

Article

Learning Underwater Intervention Skills Based on Dynamic Movement Primitives

Xuejiao Yang ^{1,2,3} , Yunxiu Zhang ^{1,2}, Rongrong Li ⁴, Xinhui Zheng ^{1,2,3} and Qifeng Zhang ^{1,2,*}

¹ State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

² Key Laboratory of Marine Robotics, Liaoning Province, Shenyang 110169, China

³ University of Chinese Academy of Sciences, Beijing 100049, China

⁴ College of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110159, China

* Correspondence: zqf@sia.cn

Abstract: Improving the autonomy of underwater interventions by remotely operated vehicles (ROVs) can help mitigate the impact of communication delays on operational efficiency. Currently, underwater interventions for ROVs usually rely on real-time teleoperation or preprogramming by operators, which is not only time-consuming and increases the cognitive burden on operators but also requires extensive specialized programming. Instead, this paper uses the intuitive learning from demonstrations (LfD) approach that uses operator demonstrations as inputs and models the trajectory characteristics of the task through the dynamic movement primitive (DMP) approach for task reproduction as well as the generalization of knowledge to new environments. Unlike existing applications of DMP-based robot trajectory learning methods, we propose the underwater DMP (UDMP) method to address the problem that the complexity and stochasticity of underwater operational environments (e.g., current perturbations and floating operations) diminish the representativeness of the demonstrated trajectories. First, the Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) are used for feature extraction of multiple demonstration trajectories to obtain typical trajectories as inputs to the DMP method. The UDMP method is more suitable for the LfD of underwater interventions than the method that directly learns the nonlinear terms of the DMP. In addition, we improve the commonly used homomorphic-based teleoperation mode to heteromorphic mode, which allows the operator to focus more on the end-operation task. Finally, the effectiveness of the developed method is verified by simulation experiments.

Keywords: learning from demonstration; underwater intervention; ROV; dynamic movement primitives



Citation: Yang, X.; Zhang, Y.; Li, R.; Zheng, X.; Zhang, Q. Learning Underwater Intervention Skills Based on Dynamic Movement Primitives. *Electronics* **2024**, *13*, 3860. <https://doi.org/10.3390/electronics13193860>

Academic Editor: Cecilio Angulo

Received: 20 August 2024

Revised: 26 September 2024

Accepted: 26 September 2024

Published: 29 September 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/>).

1. Introduction

In the late 1970s, remotely operated vehicles (ROVs) became the main tool for underwater interventions, and operators located on the mother ship or shore base remotely operated the ROV to accomplish underwater interventions such as sampling, operating valves, welding, etc. [1,2]. The joystick currently used to control the underwater manipulator and the underwater manipulator has the same configuration of degrees of freedom, similar shapes, and proportional sizes, and the operator remotely operates the underwater manipulator directly in the joint space by referring to the underwater video of the feedback, while the ROV adopts a handle for remote operation [3]. This model has very high cognitive requirements for the operator, especially during delicate operations [4]. Additionally, as ocean exploration moves toward deeper and longer-term operations, the concept of resident ROVs has been introduced to reduce personnel and costs at sea by changing communication links [5,6]. However, the long-distance transmission of signals and bandwidth limitations add latency to the underwater operating system, which may

lead to duplicate or overcorrected commands from the operator, affecting the efficiency of underwater interventions.

Enhancing the autonomy of underwater interventions reduces the operator's operational burden while also helping to mitigate the impact of communication latency on operational efficiency. In this model, the operator only needs to send out high-level commands, and the slave terminal receives the commands and performs the control tasks in its own circuits, separating the master and slave control circuits to solve the problems caused by the time delay. Underwater autonomous intervention has also been a hot research topic in the last decade, and some research projects have begun to demonstrate underwater autonomous intervention capabilities. The SAUVIM project [7] proposes the first underwater vehicle able to intervene autonomously on a floating pedestal, where the operator only needs to confirm that the seafloor recovery target is within the searchable area during the entire operation process. In the TRITON project [8], vision-based servoing completes automated docking with underwater panels with a priori knowledge, as well as valve attachment, rotation, and extraction motions for fixed bases, a process that uses a task prioritization framework to control the underwater vehicle and manipulator to complete the gripping task. The PANDORA project [9] aims to provide the underwater vehicle with the ability to be continuously autonomous, thereby reducing the frequency of requests for assistance from the mother ship. The project uses a learning by demonstration (LbD) approach to learn and reproduce the operation process, where a small number of operator demonstrations based on dynamic movement primitive (DMP) learning are used to generalize the learned task knowledge into a model, and a hybrid force/motion controller is used to perform the rotary valve task. To the best of our knowledge, this would be the first application of demonstration learning techniques for underwater interventions. In the DexROV project of the EU [10,11], the operator interacts with the real-time simulation environment through a wearable manipulator with force feedback to complete the task, and the remote underwater ROV receives simple high-level semantic commands to complete the task autonomously, which learns the representation of the task based on a parameterized hidden semi-Markov model (TP-HSMM).

Due to the dynamic conditions of the underwater environment (current disturbances, reduced visibility, etc.), realizing autonomous underwater interventions is a considerable challenge. In addition, different intervention tasks require a lot of programming work. LbD is one of the most direct and effective skill-learning methods and can be based on existing underwater teleoperation technology, which is believed to help accelerate the process of autonomy for underwater interventions.

Most robot behavioral actions are imitated by tracking at the trajectory level. For example, in an underwater operation task, the variation in the operation task is usually between the position of the underwater manipulator and the position of the operation target, while the essence of the task is the same. Therefore, in this paper, we apply the method of DMP-based demonstration learning to underwater intervention, and we propose an underwater DMP (UDMP) method to address the problem of underwater interference affecting the demonstration trajectories. First, the Gaussian mixture model and Gaussian mixture regression (GMM–GMR) are used for feature extraction of multiple demonstration trajectories and regression to obtain typical trajectories. Second, the DMP method is used to learn that typical trajectory. By learning typical trajectories, the DMP method can be used to reproduce or generalize already-learned trajectories. This approach is compared to the one used in [12], which learns the nonlinear terms of the DMP directly. Simulation experiments show the effectiveness of the proposed method for demonstration learning for underwater intervention.

The rest of this paper is organized as follows. Section 2 summarizes LbD-related work in the robotics community and illustrates the incompatibility of existing approaches for underwater intervention applications. Section 3 describes the ROV system used for the intervention task. Section 4 presents our proposed learning framework and the main content of the UDMP approach. Section 5 shows the validity of the proposed method

through simulation and comparison experiments. Section 6 summarizes our work and provides our outlook for future research.

2. Related Work

Unstructured work environments increase the difficulty of applying pre-programmed methods. In contrast, LfD provides an efficient and intuitive way to transfer skills from humans to robots. Task-relevant prior knowledge is extracted from the demonstration, and no other prior knowledge or data are required, making it a simple and effective way to characterize the operator's actions when completing underwater intervention.

So far, many LfD algorithms have been proposed, such as DMP [13], a stable estimator of dynamical systems (SEDS) [14], the hidden Markov model (HMM) [15], probabilistic movement primitives (ProMP) [16], kernelized movement primitives (KMP) [17], and so on. The DMP method is proposed in [13] for generalizing point-to-point and periodic motions by learning a single demonstration trajectory. The method uses a spring-damping model and a nonlinear term to ensure that the generalized trajectory converges to the target point when imitating the demonstrated skill. The SEDS method, proposed in [14], is based on the dynamic systems (DS) algorithm and uses a nonlinear solver to optimize the parameters of a multi-sample GMM to ensure that the system is globally asymptotically stable under a quadratic Lyapunov function. The KMP method was proposed in [17], which minimizes the Kullback–Leibler divergence between parameterized and sample trajectories and introduces a Kernel trick to obtain a non-parametric skill learning model. Due to its generalization, stability, and robustness properties, DMP has been widely used in robotics to encode and reproduce motor behaviors such as pouring water [18], painting [19], and obstacle avoidance [20]. The DMP method has been particularly applied in the collaborative domain [21] by fusing it with impedance information obtained from electromyography (EMG)-based methods for estimating the stiffness of human limbs. In [22], collaborative skills are extracted from a single human demonstration and learned through a Riemannian DMP, where the learning process is adapted online according to human preferences and ergonomics to accomplish a human–machine collaborative handling task. In this research, the position, orientation, and stiffness of human demonstrations are learned to enable the human-like variable impedance control of robots.

Existing research has focused on manipulators for land-based applications, which usually require only one demonstration for a task to represent the characteristics of the taught trajectory since its base is usually cemented to the world. In contrast, an underwater manipulator is usually fixed to an ROV in a floating operation, and the movement of the manipulator may affect the movement of the ROV, thus changing the position of the manipulator's base coordinates. In addition, due to the current interference in the underwater environment, the operator needs to resist the motion error caused by the current interference between the ROV and the manipulator during the teleoperation demonstration, so the demonstration data usually have a different starting point and end point, which is different from land manipulator demonstration learning. Therefore, it is not feasible to directly apply the DMP method, which requires only one acquisition of a typical trajectory on land, directly to the demonstration learning of an underwater manipulator. Skill learning for underwater manipulators requires learning multiple sets of demonstrations to obtain more trajectory features, creating a "1 plus 1 is greater than 2" effect. Currently, methods based on DMP learning of multiple demonstrations simultaneously focus on estimating the nonlinear forcing term in the DMP model via GMM [12,23] or obtaining the weight of the forcing term by transforming the term into solving a linear problem [24]. Inspired by ProMP and KMP, we use the probabilistic method GMM–GMR to preprocess multiple sets of demonstration trajectories before applying the DMP method to ensure the convergence of the trajectories. The aim is to preserve the common features contained in the demonstration trajectories, which are important for underwater demonstration learning. To illustrate the infeasibility of migrating the approach for land manipulators directly to underwater, in this paper, we compare our approach with that proposed in [12].

3. ROV Teleoperation System

3.1. System Constitution

The underwater teleoperation system consists of four parts (Figure 1): an operation control center (OCC), a satellite, an ROV, and tasks.

OCC includes an operator and a six-degree-of-freedom joystick with force feedback (CLAF_mini) to capture the operator's hand movements. The slave end is an ROV that is in direct contact with the underwater environment and consists of the main frame, buoyancy material, power propulsion system, control and energy system, lighting and camera system, positioning system, and a seven-function underwater manipulator [25]. Table 1 shows the specifications of the CLAF_mini and ROV.

When the operator accomplishes the underwater intervention task, he expresses the will to move by controlling the movement of the end of the CLAF_mini. After the mapping algorithm unit maps the movement of the end of the CLAF_mini to the desired movement of the ROV, which is transmitted to the ROV via the communication link. While the ROV executes control commands, the ROV's status information and the video information of the surrounding environment are fed back to the OCC through the communication link for the operator to use as a reference for the next action.

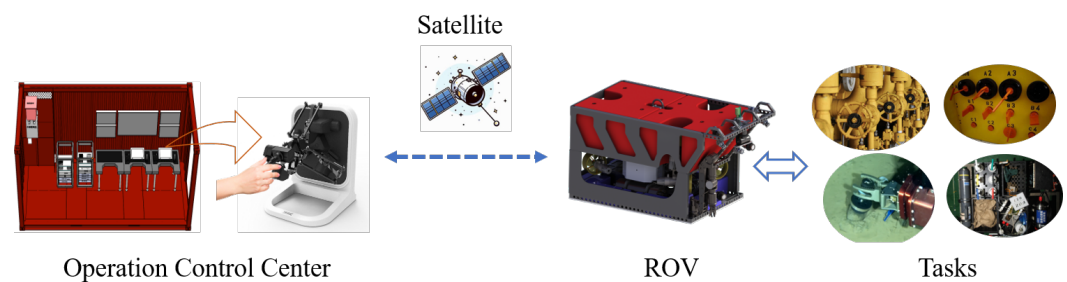


Figure 1. Components of an underwater teleoperation system.

Table 1. The specifications of the CLAF_mini and ROV system.

	Item	Value
ROV	Design depth	11,000 m
	Size (L × H × W)	2.3 m × 1.3 m × 1.5 m
	Mass	1470 kg
	Thrusters	7
	Maximum reach	1.6 m
Manipulator	Function	7
	Lift at full extension	20 kg
	Workspace	0.2 m × 0.2 m × 0.13 m
CLAF_mini	Force	8.5 N

3.2. Mapping Algorithm

Different from the traditional teleoperation of ROVs in joint space, considering the operator's operating habit, we adopt the scheme of motion mapping between the master and slave in Cartesian space. However, since the master and the slave are heterogeneous, and the shape and size of the workspace are not the same, we develop workspace mapping algorithms to extend the smaller workspace of the master to the larger workspace of the ROV and the underwater manipulator, while guaranteeing operation accuracy and efficiency.

For the motion of the ROV, we adopt a velocity-based motion mapping strategy, considering that it usually performs a wide range of underwater exploration tasks. The desired velocity of the ROV is as follows:

$$v_{rov_cmd} = k_{rov} (p_{joystick_now} - p_{joystick_init}), \tag{1}$$

where k_{rov} is the mapping gain coefficient, and $p_{joystick_now}$ and $p_{joystick_init}$ are the current end position of the joystick CLAF_mini and the initial position of the current control cycle, respectively. The desired position of the end of the underwater manipulator is as follows:

$$p_{arm_cmd} = p_{arm_now} + k_{arm} (p_{master_now} - p_{master_init}), \tag{2}$$

where k_{arm} is the mapping gain coefficient, p_{arm_now} is the current position of the end of the underwater manipulator, and p_{master_now} and p_{master_init} are the current position of the end of the manipulator and the initial position of the current control cycle, respectively.

For the orientation of the ROV and the end of the underwater manipulator, we use a 1:1 mapping method to ensure the master–slave end orientation is consistent.

4. Methods

As shown in Figure 2, our learning framework consists of three main parts: multiple demonstrations, skill learning and generalization, and ROV control.

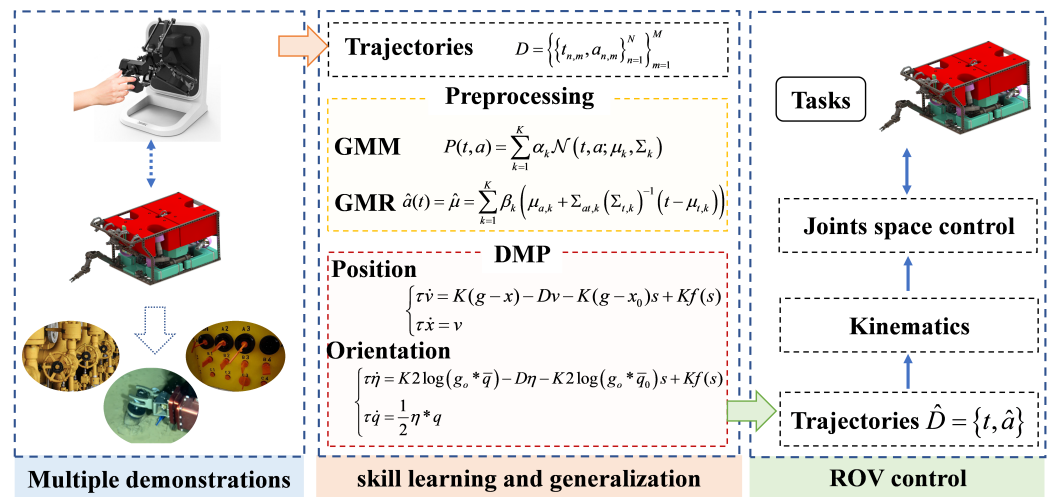


Figure 2. Overview of the learning framework.

Multiple demonstrations: The operator remotely operates the ROV through the joystick to accomplish a task multiple times. The resulting demonstration trajectories are then aligned to the same time frame as the M demonstration trajectories. $D = \left\{ \left\{ t_{n,m}, a_{n,m} \right\}_{n=1}^N \right\}_{m=1}^M$ is the output of this module, $t_{n,m}$ denotes the demonstration time, $a_{n,m}$ denotes the position, orientation, velocity and acceleration of multiple trajectories, and N denotes the length of the sample data.

Skill learning and generalization: To address the problem that underwater environments (e.g., current perturbations and floating operations) attenuate the features of a single demonstration trajectory, we first encode multiple sets of multidimensional demonstration trajectories using GMM–GMR and generate typical trajectories $\hat{a}(t)$ (position and orientation) that contain the operational features of an underwater intervention. Then, we use the extended DMP framework to model, learn, and generalize the typical trajectories of underwater interventions to obtain the desired trajectories $\hat{D} = \{t, \hat{a}\}$.

ROV control: The ROV receives the desired velocity command, and the underwater manipulator receives the desired end-effector position and orientation commands. The kinematics module is responsible for translating the end-effector position and orientation commands into the joint space, where the required control commands for each joint are computed by the underlying PID control.

4.1. GMM–GMR Preprocessing

To mitigate the influence of the underwater environment on the representativeness of the demonstration trajectories for underwater interventions, based on the assumption that each trajectory datum is composed of multiple Gaussian distributions, we use GMM–GMR to extract the features of multiple demonstration trajectories and fit the regression to produce a typical, smooth trajectory $\hat{a}(t)$ that is characteristic of underwater interventions, so as to achieve the effect of one plus one is greater than two.

4.1.1. Gaussian Mixture Model

The joint probability density of a GMM is defined as follows [26]:

$$P(t, a) = \sum_{k=1}^K \alpha_k \mathcal{N}(t, a; \mu_k, \Sigma_k), \quad (3)$$

where $\sum_{k=1}^K \alpha_k = 1$, and \mathcal{N} is the Gaussian probability distribution defined as follows:

$$\mathcal{N}(t, a; \mu_k, \Sigma_k) = \frac{e^{-0.5([t, a]^T - \mu_k)^T \Sigma_k^{-1} ([t, a]^T - \mu_k)}}{2\pi \sqrt{|\Sigma_k|}}, \quad (4)$$

where K is the number of Gaussian distributions, α_k is the weight, and μ_k and Σ_k denote the mean and the covariance matrix of the k th Gaussian component, respectively.

To better represent the dataset without overfitting because of a large number of Gaussian components or underfitting because of a small number of Gaussian components, we use the Bayesian information criterion (BIC) to determine the number of Gaussian components [27]:

$$S_{BIC} = -L + \frac{n(K)}{2} \lg N, \quad (5)$$

where $L = \sum_{j=1}^N \lg(p(t, a))$ is the log-likelihood of the model using the demonstrations as a testing set, $n(K)$ is the number of free parameters required for a mixture of K components, and $n(K) = (K - 1) + K(D + (1/2)D(D + 1))$. N is the number of D -dimensional data-points. The value of K with the smallest value of S_{BIC} is ultimately chosen to achieve a balance between the fit of the data and the number of parameters required.

The maximum expectation algorithm (EM) is used to estimate the GMM parameters, where the k-means algorithm is used to initialize the parameters α_k , μ_k , and Σ_k to mitigate the sensitivity of the EM algorithm to the initial values [28]. The EM algorithm aims to find the parameters that maximize the log-likelihood function:

$$\hat{\pi}_k = \arg \max_{\pi_k} \log(p(t, a | \pi_k)). \quad (6)$$

4.1.2. Gaussian Mixture Regression

We use time t as the query point and estimate the corresponding trajectory values \hat{a} via GMR regression [27]. The conditional probability density of a given t is as follows:

$$P(a | t) \sim \sum_{k=1}^K \beta_k \mathcal{N}(\hat{\mu}_k, \hat{\Sigma}_k). \quad (7)$$

As a result, a smooth motion trajectory extracted from a plurality of demonstration trajectories containing operator demonstration features is represented as follows:

$$\hat{a}(t) = \hat{\mu} = \sum_{k=1}^K \beta_k \left(\mu_{a,k} + \Sigma_{at,k} (\Sigma_{t,k})^{-1} (t - \mu_{t,k}) \right). \quad (8)$$

4.2. Cartesian Space Dynamic Movement Primitive

4.2.1. DMP for Position

We use a discrete DMP as the basic motion model, which consists of two parts: a stiffness damping system and a nonlinear forcing term [13], which can be expressed as follows:

$$\begin{cases} \tau\dot{v} = K_p(g - x) - D_p v + (g - x_0)f(s) \\ \tau\dot{x} = v \end{cases}, \tag{9}$$

where $x, v \in \mathbb{R}$ are the position and velocity of the system at a certain moment, $x_0 \in \mathbb{R}$ is the initial position of the system, and $g \in \mathbb{R}$ is the target position of the system. $K_p, D_p \in \mathbb{R}^+$ are the stiffness and damping term coefficients, respectively, and the system is critically damped when $D_p = 2\sqrt{K_p}$. $\tau \in \mathbb{R}^+$ is the time-scaling coefficient, $f(s)$ is the forcing term, and $s \in (0, 1]$ is a reparameterized representation of time $t \in [0, T]$, controlled by a regular system as follows:

$$\tau\dot{s} = -\alpha_p s, \tag{10}$$

where $\alpha_p \in \mathbb{R}^+$ is the exponential decay coefficient of the regular system with an initial value $s(0) = 1$. The forcing term $f(s)$ is written as follows:

$$f(s) = \frac{\sum_{i=0}^N \omega_i \psi_i(s)}{\sum_{i=0}^N \psi_i(s)} s, \tag{11}$$

where $\psi_i(s) = \exp(-h_i(s - c_i)^2)$ is the Gaussian basis function (GBF) with center c_i and width h_i . N is the number of Gaussian functions, and when the regular system converges to the target, the corresponding Gaussian functions are activated, and the forcing term takes effect.

The above DMP formulation suffers from the problem of cross-zeroing because of the coupling of the relative position between the target position and the start position with the forcing term, e.g., when the sign of the target position is changed, the learned trajectory is also mirrored. Therefore, to overcome the above drawbacks, we use the proposed extended DMP where the forcing term no longer relies on the relative position between the start and end points [29,30]:

$$\begin{cases} \tau\dot{v} = K_p(g - x) - D_p v - K_p(g - x_0)s + K_p f(s) \\ \tau\dot{x} = v \end{cases}. \tag{12}$$

The learning process focuses on calculating the weight $w_i \in \mathbb{R}$ that is closest to the desired forcing term, and we rewrite Equation as follows (12):

$$f_d(s(t)) = 1/K_p(\tau\dot{v} - K_p(g - x) + D_p v + K_p(g - x_0)s). \tag{13}$$

Thus, the minimized loss function expression is as follows:

$$J_i = \sum_{t=1}^T \psi_i(t)(f_d(t) - w_i \zeta(t))^2, \tag{14}$$

where $\zeta(t) = s(t)$, and it can be obtained by solving the local weighted regression (LWR):

$$w_i = \frac{s^T \psi_i f_d}{s^T \psi_i s}. \tag{15}$$

4.2.2. DMP for Orientation

Unlike the position, which can be decoupled into three separate one-dimensional motions, the set of orientations $SO(3)$ is a three-dimensional manifold that does not allow for the decoupling scheme described above. In addition, in contrast to rotation matrices, we use the unit quaternion representation of the orientations because it provides a singularity-

free, nonminimal representation of the orientations, where a logarithmic mapping function is used to compute the distance between two quaternions [31,32], and the conjugate of q is denoted by $\bar{q} = -u + v$:

$$d(q_1, q_2) = \begin{cases} [0, 0, 0]^T, & q_1 * \bar{q}_2 = -1 + [0, 0, 0]^T \\ 2\|\log(q_1 * \bar{q}_2)\|, & \text{otherwise} \end{cases} \quad (16)$$

The quaternion logarithm $\log, S^3 \mapsto \mathbb{R}^3$, is defined as follows:

$$\log(q) = \log(v + u) = \begin{cases} \arccos(v) \frac{u}{\|u\|}, & u \neq 0 \\ [0, 0, 0]^T, & \text{otherwise} \end{cases} \quad (17)$$

Therefore, the DMP model of orientation can be expressed as follows:

$$\begin{cases} \tau \dot{\eta} = K_o 2 \log(g_o * \bar{q}) - D_o \eta - K_o 2 \log(g_o * \bar{q}_0) s + K_o f(s) \\ \tau \dot{q} = \frac{1}{2} \eta * q \end{cases} \quad (18)$$

The relation between the quaternion derivative and angular velocity is given by $\dot{q} = \frac{1}{2} \omega * q$; we obtain $\eta = \tau \omega$, and $\omega = 2 \log(q_1 * \bar{q}_2) / dt$. Integrating the quaternionic derivative yields the following:

$$q(t + \Delta t) = \exp\left(\frac{\Delta t}{2} \frac{\eta(t)}{\tau}\right) * q(t), \quad (19)$$

where the exponential mapping of quaternions $\mathbb{R}^3 \mapsto S^3$ is defined as follows:

$$\exp(r) = \begin{cases} \cos(\|r\|) + \sin(\|r\|) \frac{r}{\|r\|}, & r \neq 0 \\ [0, 0, 0]^T + 1, & \text{otherwise} \end{cases} \quad (20)$$

5. Simulation

The simulation environment is the ROV simulation environment built in the previous study [25]. In this simulation environment, the ROV has basic video feedback and dynamic positioning capabilities, and the underwater manipulator has basic functions such as teleoperation and joint space PID control. The experiment is run under the Ubuntu 18 operating system, and Figure 3 shows the composition of the experimental system. The experimental conditions were set as follows: the ROV was dynamically positioned at a depth of 95 m, and the fluid density was set to 1028 kg/m³.

To evaluate the performance of our method for learning from demonstrations for underwater interventions, we compare it to the method in [12], which is an effective method for solving the problem of learning from multiple demonstrations, except for its application to robots on land. The approach in this paper we refer to as UDMP, and the approach in [12], which we refer to as DMP.



Figure 3. The composition of the experimental system.

5.1. Collection of Multiple Demonstration Trajectories

In this experiment, we assume that the ROV has sailed to the vicinity of the operational target, and the operator remotely operates the manipulator after turning on dynamic positioning. We collect six trajectories when the operator completes the valve-gripping task, the initial state of the ROV is similar in each trajectory, and the operator completes the underwater intervention task based on the feedback video. Figure 4 shows the position of the demonstration trajectories. Figure 5 shows the orientation of the demonstration trajectories.

The figure shows that the collected trajectories are not smooth, which is due to the fact that underwater intervention tasks are different from land operation tasks. First, the ROV operator can only use the video feedback from a limited number of cameras on the ROV as a reference during teleoperation, while most of the applications on land have global information feedback. Second, due to the effects of current disturbances and dynamic positioning errors on the ROV's position, the ROV operator needs to continuously adjust the position and orientation during teleoperation to mitigate the effects of the floating base, whereas most applications on land are based on a fixed base.

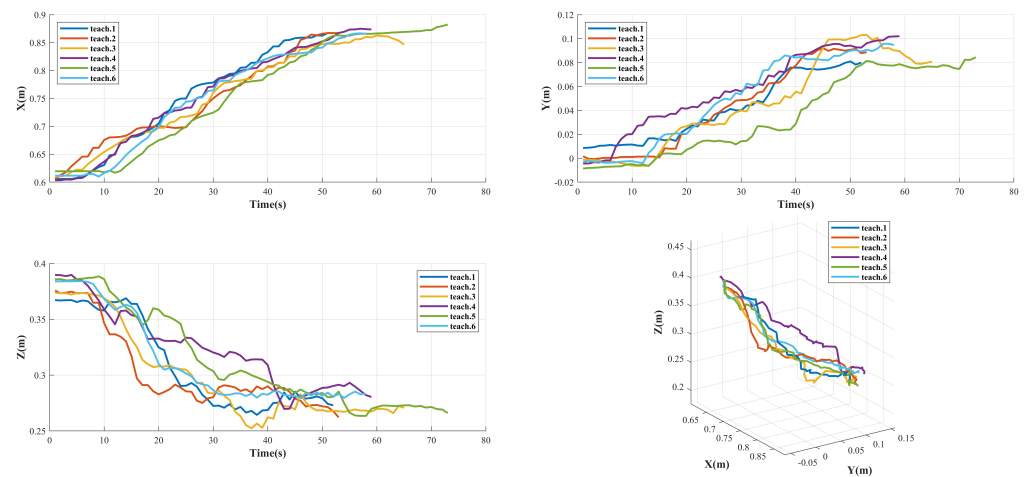


Figure 4. The position of the demonstration trajectories.

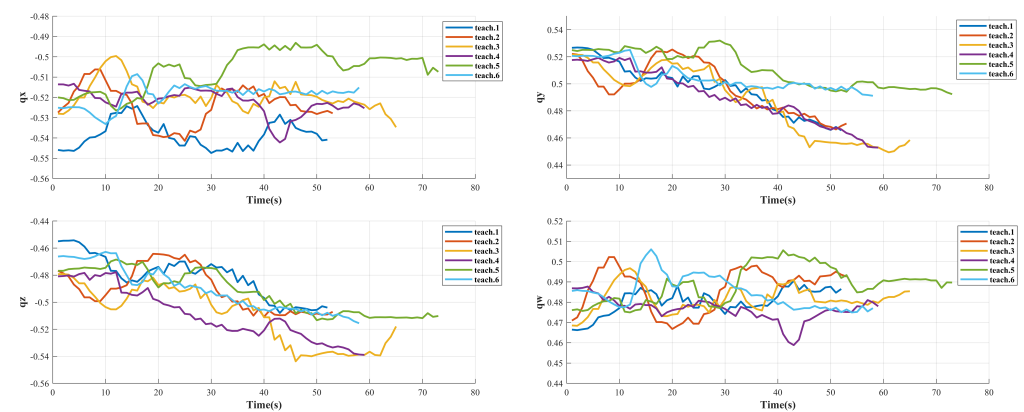


Figure 5. The orientation of the demonstration trajectories.

5.2. Learning from Multiple Demonstrations

Since the durations of multiple presentations may be different, we regularized the time of the trajectory set at the same time, after which we used GMM–GMR for operational task feature extraction as well as regression fitting, as shown in Figures 6 and 7. The Gaussian component of the position trajectories obtained using the BCI rule is 14, and the Gaussian component of the orientation trajectories is 11.

The results indicate that the variance (the width of the blue band) becomes progressively smaller as we approach the valve, indicating that the constraints become progres-

sively stronger due to the same target position and orientation for each demonstration. The trajectory variance is larger in the initial stage due to the disturbance caused by the current to the ROV, which makes the initial position and orientation different. The dark blue curve in the figure indicates the trajectory with typical characteristics obtained after GMM–GMR preprocessing, which contains the trajectory characteristics of underwater intervention.

The method in [12] is used to learn and regress the nonlinear term $f(s)$ of the DMP model directly using GMM–GMR. Figures 8 and 9 show the relationship between s and $f(s)$ that we obtained using this method. The Gaussian component of the position trajectories obtained using the BCI rule is 4, and the Gaussian component of the orientation trajectories is 8. It can be observed that the width of the blue band representing the variance of the forcing term $f(s)$ first becomes narrower and then wider and then narrower. This is consistent with the actual operation process, in which the position and orientation of the end of the underwater manipulator must satisfy the operational requirements of the initial and final moments, i.e., the constraints are strong, while the constraints of the intermediate processes are weak.

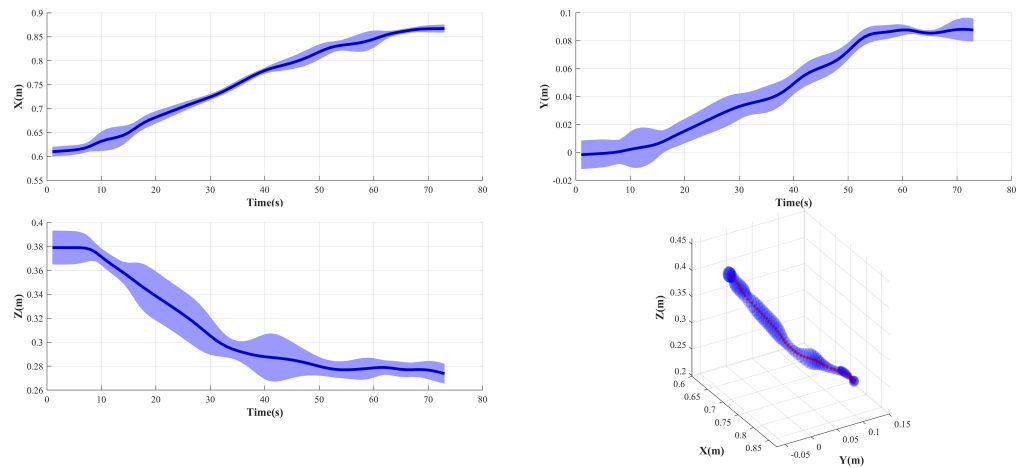


Figure 6. GMM–GMR preprocessed demonstration trajectories used to obtain the $t-a(t)$ of the position.

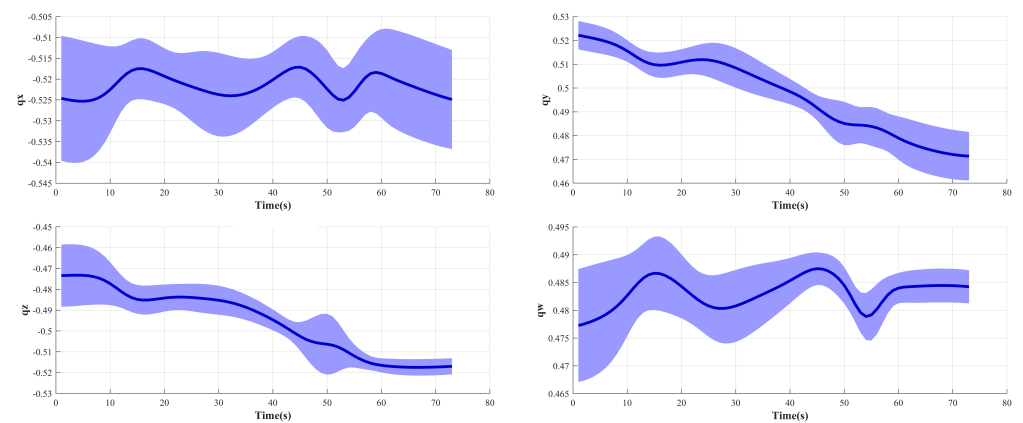


Figure 7. GMM–GMR preprocessed demonstration trajectories used to obtain the $t-a(t)$ of the orientation.

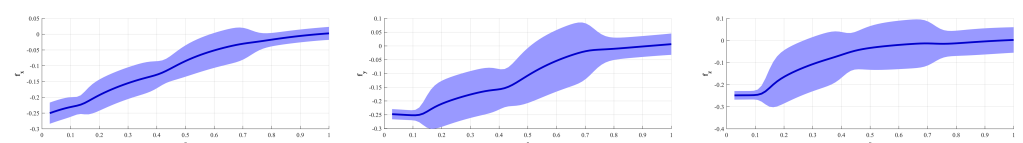


Figure 8. Nonlinear term $s-f(s)$ in DMP modeling [12] of demonstration trajectories (position).

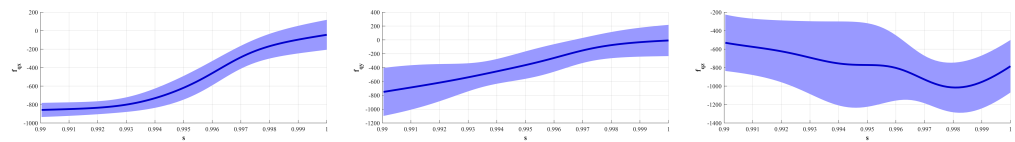


Figure 9. Nonlinear term $s-f(s)$ in DMP modeling [12] of demonstration trajectories (orientation).

5.3. Replication and Generalization of Skill

The purpose of this subsection is to validate the effectiveness of the UDMP method proposed in this paper for underwater intervention demonstration learning, as well as to compare it with the DMP method in [12].

The UDMP requires the tuning of three parameters to accommodate different trajectory behaviors, including the number of Gaussian functions of the nonlinear terms N , the spring factor K , and the attenuation coefficient α . The DMP requires the tuning of two parameters to accommodate different trajectory behaviors, including the spring factor K and the attenuation coefficient α . A higher K causes the system to respond to the target trajectory more quickly but may cause oscillations. A higher α makes the system converge to the target quickly. To ensure the smoothness and robustness of the learned trajectories, the parameter settings for this experiment are shown in Table 2.

We used the DMP and UDMP methods for demonstration reproduction, respectively, and the reproduction results for position and orientation are shown in Figures 10 and 11. The reproduction results of the two methods are shown in Table 3. From the results, it can be observed that the errors of both methods tend to be zero, but compared to the DMP method, UDMP learns better for the features of underwater interventions, especially in the orientation dimension, and UDMP can learn better from the demonstrative data which has little change in orientation.

Table 2. The parameter settings for DMP and UDMP.

	Position			Orientation		
	α_p	K_p	N_p	α_p	K_p	N_p
DMP	0.05	0.25		1	900	
UDMP	0.05	0.25	50	1	900	50

Table 3. Errors in demonstration replication using DMP and UDMP methods.

		Position			Orientation		
		x (m)	y (m)	z (m)	Roll (rad)	Pitch (rad)	Yaw (rad)
DMP	rmse	0.0037	0.0024	0.0026	0.0073	0.0084	0.0081
	max. error	0.0099	0.0053	0.0066	0.0162	0.0178	0.0168
UDMP	rmse	0.0016	0.0007	0.0007	0.003	0.0057	0.0072
	max. error	0.0043	0.0021	0.0023	0.0072	0.0139	0.0134

To further validate the learning performance of the UDMP method, we changed the position of the valve so that it could be approached without changing the orientation. This is due to the fact that the valve position, i.e., the target position, is usually changed in this type of underwater intervention task, and the orientation of the underwater manipulator end-effector when approaching the valve is usually the orientation facing the valve, i.e., it is similar to the initial orientation of the manipulator end-effector in this experiment. Still, using the parameters in Table 2, the results after the generalization of the new position using the DMP method and the UDMP method are shown in Figure 12. The trajectories obtained from the generalization of both methods converge to the desired new target position, and the generalization result of the proposed UDMP method is similar to the shape of the demonstration trajectory, i.e., the constraints in the motion process guarantee

the phase, while the DMP method is different from the shape of the demonstration trajectory. Although all of them eventually converge to the target position, the learning of the shape of the trajectory is the focus of this paper's research, so the above results show that the UDMP method proposed in this paper is effective for the application of underwater intervention demonstration learning.

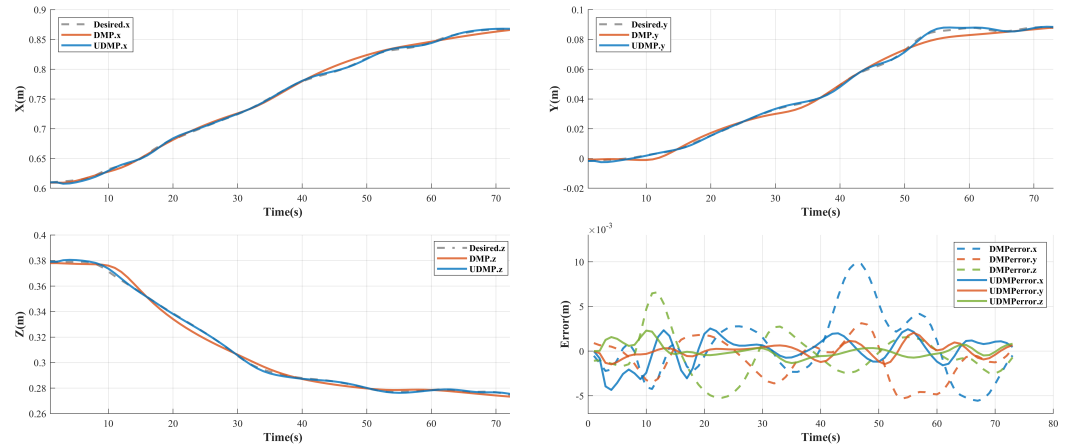


Figure 10. Position trajectories and errors reproduced by DMP and UDMP methods.

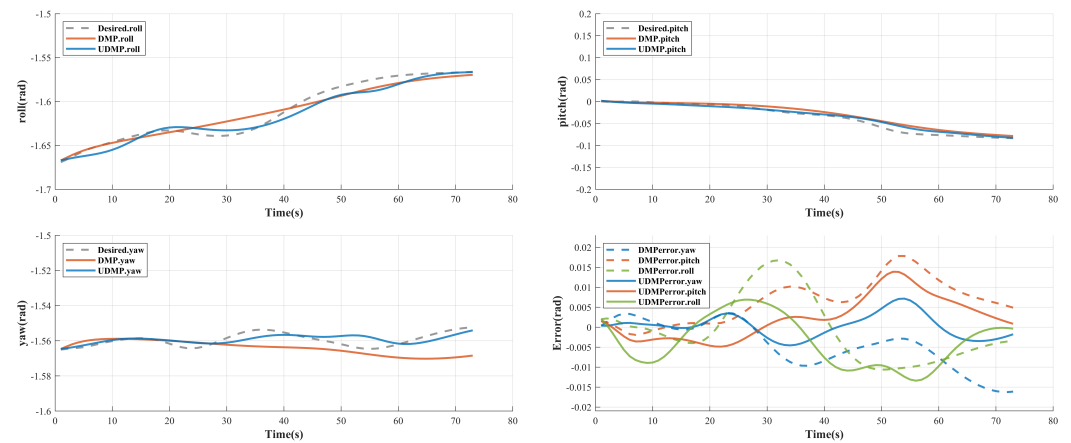


Figure 11. Orientation trajectories and errors reproduced by DMP and UDMP methods (expressed as RPY).

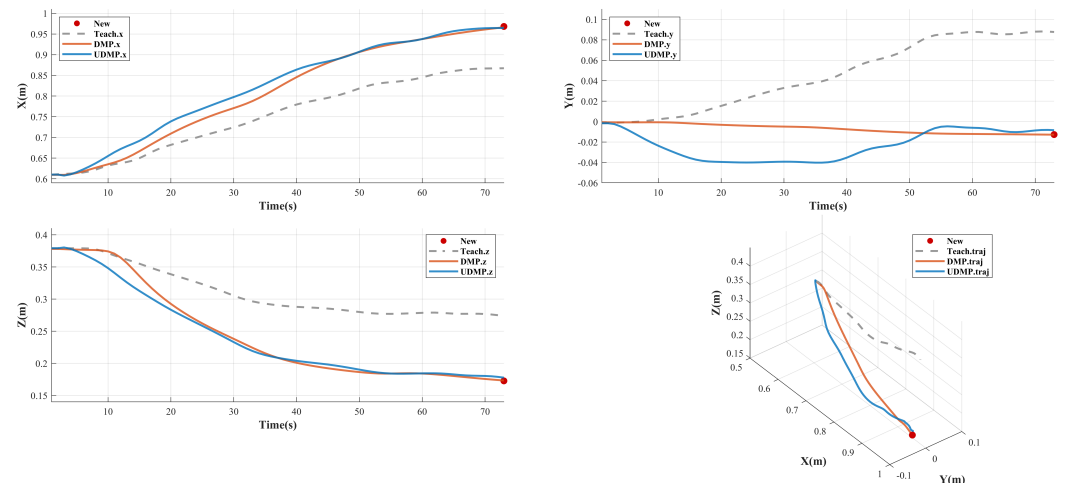


Figure 12. DMP and UDMP methods for generalizing new target position.

6. Conclusions

To mitigate the effects of communication delays on underwater intervention tasks as well as to reduce the cognitive burden on the operator, this paper adopts an intuitive LfD approach to learn operational skills from a small number of demonstrations, thereby enhancing the autonomy of underwater intervention. To address the problem that the complexity and randomness of the underwater operation environment (e.g., current disturbance and floating operation) diminish the representativeness of the demonstration trajectories, we propose the UDMP method, in which multiple demonstration trajectories are feature-extracted using GMM-GMR, a typical trajectory is obtained, and then, the trajectory is modeled using the DMP method. Experiments show that the proposed UDMP method can extract more motion features than the existing methods that learn the nonlinear terms of DMP. This is due to the fact that demonstration trajectories of underwater intervention are noisy, and the DMP method that learns the nonlinear term indirectly will lose some trajectory features, while the proposed UDMP method directly extracts the features of the taught trajectory first and then proceeds to the learning of the DMP model to retain more motion features, which is exactly what is required for the learning of demonstration trajectories of underwater intervention.

The underwater intervention in this paper does not consider operational tasks that require contact force. In future work, we will consider learning the contact force during underwater intervention as a way to adapt to more operational tasks.

Author Contributions: Conceptualization and methodology, X.Y.; software, X.Y., X.Z. and R.L.; validation, X.Y.; data curation, X.Y.; writing—original draft preparation, X.Y.; writing—review and editing, Y.Z. and Q.Z.; supervision, Q.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA22040102), the fundamental research project of SIA, the Natural Science Foundation of Liaoning Province of China (2022-MS-039), the China Postdoctoral Science Foundation (2022M713297), and the Youth Innovation Promotion Association, Chinese Academy of Sciences (2023208).

Data Availability Statement: All data underlying the results are available as part of the article and no additional source data are required.

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

References

1. Sivčev, S.; Coleman, J.; Omerdić, E.; Dooly, G.; Toal, D. Underwater manipulators: A review. *Ocean Eng.* **2018**, *163*, 431–450. [[CrossRef](#)]
2. Schjølberg, I.; Gjersvik, T.B.; Transeth, A.A.; Utne, I.B. Next generation subsea inspection, maintenance and repair operations. *IFAC-PapersOnLine* **2016**, *49*, 434–439. [[CrossRef](#)]
3. Shim, H.; Jun, B.H.; Lee, P.M.; Baek, H.; Lee, J. Workspace control system of underwater tele-operated manipulators on an ROV. *Ocean Eng.* **2010**, *37*, 1036–1047. [[CrossRef](#)]
4. Sivčev, S.; Rossi, M.; Coleman, J.; Dooly, G.; Omerdić, E.; Toal, D. Fully automatic visual servoing control for work-class marine intervention ROVs. *Control Eng. Pract.* **2018**, *74*, 153–167. [[CrossRef](#)]
5. Gilmour, B.; Niccum, G.; O'Donnell, T. Field resident AUV systems—Chevron's long-term goal for AUV development. In Proceedings of the 2012 IEEE/OES Autonomous Underwater Vehicles (AUV), Southampton, UK, 24–27 September 2012; pp. 1–5.
6. Teigland, H.; Hassani, V.; Møller, M.T. Operator focused automation of ROV operations. In Proceedings of the 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), St. Johns, NL, Canada, 30 September–2 October 2020; pp. 1–7.
7. Marani, G.; Choi, S.K.; Yuh, J. Underwater autonomous manipulation for intervention missions AUVs. *Ocean Eng.* **2009**, *36*, 15–23. [[CrossRef](#)]
8. Palomeras, N.; Nagappa, S.; Ribas, D.; Gracias, N.; Carreras, M. Vision-based localization and mapping system for AUV intervention. In Proceedings of the 2013 MTS/IEEE OCEANS-Bergen, Bergen, Norway, 10–14 June 2013; pp. 1–7.
9. Carrera, A.; Palomeras, N.; Hurtós, N.; Kormushev, P.; Carreras, M. Cognitive system for autonomous underwater intervention. *Pattern Recognit. Lett.* **2015**, *67*, 91–99. [[CrossRef](#)]
10. Havoutis, I.; Calinon, S. Learning from demonstration for semi-autonomous teleoperation. *Auton. Robot.* **2019**, *43*, 713–726. [[CrossRef](#)]

11. Gancet, J.; Weiss, P.; Antonelli, G.; Pfingsthorn, M.F.; Calinon, S.; Turetta, A.; Walen, C.; Urbina, D.; Govindaraj, S.; Letier, P.; et al. Dexterous undersea interventions with far distance onshore supervision: The DexROV project. *IFAC-PapersOnLine* **2016**, *49*, 414–419. [[CrossRef](#)]
12. Yang, C.; Chen, C.; He, W.; Cui, R.; Li, Z. Robot learning system based on adaptive neural control and dynamic movement primitives. *IEEE Trans. Neural Networks Learn. Syst.* **2018**, *30*, 777–787. [[CrossRef](#)] [[PubMed](#)]
13. Ijspeert, A.J.; Nakanishi, J.; Schaal, S. Movement imitation with nonlinear dynamical systems in humanoid robots. In Proceedings of the Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, USA, 11–15 May 2002; Volume 2, pp. 1398–1403.
14. Khansari-Zadeh, S.M.; Billard, A. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Trans. Robot.* **2011**, *27*, 943–957. [[CrossRef](#)]
15. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [[CrossRef](#)]
16. Paraschos, A.; Daniel, C.; Peters, J.R.; Neumann, G. Probabilistic movement primitives. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2616–2624.
17. Huang, Y.; Rozo, L.; Silvério, J.; Caldwell, D.G. Kernelized movement primitives. *Int. J. Robot. Res.* **2019**, *38*, 833–852. [[CrossRef](#)]
18. Liao, Z.; Jiang, G.; Zhao, F.; Wu, Y.; Yue, Y.; Mei, X. Dynamic skill learning from human demonstration based on the human arm stiffness estimation model and Riemannian DMP. *IEEE/ASME Trans. Mechatronics* **2022**, *28*, 1149–1160. [[CrossRef](#)]
19. Lu, Z.; Wang, N.; Li, Q.; Yang, C. A trajectory and force dual-incremental robot skill learning and generalization framework using improved dynamical movement primitives and adaptive neural network control. *Neurocomputing* **2023**, *521*, 146–159. [[CrossRef](#)]
20. Sidiropoulos, A.; Papageorgiou, D.; Doulergi, Z. A novel framework for generalizing dynamic movement primitives under kinematic constraints. *Auton. Robot.* **2023**, *47*, 37–50. [[CrossRef](#)]
21. Yu, X.; Liu, P.; He, W.; Liu, Y.; Chen, Q.; Ding, L. Human-robot variable impedance skills transfer learning based on dynamic movement primitives. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6463–6470. [[CrossRef](#)]
22. Liao, Z.; Lorenzini, M.; Leonori, M.; Zhao, F.; Jiang, G.; Ajoudani, A. An Ergo-Interactive Framework for Human-Robot Collaboration Via Learning From Demonstration. *IEEE Robot. Autom. Lett.* **2023**, *9*, 359–366. [[CrossRef](#)]
23. Chen, C.; Yang, C.; Zeng, C.; Wang, N.; Li, Z. Robot learning from multiple demonstrations with dynamic movement primitive. In Proceedings of the 2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM), Hefei and Tai'an, China, 27–31 August 2017; pp. 523–528.
24. Ginesi, M.; Sansonetto, N.; Fiorini, P. Overcoming some drawbacks of dynamic movement primitives. *Robot. Auton. Syst.* **2021**, *144*, 103844. [[CrossRef](#)]
25. Yang, X.; Zhang, Q.; Wang, C.; Liu, X.; Zhang, Y.; Li, D. Development and Construction of a Simulation Platform for a New R-ROV. In Proceedings of the 2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Baishan, China, 27–31 July 2022; pp. 1293–1298.
26. Sung, H.G. *Gaussian Mixture Regression and Classification*; Rice University: Houston, TX, USA, 2004.
27. Burnham, K.P.; Anderson, D.R. Multimodel inference: Understanding AIC and BIC in model selection. *Sociol. Methods Res.* **2004**, *33*, 261–304. [[CrossRef](#)]
28. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22. [[CrossRef](#)]
29. 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.
30. Saveriano, M.; Abu-Dakka, F.J.; Kramberger, A.; Peternel, L. Dynamic movement primitives in robotics: A tutorial survey. *Int. J. Robot. Res.* **2023**, *42*, 1133–1184. [[CrossRef](#)]
31. Ude, A.; Nemeč, B.; Petrić, T.; Morimoto, J. Orientation in cartesian space dynamic movement primitives. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 2997–3004.
32. Abu-Dakka, F.J.; Nemeč, 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](#)]

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.