sensors

MDPI

*Article*

# Learning Compliant Box-in-Box Insertion through Haptic-Based Robotic Teleoperation

**Sreekanth Kana, Juhi Gurnani, Vishal Ramanathan, Mohammad Zaidi Ariffin, Sri Harsha Turlapati and Domenico Campolo ***

School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore; ksreekanth384@gmail.com (S.K.); juhigurnani@gmail.com (J.G.); vishal.pr@ntu.edu.sg (V.R.); mohd.zaidi@ntu.edu.sg (M.Z.A.); turl0001@e.ntu.edu.sg (S.H.T.)
* Correspondence: d.campolo@ntu.edu.sg

**Abstract:** In modern logistics, the box-in-box insertion task is representative of a wide range of packaging applications, and automating compliant object insertion is difficult due to challenges in modelling the object deformation during insertion. Using Learning from Demonstration (LfD) paradigms, which are frequently used in robotics to facilitate skill transfer from humans to robots, can be one solution for complex tasks that are difficult to mathematically model. In order to automate the box-in-box insertion task for packaging applications, this study makes use of LfD techniques. The proposed framework has three phases. Firstly, a master–slave teleoperated robot system is used in the initial phase to haptically demonstrate the insertion task. Then, the learning phase involves identifying trends in the demonstrated trajectories using probabilistic methods, in this case, Gaussian Mixture Regression. In the third phase, the insertion task is generalised, and the robot adjusts to any object position using barycentric interpolation. This method is novel because it tackles tight insertion by taking advantage of the boxes' natural compliance, making it possible to complete the task even with a position-controlled robot. To determine whether the strategy is generalisable and repeatable, experimental validation was carried out.

**Keywords:** box-in-box insertion; compliant insertion; Learning from Demonstration; teleoperation; haptic feedback; human–robot collaboration; Gaussian mixture regression; barycentric interpolation; robotic automation

check for updates

**Citation:** Kana, S.; Gurnani, J.; Ramanathan, V.; Ariffin, M.Z.; Turlapati, S.H.; Campolo, D. Learning Compliant Box-in-Box Insertion through Haptic-Based Robotic Teleoperation. *Sensors* **2023**, *23*, 8721. https://doi.org/10.3390/s23218721

Academic Editor: Bijan Shirinzadeh

Received: 11 August 2023
Revised: 30 September 2023
Accepted: 11 October 2023
Published: 25 October 2023

**Copyright:** © 2023 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

Modern logistics require box-in-box insertion for a broad range of packaging tasks involving compliant, cardboard boxes. Industries such as electronics and cosmetics heavily rely on such box packaging. Typically, smaller boxes containing a product are stacked and placed into larger boxes for transportation. However, these tasks are often carried out manually on the factory floor, leading to drawbacks such as fatigue, limited operational time, and time consumption due to the repetitive and monotonous nature of the work. To overcome these challenges, there is a need for robot automation solutions to perform these tasks. Given that most industrial products are packaged in cuboid-shaped boxes, the problem at hand can be described as 'developing an autonomous system that can perform compliant box-in-box insertion, wherein a cuboid box made of a flexible material is inserted into another larger box made of a similar deformable material' (see Figure 1).

Nonetheless, autonomous insertion is a heavily researched field in robotics, especially in the classical peg-in-hole assembly problem [1]. The existing approaches fall into contact model-based and contact model-free approaches [2]. While the former covers techniques based on the contact state modelling between the parts, the latter focuses on learning paradigms such as Learning from Demonstration and Reinforcement Learning. Contact state modelling is known to be sensitive to uncertainties such as the elasticity of the system. A model-free strategy like Learning from Demonstration looks more promising given that

the task in question requires compliant objects because humans' superior sensory systems and decision-making skills enable them to accomplish insertion tasks even under uncertain environments.

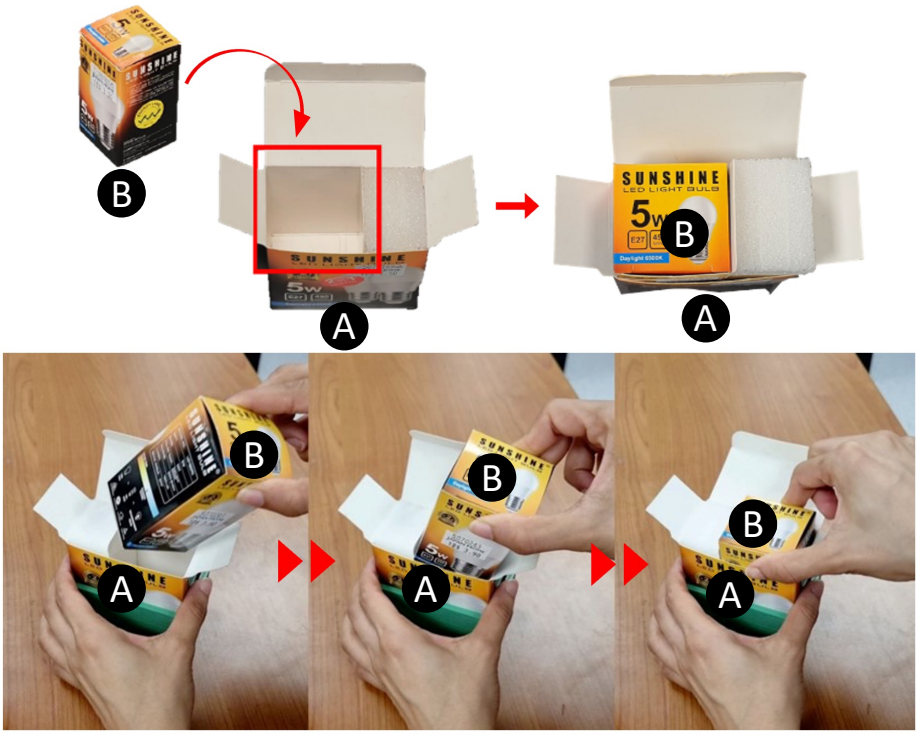


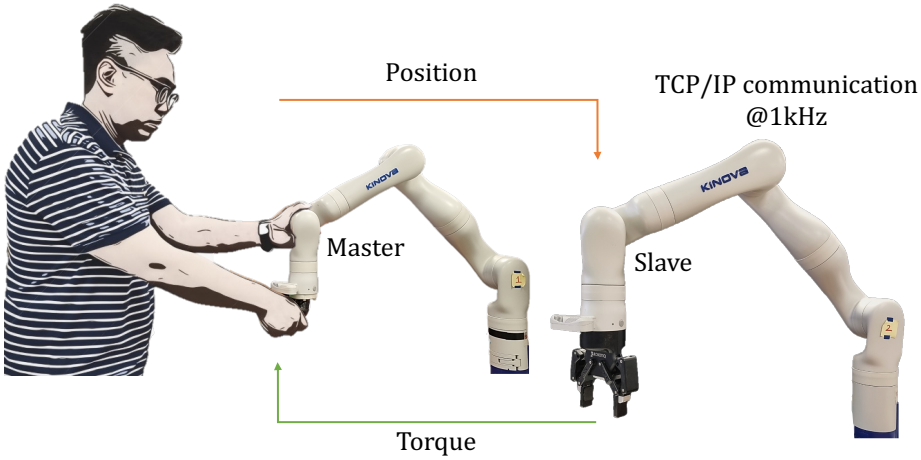

**Figure 1.** Example of a compliant box-in-box insertion where a folding carton (box B) containing an electric light bulb is to be inserted into a receptacle folding carton (box A). Manual insertion performed by a human (**bottom**).

While various research has attempted automated assembly in the past, the focus has frequently been on peg-in-hole insertion dealing with rigid objects because it is the basis of many assembly operations. To avoid the damage or wear of such objects, however, some compliance has been introduced through the use of force-guided robotic systems [3] combined with various other techniques, such as the use of machine learning [4] and vision [5], for example. Another simple approach used to safely manipulate objects with ease is to operate the manipulator in velocity control mode [6], which can then be translated to joint torques using the computed torque control method, as performed in [7], to introduce the desired compliance during manipulation. While several works take advantage of active interaction controls such as impedance/admittance control [8], compliance control [9], and hybrid force/position control [10], some other works also focus on utilising grasp planning [11] and soft grippers [12,13] to mimic the compliance of human fingers to reduce the effects of localisation uncertainties [7].

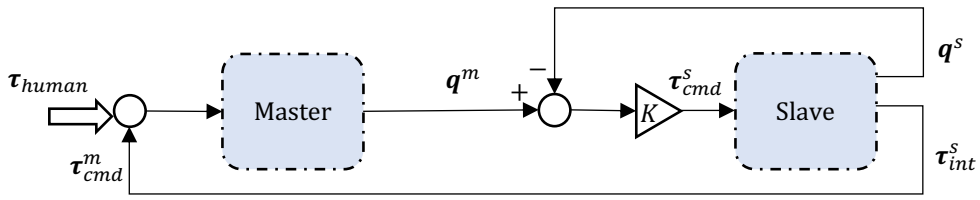
However, to the authors' knowledge, there are very few works on a real-world industrial box-in-box application. More importantly, the existing works on the automation of box-in-box insertion tasks focus on the compliance of the robot itself or that of specifically designed soft grippers but not on the compliance of the objects themselves. This paper leverages the compliance of the objects along with Learning from Demonstration techniques to learn and automate a compliant box-in-box insertion task in the context of packaging an electric light bulb. The passive compliance introduced by the deformable cardboard boxes not only assists the operator in performing the tasks but also makes the autonomous insertion more tolerant of errors in localising the pose of the boxes.

The proposed approach is divided into three phases: the (a) demonstration phase, (b) learning phase, and (c) generalisation phase. In the demonstration phase, the user

demonstrates the insertion task to the robot. The platform for teaching consists of two 7-DoF Kinova Gen3 Ultra-lightweight robot arms which form a master–slave teleoperated system whereby haptic feedback is always available to the user, as seen in Figure 2. While the active compliance provided by the master-robot enables the human to physically guides the robot effortlessly, the passive compliance introduced by the deformable cardboard boxes assists the operator in a smoother insertion by providing more lateral and angular tolerance. In the learning phase, the skill from human demonstrations is captured with probabilistic learning techniques, such as the Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR), which is then used for performing autonomous insertion. One key aspect of autonomous insertion is its capability to generalise for varying object poses. In the generalisation phase, a barycentric coordinate-based algorithm is implemented and evaluated to generalise the insertion for various object positions.




(**a**) Human user operating the master–slave teleoperated system.




(**b**) Teleoperation control block diagram.

**Figure 2.** Teleoperated system of two 7-DoF Kinova Gen3 robots.

This paper is organised in the following manner. The related works on Learning from Demonstration and teleoperation are detailed in Section 2. The proposed framework is presented in Section 3 with an overview of the underpinning control system, the task description, and the details of the learning algorithms employed. The subsequent Section 4 comprises the experimental validation of this approach, followed by the conclusion and future work in Section 5.

## 2. Related Works

Learning from Demonstration (LfD) is a policy learning approach that involves a human demonstrator performing/demonstrating a task that is to be imitated by a robot [14]. This technique ends the need for a non-expert robot user to learn how to manually program for a task, as maybe required by other assembly task implementations such as in [15]. Learning from Demonstration can be implemented on robots in several ways, of which the most common ones are kinesthetic teaching [16], teleoperation [17,18], vision-sensing [19–21], and the use of wearable sensors. This paper focuses on a compliant packaging box-in-box

insertion task using a slave robot arm that is controlled remotely by a master arm, in combination with vision-based sensing.

The datasets gathered from these demonstrations consist of task execution skills acquired from humans. These skills are then extricated (learnt) with the help of different techniques for the task to be performed by the robot [16]. One such commonly used technique is the Hidden Markov Model (HMM) [22,23], which is a robust probabilistic method to encode the spatial and temporal variabilities of human motion across various demonstrations [24] as a sequence of states. These states are defined as separate Gaussian Mixture Models (GMM) to explain the input data [25]. Ref. [23] compared the use of the HMM and Gaussian Mixture Regressions (GMR) approach vs. another, more popular Dynamic Movement Primitive (DMP) technique to allow robots to acquire skills through imitation. DMPs are basically units of action in terms of attractor dynamics of non-linear differential equations that encode a desired movement trajectory [26]. This method allows one to learn a control policy for a task from the demonstrations provided. However, standard DMP learning is prone to existing noise in human demonstrations [27].

It was also concluded by [23] that in the context of separated learning and reproduction, the HMM was more systematic in generalising motion than DMP. It allowed the demonstrator to provide partial demonstrations for a specific segment of the task instead of repeating the whole task again. This is an important feature when the aim is to refine one part of the movement.

Another study by [28], on the robotic assembly of mixed deformable and rigid objects, leveraged the use of haptic feedback with position and velocity controlled robots to make them compliant without explicitly engaging joint torque control. This information was integrated into a reinforcement learning (RL) framework as the insertion hole was smaller than the peg in diameter and was deformable. Thus, the contact mechanics for it were unknown, making it difficult to design a feedback control law. However, the lack of a vision system in their setup caused the insertion to sometimes fail. Another limitation of this method was that it only worked when the peg was relatively close to the hole. Research about automatic control helps with designing adaptive control [29,30] in cases where the peg position is different. Several other works utilising RL to adapt to varying dynamics of the environment during contact-rich robotic manipulation of non-rigid, deformable objects were reviewed by [31].

While the HMM and RL are widely used techniques for encoding task trajectories, these were not chosen for this paper due to the following shortcomings. RL requires a large dataset of demonstrations for training, which is an issue when there are time constraints, and the HMM method works by interpolating between discrete sets which are unsuitable for the continuous process of encoding tasks [32]. Also, the HMM relies on the proper choice of gains for stability, which requires estimating perturbations and the range of initial positions that the system can handle in advance, which can lead to inaccuracies [23]. Thus, in order to overcome these limitations, this work proposes a technique involving the use of GMMs for encoding training data and GMR for trajectory learning in combination with Barycentric coordinates for generalisation.

The GMM is a probabilistic method that is simple and robust enough for skill learning at the trajectory level. It enables the extraction of all trajectory constraints [16,21]. A combination of the GMM and GMR has been used to encode and generalise trajectories in previous independent works, such as [21,22,33]. The work by [22] presented a Learning by Demonstration framework for extracting different features of various manipulation tasks taught to a humanoid robot via kinesthetic teaching. The collected data were first projected onto a latent space for dimensionality reduction through the Principal Component Analysis (PCA) technique. Then, the signals were temporally aligned using the Dynamic Time Warping (DTW) approach for probabilistic encoding in GMMs. This study was able to successfully reproduce the important features of the performed tasks for different initial conditions. Ref. [21] made use of GMMs to encode upper-body gesture learning through the use of vision sensors and generalised the data through GMR.

While the GMM has been used to encode the training data for box-in-box insertion tasks, the proposed framework introduces a novel generalisation method based on GMR in Barycentric coordinates. Barycentric coordinates are commonly used in computer graphics and computational mechanics to represent a point inside a simplex as an affine combination of all of its vertices. This technique can be generalised to arbitrary polytopes [34] and is implemented in the generalisation approach of the training datasets. GMR is used to learn the robot trajectories that were demonstrated to the robot by the human operator and encoded by Gaussians. Regression makes it possible to retrieve a smooth signal analytically [22].

To perform human demonstrations of contact tasks like box-in-box assembly, it is important to consider how the task forces are being recorded and, more importantly, separated from the human forces acting on the system. Traditional kinesthetic teaching [35], i.e., by a human directly interacting with a robot and physically guiding it through the contact task, would result in the recording of the human forces coupled with the contact task forces. It is difficult to isolate the contact task forces only to perform force control later when generalising. One way to achieve this is to use two robots coupled through teleoperation control [36] so that the human interacts with the master robot and the slave robot deals with the contact task forces. Here, the term 'Master-Slave' simply means that the slave robot follows the motion of the master robot in the joint space. That is, to perform a task in a remote fashion, the human operator physically guides the master robot, thereby controlling the action of the slave robot which is in direct interaction with the environment. In fact, one of the first applications of teleoperation control was manipulation, specifically to handle radioactive materials from a safe distance in nuclear research [37].

For contact-rich tasks such as tight tolerance insertion, haptic feedback improves performance in the manipulation of remote objects [38]. Sensing in force-based teleoperation can be implemented through two possibilities: (a) a force/torque sensor attached to the flange of the robot to sense the end-effector forces and (b) a torque sensor built into every manipulator joint itself [39]. The latter approach was preferred due to the expensive nature of standalone six-axis F/T sensors. Instead, the robot construction was performed with strain gauges at every joint of an industrial robot manipulator [40]. Instead of using explicit force-control to track the contact task haptics, the teleoperation's impedance characteristics are used to capture the forces that the slave should exert on the environment. This is inherent in the master kinematics and is a key aspect of the proposed approach, which allows for an open-loop playback of the assembly demonstrations given the task conditions do not change.

## 3. Methodology

### *3.1. Demonstration Phase*

The goal in the demonstration phase is to successfully demonstrate the insertion of a folding carton (box B) containing an electric light bulb into the container (box A) shown in Figure 1. Kinesthetic teaching is one of the most widely used approaches in Learning from Demonstration paradigms for kinematic applications such as pick place and trajectory tracing. However, it is difficult to isolate the interaction forces purely due to insertion in the context of a contact task because robots with integrated joint torque sensors (such as the Kinova Gen3) also sense the forces due to human–robot physical interaction.

To allow the force feedback due to the contact task to be distinguished from the human forces, we choose to implement a master–slave teleoperated robot system using two Kinova Gen3 7-DoF robots. Continuous haptic feedback is provided to the human who interacts with the master side. The position information from the master side is used as the control input to the slave side. Consequently, all haptic feedback of the task on the slave side is fed back to the human operating the master side, as shown in Figure 2. As a result, the user is able to distinguish the interaction between geometric features such as the edges and faces of the cartons during the demonstration. The key advantage of this approach is the recording of the haptic feedback from a successful assembly task.

3.1.1. Teleoperation Control

Two identical robots are teleoperated in a master–slave configuration to demonstrate the insertion task. The user controls the master, which in turn controls the slave to carry out the desired action. Figure 3 depicts a block diagram for a single robot arm. The detailed block at the top of Figure 3 depicts the robot dynamic compensation, while the condensed block below is a simplified version of the same, which will be further used to represent each robot arm in the system. The robot feedback for the 7-DoF robot arm used in this work comprises the (sensed) joint position $\boldsymbol{q}$, joint velocitie $\dot{\boldsymbol{q}}$, and joint torque $\boldsymbol{\tau} \in \mathbb{R}^7$, respectively. Each robot is calibrated [41] using an oracle to determine the robot parameters, i.e., joint offsets and link lengths, that will be used in the computation of robot parameters, including inertia $\boldsymbol{M}$, Coriolis terms $\boldsymbol{C} \in \mathbb{R}^{7\times7}$, and the gravity term $\boldsymbol{G} \in \mathbb{R}^7$, used in the standard robotic control Equation (1) later in this section.

For a master–slave system to be reliable, it is important that the master offers minimal resistance to the user for smooth guidance. Therefore, we define the torque $\boldsymbol{\tau}_{dyn}$ (see Figure 3) originating from the dynamic effects (such as Inertia, Coriolis, and Gravity) that must be compensated during motion. Dynamic compensation is also crucial on the slave side, as the slave's motion is to be solely dictated (ideally) by an external command torque ($\boldsymbol{\tau}_{cmd}$) which will be a function of the master's state variable, as defined later on.

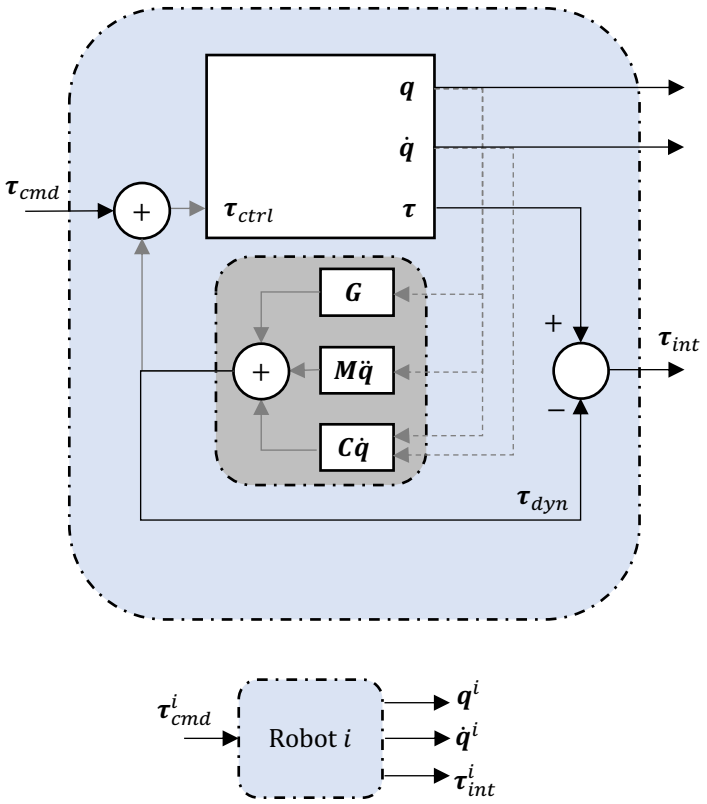



**Figure 3.** Torque controlled robot with dynamic compensation.

Therefore, with reference to Figure 3, the resultant robot dynamics can be written as:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}_{ctrl} \tag{1}$$

where $\boldsymbol{\tau}_{ctrl} = \boldsymbol{\tau}_{dyn} + \boldsymbol{\tau}_{cmd}$ is the control torque issued to the robot, which should comprise (i) the dynamic compensation $\boldsymbol{\tau}_{dyn}$, i.e., rendering the robot free to move in space, and (ii) external command torque $\boldsymbol{\tau}_{cmd}$.

The block at the bottom of Figure 3 is a simple representation of the control block of any robot $i$, with the associated control and feedback signals marked by the superscript $(\cdot)^i$. In order to establish a master–slave behaviour, two such robots are connected together as

depicted in the block diagram in Figure 4. As per the notation in the control block diagram above, we mark the control and feedback signals for master and slave with superscripts $m$ and $s$, respectively (for example, the feedback joint configuration for the slave is represented as $\boldsymbol{q}^s$).

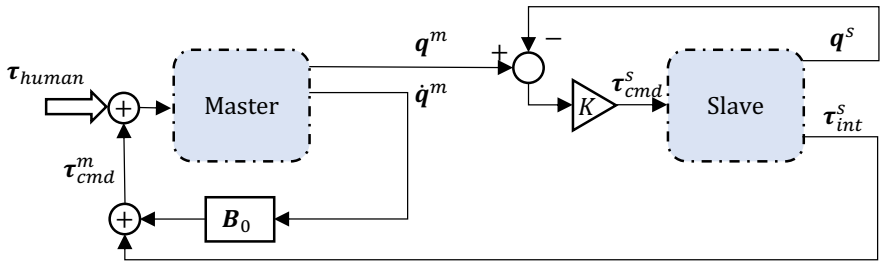



**Figure 4.** Master–slave control diagram for teleoperation with physical interaction with humans, captured by $\boldsymbol{\tau}_{human}$.

Here, the external command torque to the slave is elastic in nature and is proportional to the joint position (vector) differences between the master and the slave. That is,

$$\boldsymbol{\tau}^s_{cmd} = \boldsymbol{K}(\boldsymbol{q}^m - \boldsymbol{q}^s) \tag{2}$$

where $\boldsymbol{K} \in \mathbb{R}^{7\times7}$ is the stiffness matrix. A torque-controlled slave robot obeying the above control law always follows the master, which is physically guided by a user. Through proper modulation of the stiffness elements in $\boldsymbol{K} = \text{diag}([K_1\ K_2 \dots K_7])$, we programme the compliant behaviour of the slave robot, limiting the forces of interaction with its environment to mitigate the damage in the event of a collision.

For a master–slave system to be reliable, it is important that the master offers minimal resistance to the user for greater haptic transparency, i.e., rendering (as closely as possible) of slave-side contact forces. In addition to the dynamic compensation being conducted on both robots in the teleoperation scheme, we impose positive feedback on the master side to overcome the damping in the joints.

To that end, the external command torque on the master side is defined as follows:

$$\boldsymbol{\tau}^m_{cmd} = \boldsymbol{\tau}^s_{int} + \boldsymbol{B}_0\dot{\boldsymbol{q}}^m \tag{3}$$

where $\boldsymbol{\tau}^s_{int} = \boldsymbol{\tau}^s - \boldsymbol{\tau}^s_{dyn}$ is the interaction slave torque (i.e., the torque due to physical contact on the slave robot; the reader is referred to Figure 3), that results from the difference between the sensed feedback torque and the dynamic torques for the slave robot arm. $\boldsymbol{B}_0 \in \mathbb{R}^{7\times7}$ is the diagonal damping matrix for the master side, and it was regressed from the experimental data to compensate for the joint damping torques, i.e., $\boldsymbol{B}_0\dot{\boldsymbol{q}}^m$. The structure for this damping term is given by a diagonal matrix with each value corresponding to the joint damping $\boldsymbol{B}_0 = \text{diag}([\boldsymbol{B}_1\ \boldsymbol{B}_2 \dots \boldsymbol{B}_7])$.

#### 3.1.2. Task Demonstration

With the teleoperated robot system devised above, the user demonstrates the 4 stages of the task, which are the (a) picking, (b) placing, (c) insertion, and (d) retraction of the robot. In the demonstration, a set of four distinct locations is considered for box B in the task space, while keeping box A at a pre-defined position. The combination of all the sub-tasks corresponding to every Box B location is referred to as an experiment, and for each experiment multiple trials are performed for all the four sub-tasks. The higher the number of demonstrations, the more reliable the encoding of the human skill, as probabilistic approaches such as GMM and GMR tend to identify the most common trends in the demonstrations. Over the course of the demonstrations, the spatial vector containing the robot Cartesian poses $\boldsymbol{d}_s = [x, y, z, \alpha, \beta, \gamma] \in \mathbb{R}^{6\times1}$ (where $x$, $y$, and $z$ are the Cartesian position coordinates and $\alpha$, $\beta$, and $\gamma$ are the Euler angles about $X$, $Y$, and $Z$ axes, respectively)

is to be collected separately for each of the sub-tasks indexed against the timestamp data $d_t \in \mathbb{R}$. The collected data from multiple trials are then used to learn how each sub-task is performed by the human.

### *3.2. Learning Phase*

In the learning phase, we use GMM and GMR to obtain the statistical average across multiple trials in the demonstration, as discussed in this section.

#### 3.2.1. K-Means Clustering

To fit a Gaussian Mixture Model to a dataset, an initial guess for the mean and the covariance of the Gaussians is required. For this purpose, K-means clustering is used on the dataset. Note that the learning is performed for each sub-task (with a specific initial location of box B and the destination of box A) separately so that the learnt individual sub-tasks can be reused separately in the future.

The data for the robot Cartesian pose (for a sub-task) for a total of $N$ observations can be expressed as:

$$\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N] \tag{4}$$

where each observation (also called the data point) $\boldsymbol{d}_j = [d_t, \boldsymbol{d}_s]^T \in \mathbb{R}^n$ is indexed against the timestamp $d_t \in \mathbb{R}$ and a spatial component $\boldsymbol{d}_s \in \mathbb{R}^{(n-1)}$, where $n$ is the dimension of the data point. In this work, $n = 7$, because the Cartesian pose is described by 6 kinematic variables.

The parameters for the Gaussian Mixture Model (GMM) are initialised by performing the K-means clustering algorithm.

$$[\boldsymbol{\pi}_{ini}, \boldsymbol{\mu}_{ini}, \boldsymbol{\Sigma}_{ini}] = \textbf{Kmeans}\,(\boldsymbol{D}, K) \tag{5}$$

where $K$ is the number of clusters.

Here, $\boldsymbol{\mu}_{ini} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K]$, with initial mean $\boldsymbol{\mu}_j \in \mathbb{R}^n$ and $\boldsymbol{\Sigma}_{ini} = [\boldsymbol{\Sigma}, \ldots, \boldsymbol{\Sigma}_K]$ and with initial covariance $\boldsymbol{\Sigma}_j \in \mathbb{R}^{n \times n}$ for the $K$ Gaussians with probability densities $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \ldots, \mathcal{N}(\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$, and $\mathcal{N}_j \in \mathbb{R}^n \times Sym^+(n)$ ($Sym^+(j)$ denotes the manifold of $j \times j$ Symmetric Positive-Definite (SPD) matrices. Finally, we have $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_k]$, $\pi_j \in \mathbb{R}$ representing the prior probabilities for each of the Gaussian components.

#### 3.2.2. Expectation Maximisation (EM)

The process of Expectation Maximisation [42] estimates the parameters for the Gaussian Mixture Model by performing a recursive 2-stage (the E-step and the M-step) computation until convergence. The expectation (E-step) computes the contribution (known as the weight) of each of the Gaussians in the occurrence of each of the data points, while maximisation (the M-step) updates the initial guess for the Gaussian parameters based on the computed weights. Hence,

$$[\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}] = \textbf{EM}(\boldsymbol{D}, \boldsymbol{\pi}_{ini}, \boldsymbol{\mu}_{ini}, \boldsymbol{\Sigma}_{ini}) \tag{6}$$

where $\boldsymbol{\pi} \in \mathbb{R}^n, \boldsymbol{\mu} \in \mathbb{R}^{n \times K}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n \times K}$ are the updated parameters (Refer to Appendix A).

As a result, a Gaussian Mixture Model (GMM) with $K$ components can be used to model the dataset, with the following probability density function

$$p(\boldsymbol{d}_j) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{d}_j; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{7}$$

where

$$\boldsymbol{\mu}_j = [\mu_{t,j}, \boldsymbol{\mu}_{s,j}], \qquad \boldsymbol{\Sigma}_j = \begin{bmatrix} \Sigma_{tt,j} & \boldsymbol{\Sigma}_{ts,j} \\ \boldsymbol{\Sigma}_{st,j} & \boldsymbol{\Sigma}_{ss,j} \end{bmatrix} \tag{8}$$

and $\mu_{t,j} \in \mathbb{R}$, $\boldsymbol{\mu}_{s,j} \in \mathbb{R}^{n-1}$, $\Sigma_{tt,j} \in \mathbb{R}$, $\boldsymbol{\Sigma}_{ts,j} \in \mathbb{R}^{n-1}$, and $\boldsymbol{\Sigma}_{ss,j} \in \mathbb{R}^{(n-1)\times(n-1)}$.

#### 3.2.3. Gaussian Mixture Regression (GMR)

Once GMM is fitted to the obtained dataset, the goal is to extract a single average trajectory through GMR; the trajectory is characterised by all the demonstrations but weighted by the probability of their occurrence. In addition, GMR provides a smooth trajectory despite any discontinuous or jerky motions present during the demonstration.

A regression of the obtained Gaussian mixture model over a temporal vector $\boldsymbol{T} = [t_1, t_2, \ldots, t_m]$ can be performed as [22]

$$[\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r] = \mathbf{GMR}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{T}) \tag{9}$$

where $\boldsymbol{\mu}_r \in \mathbb{R}^{(n-1)\times m}$ and $\boldsymbol{\Sigma}_r \in \mathbb{R}^{(n-1)\times(n-1)\times m}$ are the conditional expectation of the data and associated covariances corresponding to the timestamp values in $\boldsymbol{T}$ (refer to Appendix B).

Performing GMR outputs a trajectory $\boldsymbol{r}(t) = [(t_1, \boldsymbol{\mu}_1), (t_2, \boldsymbol{\mu}_2), \ldots, (t_m, \boldsymbol{\mu}_m)]$ for each sub-task, and the overall robot trajectory for the $k^{th}$ experiment is characterised by executing the trajectories for all the subtasks, namely pick, place, insert, and retract ($\boldsymbol{r}_{p_k}(t)$, $\boldsymbol{r}_{pl_k}(t)$, $\boldsymbol{r}_{in_k}(t)$, and $\boldsymbol{r}_{re_k}(t)$, respectively) in order.

### *3.3. Generalisation Phase*

The demonstrated trials, though performed for a selected set of object poses, can be generalised to arbitrary object positions. Barycentric coordinates, a widely used notion in computer graphics, especially in texture mapping and ray tracing applications, can be utilised for this.

**Note:** This work does not address the problem of generalisation for varying object orientations.

#### Trajectory Generalisation Using Barycentric Coordinates

For a 2D convex polygon $P$ with $n$ vertices $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$, one can define a set of barycentric coordinates $\phi = [\phi_1, \phi_2, \ldots, \phi_n] : P \to \mathbb{R}^n$ such that an arbitrary object position $\boldsymbol{x} \in \mathbb{R}^2$ within the convex hull of $P$ can be expressed [34] as

$$\sum_{i=1}^{n} \phi_i(\boldsymbol{x}) \boldsymbol{v}_i = \boldsymbol{x} \quad \forall\, \boldsymbol{x} \in P \tag{10}$$

from the linear reproduction property wherein

$$\sum_{i=1}^{n} \phi_i(\boldsymbol{x}) = 1 \quad \forall\, \boldsymbol{x} \in P \tag{11}$$

from the partition of unity property.

Bringing this to the task perspective (See Figure 5), assuming that the initial object positions in the 2D plane, which are observed by the robot at time $t_p$ (in the picking phase of the trial) across $m$ trials, i.e., $\boldsymbol{p} = [\boldsymbol{p}_1(t_p), \boldsymbol{p}_2(t_p), \ldots, \boldsymbol{p}_m(t_p)]$: $\boldsymbol{p}_i(t_p) \in \mathbb{R}^2$, form a convex polygon, for $m$ different initial positions of box B, any arbitrary object position $\boldsymbol{p}_0$ (for brevity, henceforth, we omit indicating that the barycentric coordinates are being evaluated on the demonstration data at time $t_p$, i.e., for different initial positions of box B at the time of picking) within the polygon can be represented in terms of a set of barycentric coordinates $w_1, \ldots w_m$ such that:

$$\boldsymbol{p}_0 = \sum_{i=1}^{m} w_i \, \boldsymbol{p}_i \tag{12}$$

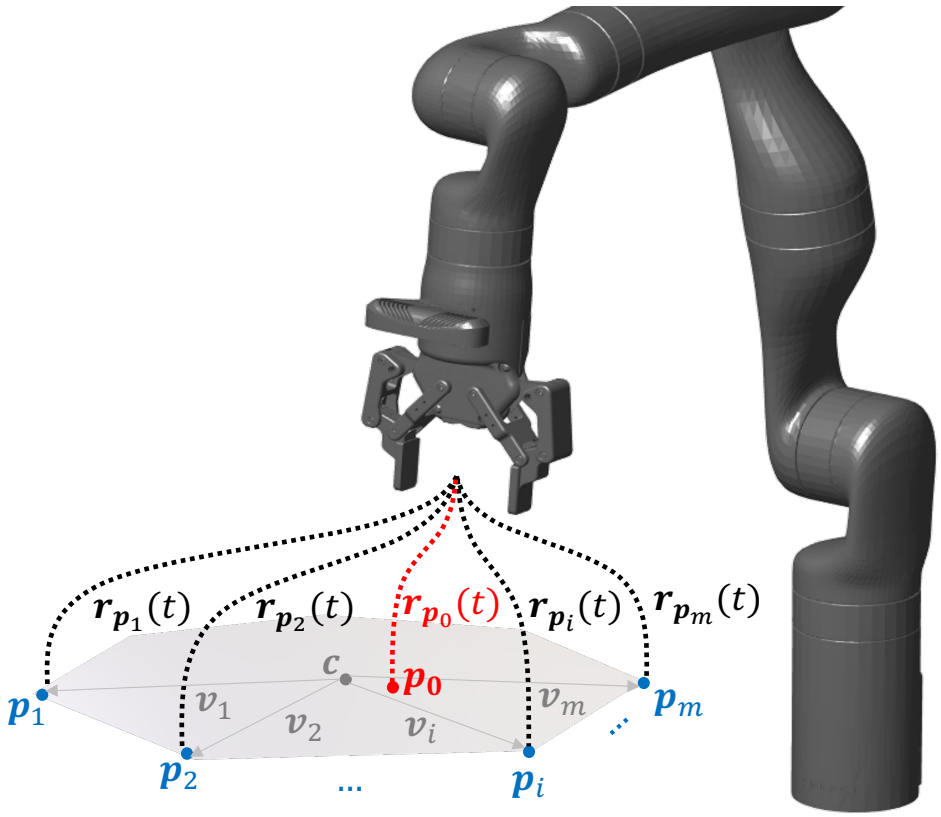



**Figure 5.** For an arbitrary object pick-position $p_0$ within the convex hull of the polygon formed by the demonstrated positions, the trajectory of the robot (red) can be represented as an affine combination of the demonstrated trajectories (blue).

To ensure that $\sum_{i=1}^{m} w_i = 1$, out of all the possible sets of barycentric coordinates, the considered affine functions possessing the minimal l-2 norm [43] are computed as:

$$w_i = \langle \boldsymbol{p}_0 - \boldsymbol{c}, (VV^T)^{-1} \boldsymbol{v}_i \rangle + \frac{1}{m} \tag{13}$$

where $\langle \cdot, \cdot \rangle$ is the inner product and $V = [\boldsymbol{v}_1, \ldots \boldsymbol{v}_m] \in \mathbb{R}^{2 \times m}$ for each of the pick-phase object positions from demonstrations, with $\boldsymbol{v}_i = \boldsymbol{p}_i - \boldsymbol{c}$ and $\boldsymbol{c} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{p}_i$.

In this work, to ensure the geometric nature of the generalisation capability to pick the box from a new position, we consider the use of barycentric coordinates. This is guided by the intuition that if the box is placed in the exact same position as in the original demonstration, the reproduction trajectory should be no different from the demonstration. Conventional generalisation techniques such as TP-GMM compute the reproduction trajectories by way of products of task-parameterised Gaussians (using a quadratic error cost, i.e., Equation (3) [44] to prioritise the frame that contributed more to the product Gaussian's mean at a certain time). For this reason, TP-GMM would not generate an exactly identical trajectory given an exactly identical object position. Instead, the reproduction trajectory will be a weighted average, with the weights chosen based on probabilities and not geometry. Thus, it is a probabilistic approach that allows one to prioritise which task frame was important at different times of the task [45] based on the variance in the demonstration as observed from that frame. For this reason the TP-GMM formulation works best when the number of initial poses shown are maximised, since their contribution is weighted in the computation of the task-parameterised variance. Therefore, the reproduction trajectory will not be identical to the original demonstration even if the new object pose is identical. Using the geometric nature of barycentric coordinates, we overcome this challenge. Moreover, the use of barycentric coordinates may be deployed in rotations as well, since this method

is geometrically invariant (i.e., the coordinates $w_i$ may be applied to any vector-space extrapolation, not necessarily in the Euclidean space of position vectors $\boldsymbol{p}_i$), allowing one to expand the generalisation capability to rotation-only generalisation, e.g., when the box is at the same position but just rotated, making the use of barycentric coordinates a novel contribution to generalisation capabilities in LfD scenarios.

## 4. Experimental Validation

In this section, the proposed framework is validated by performing an autonomous box-in-box insertion test followed by a repeatability test with arbitrary box B positions. Note that these experiments focus on how well this framework performs for varying object positions while considering the orientation to be constant. The master–slave setup implemented in this paper involves two 7-DoF Kinova Gen3 ultra-lightweight robotic arms. The master robot arm is mounted with a customised 3D printed handle to provide the user with the convenience of guiding, whereas the slave is equipped with a 2F-85 Robotiq gripper for object manipulation. Both the robots are controlled in torque control mode from a single workstation at a frequency of 1 kHz through TCP/IP. Before the task demonstration was performed, the procedure for fast kinematic re-calibration of cobots [41] was performed on the slave robot to account for any joint offsets that need to be regressed. The results from this experiment are listed in Table 1. Additionally, the gravity compensation module on both the robots was tested to verify there is no drift in the robots. In case of drift, the joint torque offsets of the robots were reset after initialising the robot to the zero configuration. This is an important step to ensure the human does not feel unnecessary forces originating from torque sensor offsets fed back because of the torque controller, i.e., the intended function is so that the human feels the haptic feedback from the task only. Once the robots are initialised with these steps, the demonstration of the insertion task of the box-in-box assembly is carried out.

**Table 1.** Experimental parameters and list of acronyms.

| Quantity Name | Symbol | Value and Units |
|---|---|---|
| Stiffness for joint $i$ | $K_i$ | 15 Nm/rad |
| Joint 1 damping for master | $\boldsymbol{B}_1$ | 3.3132 Ns |
| Joint 2 damping for master | $\boldsymbol{B}_2$ | 3.4284 Ns |
| Joint 3 damping for master | $\boldsymbol{B}_3$ | 3.1245 Ns |
| Joint 4 damping for master | $\boldsymbol{B}_4$ | 3.2895 Ns |
| Joint 5 damping for master | $\boldsymbol{B}_5$ | 2.836 Ns |
| Joint 6 damping for master | $\boldsymbol{B}_6$ | 2.8579 Ns |
| Joint 7 damping for master | $\boldsymbol{B}_7$ | 7.6092 Ns |
| Pick time | $t_p$ | ≈11 s |
| Place time | $t_{pl}$ | ≈25 s |
| Insert time | $t_{in}$ | ≈34 s |
| Retract time | $t_{re}$ | ≈40 s |
| **Full form** | | **Acronym** |
| Learning from Demonstration | | LfD |
| Gaussian Mixture Model | | GMM |
| Gaussian Mixture Regression | | GMR |
| Expectation Maximisation | | EM |
| Internet Protocol Suite | | TCP/IP |

### *4.1. Task Demonstration*

The overall insertion task was divided into four sub-tasks: (see Figure 6). Here, the geometry of box B is that of a square prism and is placed in such a way that it rests on one of its bases.

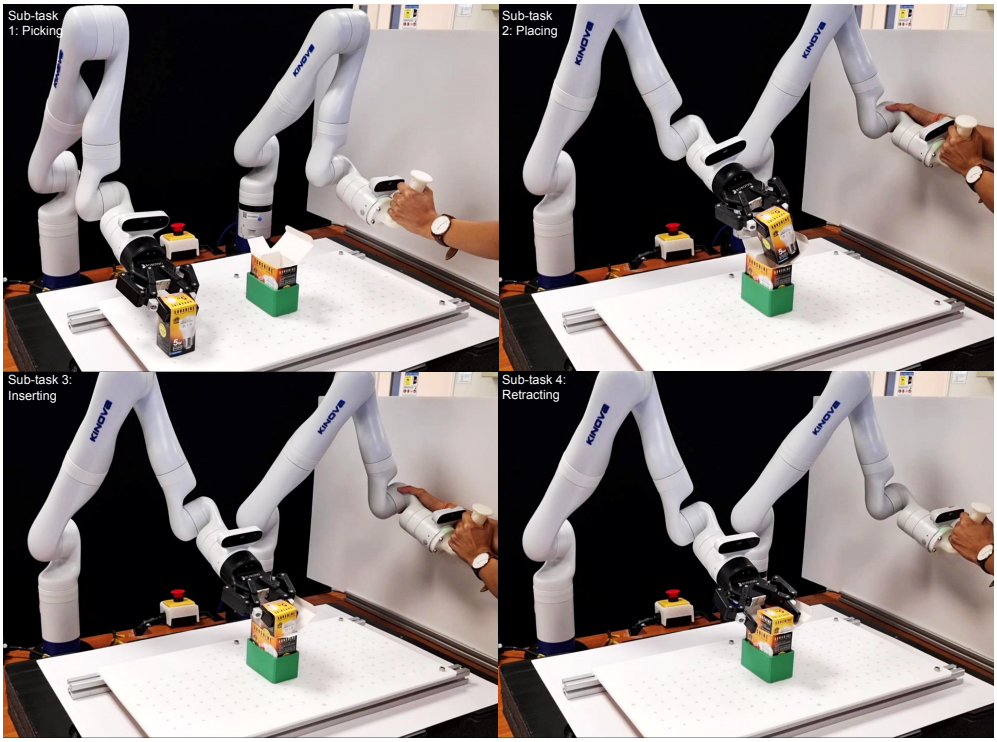



**Figure 6.** Demonstration of pick and insert task performed using master–slave teleoperation between two KINOVA Gen3 robotic arms. User moves the master robot to guide the slave robot to perform the following sequence of tasks: (video: refer to Supplementary Materials section, link 1).

In the picking sub-task, both the master and the slave robots initially assume a pre-defined home position. The teleoperated setup enables the user to control the slave motion by guiding the master robot. The user guides the master in such a way that the slave approaches box B with its gripper oriented at a suitable angle for grasping. As soon as the robot is ready to pick, the gripper is closed, ensuring a firm grasp.

Once box B is picked up, the user enters into the insertion sub-task, where the user guides the slave towards box A. The placing and insertion sub-tasks are interlinked, as the mechanism of insertion determines the way box B is placed at the opening of box A.

Considering that the task at hand is a tight-tolerance insertion task, insertions performed with a vertical approach vector are highly likely to fail. As such, the passive compliance provided by the boxes is exploited to perform a successful insertion. This can be seen in the insertion sub-task, divided into three sub-stages, with reference to Figure 7.

At the end of the placing sub-task, box B is oriented in such a way that one of its vertices is placed around the centre of the opening of box A.

Subsequently, the vertex of box B is pushed against the edge, followed by a simultaneous rotation about the $X, Y$ axes and a translation along the $Z$ axis. Here, the compliance introduced by the elastic nature of box B guides box A into B, assisting with the insertion task. Once the insertion is performed, the gripper is opened and the retraction sub-task is executed to move the robot away to a safe position.

In this demonstration, a set of distinct locations is considered for box B in the task space, while keeping box A at a pre-defined position. For each experiment, a new box location is considered, and for each box location, multiple demonstrations are performed (referred to as trial) for all the four sub-tasks. The higher the number of demonstrations, the more reliable the encoding of the human skill, as probabilistic approaches such as the GMM and GMR tend to identify the most common trends in the demonstrations. In addition, humans learn and adapt to a task (such as insertion) after repeated trials, identifying a successful strategy of action. Over the course of the demonstrations, the robot Cartesian pose data $[x, y, z, \alpha, \beta, \gamma]$ (where $x, y$, and $z$ are the Cartesian position coordinates and $\alpha, \beta$,

and $\gamma$ are the Euler angles about the $X, Y$, and $Z$ axes, respectively) are logged separately for each of the sub-tasks along with the temporal data $t$ at a frequency of 1 kHz (refer to Figure 8).




**Figure 7.** The insertion sub-task is performed in 3 sub-stages. In the first stage, box B is oriented such that one of the corners is placed around the centre of box A. This is followed by aligning the corners of the boxes, followed by the final insertion.

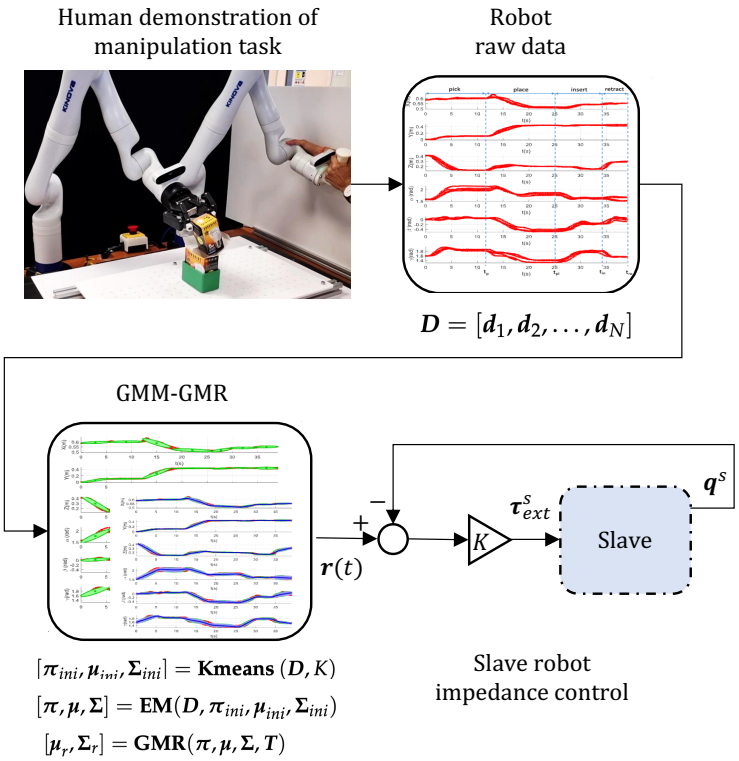



**Figure 8.** From the human demonstration of manipulation task, sample dataset with $\boldsymbol{d}_i = [t, x, y, z, \alpha, \beta, \gamma]^T$ representing the robot position $(x, y, z)$ and orientation $(\alpha, \beta, \gamma)$ in the world frame is collected. Here, $t_p, t_{pl}, t_{in}$, and $t_{re}$ represent the time taken for each of the sub-tasks (pick, place, insert, and retract, respectively). Then, the GMM fitting is performed with $K = 7$ components and the corresponding Gaussian regression allows one to compute the trajectory $\boldsymbol{r}(t)$, which is fed as the input to slave robot impedance control.

### *4.2. Learning Autonomous Insertion with GMM and GMR*

The resulting Gaussian Mixture Regression is carried out for robot-pose data and is depicted in Figure 8 with the number of clusters, $K = 7$, for the fitting of the Gaussian Mixture Model.

The fitted Gaussian Mixture Regression is then used to determine whether the slave robot (in position control mode) can successfully insert arbitrary object (box B, in our case) positions. An insertion is marked as successful if and only if box B is fully inserted in box A causing no damage to either of the boxes. In the event of a partial insertion, no insertion, or insertion causing damage to the boxes, the attempt is considered unsuccessful. Throughout this experiment, box A assumes the same position as it did in the demonstration phase. The various sub-tasks performed autonomously by the robot for a single successful trial are illustrated in Figure 9.

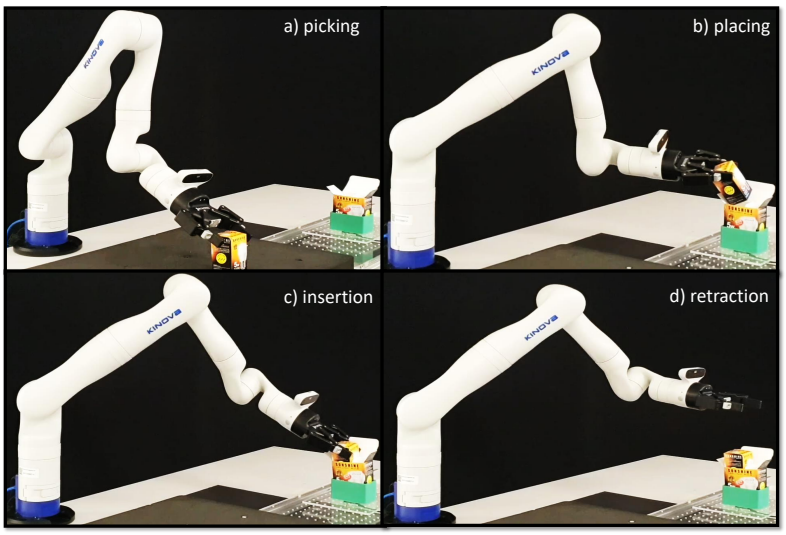



**Figure 9.** Automated insertion based on the demonstrated trials for a new box B position (video: refer to Supplementary Materials section, link 2).

### *4.3. Generalisation and Repeatability Test*

Six trials of autonomous insertion were carried out, with box B assuming six different positions within the convex polygon formed by the demonstrated positions. For each trial, the built-in RGB and depth camera on the Kinova Gen3 were used to identify the position of the box, which was followed by the barycentric coordinate-based trajectory adaptation. It was observed that the robot performed successful insertion for all six trials.

The repeatability of insertion was tested out by performing 10 trials corresponding to one of the box positions. The object was always oriented in such a way that its edges were in parallel with the base frame $X$-$Y$ axes. Rather than just performing a simple repeatability test, it was important to analyse how well this framework performs under slight variations in the camera-identified object pose. For instance, if the object is manually reinstated back to its original pose after each trial, there exist slight deviations from the initially identified pose (by the camera). To test how this affects the insertion task, the robot assumes that the object pose is the same as the one identified in the first trial, where, in fact, slight variations were introduced in successive trials due to the manual placement.

In all 10 of the trials, the robot (operating in position control mode), successfully performed an insertion (for the video depicting the repeatability tests for two different box positions inside the polygon: video: refer to Supplementary Materials section, link 3). Even though the success of insertion highly depends on the repeatability of the robot itself, the results show that despite the slight variations in the object position and orientation arising from the manual placement (which is not captured through vision), the robot is able to perform the insertion successfully. The passive pose adjustment of the object when the gripper closes and the compliance provided by the boxes during the insertion allows such a position-controlled robot to accomplish these insertions successfully.

### *4.4. Discussion*

Beyond the trivial advantages of having ease of programming with quick human demonstrations, our choice in using the GMM as the learning method allows for fast training compared with deep neural networks. This is because of the low dimensionality of the GMM, i.e., in the parameters used to train the model. From a practitioner's perspective, no more than 10 demonstrations, each less than a minute long, were sufficient to automate the task of box-in-box insertion shown in this work.

The choice of the GMM also makes the model more explainable. Specifically, with time as one of the parameters, the choice of the number of Gaussian kernels directly relates to the number of "phases" in the experiment. For instance, a practitioner may directly choose this parameter based on their experience of the task. This contrasts with deep neural networks, where choosing the number of parameters/layers/activation functions is an open problem. The generalisation in this work was conducted using barycentric coordinates, which allow one to generate robot motion for a new object pose given visual feedback. However, this generalisation does not account for any adaptation needed in case of a task failure. This is one of the future threads to explore.

The importance of compliance has been studied in assembly tasks from the early 1980s [46] and, more recently, in [47]. Typically, lateral and angular compliances are required for at least one of the mating parts, to account for the errors encountered due to various sources of uncertainties in the task. The concept of a remote centre of compliance allows for free rotation/translation to reduce resistance and may be implemented using passive mechanisms [46], active compliance control [48], and by learning compliant motion primitives [49] from demonstrations that encode the correlation between compliance and tolerance required in the task. In this work, we rely on the passive compliance inherent in the paper boxes used for assembly instead of allowing for tolerance errors during insertion.

## 5. Conclusions and Future Work

Autonomous insertion tasks hold a great deal of significance in industrial packaging applications. This work presented a box-in-box insertion framework that uses the Learning from Demonstration paradigm in contrast to the conventional programming-intensive approaches. To respect the contact-rich nature of the insertion task, a master–slave teleoperated robot system was developed where the forces of interaction with the environment are effectively perceived by the user through continuous haptic feedback. The human skill from these demonstrations is captured with the help of the Gaussian Mixture Model and Gaussian Mixture Regression. The learned skill can be applied to different positions (of the box to be inserted) in the task space and is only constrained by the boundaries of a polygon formed by the demonstrated object positions, as shown by the experimental analysis. In the repeatability test conducted, it was observed that despite the presence of slight variations in the estimated object pose, the robot was able to successfully perform the insertion.

The proposed approach mainly relies on the compliance offered by both the folding cartons, which facilitated smooth guidance during the insertion sub-task. In other words, compliance is provided by the environment rather than the robot, which enables successful insertion. This work only focused on how the demonstrated skill can be adapted to varying object positions. Our future work will incorporate the variation in the object orientation as well into the framework. Additionally, impedance control can potentially be implemented on the robot to achieve minimal contact forces in the event of an unsuccessful insertion.

**Supplementary Materials:** Video links for the experiments performed can be found below. 1. Demonstration of teleoperated pick and insert task: https://clipchamp.com/watch/K2rOLixNQsE (accessed on 29 September 2023); 2. Autonomous insertion: https://drive.google.com/file/d/1DW4SmIeCZMo1kObChQ6XcDi96-lu1afd/view?usp=share_link (accessed on 29 September 2023); 3. Repeatability test: https://drive.google.com/file/d/17csYMthMBpPtgOFTxGc5KR1r0cUocfST/view?usp=share_link (accessed on 29 September 2023).

**Author Contributions:** Conceptualisation, S.K., J.G. and D.C.; methodology, S.K., J.G. and D.C.; software, S.K., J.G. and V.R.; experimental validation, S.K., J.G. and M.Z.A.; data curation, S.K. and J.G.; writing—original draft preparation, S.K. and J.G.; writing—review and editing, J.G., M.Z.A. and S.H.T.; project administration, J.G. and D.C.; funding acquisition, D.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was conducted under project WP3 within the Delta-NTU Corporate Lab with funding support from A*STAR under its IAF-ICP programme (Grant no: I2201E0013) and Delta Electronics Inc.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## Appendix A. Expectation Maximisation

For a set of $K$ Gaussians with probability densities $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \ldots, \mathcal{N}(\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$ and with prior probabilities as $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_k]$, $\pi_k \in \mathbb{R}$, the Expectation Maximisation algorithm follows the below steps in order until convergence occurs.

(a) The weight of the $j$th cluster $\mathcal{N}_j = \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$ in contributing to the $i$th data point $\boldsymbol{d}_i$ is computed as

$$w_{ij} = P(\boldsymbol{d}_i \text{ belongs to } \mathcal{N}_j) = \frac{\pi_j \mathcal{N}(\boldsymbol{d}_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{d}_i, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \tag{A1}$$

(b) Update the priors, means, and covariances as:

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} w_{ij} \tag{A2}$$

$$\boldsymbol{\mu}_j = \frac{1}{N\pi_j} \sum_{i=1}^{N} w_{ij} \boldsymbol{d}_i \tag{A3}$$

$$\boldsymbol{\Sigma}_j = \frac{1}{N\pi_j} \sum_{i=1}^{N} w_{ij} (\boldsymbol{d}_i - \boldsymbol{\mu}_j)(\boldsymbol{d}_i - \boldsymbol{\mu}_j)^T \tag{A4}$$

## Appendix B. Gaussian Mixture Regression

For each Gaussian component $j$, for a given temporal value $t_i$, the expected distribution of the spatial value $\boldsymbol{d}_{st_i}$ can be obtained as

$$p(\boldsymbol{d}_{st_i} | t_i, j) = \mathcal{N}(\boldsymbol{d}_{st_i}; \hat{\boldsymbol{d}}_{s,j}, \hat{\boldsymbol{\Sigma}}_{ss,j}) \tag{A5}$$

$$\hat{\boldsymbol{d}}_{s,j} = \boldsymbol{\mu}_{s,j} + \boldsymbol{\Sigma}_{st,j} (\Sigma_{tt,j})^{-1} (t_i - \mu_{t,j}) \tag{A6}$$

$$\hat{\boldsymbol{\Sigma}}_{ss,j} = \boldsymbol{\Sigma}_{ss,j} - \boldsymbol{\Sigma}_{st,j} (\Sigma_{tt,j})^{-1} \boldsymbol{\Sigma}_{ts,j} \tag{A7}$$

Looking at the complete model, the expected distribution is given by

$$p(\boldsymbol{d}_{st_i} | t_i) = \sum_{j=1}^{K} \beta_j \mathcal{N}(\boldsymbol{d}_{st_i}; \hat{\boldsymbol{d}}_{s,j}, \hat{\boldsymbol{\Sigma}}_{ss,j}) \tag{A8}$$

where $\beta_j = p(j|t_i)$ is the probability that the $j^{th}$ Gaussian component is responsible for $t = t_i$ and is obtained as:

$$\beta_j = \frac{p(j)p(t_i|j)}{\sum_{k=1}^{K} p(k)p(t_i|k)} = \frac{\pi_j \mathcal{N}(t_i; \mu_{t,j}, \Sigma_{tt,j})}{\sum_{k=1}^{K} \pi_k \mathcal{N}(t_i; \mu_{t,k}, \Sigma_{tt,k})} \tag{A9}$$

The linear transformation property of Gaussian distribution allows us to write the conditional probability distribution as $p(\boldsymbol{d}_{st_i}|t_i) \sim \mathcal{N}(\hat{\boldsymbol{d}}_s, \hat{\Sigma}_{ss})$, where the parameters of the Gaussian is defined as:

$$\hat{\boldsymbol{d}}_s = \sum_{j=1}^{K} \beta_j \, \hat{\boldsymbol{d}}_{s,j}, \; \hat{\boldsymbol{\Sigma}}_{ss} = \sum_{j=1}^{K} \beta_j^2 \, \hat{\boldsymbol{\Sigma}}_{ss,j} \tag{A10}$$

Hence, the regression of the mixture model over the temporal vector $T$ yields the conditional expectation of the data, given by $\boldsymbol{\mu}_r = [\boldsymbol{d}_{r1}, \boldsymbol{d}_{r2}, \ldots, \boldsymbol{d}_{rm}]$, where $\boldsymbol{d}_{ri} = \hat{\boldsymbol{d}}_s|_{t=t_i}$, $\hat{\boldsymbol{d}}_s \in \mathbb{R}^{(n-1)}$ and the associated covarince matrices $\boldsymbol{\Sigma}_r = [\boldsymbol{\Sigma}_{r1}, \boldsymbol{\Sigma}_{r2}, \ldots, \boldsymbol{\Sigma}_{rm}], \boldsymbol{\Sigma}_{ri} = \hat{\boldsymbol{\Sigma}}_{ss}|_{t=t_i}$, $\hat{\boldsymbol{\Sigma}}_{ss} \in \mathbb{R}^{(n-1)\times(n-1)}$.

## References


1. Whitney, D.E. *Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development*; Oxford University Press: New York, NY, USA, 2004; Volume 1.
2. Xu, J.; Hou, Z.; Liu, Z.; Qiao, H. Compare contact model-based control and contact model-free learning: A survey of robotic peg-in-hole assembly strategies. *arXiv* **2019**, arXiv:1904.05240.
3. Chin, K.S.; Ratnam, M.M.; Mandava, R. Force-guided robot in automated assembly of mobile phone. *Assem. Autom.* **2003**, *23*, 75–86. [CrossRef]
4. Li, X.; Xiao, J.; Zhao, W.; Liu, H.; Wang, G. Multiple peg-in-hole compliant assembly based on a learning-accelerated deep deterministic policy gradient strategy. *Ind. Robot. Int. J. Robot. Res. Appl.* **2021**, *49*, 54–64. [CrossRef]
5. Song, R.; Li, F.; Fu, T.; Zhao, J. A robotic automatic assembly system based on vision. *Appl. Sci.* **2020**, *10*, 1157. [CrossRef]
6. Navarro-Alarcon, D.; Liu, Y.H.; Romero, J.G.; Li, P. On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments. *Int. J. Robot. Res.* **2014**, *33*, 1462–1480. [CrossRef]
7. Kazemi, M.; Valois, J.S.; Bagnell, J.A.; Pollard, N. Human-inspired force compliant grasping primitives. *Auton. Robot.* **2014**, *37*, 209–225. [CrossRef]
8. Hogan, N. The mechanics of multi-joint posture and movement control. *Biol. Cybern.* **1985**, *52*, 315–331. [CrossRef]
9. Mason, M.T. Compliance and force control for computer controlled manipulators. *IEEE Trans. Syst. Man Cybern.* **1981**, *11*, 418–432. [CrossRef]
10. Xing, D.; Liu, X.; Liu, F.; Xu, D. Efficient insertion strategy for precision assembly with uncertainties using a passive mechanism. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1263–1273. [CrossRef]
11. Wermelinger, M.; Johns, R.; Gramazio, F.; Kohler, M.; Hutter, M. Grasping and Object Reorientation for Autonomous Construction of Stone Structures. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5105–5112. [CrossRef]
12. Wang, T.; Joo, H.J.; Song, S.; Hu, W.; Keplinger, C.; Sitti, M. A versatile jellyfish-like robotic platform for effective underwater propulsion and manipulation. *Sci. Adv.* **2023**, *9*, eadg0292. [CrossRef]
13. Zhai, Y.; Boer, A.D.; Yan, J.; Shih, B.; Faber, M.; Speros, J.; Gupta, R.; Tolley, M.T. Desktop fabrication of monolithic soft robotic devices with embedded fluidic control circuits. *Sci. Robot.* **2023**, *8*, adg3792. [CrossRef] [PubMed]
14. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A survey of robot learning from demonstration. *Robot. Auton. Syst.* **2009**, *57*, 469–483. [CrossRef]
15. Polverini, M.P.; Zanchettin, A.M.; Rocco, P. A constraint-based programming approach for robotic assembly skills implementation. *Robot. Comput. Integr. Manuf.* **2019**, *59*, 69–81. [CrossRef]
16. Kyrarini, M.; Haseeb, M.A.; Ristić-Durrant, D.; Gräser, A. Robot learning of industrial assembly task via human demonstrations. *Auton. Robot.* **2019**, *43*, 239–257. [CrossRef]
17. Havoutis, I.; Calinon, S. Learning assistive teleoperation behaviors from demonstration. In Proceedings of the 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Lausanne, Switzerland, 23–27 October 2016; pp. 258–263. [CrossRef]
18. Pervez, A.; Ali, A.; Ryu, J.H.; Lee, D. Novel learning from demonstration approach for repetitive teleoperation tasks. In Proceedings of the 2017 IEEE World Haptics Conference (WHC), Munich, Germany, 6–9 June 2017; pp. 60–65. [CrossRef]
19. Huang, B.; Ye, M.; Lee, S.L.; Yang, G.Z. A Vision-Guided Multi-Robot Cooperation Framework for Learning-by-Demonstration and Task Reproduction. 2017. Available online: http://xxx.lanl.gov/abs/1706.00508 (accessed on 29 September 2023).

20. Lin, L.; Yang, Y.; Song, Y.; Nemec, B.; Ude, A.; Rytz, J.; Buch, A.; Krüger, N.; Savarimuthu, T. Peg-in-Hole assembly under uncertain pose estimation. In Proceedings of the Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 2842–2847. [CrossRef]
21. Sabbaghi, E.; Bahrami, M.; Ghidary, S.S. Learning of gestures by imitation using a monocular vision system on a humanoid robot. In Proceedings of the 2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 15–17 October 2014; pp. 588–594. [CrossRef]
22. Calinon, S.; Guenter, F.; Billard, A. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2007**, *37*, 286–298. [CrossRef] [PubMed]
23. Calinon, S.; Sauser, E.L.; Billard, A.G.; Caldwell, D.G. Evaluation of a probabilistic approach to learn and reproduce gestures by imitation. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2671–2676.
24. Lee, S.H.; Suh, I.H.; Calinon, S.; Johansson, R. Autonomous framework for segmenting robot trajectories of manipulation task. *Auton. Robot.* **2015**, *38*, 107–141. [CrossRef]
25. Brown, J.C.; Smaragdis, P. Hidden Markov and Gaussian mixture models for automatic call classification. *J. Acoust. Soc. Am.* **2009**, *125*, EL221–EL224. [CrossRef]
26. Meier, F.; Schaal, S. A probabilistic representation for dynamic movement primitives. *arXiv* **2016**, arXiv:1612.05932.
27. Ghalamzan E, A.M.; Paxton, C.; Hager, G.D.; Bascetta, L. An incremental approach to learning generalizable robot tasks from human demonstration. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5616–5621. [CrossRef]
28. Luo, J.; Solowjow, E.; Wen, C.; Ojea, J.A.; Agogino, A.M. Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2062–2069.
29. Chi, R.; Li, H.; Shen, D.; Hou, Z.; Huang, B. Enhanced P-Type Control: Indirect Adaptive Learning From Set-Point Updates. *IEEE Trans. Autom. Control* **2023**, *68*, 1600–1613. [CrossRef]
30. Roman, R.C.; Precup, R.E.; Petriu, E.M. Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems. *Eur. J. Control* **2021**, *58*, 373–387. [CrossRef]
31. Elguea-Aguinaco, Í.; Serrano-Muñoz, A.; Chrysostomou, D.; Inziarte-Hidalgo, I.; Bøgh, S.; Arana-Arexolaleiba, N. A review on reinforcement learning for contact-rich robotic manipulation tasks. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102517. [CrossRef]
32. Luo, J.; Yang, C.; Li, Q.; Wang, M. A Task Learning Mechanism for the Telerobots. *Int. J. Humanoid Robot.* **2019**, *16*, 1950009. [CrossRef]
33. Calinon, S.; Billard, A. A framework integrating statistical and social cues to teach a humanoid robot new skills. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop on Social Interaction with Intelligent Indoor Robots, Pasadena, CA, USA, 23–27 May 2008; number CONF.
34. Hormann, K.; Sukumar, N. *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*; CRC Press: Boca Raton, FL, USA, 2017.
35. Caccavale, R.; Saveriano, M.; Finzi, A.; Lee, D. Kinesthetic teaching and attentional supervision of structured tasks in human–robot interaction. *Auton. Robot.* **2019**, *43*, 1291–1307. [CrossRef]
36. Kuchenbecker, K.J.; Niemeyer, G. Induced master motion in force-reflecting teleoperation. *J. Dyn. Sys. Meas. Control* **2006**, *128*, 800–810. [CrossRef]
37. Niemeyer, G.; Preusche, C.; Stramigioli, S.; Lee, D. Telerobotics. In *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1085–1108.
38. Das, H.; Zak, H.; Kim, W.S.; Bejczy, A.K.; Schenker, P.S. Operator performance with alternative manual control modes in teleoperation. *Presence Teleoperators Virtual Environ.* **1992**, *1*, 201–218. [CrossRef]
39. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Force Control*; Springer: Berlin/Heidelberg, Germany, 2009.
40. Luh, J.; Fisher, W.; Paul, R. Joint torque control by a direct feedback for industrial robots. *IEEE Trans. Autom. Control* **1983**, *28*, 153–161. [CrossRef]
41. Kana, S.; Gurnani, J.; Ramanathan, V.; Turlapati, S.H.; Ariffin, M.Z.; Campolo, D. Fast kinematic re-calibration for industrial robot arms. *Sensors* **2022**, *22*, 2295. [CrossRef]
42. Bishop, C.M. Pattern recognition. *Mach. Learn.* **2006**, *128*, 9.
43. Waldron, S. Affine generalised barycentric coordinates. *Jaen J. Approx.* **2011**, *3*, 209–226.
44. Calinon, S. A tutorial on task-parameterized movement learning and retrieval. *Intell. Serv. Robot.* **2015**, *9*, 1–29. [CrossRef]
45. Huang, Y.; Silverio, J.; Rozo, L.; Caldwell, D.G. Generalized Task-Parameterized Skill Learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018. [CrossRef]
46. Whitney, D.E. Quasi-Static Assembly of Compliantly Supported Rigid Parts. *J. Dyn. Syst. Meas. Control* **1982**, *104*, 65–77. [CrossRef]
47. Zhang, Y.; Lu, H.; Pham, D.T.; Wang, Y.; Qu, M.; Lim, J.; Su, S. Peg–hole disassembly using active compliance. *R. Soc. Open Sci.* **2019**, *6*, 190476. [CrossRef] [PubMed]

48. Beltran-Hernandez, C.C.; Petit, D.; Ramirez-Alpizar, I.G.; Harada, K. Variable Compliance Control for Robotic Peg-in-Hole Assembly: A Deep-Reinforcement-Learning Approach. *Appl. Sci.* **2020**, *10*, 6923. [CrossRef]
49. Suomalainen, M.; Kyrki, V. Learning 6-D compliant motion primitives from demonstration. *arXiv* **2018**, arXiv:1809.01561.

**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.