# Industrial Robot: An International Journal

Robot Programming by Demonstration using Teleoperation through Imitation

Abhishek Jha Shital S Chiddarwar

## Article information:

To cite this document:
Abhishek Jha Shital S Chiddarwar , (2017)," Robot Programming by Demonstration using Teleoperation through Imitation ", Industrial Robot: An International Journal, Vol. 44 Iss 2 pp. -
Permanent link to this document:
http://dx.doi.org/10.1108/IR-03-2016-0114

# Robot Programming by Demonstration using Teleoperation through Imitation

## 1. Introduction

Robotic research is finding successful implementation in core industries as well as science industries. The new applications of the robotics involve the cluster of various technologies existing in multiple disciplines (Andulkar *et al.,* 2015; Andulkar and Chiddarwar, 2015). The diversity of application demands more complex skills and behavior such as grasping, handling and coordination. These are hard to program explicitly to the robot using conventional methods. Robot Programming by Demonstration (PbD) or Learning from Demonstration (LfD) are the upcoming techniques. These techniques help to build ability in the robot to communicate and learn from human teacher systems or other robots. The end user can build new skill behavior and motions in the robot from the beginning. The motive is to have the robot with an ability to learn from the performed demonstrations regardless of end user's knowledge of robotics or programming.

LfD or PbD based planner essentially learns new skills or motion behaviors through the guidance of the human teacher. During this process, four vital questions which a typical trajectory planner should answer are, What to imitate? How to imitate? When to imitate? and Whom to imitate? (Nehaniv and Dautenhahn, 2007).The researchers have tried to give appropriate answers to the first two questions, but later two questions have subjective solutions hence they cannot be generalized. To answer the first question, the human teacher shows the specifics of the new skills or motion behavior which the robot needs to learn through its perception. The robot extracts information from these demonstrations using an internal trajectory planner (Argall *et al*., 2009; Billard *et al.,* 2008; Calinon *et al.,* 2010; Pardowitz *et al.,* 2007). The answer to the second question is the goal-driven programming rather than simple point to point programming. The use of trajectories for transfer of knowledge between a human demonstrator and robot can be an alternate approach.

In the current work, acquiring skills from human demonstration is done at the trajectory level as it contains rich information (Aleotti and Caselli, 2006). This information includes all vital execution level kinematics data like velocities, accelerations and workcell information like task space constraints, obstacle avoidance requirements, grasping a target object, and pose of the robot end-effector for approaching the work piece (Billard *et al.,* 2008; Calinon *et al.,* 2010; Vakanski *et al*., 2012). Learning a skill at the trajectory level requires extraction of the generalized representation of the demonstrations. For this, several methods like Hidden Markov Model (Yang *et al.,* 1997), Gaussian Mixture Model (Vukovic *et al.,* 2015), Conditional random field (Lafferty *et al*., 2001), Artificial Neural Network (Bullock and Grossberg,1988), Fuzzy neuro reinforcement learning (Obayashi *et al.,*2008, Watanabe and Wada,2008) and hand state approach (Skoglund *et al.,* 2010) are reported in the literature. All these approaches are probabilistic in nature. Even though these methods show very good learning from the demonstrations and the generalization of the workspace abilities, the probabilistic approach normalizes the information recorded from the demonstrations. The most of these methods are constrained to use recorded raw data instead of the reliable data collected from the teleoperation.

The essential assumption for the probabilistic methods is that the human teacher always demonstrates in the best working condition and performs the best demonstration. Due to this, there is no scope to correct the demonstrations as per the response of the robot. On the other hand, in the teleoperation, the demonstrator can modify the demonstrations if certain configurations are unachievable for the robot due to inevitable variations between physical epitome of the human and the robot. For this purpose, a trajectory planner blended with incremental inverse kinematics, singularity avoidance and framed within the kinematic constraints of the embodiments can be an efficient tool. This enables trajectory planning ahead of the current state unlike conventional PbD approaches.

The first attempt of using incremental inverse kinematics was reported by (Hersch and Billard, 2008). They combined the two controllers in parallel, in which, one controller acted in the joint space and other in the Cartesian space. In our approach, a human demonstrator guides the robot to grasp the object and complete the pick and place operation. This is achieved by combining the demonstrated path with the robot's own plan constrained by the kinematic and workcell constraints. According to (Ijspeert *et al.,* 2002), most of the approaches reported in literature use either the joint space or the Cartesian space without bothering for non-linear relationship between these two spaces. On the contrary, our approach has given due consideration to this issue.

Additionally, in this work, the skill transfer is done from the human teacher to the industrial robot where correspondence is critical as compared to the scenario with skill transfer from human to a humanoid as described by (Ijspeert *et al.,* 2002; Hersch and Billard, 2008). Moreover, it is shown here that the skill transfer from human teacher to industrial robot is possible despite inherent variation in their morphologies. Furthermore, teleoperation bypasses the correspondence problem which naturally arises as a critical issue in the PbD/LfD framework. As clearly indicated by (Narayanan *et al.,* 2011), "The human interaction with the environment is more intuitive because of the inherent situational awareness to look ahead and foresee obstacles in the executing motor decisions."

The main motivation behind this work is to develop a self-sufficient system for robot programming by utilizing the natural human motion. This work presents a simple and inexpensive architecture for robot programming by exploiting the implicit knowledge of the human operator to demonstrate through examples. To illustrate the approach, a pick and place operation is performed under different scenarios where the goal-directed human demonstrations are reproduced by the robot. The contributions of this work are as follows:

1) A novel incremental motion planner based on imitation through teleoperation is introduced to map the human hand demonstrations to the robot trajectories.

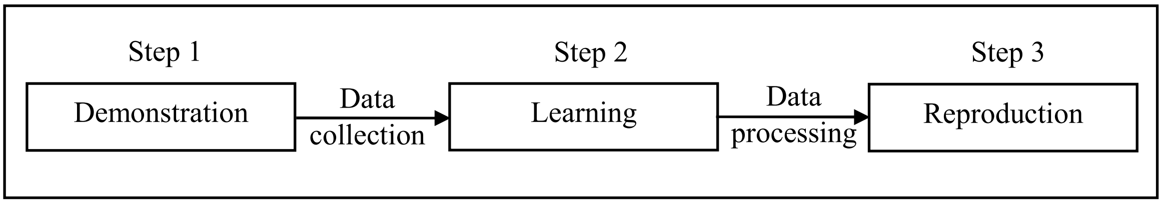
2) The concept of real time NURBS fitting and optimization is used to generate the optimal trajectory for the reproduction.
3) The combination of teleoperation and incremental motion kinematics provides a robust approach to address the correspondence problem. The experiments reveal how effectively the robot can generalize demonstrations despite unavoidable fundamental differences between human arm (7 degrees of freedom) and SCORBOT (5 degrees of freedom).

The rest of this paper is organized as follows. Section 2 describes the proposed planner based on teleoperation through imitation. The experimental setup and the procedures are reported in the section 3. The section 4 describes the results and discussions of this work followed by the concluding remarks in the section 5.

## 2. Outline of the teleoperation through imitation (TTI) planner

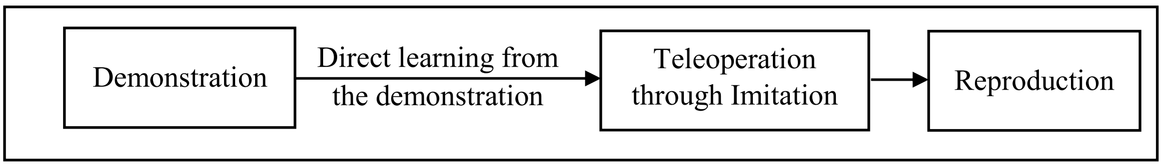
The path or trajectory imitation is considered as an effective medium to teach a robot from the task demonstrations. This low-level skill learning technique provides the implicit information about the task to be performed and simplifies the robot programming. The conventional PbD frameworks make use of architecture as shown in Figure 1. This essentially requires separate demonstration and the learning phase, and eventually the reliability of the planner is governed by the correctness of the collected data.

**Figure 1** Conventional PbD approach




To avoid this dependency and to make the proposed planner self-sufficient and robust, the real time motion tracking of the demonstrator is integrated with the teleoperation of the robot for the direct learning from the demonstrations as shown in Figure 2. It operates in two phases: (1) Trajectory generation, and (2) Reproduction of the generated trajectory.

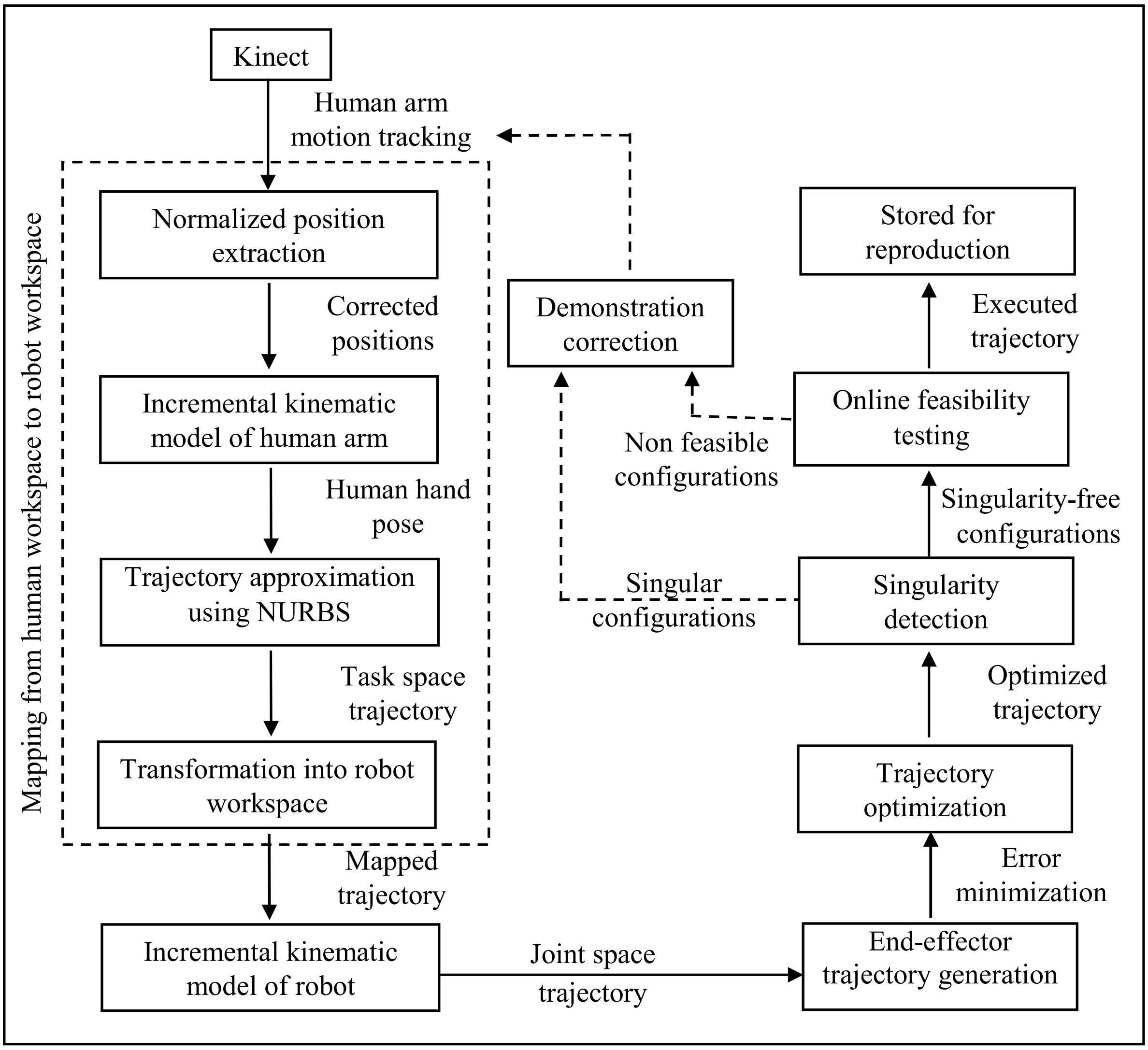
**Figure 2** Proposed approach




### 2.1 Phase I: Trajectory generation

The overview of the strategy developed for trajectory generation phase is given in Figure 3.

**Figure 3** Trajectory generation scheme of the TTI planner




The human arm Cartesian configurations captured by Kinect sensor are mapped to robot end-effector pose (position and orientation) with the help of methodology shown in Figure 4. The human arm home position and orientation in Cartesian ($X_0$, $Y_0$, $Z_0$, $Roll_0$, $Pitch_0$, $Yaw_0$) as well as joint angle space ($q_0 = (\theta_i)_0$ where i=1,2,3,4,5,6,7 ) is known a priori. The trajectory generation phase requires coordinate frames of the Kinect sensor, human arm and the Robot arm. These coordinate frames are assigned as shown in Figure 5.

The Kinect sensor senses the Cartesian coordinates of the shoulder, elbow and hand of the human arm. For teleoperation purpose, Cartesian coordinates of hand with respect to shoulder of the demonstrator are required. These coordinates are necessarily to be corrected in order to eliminate the probable motion tracking errors (Rosado *et al.,* 2014).

**Figure 4** Developed mapping strategy for the TTI planner

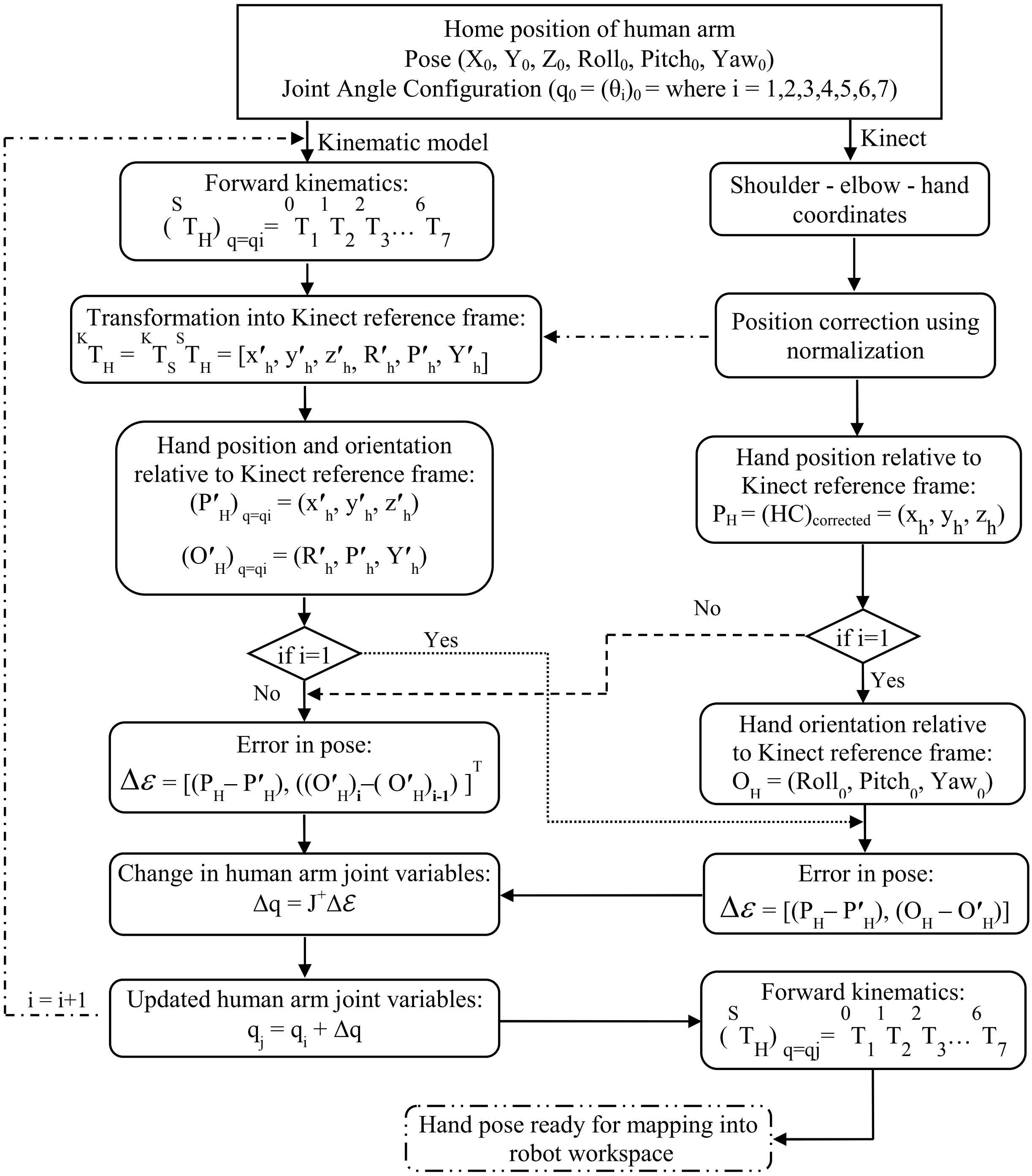
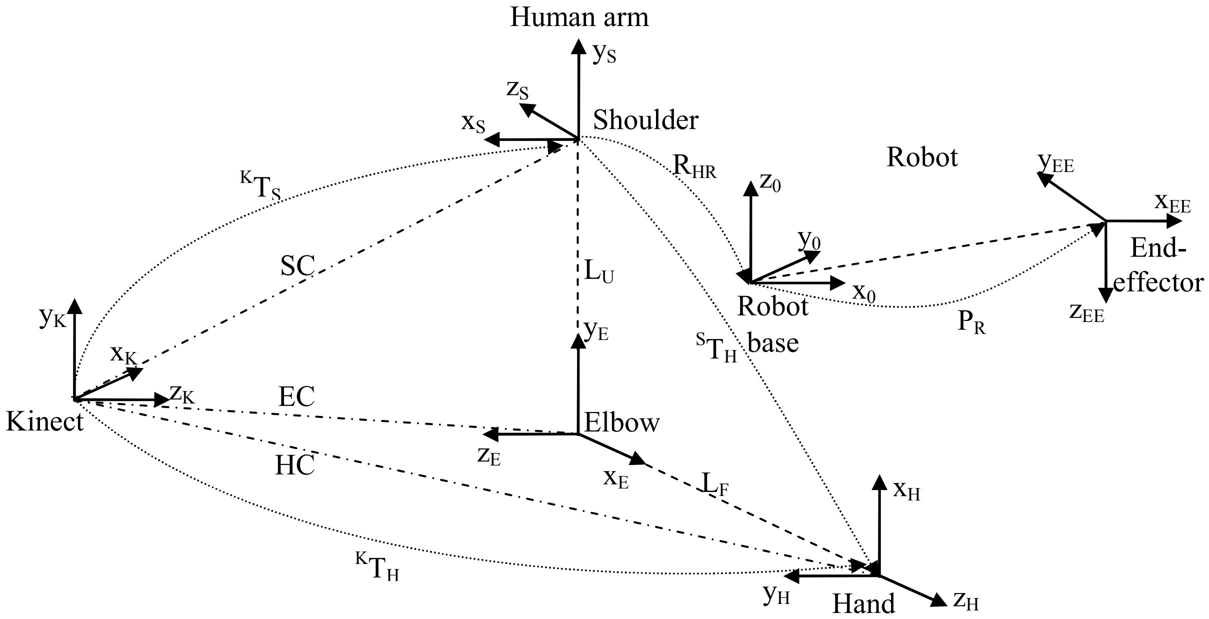


**Figure 5** Geometric representation of mapping




**Remark 1: Correction of hand coordinates**

The correction is necessary to address the variability of human arm dimensions. For this purpose, normal vectors between the shoulder and the elbow ($N_{SE}$) and the elbow and the hand ($N_{EH}$) are computed as shown in equation (1).

$$N_{SE} = \frac{SC - EC}{|SC - EC|}$$
$$N_{EH} = \frac{EC - HC}{|EC - HC|} \tag{1}$$

Where, *SC*, *EC*, and *HC* are the Cartesian coordinates of shoulder, elbow and hand of the human arm. Using these relations and the actual upper arm and forearm lengths, the corrected elbow coordinates $(EC)_{corrected}$ and hand coordinates $(HC)_{corrected}$ are computed as shown in equation (2).

$$(EC)_{corrected} = SC + N_{SE} L_U$$
$$(HC)_{corrected} = (EC)_{corrected} + N_{EH} L_F \tag{2}$$

Where, $L_U$ and $L_F$ are the upper and forearm lengths. The corrected position of hand ($P_H = (HC)_{corrected} = (x_h, y_h, z_h)$) along with the initial orientation of hand ($O_H = (Roll_0, Pitch_0, Yaw_0)$) describes the hand pose at starting position relative to the Kinect coordinate frame. Further, a priori known home position of the human arm is substituted in the forward kinematic relationship of human arm to obtain a transformation matrix ${}^{S}T_{H}$ which gives pose of hand with respect to the shoulder of the demonstrator. Afterward, the transformation matrix ${}^{S}T_{H}$ is transformed in the Kinect coordinate frame as ${}^{K}T_{H} = {}^{K}T_{S} \cdot {}^{S}T_{H}$, where, ${}^{K}T_{S}$ is the homogeneous transformation matrix relating the shoulder and the Kinect coordinate frames. The ${}^{K}T_{H}$ is converted into [$x'_{h}$, $y'_{h}$, $z'_{h}$, $Roll'_{h}$, $Pitch'_{h}$, $Yaw'_{h}$], where position is $(P'_{H}) = (x'_{h}, y'_{h}, z'_{h})$ and orientation is $(O'_{H}) = (Roll'_{h}, Pitch'_{h}, Yaw'_{h})$. For first instance, i.e. i=1, error in the obtained pose is computed as $\Delta\varepsilon = [(P_{H} - P_{H}^{'}), (O_{H} - O_{H}^{'})]^{T}$, and for the next instance, i.e. i=i+1, $\Delta\varepsilon = [(P_{H} - P_{H}^{'}), ((O_{H}^{'})_{i} - (O_{H}^{'})_{i-1})]^{T}$. The change in human arm joint configuration is computed using Jacobian pseudo inverse as $\Delta q = J^{+}\Delta\varepsilon$.

**Remark 2: Jacobian Pseudo Inverse**

For the home position of human arm, the difference between the consecutive hand poses relative to the Kinect reference frame is computed as the pose error ($\Delta\varepsilon$). It ensures the incremental motion of the human hand in task space. To determine the corresponding incremental motion in the joint space of human arm, the concept of Jacobian pseudo inverse is used (Chevallereau and Khalil, 1987; Buss, 2004; Flash *et al.,* 2013) as $\Delta q = J^{+}\Delta\varepsilon$ , where, $\Delta q$ represents the incremental change in the current joint space variables and $\Delta\varepsilon$ is the error in pose for two consecutive configurations of the human hand in task space. The term $J^{+}$ denotes the pseudo inverse of the human arm Jacobian matrix $J(q) = \left(\frac{\partial f_i}{\partial q_j}\right)_{i,j}$. It is obtained by taking the time derivative of the kinematics equations, which relates the joint rates to the linear and angular velocity of the end-effector. This also indicates the incremental motion in joint as well as Cartesian space.

Further, the human arm joint variables are updated as $q_{i+1} = q_{i} + \Delta q$. These are substituted in forward kinematics of the human hand to obtain ${}^{S}T_{H}$. This is the updated pose of human hand ready to map into the robot workspace. The set of such discrete updated poses form the human hand trajectory to be mapped to robot workspace. Hence, to bring the continuity in it, the trajectory thus obtained is fitted with Non Uniform Rational B-Splines (NURBS) as they are considered to be the best parametric curve for path planning applications and provide a flexible way to model the trajectory (Aleotti *et al.,* 2005). Moreover, it gives compact, noise free and the best approximation to its original counterpart. The generalized representation of NURBS curves is as given by equation (3)

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u)\, w_i} \qquad a \le u \le b \tag{3}$$

where, $P_i$ are the control points, $w_i$ are the weights and $N_{i,p}(u)$ are the $p$th degree B-Spline basis functions defined on the non-periodic and non-uniform knot vector

$$U = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b \dots b\} \quad (4)$$

The normalized B-Spline basis functions for degree $p$ polynomial are defined recursively as

$$N_{i,o}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (6)$$

The trajectory obtained after NURBS fitting corresponds to the demonstrator's workspace. This is further transformed into the robot's frame of reference using mapping shown in Figure 5. The relationship $P_R = R_{HR} P_{HA}$ is utilized for mapping where $P_R$ is the pose in the robot's reference frame; $R_{HR}$ is the homogeneous transformation matrix which maps the pose of the human hand to the robot workspace, and $P_{HA}$ is the matrix representing the pose of the human hand estimated using NURBS fitting. Eventually, the transformation $R_{HR}$ establishes mapping between the two workspaces by considering the morphological differences and workspace limitations of the robot. The Cartesian trajectory in the form of $P_R$ is converted into robot joint angle trajectory using relationship $\Delta\theta = J_R^+ \Delta P_R$, where $J_R^+$ the pseudo inverse of the robot Jacobian matrix and $\Delta P_R$ is is the task space error between the mapped and the current pose of the robot. During this mapping process, the morphological difference between two embodiments i.e. human arm and robot is responsible for the error, $e(\theta)$, between the mapped pose and the pose obtained from the incremental inverse kinematic model of the robot. To make the mapping robust and accurate, this error needs to be minimized. For this purpose, a nonlinear multivariable constrained optimization problem is formulated and constrained for the joint angle limitations of the robot. The optimized set of joint variables of the robot obtained from this process best represents end-effector pose in relation with the mapped pose. The imposed constraints ensure that the optimized joint space variables lie within the kinematic constraints and reachable workspace of the robot. Each optimized set of joint space variables of the robot is first checked for the singularity. Then it is further subjected to the feasibility check. If it is singularity free and feasible then saved for the reproduction phase else the demonstrator modifies the demonstration until singularity free and feasible configuration is obtained. The set of such configurations is nothing but the mapped trajectory to be utilized during the reproduction phase.

### 2.2 Phase II: Reproduction of the generated trajectory

The multiple demonstrations of the same task are necessary to minimize the ambiguity in the demonstrated actions due to dexterous nature of the human arm. Moreover, the inconsistency associated with the human arm motion makes the multiple demonstrations of the same task to differ from each other. Due to multiple demonstrations, multiple trajectories come in the reproduction phase. This is the final phase in which the trajectory obtained from the planner is given input to the robot for task execution. The selection of the best trajectory for reproduction can be done on the basis of a metric which is governed by the application. The metric is defined on the length of path to be traversed by the robot and torque required for the task execution. The path length provides a definite measure for the performance assessment in path planning applications (Chiddarwar and Babu, 2011), whereas the joint torques minimization is customarily used for optimal trajectory selection (Luo *et al.*, 2015). For computation of the torque, the Newton-Euler dynamic model of the robot was utilized. However, consideration of the robot dynamics model during demonstrations is computationally complex and hence neglected during demonstrations. In such a case, the evaluation metric based on the path length and the joint torque assesses the trajectories and provides a rationale for the appropriate trajectory selection.

The TTI planner proposed here has the direct learning ability, and it helps to generalize from multiple demonstrations. Here, the demonstrator can provide information to the robot about the goodness of the learned motion because of the bidirectional communication between the robot and the demonstrator. The feasibility of the trajectory obtained using TTI planner is checked in the teleoperation scenario where a demonstrator guides the robot in a real time environment. For imitation of the demonstrated trajectory via teleoperation, an industrial manipulator SCORBOT ER-4u with five degrees of freedom is used. The inclusion of TTI planner in the PbD architecture offered many advantages. Firstly, the direct trajectory learning via teleoperation simplified the robot programming, since it is much easier to demonstrate the trajectory instead of programming it offline. The human demonstrated trajectories encoded vital information, which is difficult to specify by other means. Such information includes the preferential direction of movement to approach object, the regions of workspace to be avoided and the obstacle-free path (Aleotti and Caselli, 2006). Unlike pre-recorded demonstrations, the teleoperation process enabled the user to correct or adjust his actions for non-feasible regions and/or configurations of the robot, during the task demonstration itself. The experiments performed in support of these observations are discussed in the next section.

## 3. Experiments

The experimental setup for the object manipulation task comprised of a Kinect based motion capture system, SCORBOT ER-4u manipulator and a human demonstrator. The layout of the experimental setup is shown in Figure 6. For the experiments, the human demonstrations of the task trajectory were captured using a Kinect based motion capture system. The Kinect sensor is capable of generating color, depth and skeleton frames for the real time motion tracking applications. It provides an inexpensive and reliable alternative to the other motion tracking systems (Han *et al.,* 2013).

For human arm motion tracking, the skeleton tracking feature of the Kinect was accessed using the image acquisition toolbox of the MATLAB. This feature provided the position data of the human arm with a frame rate of 30 fps. The Kinect camera was placed 2 m apart in front of the demonstrator. The starting postures of the demonstrator were initially calibrated with respect to the Kinect (Figure 7) and kept fixed at the start of demonstrations.

**Figure 6** System architecture for the TTI planner

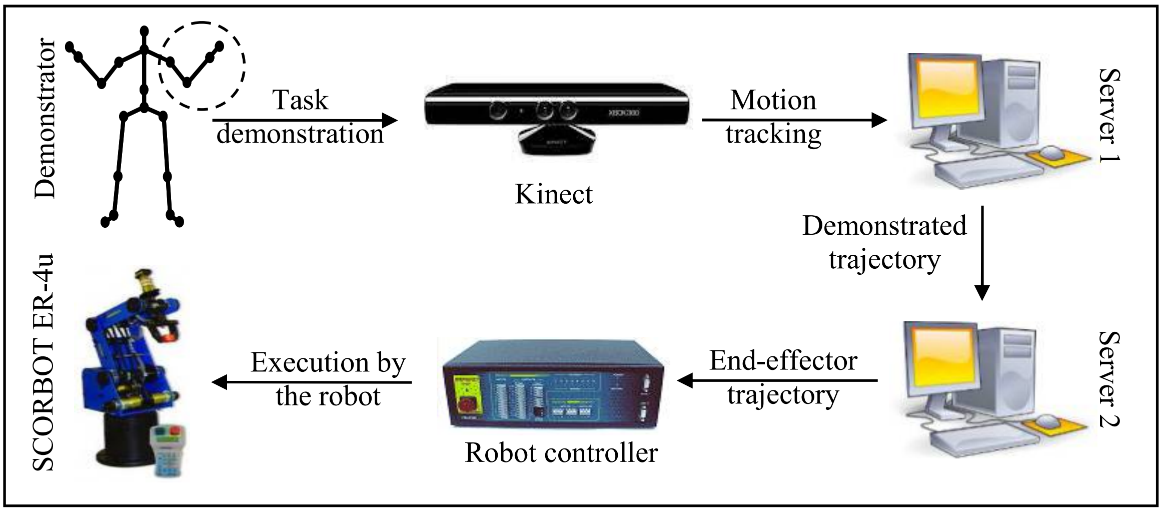
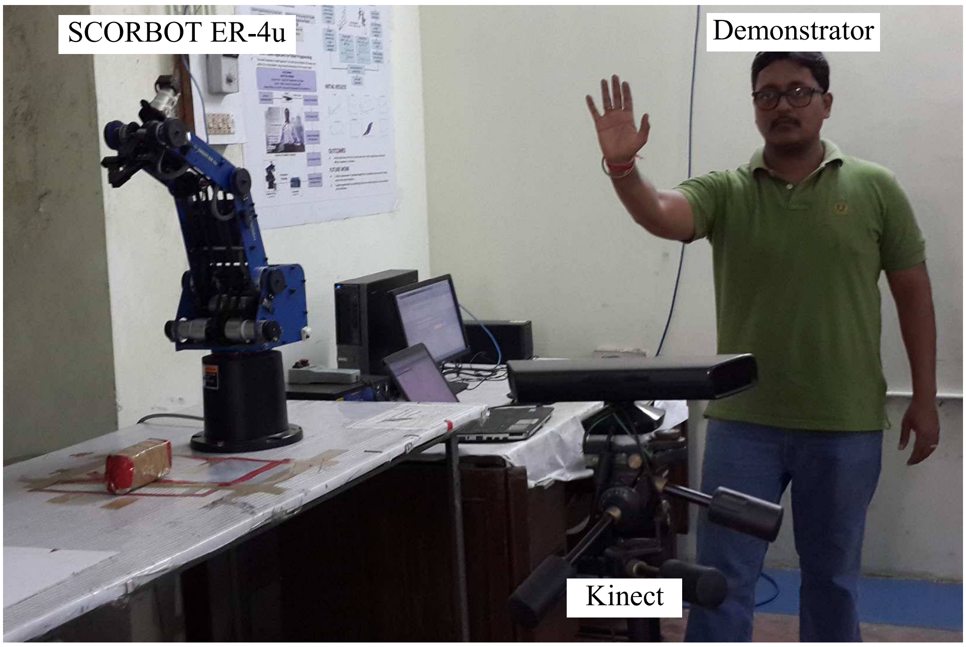



**Figure 7** Starting posture for the demonstration




To imitate the human arm motion, the five degrees of freedom vertical articulated manipulator SCORBOT ER-4u with two finger gripper was used. Compared to conventional six degrees of freedom robots, it is easier to control the five degrees of freedom structure of the manipulator, and it is also suitable for the repetitive nature of the object manipulation task. The two degrees of freedom wrist of the manipulator enabled it to grasp the object within its reachable workspace. The gripper actuation is achieved by modeling the distance to the target as the indicator for the opening-closing of the gripper. The distance based function thus obtained enabled the gripper to grasp and release the object at the specified positions.

During experiments, the human arm position data obtained from the motion capture system were processed in the server 1 using the TTI planner to generate a human hand trajectory. This hand trajectory was serially transferred to server 2 for generation of the robot trajectory using TTI planner. Both the servers worked on the Matlab environment, and the serial communication ensured the real-time data transfer between them. The robot trajectory generated by the server 2 was given to the connected robot controller for task execution by the robot. The real-time communication between the systems defined the teleoperation architecture and provided real time control over the robot motion. During experiments, a time lag of 1-2.5 seconds was observed between the demonstrator and robot movements. This lag is essentially due to the computational time required for the convergence of the optimization problem, and the response time of the robot controller. Being local to demonstration phase, this lag never affects the reproduction of the trajectory by the robot.

To perform the pick and place operation, the demonstrator teleoperated the robot from the pick position to goal position. Multiple demonstrations for the same task were given to the robot. The experiment was repeated for different pick and place positions in the robot workspace, distinct starting positions of the robot and different starting postures of the demonstrator. The results of the experiments performed for the object manipulation task, under different operational conditions are reported in the next section.



## 4. Experimental Results and Discussions

For the designed set of experiments, the TTI planner is evaluated by considering the following aspects:

1. Feasibility of the developed planner for PbD framework.
2. Effectiveness of the planner for generalization of the robot workspace.
3. Skill improvement of the robot.
4. Assessment of the developed mapping strategy.

### 4.1 Feasibility of the TTI planner for the PbD framework

In this set of experiments, the pick and the goal positions of the object were pre-defined. Moreover, the starting pose of the robot and the demonstrator was also fixed. The demonstrator teleoperated the robot for execution of the pick and place task as shown in Figure 8.

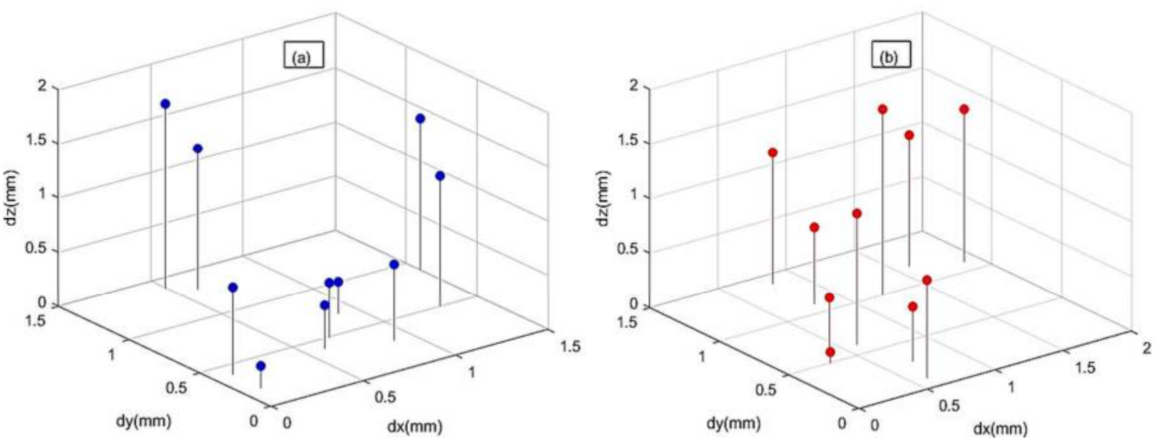
**Figure 8** Task demonstration to the robot

For this experiment, total 18 demonstrations were performed out of which 2 were discarded due to occlusion and insufficient data generation. For the remaining 16 demonstrations, 10 attempts were successful where teleoperation for the object manipulation was achieved whereas 6 demonstrations failed. The trajectories generated for the 10 successful demonstrations can readily be used for reproduction as they satisfy the kinematic and the workspace constraints imposed in the TTI planner. The 6 failed demonstrations were those where the demonstrator was not able to complete the object manipulation task due to inconsistency in the demonstrations. The inconsistency in the demonstrations may be due to the involuntary hand movements and variability associated with the human arm. The inconsistent demonstrations cause the robot to deviate from the desired path and result in a non-reachable or non-feasible configuration of the end-effector. Since, every demonstration starts from a predefined posture of the human hand, the planner generates a trajectory as per the movement of the hand during task demonstration. It further causes the Jacobian based inverse kinematic solution of the robot to be unstable and, ultimately leads to the failure of demonstration. Being five degrees of freedom structure, the robot is more susceptible to the variations in the input motion profile rather than the human arm model. This can be a shortcoming of the demonstration strategy adopted here. Another reason for the demonstration failure is the occlusion of the skeletal joints during motion tracking. During demonstration, the occlusion of the skeleton joints may result in erroneous joint positions. Due to real-time interaction between the demonstrator and the robot, the erroneous positions make the robot to diverge from the desired path. This phenomenon requires attention during demonstration, especially near the pick and the goal positions for successful grasp and release of the object. These aspects can be taken care by imparting proper training to the demonstrator, changing the home position of the robot and providing more demonstrations for the task.

However, changing the starting location of the robot is not advisable because robots are programmed for starting its work from the home position. On the contrary, the number of demonstrations can be increased at the stake of the computational load. In view of above aspects, from this experiment it can be inferred that the TTI planner has a good ability to learn directly from the demonstrations as it shows 62.5% of the demonstrated trajectories were successfully executed by the robot.

For the ten trajectories generated by the planner, variations in the position of the end-effector from the predefined pick and the goal positions are shown in Figure 9. The average variation for the ten trajectories is observed to be 0.72 mm, 0.66 mm and 0.85 mm along the x, y and z directions respectively. This indicates that the positioning achieved during teleoperation is accurate enough for the object manipulation task, and it justifies the feasibility of the trajectory planner.

**Figure 9** Variation in end-effector position (a) pick position (b) goal position




### 4.1.1 Selection of the best trajectory for reproduction

All the successfully generated trajectories via teleoperation differ from each other in terms of the path length, form of the trajectory and efforts required to execute the trajectory due to variability present in the human motion. For optimal trajectory selection out of multiple demonstrations, various methods like trajectory averaging (Lee and Nakamura 2006), trajectory clustering (Aleotti and Caselli, 2006) or log likelihood based method (Vakanski *et al*., 2012) are reported. These approaches further modify the recorded trajectories to define an optimal one. In this work, the selection of the best trajectory is done based on the minimum path length and the minimum joint torques to preserve the original content of the learned trajectory. Table 1 shows the path lengths, the mean joint torques and the execution time for reproduction for the 10 successfully generated trajectories during the experiment.

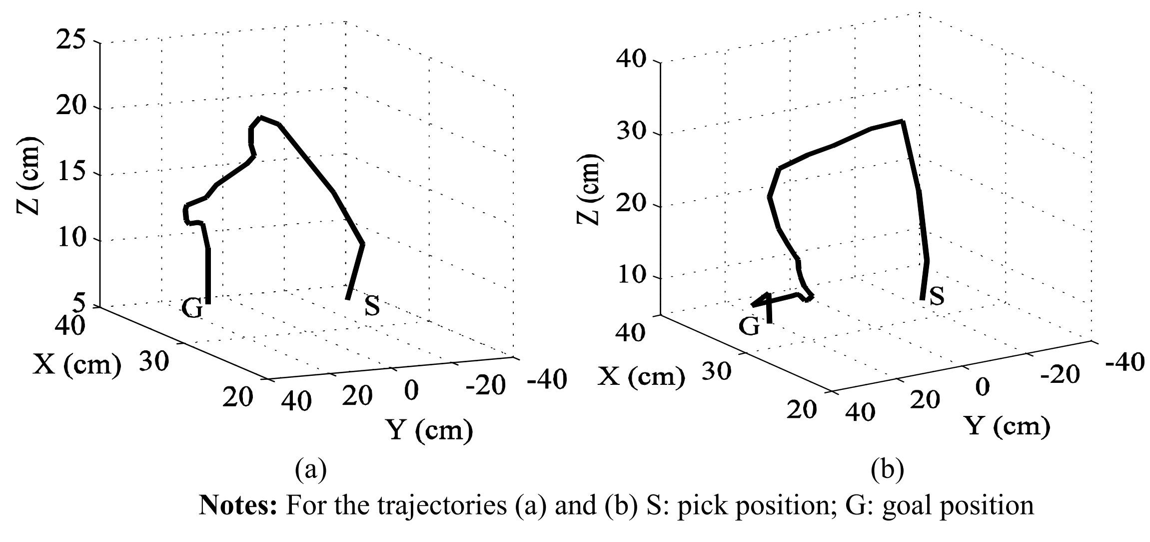
It can be seen from Table 1 that, as per the minimum path length criterion, the trajectory obtained for the experiment 1 is considered to be the best. For this trajectory, minimum execution time is observed during the reproduction phase. The minimum path length is directly responsible for the minimum cycle time during completion of the task by robot. This is a desirable criterion for the lean manufacturing systems and the rapid response manufacturing where emphasis is on quick and efficient work cycles.

**Table 1** Path length, mean joint torques and execution time for the generated trajectories

| Exp. No. | Path Length (cm) | Joint torques | | | | Execution time (seconds) |
|---|---|---|---|---|---|---|
| | | $\tau_1$ (Nm) | $\tau_2$ (Nm) | $\tau_3$ (Nm) | $\tau_4$ (Nm) | |
| 1 | 73.100 | 0.142 | 16.541 | 15.321 | 2.213 | 17.94 |
| 2 | 98.860 | 0.570 | 20.031 | 11.690 | 0.230 | 23.50 |
| 3 | 80.710 | 0.485 | 19.767 | 6.488 | 0.316 | 19.16 |
| 4 | 109.730 | 0.452 | 20.338 | 13.931 | 1.317 | 24.87 |
| 5 | 138.220 | 0.599 | 20.828 | 11.922 | 2.908 | 55.97 |
| 6 | 122.880 | 0.813 | 21.303 | 8.157 | 0.786 | 29.07 |
| 7 | 151.340 | 0.749 | 22.824 | 13.558 | 2.798 | 60.38 |
| 8 | 108.650 | 0.594 | 15.989 | 4.228 | 2.864 | 25.89 |
| 9 | 88.693 | 0.778 | 21.493 | 13.259 | 2.947 | 21.88 |
| 10 | 102.967 | 0.792 | 20.156 | 11.501 | 2.793 | 22.12 |

On the other side, the trajectory generated in experiment number 8 seems to be best as per the minimum joint torque criterion. This is a desirable criterion when minimum energy consumption is required as in the green manufacturing systems. Based on the two criteria discussed above, the best trajectories are shown in Figure 10 and can be utilized for reproduction. The considered criteria provide additional flexibility to the user, as the choice of trajectory selection specifically depends on his intention.

**Figure 10** Best trajectories obtained (a) minimum path length (b) minimum joint torques




(a) (b)

**Notes:** For the trajectories (a) and (b) S: pick position; G: goal position

### 4.2 Generalization of the robot workspace

In this experiment, the trajectory planner was tested on how well it adapts to the changes made in the operating conditions. The test analyzed the ability of the planner to address the correspondence problem by characterizing the different set of configurations that the robot can adapt within the reachable workspace. For this purpose, three different sets of experiments for the pick and place operations were performed with the different operating conditions in each case. The trajectories obtained for each case are shown in Figure 11.

Case 1: For this set of experiment, the starting position of the robot was fixed. Three different sets of locations in the robot workspace were used as the pick and goal positions as shown in Table 2.

**Table 2** Experiment statistics for the case 1

| Expt. No | Pick Position | | | Goal Position | | | Demonstrations | | |
|---|---|---|---|---|---|---|---|---|---|
| | x (cm) | y (cm) | z (cm) | x (cm) | y (cm) | z (cm) | Performed | Successful | Failed |
| 1 | 17.71 | 30.58 | 5.05 | 33.63 | -21.82 | 5.05 | 8 | 5 | 3 |
| 2 | 33.63 | -21.82 | 5.05 | 39.58 | 3.94 | 11.35 | 10 | 5 | 5 |
| 3 | 33.63 | -21.82 | 5.05 | 17.71 | 30.58 | 5.05 | 6 | 5 | 1 |
| Total | | | | | | | 24 | 15 | 9 |

During experiments, the demonstrator successfully teleoperated the robot for the different locations in the robot workspace and completed the manipulation task. Figure 11((a) - (c)) shows the trajectories obtained for the three locations within the robot workspace. However, different numbers of demonstrations were required for each location for successful execution of the task.

Case 2: In this experiment, the starting position of the robot was different for each demonstration. The different starting positions were defined by employing different home positions of the robot. The object was placed at a fixed position in the robot workspace. The five random home positions used during the experiment are shown in Table 3.

**Table 3** Different home positions used for the robot

| Position | x (cm) | y (cm) | z (cm) |
|---|---|---|---|
| Pick position of the object | 33.63 | -21.82 | 5.05 |

| | | | |
|---|---|---|---|
| Place position of the object | 17.71 | 30.58 | 5.05 |
| Robot home position 1 | 0.00 | 17.00 | 50.00 |
| Robot home position 2 | 60.31 | 0.00 | 34.90 |
| Robot home position 3 | 17.25 | 0.00 | 49.63 |
| Robot home position 4 | 8.62 | -14.94 | 49.63 |
| Robot home position 5 | 4.46 | -16.66 | 49.00 |

In the experiment, it took seven demonstrations to successfully execute the task five times. In this case, the reaching motion from the start point of the end-effector to the pick position is different for each experiment as shown in Figure 11((d) – (h)). It can be observed from the Table 3 that the change in starting position of the robot does not affect the planner, and the robot successfully executed the task with five different initial configurations.

Case 3: For this experiment, the demonstrator altered the starting posture of the hand for each demonstration. For each posture, the initial position and orientation of the hand was different. The teleoperation was performed with five different postures. During experiment, the end-effector successfully imitated the demonstrated trajectory and accomplished the task. In this case, ten demonstrations were required to generate five successfully executed trajectories. The start positions of the end-effector in the robot workspace corresponding to the different starting postures of the demonstrator are shown in the Table 4, and the trajectories obtained are shown in the Figure 11((i) – (m)).

**Table 4** End-effector start positions relative to different starting postures of the demonstrator

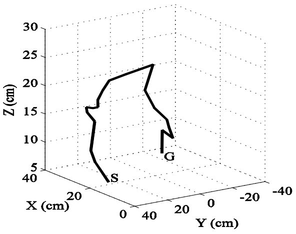
| Position | x (cm) | y (cm) | z (cm) |
|---|---|---|---|
| Pick position of the object | 33.63 | -21.82 | 5.05 |
| Place position of the object | 33.63 | 21.82 | 5.05 |
| End-effector start point 1 | 42.02 | -29.19 | 52.47 |
| End-effector start point 2 | 41.29 | -10.60 | 51.49 |
| End-effector start point 3 | 39.42 | -36.24 | 48.66 |
| End-effector start point 4 | 20.37 | -38.69 | 47.50 |
| End-effector start point 5 | -37.87 | -33.45 | 53.77 |

From the results obtained for case 3, it can be inferred that the TTI planner effectively incorporated the different starting postures of the demonstrator and generated robot trajectories. This fact is of importance for demonstrating a specific trajectory for a particular application. For the three cases presented here, it can be deduced that the TTI planner adapted the changes made in the operating conditions. The overall results for workspace generalization are summarized in the Table 5.

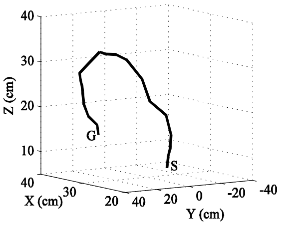
**Table 5** Experiments statistics for the workspace generalization

| Case No | Demonstrations Performed | Demonstrations Successful | Demonstrations Failed | Success rate (%) |
|---|---|---|---|---|
| 1 | 24 | 15 | 9 | 62.5 |
| 2 | 7 | 5 | 2 | 71.4 |
| 3 | 10 | 5 | 5 | 50.0 |
| Total | 41 | 25 | 16 | 60.9 |

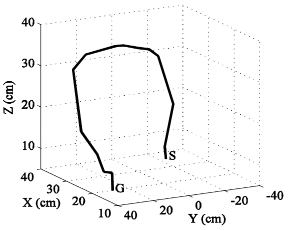
**Figure 11** Trajectories obtained during the experiments for workspace generalization



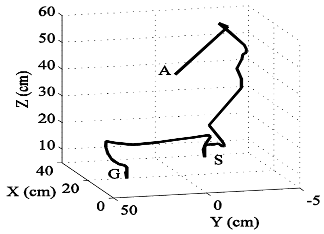
(a) Case 1: Experiment 1



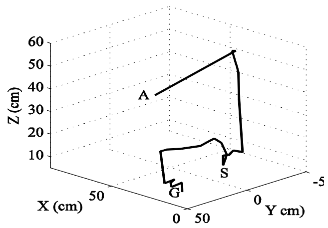
(b) Case 1: Experiment 2



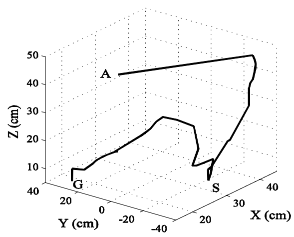
(c) Case 1: Experiment 3



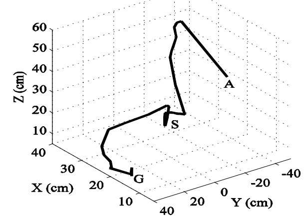
(d) Case 2: Experiment 1



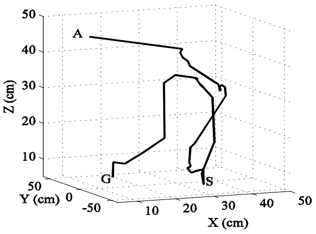
(e) Case 2: Experiment 2



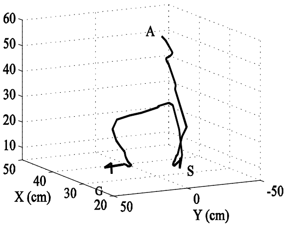
(f) Case 2: Experiment 3



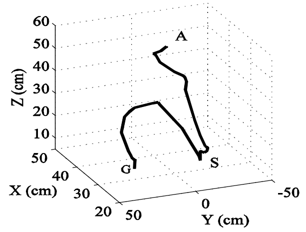
(g) Case 2: Experiment 4



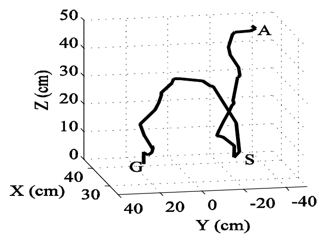
(h) Case 2: Experiment 5



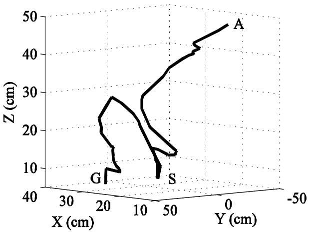
(i) Case 3: Experiment 1



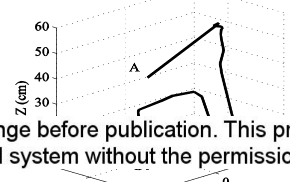
(j) Case 3: Experiment 2



(k) Case 3: Experiment 3



(l) Case 3: Experiment 4



The experiments performed for the workspace generalization tested ability of the TTI planner to tackle the correspondence problem. The case 1, where different locations were used as the target positions for the object manipulation reflected the reachable workspace utilization by the planner. The planner configured distinctive trajectories within the different regions of the reachable workspace by imitating in accordance with the input trajectories. The case 2, where random home positions of the robot were involved to reach a fixed location, outlined the independency of the planner to a particular state or configuration. The case 1 and 2 together constitute an important outcome for the more complex robot application like spray painting, where a specific trajectory is to be followed in a definite region of the robot workspace (Ferreira *et al*., 2014). In case 3, five different starting postures of the human demonstrator were used to execute the task. To obtain these postures, the initial position and orientation of the human hand were changed. In this case, the input reference trajectories to the planner differed in each case, and the planner successfully handled these variations during robot trajectory generation. Based on the obtained results from Table 5 it can be observed that, 60% of the demonstrations were successfully converted into executable trajectories by the planner. From the above discussion for all the set of experiments, it can be concluded that the planner characterizes well in the tested states to deal with the correspondence problem. Moreover, the failed demonstrations need to be addressed in the future work to explore how the teleoperation can be further improved from the demonstrator's view point to minimize the failure.

### 4.3 Skill improvement of the robot

From the experiments, it is evident that the robot can execute the demonstrated trajectory successfully. This experiment was performed to check the robustness of the planner to deal with the variation in the available information for trajectory generation. The experiment evaluated the ability of the planner to generate the executable trajectory for the robot with lesser information.

For this purpose, the four demonstrated trajectories for the pick and place operation were randomly selected from the experiments described in section 4.1. For generating the executable trajectory for the robot, the selected trajectories were given input to the planner in offline mode in the following manner: 1) 100% sampled points were given input to the planner, 2) 50% sampled points were given input to the planner, 3) 25% sampled points were given input to the planner. In each case, a reference robot trajectory was generated by the planner. The trajectories such generated were tested with the robot for pick and place operation. The pick and place positions were kept fixed for all the trajectories. In order to assess the generated trajectories, mean deviation of the trajectories from the target position was estimated.

The reference trajectories generated were successfully executed by the robot for each case. The reduction in the number of sampled points did not affect the execution of the object manipulation task. The quantitative results from this experiment indicated that the mean deviation is between 1-22 mm along all 3 axes. This means that the planner can generate an executable and feasible trajectory for the robot with the limited number of sampled points. Due to this ability, the proposed planer can be comfortably used to teach the robot in the cluttered and confined workspace like machine tending applications. During teleoperation, due to real time sampling of the points, the planner has more flexibility to generate a best-fit trajectory. This ability resembles human learning. Once taught the human being develops a skill and can work with limited information in the future. In the similar lines, the planner proposed here has a capability to learn a skill and improve it.

### 4.4 Assessment of the developed mapping strategy

For any PbD system, two desirable properties are the faster learning rate and the minimal tuning of the generated parameters. In this work, the teleoperation based direct learning approach provided a quick learning rate as no separate learning phase was involved in the approach and it fulfilled the former condition. In the second condition, the minimal tuning of parameters depends on the adopted mapping strategy. Here, the kinematics based transformation technique is utilized to map the demonstrated trajectory into the robot workspace. Moreover, the adapted mapping strategy is used to achieve the imitation of the human hand trajectory by the robot end-effector.

To test the mapping ability of the trajectory planner, the data set generated for 10 successful experiments described in section 4.1.1 are utilized. The data set generated during the experiment comprised of the number of sampled points obtained for a trajectory from the Kinect based motion tracking system, the number of points mapped to the robot workspace, and the number of points executed by the robot. To quantify the extent of imitation the Procrustes analysis (Goodall, 1991) is used. The Procrustes analysis is one of the popular statistical techniques used for shape feature matching in the domain of image analysis (Goodall, 1991) and can be used for similarity assessment of the trajectories (Jha *et al.,* 2015). Using this technique, a standardized dissimilarity measure is computed to assess the extent of geometric similarity between the trajectory mapped to the robot workspace and the trajectory followed by the robot end-effector. The computed dissimilarity measure represents the difference between the two shapes under comparison and for the maximum similarity between the shapes its value closer to zero is desirable. Further, the mean square errors between the mapped and the executed trajectories are also computed to judge the accuracy of the mapping in the robot workspace. The results obtained from the analysis are shown in the Table 6.

From the obtained results, it can be inferred that the mapped trajectories are reproduced quite closely during teleoperation. The mean dissimilarity measure is 0.0943, which reflects an average similarity of about 90% between the mapped and the executed trajectories. The close geometrical similarity between the two trajectories shows the effectiveness of the mapping strategy as the robot closely reproduced the demonstrated trajectory despite differences in the motion spaces and the kinematic structure. The mean execution rate observed during the experiments is 76%, which shows that 24% non-feasible points were rejected during execution. Moreover, the minimized errors between mapped and executed trajectories show the accuracy of the mapping strategy followed by the planner in the developed approach.

**Table 6** Parameters for assessment of the mapping strategy

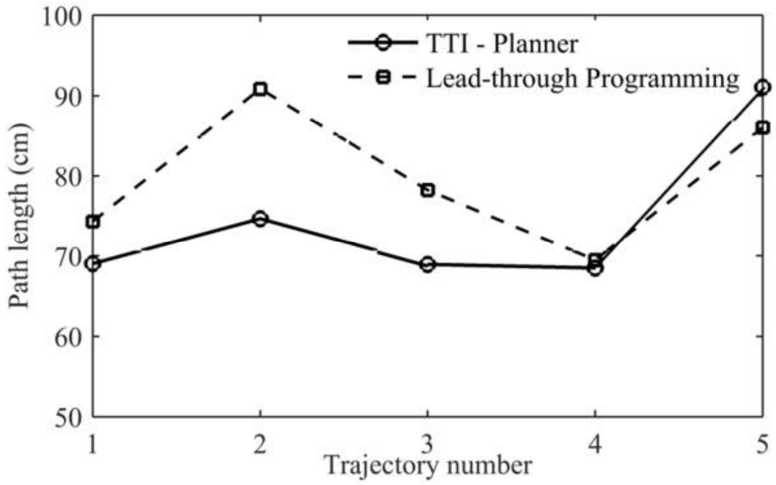
| Expt. No. | No. of Sampled Points | No. of Mapped Points | No. of Executed Points | Execution (%) | Dissimilarity Measure | Mean Squared Errors | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | $E_x$ (mm) | $E_y$ (mm) | $E_z$ (mm) |
| 1 | 349 | 113 | 113 | 100 | 0.0030 | 0.30 | 3.50 | 4.20 |
| 2 | 340 | 110 | 88 | 80.00 | 0.1161 | 0.60 | 3.50 | 0.80 |
| 3 | 337 | 109 | 81 | 74.31 | 0.0881 | 0.80 | 2.30 | 0.50 |
| 4 | 355 | 115 | 87 | 75.65 | 0.5755 | 0.80 | 9.00 | 3.40 |
| 5 | 364 | 118 | 87 | 73.72 | 0.0350 | 1.20 | 2.40 | 4.90 |
| 6 | 190 | 60 | 52 | 86.66 | 0.0205 | 0.10 | 0.40 | 0.70 |
| 7 | 244 | 78 | 67 | 85.89 | 0.0026 | 0.01 | 0.04 | 0.08 |
| 8 | 415 | 135 | 74 | 54.81 | 0.0431 | 0.60 | 6.90 | 4.60 |
| 9 | 442 | 144 | 47 | 32.63 | 0.0032 | 0.20 | 3.80 | 3.00 |
| 10 | 193 | 61 | 59 | 96.72 | 0.0563 | 0.20 | 1.30 | 0.50 |
| Average | | | | 76.04 | 0.0943 | 0.48 | 3.31 | 2.26 |

From these results, it can be concluded that the developed kinematic transformations based mapping strategy for teleoperation scenario worked well. The robot successfully imitated the demonstrated trajectory with close geometric similarity and minimized errors under the tested conditions.

## 4.5 Comparison with the existing industrial robot programming method

To compare the performance of the proposed approach, another set of experiments were performed by programming the robot for the same task using lead-through programming method. It is a commonly used method for teaching industrial robots using the teach pendent to guide the robot through the desired path. The relevant configurations of the end-effector are directly recorded by the robot controller. These recorded configurations are used to generate a robot program which can be later used for the task execution (Pan *et al.*, 2012). Similar procedure was adapted to generate a program for Case 1 experiment 2 (Section 4.2). During experiment, each position and orientation of the end-effector was recorded. These configurations and their sequence were used to write a program in SCORBASE language. This program was given input to the robot controller for execution. The teaching process was repeated for five times to account for the variability present in the lead-through programming approach and thus five executable trajectories were generated for the task. For each trajectory, the path length, mean joint torques and the variation of the end-effector from the specified pick and goal positions were computed. These parameters are compared against the five successful trajectories obtained using the TTI planner as described in Section 4.2 Case 1 experiment 2. Figure 12 shows the comparison of the path lengths of the five trajectories obtained for both the approaches.

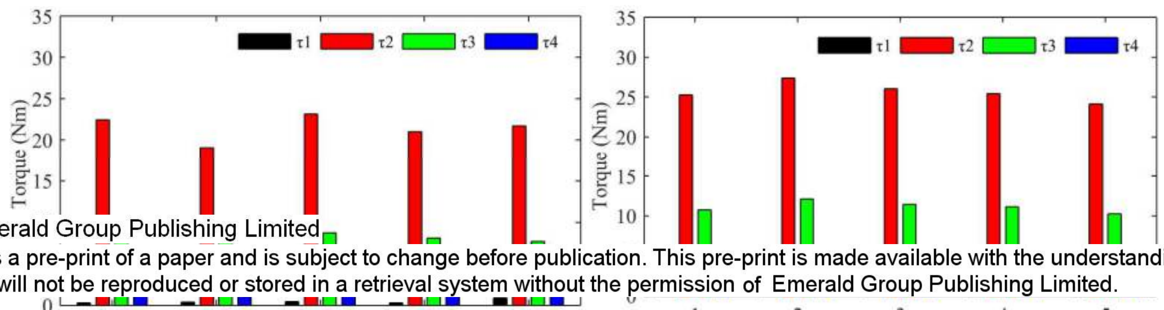
**Figure 12** Comparison of the path lengths




The minimum path length obtained using TTI planner is 68.49 cm, whereas using lead-through technique it is observed to be 69.48 cm. The marginal difference between the path-lengths reflects the capability of the TTI planner to generate a trajectory, which is directly comparable with the lead-through programming technique. Since, both the approaches primarily depend on the skill of the user, by utilizing multiple demonstrations of the same task; TTI planner can generate an optimum trajectory for the given task.

The main advantage of using TTI planner is that the planner performs singularity checks during demonstration itself to ensure singularity free configuration. On the contrary, in lead-through programming technique, the recorded configurations are needed to be checked for singularity before writing the program. Apart from being a time consuming activity, the presence of singular configuration demands for new demonstration. This can be very well observed from the fact that the joint torque requirement for the trajectories generated using TTI planner is lesser than the trajectories generated using lead-through approach (Figure 13).

**Figure 13** Joint torques comparison for the two approaches

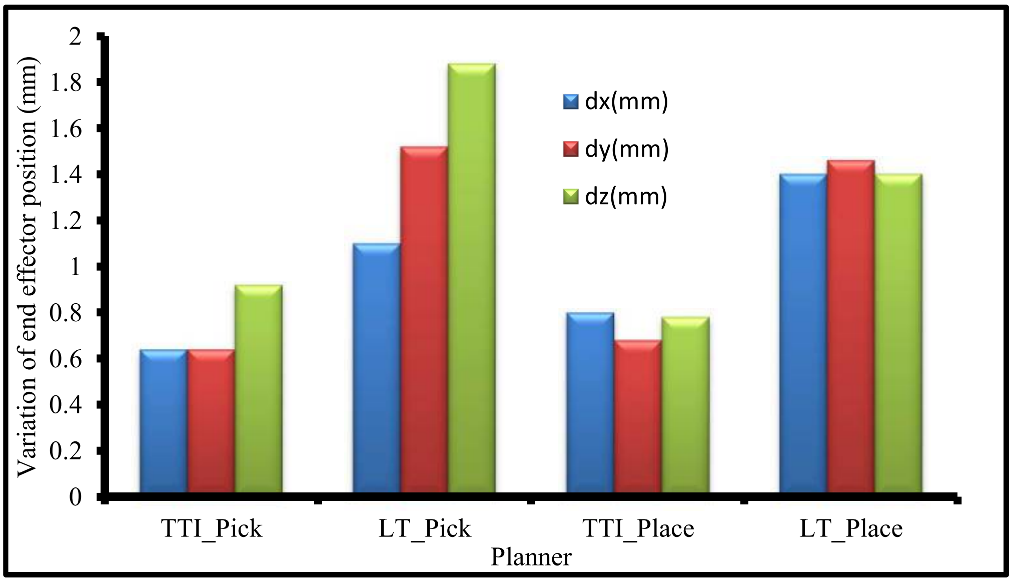


The minimized torque requirement indicates the energy efficiency of the TTI planner. From Figure 12 and 13, it is evident that trajectory 4 is of equal path length for both the approaches, however, the joint torques required for execution by the TTI planner based trajectory is around 6-27% lesser than the lead-through approach based trajectory.

Another advantage of the TTI planner over the lead-through programming technique is the utilization of NURBS for the trajectory approximation. It ensures smooth transition between the consecutive configurations and results in to a smooth trajectory. In lead-through programming technique, the transition between two recorded configurations is controller dependent. To ensure a smooth trajectory, further modifications in the recorded configurations are generally required. Moreover, in lead-through approach, the accuracy of the desired motion relies on the jogging of the robot which is not intuitive as it involves transition between many coordinate systems (Pan *et al.,* 2012). The TTI planner tackles this issue very efficiently by considering the joint space as well the Cartesian space configurations of the robot for trajectory generation.

Figure 14 shows the variation in the end-effector position from the predefined pick and goal positions for both the approaches. For TTI planner, the mean positioning accuracy of the five trajectories is observed to be 0.72 mm, 0.66 mm and 0.85 mm along the x, y and z directions respectively. On the contrary, for lead-through approach, the mean positioning accuracy is observed to be 1.25 mm, 1.49 mm, and 1.64 mm along the three directions. This means that the positioning achieved using TTI planner is accurate enough to enable successful execution of the object manipulation task. The achieved level of positioning accuracy is more for TTI planner because of substantially optimized teaching process.

**Figure 14** Variation in the end-effector position




From the above discussion, it can be inferred that the proposed TTI planner performs better as compared with conventional robot programming method. Moreover, it offers multiple advantages over the lead-through programming method, and hence can be used for programming of variety of robots involved in complex industrial applications.

## 5. Conclusions

In this paper, a method for trajectory learning and reproduction using teleoperation based programming by demonstration is presented. The demonstration phase is very strong as it utilizes teleoperation mode to demonstrate trajectories to the robot. The use of Cartesian as well as joint space configurations in the planner has made it robust to the mapping issues. The incremental kinematic model developed for converting human arm configurations to robot configurations considered singularity problem and resulted into better utilization of workspace as well as a generalization in the workspace. The robustness of the learning process is enhanced by using the optimization criteria. Moreover, the criteria used for selection of the optimal trajectory for reproduction phase, the minimum path length and joint torque are more realistic for a real time application of the planner. This gives flexibility to the user to select an optimal trajectory based on the requirements. Compared to probabilistic and conventional methods, teleoperation based learning predicts the feasibility of demonstration for robot immediately during demonstration. It becomes a closed-loop training and learning system with the advantage of realistic and application oriented learning. Further, when compared with the existing robot programming approaches, the TTI planner performed better in many aspects. However, applicability of the current approach can be further validated using it on the six degrees of freedom robot and for complex application like spray painting.



## References

Aleotti, J., Caselli, S. and Maccherozzi, G. (2005), "Trajectory reconstruction with nurbs curves for robot programming by demonstration", In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, (CIRA 2005),* pp. 73-78.

Aleotti, J. and Caselli, S. (2006) "Robust trajectory learning and approximation for robot programming by demonstration", *Robotics and Autonomous Systems*, *54*(5), pp.409-413.

Andulkar, M.V. and Chiddarwar, S.S. (2015), "Incremental approach for trajectory generation of spray painting robot", *Industrial Robot: An International Journal*, *42*(3), pp. 228-241.

Andulkar, M.V., Chiddarwar, S.S. and Marathe, A.S. (2015), "Novel integrated offline trajectory generation approach for robot assisted spray painting operation", *Journal of Manufacturing Systems*, 37(1), pp. 201-216.

Argall, B.D., Chernova, S., Veloso, M. and Browning, B. (2009), "A survey of robot learning from demonstration", *Robotics and Autonomous systems*, *57*(5), pp.469-483.

Billard, A., Calinon, S., Dillmann, R. and Schaal, S. (2008), "Robot programming by demonstration", In *Springer handbook of robotics,* pp. 1371-1394.

Bullock, D. and Grossberg, S. (1988), “Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation”, *Psychological review*, *95*(1), pp.49-90.

Buss, S.R. (2004), “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods”, *IEEE Journal of Robotics and Automation*, *17*(1-19), p.16.

Calinon, S., D'halluin, F., Sauser, E.L., Caldwell, D.G. and Billard, A.G. (2010), “Learning and reproduction of gestures by imitation”, *Robotics & Automation Magazine, IEEE*, *17*(2), pp.44-54.

Chevallereau, C. and Khalil, W. (1987), “Efficient method for the calculation of the pseudo inverse kinematic problem”, In *Proceedings of IEEE International Conference on Robotics and Automation.* Vol. 4, pp. 1842-1848.

Chiddarwar, S.S. and Babu, N.R. (2011), “Conflict free coordinated path planning for multiple robots using a dynamic path modification sequence”, *Robotics and Autonomous Systems*, *59*(7), pp.508-518.

Ferreira, M., Costa, P., Rocha, L. and Moreira, A.P. (2014), “Stereo-based real-time 6-DoF work tool tracking for robot programing by demonstration”, *The International Journal of Advanced Manufacturing Technology*, pp.1-13.

Flash, T., Meirovitch, Y. and Barliya, A. (2013), “Models of human movement: Trajectory planning and inverse kinematics studies”, *Robotics and Autonomous Systems*, *61*(4), pp.330-339.

Goodall, C., (1991), “Procrustes methods in the statistical analysis of shape”, *Journal of the Royal Statistical Society Series B (Methodological)*, pp.285-339.

Han, J., Shao, L., Xu, D. and Shotton, J. ( 2013), “Enhanced computer vision with microsoft kinect sensor: A review”, *IEEE Transactions on Cybernetics,* 43(5), pp.1318-1334.

Hersch, M. and Billard, A.G. (2008), “Reaching with multi-referential dynamical systems”, *Autonomous Robots*, *25*(1-2), pp.71-83.

Ijspeert, A.J., Nakanishi, J. and Schaal, S. (2002), “Movement imitation with nonlinear dynamical systems in humanoid robots”, In *Proceedings of IEEE International Conference on* Robotics *and Automation ICRA'02,* Vol. 2, pp. 1398-1403.

Jha, A., Chiddarwar, S.S. and Andulkar, M.V. (2015), “An integrated approach for robot training using Kinect and human arm kinematics”, *In Proceedings of IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI 2015),* pp. 216-221.

Lafferty, J., McCallum, A. and Pereira, F.C. (2001), "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pp. 282-289.

Lee, D. and Nakamura, Y. (2006), “Stochastic model of imitating a new observed motion based on the acquired motion primitives”, In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 4994-5000.

Luo, L.P., Yuan, C., Yan, R.J., Yuan, Q., Wu, J., Shin, K.S. and Han, C.S. (2015), "Trajectory planning for energy minimization of industry robotic manipulators using the Lagrange interpolation method", *International Journal of Precision Engineering and Manufacturing*, 16(5), pp.911-917.

Narayanan, K.K., Posada, L.F., Hoffmann, F. and Bertram, T. (2011), “Situated learning of visual robot behaviors”, *In Proceedings of the Fourth International Conference on Intelligent Robotics and Applications (ICIRA 11),* pp. 172-182.

Nehaniv, C.L. and Dautenhahn, K. eds. (2007), “*Imitation and social learning in robots, humans and animals: behavioural, social and communicative dimensions”,* Cambridge University Press.

Obayashi, M., Kuremoto, T. and Kobayashi, K. (2008), “A self-organized fuzzy-neuro reinforcement learning system for continuous state space for autonomous robots”, *In Proceedings of IEEE International Conference on Computational Intelligence for Modelling Control & Automation,* pp. 551-556.

Pan, Z., Polden, J., Larkin, N., Van Duin, S. and Norrish, J. (2012), “Recent progress on programming methods for industrial robots”, *Robotics and Computer-Integrated Manufacturing*, *28*(2), pp.87-94.

Pardowitz, M., Knoop, S., Dillmann, R. and Zöllner, R.D. (2007), “Incremental learning of tasks from user demonstrations, past experiences, and vocal comments”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 37*(2), pp.322-332.

Rosado, J., Silva, F. and Santos, V. (2014), “A Kinect-Based Motion Capture System for Robotic Gesture Imitation*”, In Proceedings of ROBOT2013: First Iberian Robotics Conference,* pp. 585-595.

Skoglund, A., Iliev, B. and Palm, R.(2010), “Programming-by-Demonstration of reaching motions—A next-state-planner approach”, *Robotics and Autonomous Systems*, *58*(5), pp.607-621.

Vakanski, A., Mantegh, I., Irish, A. and Janabi-Sharifi, F. (2012), “Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*(4), pp.1039-1052.

Vuković, N., Mitić, M. and Miljković, Z. (2015), “Trajectory learning and reproduction for differential drive mobile robots based on GMM/HMM and dynamic time warping using learning from demonstration framework”, *Engineering Applications of Artificial Intelligence*, Vol. 45, pp.388-404.

Watanabe, T. and Wada, T. (2008), "Reinforcement learning based on modular fuzzy model with gating unit", *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC 2008),* pp. 1806-1811.

Yang, J., Xu, Y. and Chen, C.S. (1997), "Human action learning via hidden Markov model", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* 27(1), pp. 34-44.