N, respectively. Both  $F_x$  and  $F_y$  are within the safe range of the force sensor and robot payload. On the other hand,  $F_z$  decreases rapidly. The second phase is the cutting phase.  $F_y$  rises from approximately 5 N to 25 N and maintains the great pressure within the safety range of the force sensor and robot load. In this phase,  $F_y$  is mainly used to apply pressure on the plastic plate to ensure effective cutting of the thin layer in the boundary between the foam and the plastic plate and to compensate for the resistance of the plastic backing plate to the cutting tool.  $F_x$  further dropped from −5 N to −10 N.  $T_y$  and  $F_x$  have similar trends. Both  $F_x$  and  $T_y$  are close to the data during human demonstration.  $F_x$  and  $T_y$  are mainly derived from the demonstration. When demonstrating, the plane of the blade and the plastic plate are not perpendicular. The third stage is the release stage. The cutting force of each axis gradually returns to 0 N. From the perspective of motion, there is only a minor deviation between the

test trajectories and the demonstration trajectories. This shows that after the cutting blade leaves the plastic plate and foam, the IVAC algorithm can effectively adjust the motion trajectories and reduce the offset between the target trajectory and the actual trajectory.

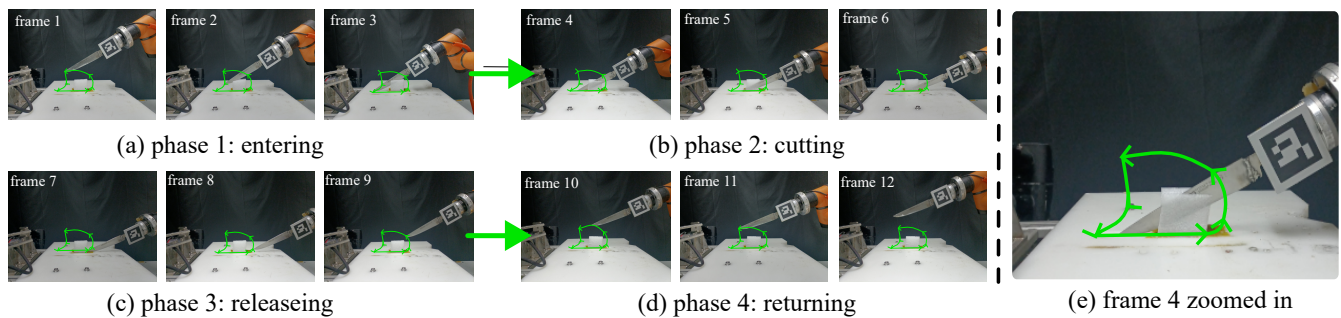


**Figure 5.** Scheme of foam cutting process. (a) The cutting tool cuts into the foam toward the front and comes into contact with the plastic plate in the entering phase. (b) The robot then splits the foam from front to back in the cutting phase. (c) After cutting, the blade leaves the plastic plate and foam. (d) Finally, the robot returns to the start position and orientation through free movement. (e) frame 4 zoomed in showing the cutting trajectory.

Table 2 presents the maximum, minimum, and average forces on each axis of the force sensor during robot cutting at different platform heights, as well as the reference values measured during human teaching. When the platform height is lower than  $-10$  mm, its amplitude is close to the human demonstration. As the platform height increases, The amplitude of  $F_z$  is also larger. However, in each test,  $F_z$  is still within the safe range of the force sensor and robot load.

**Table 2.** Foam cutting forces during robot cutting at different platform heights.

|       |         | +30 mm  | +20 mm  | +10 mm  | +00 mm  | −10 mm  | −20 mm  | −30 mm  | Human   |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| $F_x$ | maximum | 2.459   | 1.646   | 0.285   | 0.233   | 0.245   | 0.248   | 0.317   | 0.682   |
|       | minimum | −9.376  | −9.301  | −9.054  | −8.683  | −8.583  | −8.548  | −8.235  | −10.271 |
|       | average | −2.560  | −2.582  | −2.579  | −2.383  | −2.286  | −2.227  | −2.158  | −3.177  |
| $F_y$ | maximum | 34.400  | 33.778  | 32.592  | 31.604  | 30.917  | 30.180  | 29.119  | 36.452  |
|       | minimum | −0.063  | −0.155  | 0.001   | −0.031  | −0.031  | −0.133  | −0.237  | −0.189  |
|       | average | 12.463  | 11.747  | 11.136  | 10.558  | 10.200  | 9.749   | 9.202   | 12.985  |
| $F_z$ | maximum | 20.068  | 20.783  | 13.843  | 12.388  | 13.660  | 15.097  | 13.447  | 13.313  |
|       | minimum | −21.865 | −19.712 | −22.068 | −19.825 | −16.831 | −13.500 | −12.252 | −11.229 |
|       | average | 2.289   | 2.698   | 1.468   | 0.923   | 1.467   | 2.161   | 2.039   | 2.084   |
| $T_x$ | maximum | 0.028   | 0.004   | 0.004   | 0.003   | 0.003   | 0.004   | 0.008   | 0.016   |
|       | minimum | −9.282  | −9.112  | −8.871  | −8.682  | −8.548  | −8.392  | −8.178  | −8.478  |
|       | average | −3.237  | −3.105  | −2.940  | −2.827  | −2.741  | −2.658  | −2.540  | −2.747  |
| $T_y$ | maximum | 0.727   | 0.485   | 0.046   | 0.044   | 0.044   | 0.048   | 0.061   | 0.041   |
|       | minimum | −2.371  | −2.335  | −2.315  | −2.280  | −2.259  | −2.249  | −2.215  | −2.241  |
|       | average | −0.653  | −0.643  | −0.635  | −0.619  | −0.607  | −0.598  | −0.571  | −0.657  |
| $T_z$ | maximum | 0.003   | 0.003   | 0.002   | 0.006   | 0.004   | 0.002   | 0.003   | 0.001   |
|       | minimum | −0.227  | −0.227  | −0.204  | −0.217  | −0.213  | −0.204  | −0.192  | −0.202  |
|       | average | −0.047  | −0.040  | −0.036  | −0.036  | −0.038  | −0.037  | −0.033  | −0.041  |

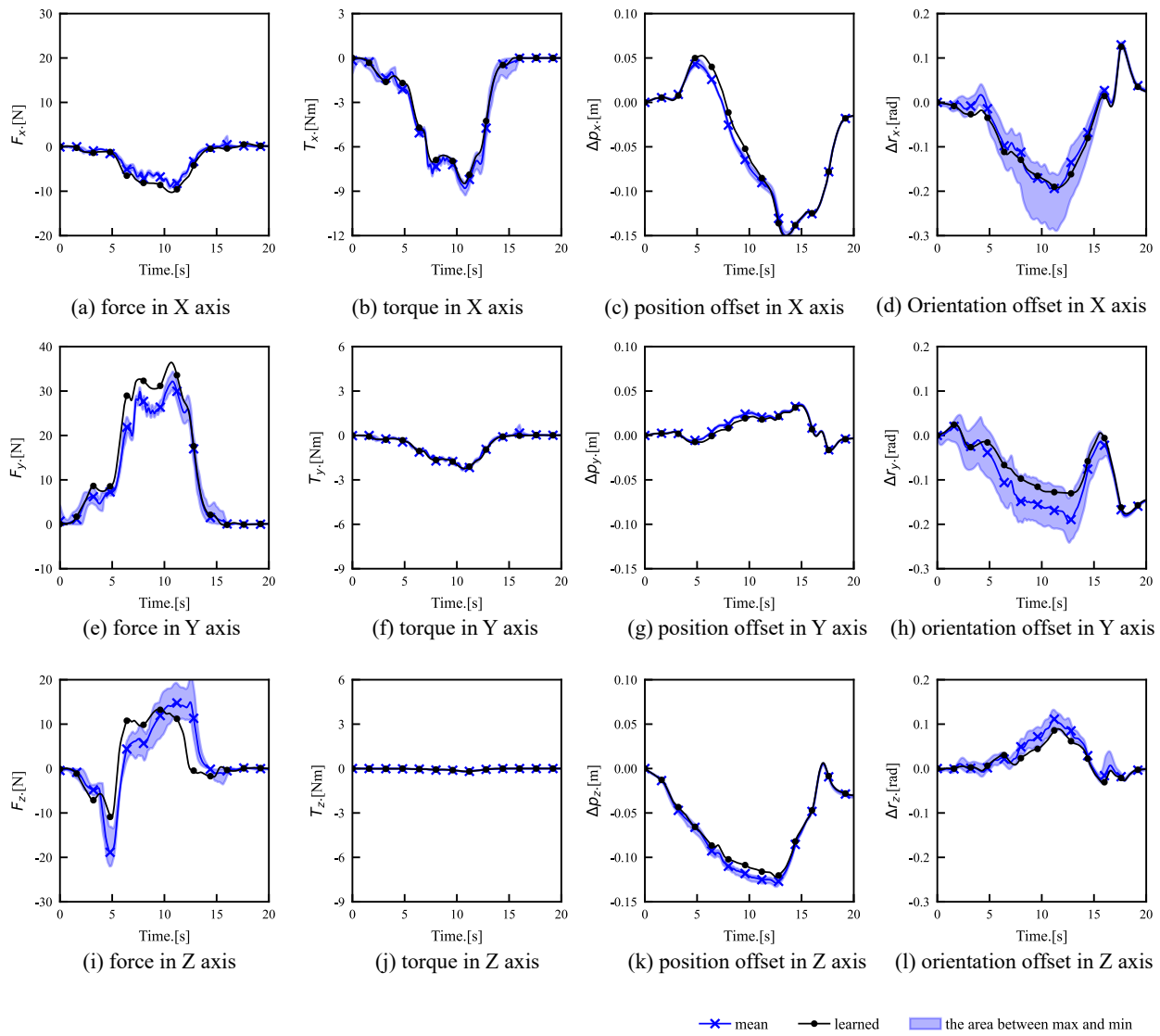


**Figure 6.** Robot cutting force and motion. (a) force in X axis, (b) torque around X axis, (c) position in X axis, (d) rotation angle around X axis, (e) force in Y axis, (f) torque around Y axis, (g) position in Y axis, (h) rotation angle around Y axis, (i) force in Z axis, (j) torque around Z axis, (k) position in Z axis, (l) rotation angle around Z axis. ‘mean’ denotes the average curves of cutting forces, torques, positions, and orientations. ‘learned’ denotes the desired curves of cutting forces, torques, positions, and orientations learned from the demonstration. The first and second columns show the robot cutting forces and torques at different platform heights. The third and fourth columns show the position and orientation of the robot end effector at different platform heights.

The above results suggest that the proposed IVAC algorithm can adapt to different platform heights quickly by changing its motion trajectory and has a wide adjustment range. The proposed framework can ensure that the cutting force is within a safe range, which is suitable for cutting when the position of the rigid medium under soft tissue is uncertain. It can also be found from the results that the cutting force errors of the same axis are quite different at different execution phases. It suggests that the optimal stiffness differs from phases and axes during execution. Follow-up research can consider the optimization of the stiffness coefficients of IVAC through reinforcement learning or variable admittance control.

### 3.3. Sheep Hindquarters Separation Test

This study carries out a sheep hindquarter separation test to evaluate the effectiveness of the proposed framework in the meat cutting scenario. The separation processes of the left hind leg of the hindquarter are selected as the test scenarios. In the test, the robot needed to cut the leg muscles along the hip bone (ilium and ischium) of the hindquarter to separate the left hind-leg from the hip bone. To complete the separation, the robot uses the proposed IVAC algorithm and the learned skill to perform cutting in Scenarios 1–3 sequentially. We prepare two sheep hindquarters for testing.

The separation effects are shown in Figure 7; the robot successfully separates the hip bone and hind-leg of the sheep carcass. The hind-leg of the divided sheep carcass is relatively complete, and there is only a little meat left on the hip bone. Moreover, no bone cracks appeared during the separation processes, and there is no collision accident between the cutting tool and the bone. This indicates that the cutting by the robot is compliant and accurate.

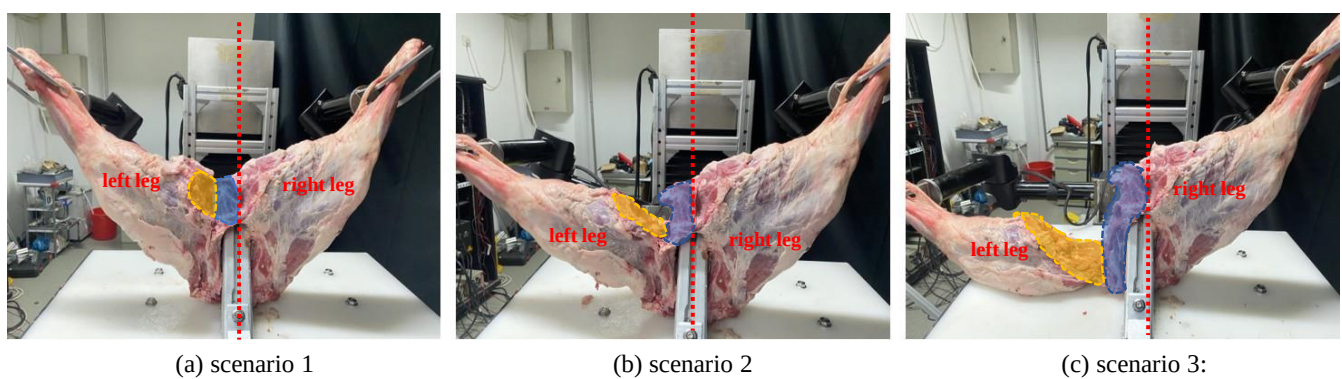


**Figure 7.** Effects of hindquarter separation. (a) Scenario 1: cutting the medial hind-leg muscles attached with the ischium ventral. (b) Scenario 2: cutting the lateral hind-leg muscles attached with the ischium dorsal. (c) Scenario 3: cutting the lateral hind-leg muscles attached with the ilium dorsal. The yellow-colored area represents the cut muscles, and the blue-colored area represents the ischium bone.

Table 3 shows the meat residual rate (RR), maximum residual meat thickness, and maximum absolute value of cutting force. The maximum residual meat thickness is used to characterize the adaptive performance of IVAC. The smaller the maximum residual meat thickness, the better the adaptive performance of IVAC when there are differences in shape and size between the test sample and the teaching sample. The maximum absolute value of the cutting force reflects the cutting resistance under the premise of completing the cutting. The smaller the value, the higher the cutting efficiency. Moreover, RR is defined as the ratio of the weight of the residual hind leg meat on the hip bone of the sheep carcass to the total weight of the separated hind-leg. RR evaluates the loss of high-value leg meat during the cutting process. The smaller the value, the less the loss. RR is calculated with Equation (41)

$$RR = \frac{2.0 \times m_{\text{res}}}{m_{\text{res}} + m_{\text{leg}}} \times 100\%, \quad (41)$$

where  $RR$  denotes the residual rate,  $m_{\text{res}}$  denotes the weight of the residual hind-leg meat on the hip bone, and  $m_{\text{leg}}$  denotes the weight of the separated hind-leg.

According to Table 3, the maximum residual meat thickness of sample 1 is 3.1 mm and sample 2 is 3.8 mm, both of which are less than 5 mm. The segmentation residual rates of samples 1 and 2 are 5.6% and 4.8%. The results indicate that the proposed method can segment sheep carcasses well and has good generalization and effectiveness. In addition, the maximum residual meat thickness and segmentation residual rate obtained in the experiment are small, indicating that the segmentation accuracy is high enough to segment

the hind-legs of sheep carcasses. The last four rows of Table 3 are the absolute maximum value of the cutting force in the directions of  $F_x$ ,  $F_y$ ,  $F_z$ , and  $T_x$ . It can be observed from Table 3 that the largest absolute values of the cutting force in each axis of samples 1 and 2 under the three scenarios are all within the safe range of the robot and the force sensor. This suggests that the proposed method can not only ensure that the cutting depth and residual rate meet the process requirements but also avoid excessive cutting force through adaptation. The proposed framework can realize the human–robot skill transfer in the sheep hindquarters separation test.

**Table 3.** Performance of sheep carcass hindquarter cutting test from 2 samples.

| Crie                   | Sample 1   |           |            | Sample 2   |            |            |
|------------------------|------------|-----------|------------|------------|------------|------------|
|                        | Scenario 1 | Senario 2 | Scenario 3 | Scenario 1 | Scenario 2 | Scenario 3 |
| Weight [kg]            |            | 5.18      |            |            | 6.30       |            |
| RR [%]                 |            | 5.6       |            |            | 4.8        |            |
| Maximal thickness [mm] |            | 3.1       |            |            | 3.8        |            |
| $ F_x _{\max}$ [N]     | 6.461      | 3.056     | 4.072      | 7.655      | 7.441      | 4.787      |
| $ F_y _{\max}$ [N]     | 30.719     | 24.780    | 15.972     | 31.981     | 27.404     | 20.432     |
| $ F_z _{\max}$ [N]     | 17.968     | 21.855    | 4.185      | 30.686     | 22.858     | 5.004      |
| $ T_x _{\max}$ [N]     | 5.708      | 4.971     | 3.803      | 5.726      | 5.2870     | 4.808      |

Figure 8 shows the cutting forces and torque curves of cutting sample 1 to illustrate the similarity between robot cutting and the human demonstration. The first column of Figure 8 shows the results of cutting scenario 1. (a) The robotic cutting force tendency of each axis is consistent with the reference curve measured in the human demonstration, indicating that the robot can imitate the human cutting behavior well. (b) The robot cutting force measured is close to the humans demonstration. (c) The maximum robot cutting force in the  $F_z$  direction (200 steps) is greater than the human cutting force. However, it is still within the safe range of the robot and force sensor. This suggests that the proposed method can perform cutting efficiently and avoid excessive cutting force.

The values in the second column of Figure 8 are curves in scenario 2. Meanwhile, the tendency and amplitude of the robot's cutting force  $F_y$ ,  $F_z$ ,  $T_x$  are very close to that of human cutting. As is shown in the third column of Figure 8, the tendency and amplitude of the robot cutting force are near human, and the vibration of robot cutting is greater in scenario 3. Additionally, Figure 8 shows that the total separation time of the hind-leg is nearly 30 s. This suggests that the proposed method has practical efficiency compared with manual cutting.

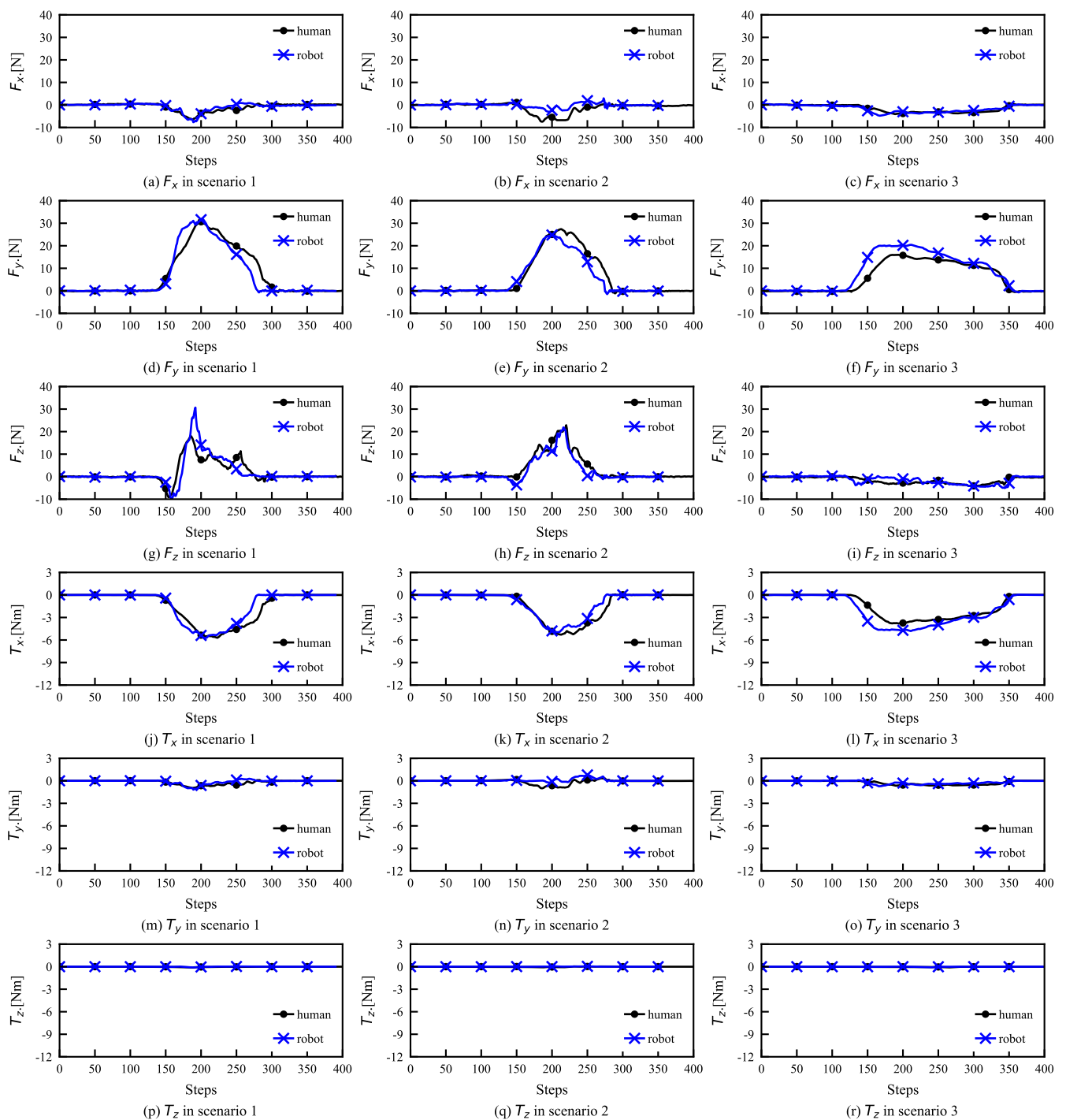


**Figure 8.** Cutting force and torque variation curve of cutting sample 1. (a)  $F_x$  in Scenario 1, (b)  $F_x$  in Scenario 2, (c)  $F_x$  in Scenario 3, (d)  $F_y$  in Scenario 1, (e)  $F_y$  in Scenario 2, (f)  $F_y$  in Scenario 3, (g)  $F_z$  in Scenario 1, (h)  $F_z$  in Scenario 2, (i)  $F_z$  in Scenario 3, (j)  $T_x$  in Scenario 1, (k)  $T_x$  in Scenario 2, (l)  $T_x$  in Scenario 3, (m)  $T_y$  in Scenario 1, (n)  $T_y$  in Scenario 2, (o)  $T_y$  in Scenario 3, (p)  $T_z$  in Scenario 1, (q)  $T_z$  in Scenario 2, (r)  $T_z$  in Scenario 3. The first column shows the results of Scenario 1. The second column shows the results of Scenario 2. The third column shows the results of Scenario 3.

#### 4. Conclusions

The meat industry requires automation due to labor shortages and inefficiencies. Traditional 2D methods are limited when cutting complex 3D anatomy structures, such as sheep hindquarters. Compliant cutting in a 3D space can precisely separate the meat tissue

from the bones but lacks flexibility. A multi-demonstration HRST framework is proposed to address these challenges, reduce cutting force, and minimize meat loss. This method potentially benefits flexible and generalized robot skill learning in interaction tasks, which provides an alternative approach for harvesting high-value cuts and increasing yields in meat industries. Multi-demonstrations of cutting are encoded through the latent space GMM-GMR module to generate the target cutting behaviors for robot, which enables the generalization to meat parts and sizes. The IVAC-DMP control provides the flexibility of modulating robot joint motion during cutting and avoids excessive cutting forces.

The experimental results show that the latent space GMM-GMR can effectively extract cutting skills from multi-demonstrations. The foam-cutting test results show that the IVAC module can adapt to the position changes of the rigid medium under soft tissue reliably and efficiently. The sheep hindquarters separation test result shows that the maximum residual meat thickness of sample 1 and 2 are 3.1 mm and 3.8 mm. The residual rates of samples 1 and 2 are 5.6% and 4.8%. Both meet the requirements of the sheep hindquarter separation.

There are several limitations to the proposed framework. First, the current PCA representation of cutting behavior may not be completely accurate. In the future, it may be beneficial to use models with stronger representation capabilities, e.g., variational autoencoder to improve the accuracy of the latent space representation. Second, the demonstrating scenarios used in this study are limited to cutting sheep hindquarters and foam blocks. To extract more generalizable semantic cutting skills, future studies could apply the framework to broader tissue-cutting tasks by training with larger-scale datasets from various meat pieces over longer epochs. Finally, the IVAC module utilized fixed admittance parameters. To improve interaction flexibility, future studies could apply variable AC or IC to provide optimal stiffness at different stages of the cutting process and in different cutting directions.

**Author Contributions:** Conceptualization, K.L. and B.X.; methodology, K.L. and B.X.; software, K.L.; validation, K.L. and Z.C.; formal analysis, K.L.; investigation, K.L. and Z.C.; resources, Z.C.; data curation, Z.C.; writing—original draft preparation, K.L.; writing—review and editing, K.L., B.X., Z.L. and Z.G.; visualization, K.L. and S.J.; supervision, B.X.; project administration, B.X.; funding acquisition, B.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key Research and Development Plan of China (grant number 2018YFD0700804).

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

|      |                                     |
|------|-------------------------------------|
| HRST | Human–robot skill transfer          |
| AC   | Admittance controller               |
| IC   | Impedance controller                |
| IV   | Inverse velocity                    |
| IKS  | Inverse kinematics solving          |
| IVAC | Inverse velocity admittance control |
| DMP  | Dynamic movement primitive          |
| CMP  | Compliant movement primitive        |
| DLB  | Dementia with Lewy bodies           |
| GMM  | Gaussian mixture model              |
| GMR  | Gaussian mixture regression         |
| PCA  | Principal components analysis       |
| LWR  | Locally weighted linear regression  |

|      |                                       |
|------|---------------------------------------|
| ROS  | Robot operating system                |
| MLA  | Meat and Livestock Australia          |
| AMPC | Australian Meat Processor Corporation |
| EM   | Expectation maximization              |
| DTW  | Dynamic time warping                  |
| CMP  | Compliant movement primitive          |
| SEDS | Stable Estimator of Dynamical Systems |
| DMRI | Danish Meat Research Institute        |
| sEMG | Surface electromyography              |

## References

- Li, J.; Xie, B.; Zhai, Z.; Zhang, P.; Hou, S. Research progress of intelligent equipment and technology for livestock and poultry slaughter and processing. *Food Mach.* **2021**, *37*, 226–232. [\[CrossRef\]](#)
- Xu, W.; He, Y.; Li, J.; Zhou, J.; Xu, E.; Wang, W.; Liu, D. Robotization and Intelligent Digital Systems in the Meat Cutting Industry: From the Perspectives of Robotic Cutting, Perception, and Digital Development. *Trends Food Sci. Technol.* **2023**, *135*, 234–251. [\[CrossRef\]](#)
- Arvidsson, I.; Balogh, I.; Hansson, G.Å.; Ohlsson, K.; Åkesson, I.; Nordander, C. Rationalization in Meat Cutting—Consequences on Physical Workload. *Appl. Ergon.* **2012**, *43*, 1026–1032. [\[CrossRef\]](#)
- Echegaray, N.; Hassoun, A.; Jagtap, S.; Tetteh-Caesar, M.; Kumar, M.; Tomasevic, I.; Goksen, G.; Lorenzo, J.M. Meat 4.0: Principles and Applications of Industry 4.0 Technologies in the Meat Industry. *Appl. Sci.* **2022**, *12*, 6986. [\[CrossRef\]](#)
- Kim, J.; Kwon, Y.K.; Kim, H.W.; Seol, K.H.; Cho, B.K. Robot Technology for Pork and Beef Meat Slaughtering Process: A Review. *Animals* **2023**, *13*, 651. [\[CrossRef\]](#) [\[PubMed\]](#)
- Hinrichsen, L. Manufacturing Technology in the Danish Pig Slaughter Industry. *Meat Sci.* **2010**, *84*, 271–275. [\[CrossRef\]](#) [\[PubMed\]](#)
- Guire, G.; Sabourin, L.; Gogu, G.; Lemoine, E. Robotic Cell for Beef Carcass Primal Cutting and Pork Ham Boning in Meat Industry. *Ind. Robot.* **2010**, *37*, 532–541. [\[CrossRef\]](#)
- Li, Z.; Wang, S.; Zhao, S.; Bai, Y. Cutting Methods of Sheeps Trunk Based on Improved DeepLabv3+ and XGBoost. *Comput. Eng. Appl.* **2021**, *57*, 263–269.
- Khodabandehloo, K. Achieving Robotic Meat Cutting. *Anim. Front.* **2022**, *12*, 7–17. [\[CrossRef\]](#) [\[PubMed\]](#)
- Xie, B.; Jiao, W.; Wen, C.; Hou, S.; Zhang, F.; Liu, K.; Li, J. Feature Detection Method for Hind Leg Segmentation of Sheep Carcass Based on Multi-Scale Dual Attention U-Net. *Comput. Electron. Agric.* **2021**, *191*, 106482. [\[CrossRef\]](#)
- Australia, M.L. Automated Forequarter Cell Installation for Lamb [EB/OL]. Available online: <https://www.mla.com.au/research-and-development/reports/2023/automated-forequarter-cell-installation-for-lamb/> (accessed on 29 November 2023).
- AMPC. First Prototype Automation for Deboning Lamb Shoulder Stage 2 [EB/OL]. Available online: <https://ampc.com.au/research-development/advanced-manufacturing/first-prototype-automation-for-deboning-lamb-shoulder-stage-2> (accessed on 29 November 2023).
- Nabil, E.; Belhassen-Chedli, B.; Grigore, G. Soft Material Modeling for Robotic Task Formulation and Control in the Muscle Separation Process. *Robot. Comput. Integr. Manuf.* **2015**, *32*, 37–53. [\[CrossRef\]](#)
- Maithani, H.; Corrales Ramon, J.A.; Lequievre, L.; Mezouar, Y.; Alric, M. Exoscarne: Assistive Strategies for an Industrial Meat Cutting System Based on Physical Human-Robot Interaction. *Appl. Sci.* **2021**, *11*, 3907. [\[CrossRef\]](#)
- Zeng, C.; Yang, C.G.; Li, Q.; Dai, L. Research Progress in Human-robot Skill Transfer. *Acta Autom. Sin.* **2019**, *45*, 16.
- Burdet, E.; Osu, R.; Franklin, D.W.; Milner, T.E.; Kawato, M. The Central Nervous System Stabilizes Unstable Dynamics by Learning Optimal Impedance. *Nature* **2001**, *414*, 446–449. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zeng, C.; Su, H.; Li, Y.; Guo, J.; Yang, C. An Approach for Robotic Leaning Inspired by Biomimetic Adaptive Control. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1479–1488. [\[CrossRef\]](#)
- Li, Y.; Ganesh, G.; Jarrassé, N.; Haddadin, S.; Albu-Schaeffer, A.; Burdet, E. Force, Impedance, and Trajectory Learning for Contact Tooling and Haptic Identification. *IEEE Trans. Robot.* **2018**, *34*, 1170–1182. [\[CrossRef\]](#)
- Gams, A.; Nemec, B.; Ijspeert, A.J.; Ude, A. Coupling Movement Primitives: Interaction With the Environment and Bimanual Tasks. *IEEE Trans. Robot.* **2014**, *30*, 816–830. [\[CrossRef\]](#)
- Kramberger, A.; Shahriari, E.; Gams, A.; Nemec, B.; Haddadin, S. Passivity Based Iterative Learning of Admittance-Coupled Dynamic Movement Primitives for Interaction with Changing Environments. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
- Xie, B.; Jiao, W.; Liu, K.; Wu, J.; Wen, C.; Chen, Z. Adaptive Segmentation Control Method of Sheep Carcass Hind Legs Based on Contact State Perception. *Trans. Chin. Soc. Agric. Mach.* **2023**, *54*, 306–315. [\[CrossRef\]](#)
- Gams, A.; Ude, A.; Petric, T.; Denisa, M. Learning Compliant Movement Primitives Through Demonstration and Statistical Generalization. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 2581–2594.
- Wu, R.; Billard, A. Learning From Demonstration and Interactive Control of Variable-Impedance to Cut Soft Tissues. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 2740–2751. [\[CrossRef\]](#)
- Garrido-Jurado, S.; Muñoz-Salinas, R.; Madrid-Cuevas, F.; Medina-Carnicer, R. Generation of Fiducial Marker Dictionaries Using Mixed Integer Linear Programming. *Pattern Recognit.* **2015**, *51*, 481–491. [\[CrossRef\]](#)

25. Zhang, L.; Hu, R.; Yi, W. Research on Force Sensing for the End-load of Industrial Robot Based on a 6-Axis Force/Torque Sensor. *Acta Autom. Sin.* **2017**, *43*, 439–447. [[CrossRef](#)]
26. Hersch, M.; Guenter, F.; Calinon, S.; Billard, A. Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations. *IEEE Trans. Robot.* **2008**, *24*, 1463–1467. [[CrossRef](#)]
27. Hoffmann, H.; Pastor, P.; Park, D.H.; Schaal, S. Biologically-Inspired Dynamical Systems for Movement Generation: Automatic Real-Time Goal Adaptation and Obstacle Avoidance. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 2587–2592. [[CrossRef](#)]
28. Abu-Dakka, F.J.; Nemec, B.; Jørgensen, J.A.; Savarimuthu, T.R.; Krüger, N.; Ude, A. Adaptation of Manipulation Skills in Physical Contact with the Environment to Reference Force Profiles. *Auton. Robot.* **2015**, *39*, 199–217. [[CrossRef](#)]
29. Cao, P.; Gan, Y.; Dai, X.; Duan, J. Convex Optimization Solution for Inverse Kinematics of a Physically Constrained Redundant Manipulator. *Robot* **2016**, *38*, 257–264. [[CrossRef](#)]
30. He, W.; Xue, C.; Yu, X.; Li, Z.; Yang, C. Admittance-Based Controller Design for Physical Human—Robot Interaction in the Constrained Task Space. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1937–1949. [[CrossRef](#)]
31. Yamane, K. Admittance Control With Unknown Location of Interaction. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4079–4086. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.