Imitation Learning in Industrial Robots: A Kinematics based Trajectory Generation Framework

Abhishek Jha
Department of Mechanical Engineering
V.N.I.T. Nagpur-440010
India
abhishekduttjha@gmail.com

Shital S. Chiddarwar
Department of Mechanical Engineering
V.N.I.T. Nagpur-440010
India
s.chiddarwar@gmail.com

Rohini Y. Bhute
Department of Mechanical Engineering
V.N.I.T. Nagpur-440010
India
rohinibhute6@gmail.com

Veer Alakshendra
Department of Mechanical Engineering
V.N.I.T. Nagpur-440010
India
alakshendra.veer@gmail.com

Gajanan Nikhade
Department of Mechanical Engineering
R.C.O.E.M. Nagpur-440010
India
grnikhade@rediffmail.com

Priya M. Khandekar
Department of Industrial Engineering
R.C.O.E.M. Nagpur-440010
India
priya.khandekar04@gmail.com

ABSTRACT

This paper presents a simplified approach of imitation learning for an industrial robot. The approach utilizes a teleoperation based trajectory planner to generate an end-effector trajectory through direct imitation of the human motion. The adapted planner exploits the features of the human arm kinematic model and the motion tracking system to achieve real time imitation for trajectory generation. In addition, a trajectory generalization framework, based on clustering and the closest point search is also proposed. This generic framework retrieves an optimal trajectory by utilizing all the demonstrations of the task. The approach is verified experimentally on five degrees of freedom industrial robot for a manufacturing application, where a precise trajectory is desired for execution. The experimental results reflect that the proposed approach provides an effective way to teach robots from human task demonstrations.[1]

CCS CONCEPTS

• **B.0 General**

KEYWORDS

Imitation, Kinematics, Trajectory planner, Generalization

1 INTRODUCTION

The robot programming by demonstration or imitation learning has turned out to be one of the promising approaches for teaching robots. It provides a simplified and effective way to program the robot through examples of the desired task. It has acquired significant attention in last few decades, and a number of approaches for various applications are reported in the literature [1,2].

Among various approaches, imitation of the desired action is a simplest and effective medium for learning a new skill. The majority of the research work in this domain focuses on the key issue of finding a generic and an optimal way of skill transfer for imitation [3]. For imitation, the robot utilizes its own perception to observe and learn the skills from the teacher. Imitation of the human motion has been in the core of many such approaches, and a large body of work has addressed this issue [2-4]. The natural ability of the humans to generate smooth and graceful motion makes it suitable for imitation [5]. Therefore, human motion imitation is extensively used for teaching complex motion behaviors to the humanoids. The intrinsic similarities in the kinematic structures of human and humanoids allow precise imitation ranging from pose imitation [6], upper-body motion imitation [7] to the whole-body motion imitation [8]. However, in case of industrial robots, the applicability of such methods is much restricted. For industrial robots, low level skill-learning techniques are employed to learn from human task demonstrations [9]. Generally, these techniques represent the motion or action primitives at the trajectory level, and for execution, its robotic version is acquired from the demonstrations. In the current work, the desired industrial robot motion is governed by the imitation of user defined demonstration of the actual task. To achieve this, it focuses on utilization of natural human motion for robot trajectory generation.

In industrial robots, for achieving effective imitation, the demonstrated skills must be executable by the robot. However, the morphological differences between teacher and the robot restrict the direct mapping between them. Such restriction is referred to as the correspondence problem [10]. The direct imitation of a skill requires the correspondence problem to be solved. The trajectory level imitation offers one solution to the correspondence problem,

AIR '17, June 28-July 2, 2017, New Delhi, India

ACM ISBN 978-1-4503-5294-9/17/06...$15.00
https://doi.org/10.1145/3132446.3134879

where the robot is taught to follow the requisite trajectory through its own embodiment. It also provides some vital information for the task such as, obstacle-free path, preferential directions of movement and the execution level kinematics data [11]. Such information is difficult to specify explicitly.

Learning a skill at the trajectory level requires precise representation of the demonstrated action. The generalized representation of the demonstrated action is often used to acquire the desired skill. For this purpose, the multiple demonstrations of the task are acquired to encode the skill and then the generalized version is extracted from it [12]. Several methods like Hidden Markov Model [12], Gaussian Mixture Model [13] and trajectory clustering [14] are reported in the literature for modeling random variations of human motions. Most of these approaches are probabilistic in nature, and their use is governed by the considered application. Such restrictions made imitation learning a challenging research problem to deal with.

In view of the above prospects, this work proposes an imitation learning framework to teach an industrial robot from human task demonstrations. For this purpose, a teleoperation based trajectory planner is used, which utilizes trajectory level imitation of the demonstrated action to generate an end-effector trajectory in a real time scenario. For generalization from multiple demonstrations, a generic distance based trajectory retrieval technique is proposed. This technique extracts the most relevant key-points from the recorded trajectories to retrieve an optimal trajectory for the robot. Experimental evaluation of the proposed imitation learning framework is performed on five degrees of freedom industrial robot for a simulated spray painting operation.

2 FRAMEWORK FOR IMITATION LEARNING

The proposed imitation learning framework consists of three phases, namely, trajectory generation, generalization and the reproduction phase. In trajectory generation phase, the task-specific user demonstrations are converted into the executable trajectories for the robot. Multiple trajectories generated during this phase are utilized in the generalization phase to retrieve an optimal trajectory. Trajectory such obtained represents the generalized version of the demonstrated action. In reproduction phase, this trajectory is executed by the robot for task accomplishment. The steps involved within the framework are given in Table I and Fig. 1 shows the architecture of the imitation system. The following section describes the adapted methodology in detail.

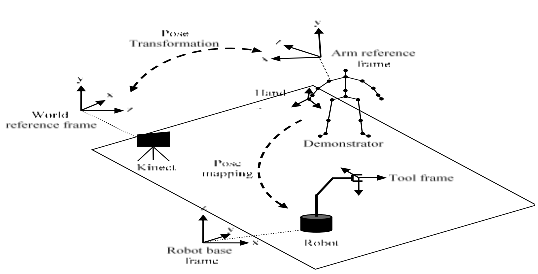


Figure 1: Overview of the imitation system

Table 1: Steps Involved in Imitation Learning Framework

1. Trajectory Generation
• Track the human arm motion during task demonstration (3D positions of arm joints)
• Perform kinematic modeling of the arm for hand trajectory generation
• Map the generated hand trajectory into the robot workspace by utilizing homogeneous transformation
• Perform incremental inverse kinematic modeling of the robot for joint space trajectory estimation
• Optimize the obtained joint space trajectory of the robot for maximum similarity in the task space
• Compute the end-effector trajectory from the optimized joint space trajectory
2. Generalization form Multiple Demonstrations
• Preprocess the recorded set of end-effector trajectories for length equalization
• Categorize preprocessed trajectories into *g*-groups
• Cluster each group separately to merge multiple trajectories
• Implement the closest point search algorithm to extract the most relevant key-points from multiple clusters
• Implement parametric curve fitting for aligning extracted key-points to form a generalized trajectory
3. Reproduction
• Execution of the generalized trajectory by the robot for task accomplishment

2.1 Trajectory Generation Phase

For end-effector trajectory generation, this work emphasizes on the trajectory level imitation of the demonstrated action. Demonstration of a requisite trajectory provides implicit information about the task to be performed and reduces the inherent complexities involved in robot programming [11]. Here, the task demonstration is based on a teleoperation based trajectory planner, which provides a direct and effective medium for teaching the particulars of the task to the robot. The adapted planner makes use of the kinematic model based strategy to generate an end-effector trajectory through direct imitation in a real time environment.

To attain this, the integration of the human arm kinematic model with the motion tracking system is used. Human arm is regarded as seven degrees of freedom redundant manipulator, whose initial configurations (joint and task space) are known a priori [15]. For task demonstration, the user starts an arm movement from a predefined starting posture of the arm. The perception of the demonstrated action is captured through a Kinect based motion capture system. It provides 3D positions of the arm joints at each instant. The positions thus obtained are used for representing the arm motion for the specific task. By incorporating the transformation between Kinect reference frame and the arm reference frame (shoulder frame) as shown in Fig. 1, the kinematic parameters of the arm can be described in either of the reference frames. The forward kinematic model of the arm provides hand pose relative to the arm reference frame and can be given by (1).

$$A_H^S(q) = \left[\begin{array}{cccc} n_H^S & o_H^S & a_H^S & p_H^S \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (1)$$

In (1), q is (7x1) vector of the joint variables, terms n_H^s , o_H^s, and a_H^s represent the unit vectors of hand frame and p_H^s is the position vector of hand. During demonstration, (1) describes the hand pose at a particular instant. To estimate the relative change in the joint variables of the arm model, its incremental inverse kinematic model is used, which can be defined as

$$\Delta q_i = J_H^{\#} \Delta \varepsilon_H \quad (2)$$

where, Δq_i is relative change in the joint variables for a small change $\Delta \varepsilon_H$ in the hand pose and $J_H^{\#}$ is pseudoinverse of the human arm Jacobian matrix. Here, the fusion of the arm kinematic model with the motion tracking system provides discrete poses of the hand. Data such obtained is fitted with Non Uniform Rational B-Splines (NURBS) to form the demonstrated trajectory of the hand. The generalized form of the NURBS can be defined as

$$C(u) = \frac{\sum N_{i,p}(u) w_i P_i}{\sum N_{i,p}(u) w_i} \quad a \le u \le b \quad (3)$$

where, P_i are the control points, $w_i$are the weights and $N_{i,p}(u)$ is the pth degree B-Spline basis function. This parametric curve fitting provides a smooth and noise-free approximation of the demonstrated trajectory. It is considered as a reference trajectory for the robot end-effector to imitate. To map a demonstrated trajectory into the robot workspace, homogeneous transformation between the two embodiments is used. This transformation maps each of the key-points from human arm workspace to the robot workspace and can be given by

$$A_E^B(\theta) = T_X \{A_H^S(q)\}^* \quad (4)$$

where, $A_E^B(\theta)$ is mapped pose in the robot workspace, T_X is the homogeneous transformation matrix which relates arm and the robot reference frames, and $\{A_H^s(q)\}^*$ is the hand pose obtained after NURBS fitting. Here, transformation T_X establishes the mapping between two embodiments by considering their workspace limitations and morphological differences. Trajectory thus mapped is converted into the joint space trajectory of the robot by using its incremental inverse kinematic model and it can be given by

$$\Delta \theta = J_R^{\#} \Delta \varepsilon_R \quad (5)$$

where, $\Delta\theta$ is relative change in the joint space configuration, $J_R^{\#}$ is pseudoinverse of robot Jacobian matrix, and $\Delta\varepsilon_R$ denotes the difference between two consecutive poses of the end-effector. To obtain maximum similarity in the task space, the difference between the mapped trajectory and the trajectory obtained from the kinematic model of the robot needs to be minimized. Therefore, the obtained joint space trajectory of the robot is optimized against the minimum error between the two trajectories in task space, and can be given by

$$\text{Minimize, } E(\theta) = X_{R,H} - X_{R,K} \quad (6)$$

where, $X_{R,H}$ is mapped trajectory and $X_{R,K}$ is the trajectory obtained from the kinematic model of the robot. Equation (6) is constrained for joint space limitations and the difference in end-effector orientation of the two structures. This error minimization provides an optimized joint space trajectory of the robot for maximum similarity in the task space. Joint space trajectory such obtained is further assessed for singularity, and the singular configurations are discarded from it. Then, it is given input to the robot controller for imitation of the demonstrated action in robot workspace.

2.2 Generalization Phase

The key intent of the generalization process is to retrieve an optimal trajectory from the multiple demonstrations of the task performed during trajectory generation phase. The inconsistency associated with the human motion causes each demonstration of the same task to qualitatively differ from other. To overcome such limitation, the proposed generalization framework relies on the most relevant key-points extraction from multiple clusters of multiple trajectories.

The M-captured trajectories can be denoted by $f_m = \{f_{1,m}, f_{2,m}, \cdots \cdots f_{t,m}\}_{m=1}^{M}$, where m is demonstration index, M is the total number of demonstrations, and t is the total number of sampled points in each trajectory. In these trajectories, each sampled key-point is a five-dimensional vector $[x, y, z, pitch, roll]$ representing the end-effector pose relative to the robot's base frame. For generalization, the M-captured trajectories are first preprocessed for length equalization. This is achieved by fixing the shortest trajectory as the reference trajectory and undersampling all other trajectories based on a small displacement ϕ to match the reference length. Then, these preprocessed trajectories are randomly categorized into g-groups. Each group contains multiple trajectories. A hierarchical clustering algorithm is then implemented on each of the groups to obtain a transformed set of clusters formed by merging multiple trajectories. For this, the average link clustering method is used. The distance between two trajectories is taken equal to the average distance given by

$$d_{avg} = \frac{1}{n_i n_j} \sum_{p \in c_i} \sum_{p \in c_j} |Q - Q'| \quad (7)$$

where, $|Q - Q'|$ is the distance between two points, n_i and n_j represents the number of sampled points in the clusters C_i and C_j. At each step, the clustering algorithm merges two nearest clusters into a single cluster. Afterward, a closest point search algorithm is implemented to extract the most relevant points from the transformed sets of clusters. The search algorithm extracts the closest points based on the two distance measures, namely, Euclidean distance and angular distance, which can be defined as

$$d_{Eucl} = \sqrt{(S_R - S_C)(S_R - S_C)^T} \quad (8a)$$

$$d_{ang} = 1 - \frac{S_R S_C}{\sqrt{(S_R S_R^T)(S_C S_C^T)}} \quad (8b)$$

where, $S_R$is a vector containing reference (query) points and S_C represents the sampled points on the transformed set of clusters. This closest point search results in the two sets of points extracted based on minimum positional and the directional differences. Hereafter, the common and nearest key points from these two sets are extracted and aligned using Nearest-Neighbor algorithm [16]. Finally, the points such obtained are fitted with NURBS to form a smooth trajectory. At this stage, the NURBS fitting eliminates the scattered nature of the extracted points and provides best approximation to the demonstrated trajectory. Trajectory such obtained, satisfies all the task constraints defined during demonstration. This trajectory is executed by the robot in reproduction phase for the given task accomplishment.

The experiments performed for assessment of the proposed approach are reported in the next section

3 EXPERIMENTAL EVALUATION

3.1 Experimental Setup

In order to assess the applicability of the proposed approach, the simulated spray-painting operation is considered. The objective of the experiment is to learn precise trajectories by imitating the demonstrated action of a skilled demonstrator. The experimental setup used for this work is shown in Fig. 2. It consists of a Kinect based motion capture system, a human demonstrator and SCORBOT ER-4u five degrees of freedom industrial robot. A spray-painting gun is used as the end-effector of the robot. To define the teleoperation architecture, serial communication between trajectory planner and the robot controller is used. It provides a fast and reliable medium for real-time data transfer between the two systems.

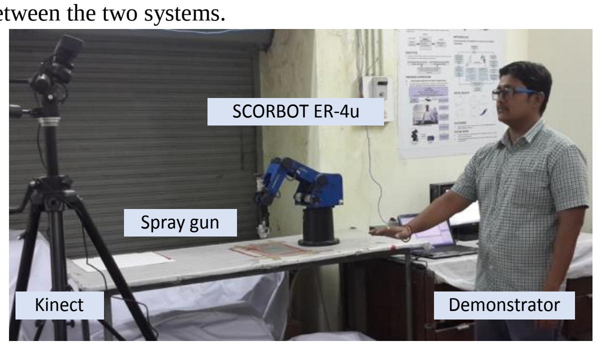


Figure 2: Experimental setup

For the considered task, a raster pattern trajectory is demonstrated by the user through teleoperating the robot over the object to be painted. The task demonstration consists of a reference trajectory for moving the spray gun from one corner of the object to another with waving motion (forward-backward and upward-downward movements). As a process requirement, the orientation of the spray-gun is kept perpendicular to the surface to be painted. The experiment is implemented for two different positions and orientations of a planar object in the robot workspace. In first set of experiments, the object is placed along the YZ- plane (vertical plane) of the robot. In second experiment, it is placed along xy-plane (horizontal plane) of the robot. For each experiment, 15 demonstrations are performed and the adapted planner generated valid and executable trajectories. The best trajectories for each case are obtained using the proposed generalization approach. The outcomes of these experiments are described next.

3.2 Results

In this work, the task execution ability of the demonstrator is used to generate a reference trajectory for the robot. Here, the teleoperation based demonstration technique successfully addresses the correspondence problem. The real time interaction between demonstrator and the robot distinguishes this approach from other works. It gives flexibility to the demonstrator to alter demonstrations as per response of the robot. Each successful demonstration results in an executable trajectory for the robot. Although, the human arm and the robot possess different kinematic structures, despite the mapped trajectory is executable by the robot. Fig. 3 shows one of the demonstrated trajectories and the corresponding trajectory generated by the planner (imitated trajectory) for second experiment.

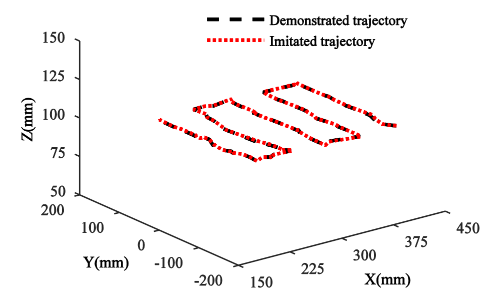


Figure 3: Comparison between demonstrated and imitated trajectories

The overlapping nature of the two trajectories reflects precise imitation of the demonstrated action. In case of motion imitation, similarities between the task and the joint space configurations of the demonstrator and imitator are regarded as the evaluation parameters [7]. Here, the two embodiments are kinematically different; hence their joint space similarities cannot be assessed. Therefore, to quantify the extent of imitation achieved in the task space trajectories, a dissimilarity index is computed using Procrustes analysis [17]. This index provides a standardized dissimilarity measure between the two trajectories, which can be used to assess the geometric similarity between them. Fig. 4 shows the dissimilarity indices of the two experiments for all the demonstrations.

The mean values of dissimilarity indices for the two experiments are 0.040 and 0.037 respectively. It reflects a geometric similarity of around 96% between demonstrated and the imitated trajectories. Moreover, the average of mean squared errors in the position between the two trajectories for all the demonstrations are observed to be 0.56 mm, 4.52 mm, and 1.82 mm for experiment 1, and 2.57 mm, 3.93 mm, and 0.03 mm for experiment 2 respectively.

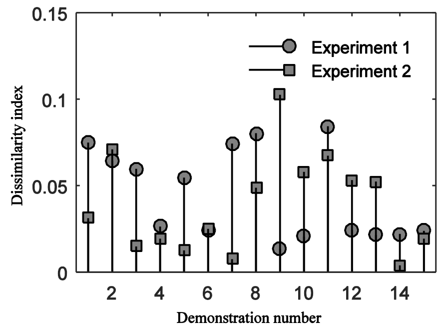


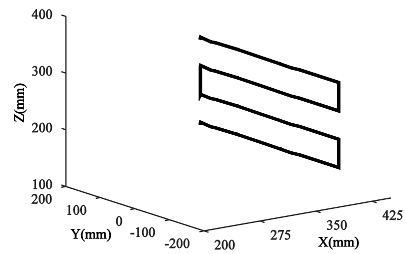
Figure 4: Dissimilarity indices for the two experiments

These results reflect that despite the differences in the kinematic structures, the teleoperation based demonstration technique resulted in geometrically similar trajectories with minimized positioning errors. It shows close replication of the demonstrated trajectory during imitation.

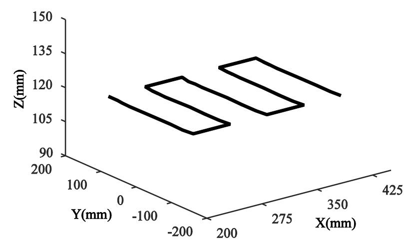
One major concern with the human task demonstrations is the inconsistency associated with the human motion. During experiments, it leads to variability in the demonstrated trajectories of the same task. To overcome this, the proposed generalization framework reconstructs a best-fit trajectory by extracting the most relevant key-points from multiple trajectories. This is achieved by clustering the recorded trajectories in multiple groups and then extracting the closest points from these clusters to generate a best-fit trajectory. The group-wise clustering ensures least modifications in the data recorded from teleoperation and maintains its originality. The generalized trajectories obtained for both the experiments, using this procedure are shown in Fig. 5.

The key feature of the proposed generalization framework is that the obtained trajectories fit better than the demonstrated ones. The adapted group-wise clustering and closest point search techniques reduce the ambiguities across multiple demonstrations by extracting the most relevant key-points for trajectory formation. For assessment of the generalization process, the jerk distribution of the generalized trajectory is compared with the average of multiple demonstrations. The average values of the jerk exerted by generalized trajectories for the two experiments are 1125.50 and 974.30 mm/s^3 respectively. On the contrary, the average jerk values of all the demonstrations are observed to be 1315.90 and 1111.70 mm/s^3 for experiment 1 and 2 respectively. This indicates that the proposed generalization strategy provides a smooth trajectory with improved performance.

In this work, imitation of the demonstrated trajectory essentially depends on the two factors, (1) accuracy of the mapping strategy, and (2) suitability of the incremental inverse kinematic model of the robot for considered application. Here, the homogeneous transformation based strategy is used to achieve mapping between the two workspaces. For appropriate mapping, every mapped key-point of the demonstrated trajectory must lie within the reachable workspace of the robot. To evaluate this, a reachable workspace of the robot is generated by using valid motion ranges of its joints. Each of the demonstrated key-points is searched within this workspace, and the average mapping error for all the demonstrated trajectories is estimated using the Nearest Neighbor algorithm. Fig. 6 shows the average mapping error obtained for the two experiments.



(a) Experiment 1



(b) Experiment 2

Figure 5: Generalized trajectories obtained for the two experiments

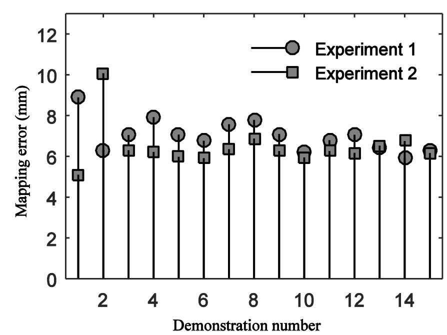
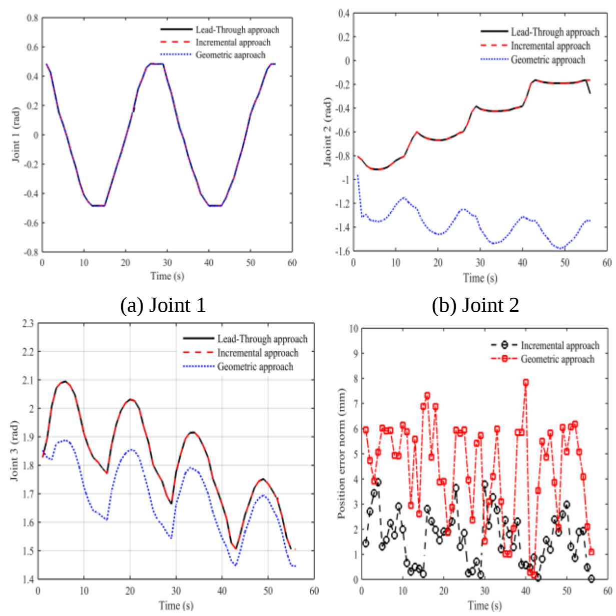


Figure 6: Mapping error for the two experiments

For first experiment, the average mapping error is 7 mm, while for the second experiment, it is 6.50 mm. These smaller values suggest that the proposed mapping strategy is capable of providing higher similarity in the end-effector motion. It is notable that, due to limitations of the incremental inverse kinematics algorithm; the mapping errors may not necessarily reach zero.

Furthermore, to assess suitability of the adapted incremental inverse kinematics model of the robot, another set of experiment is performed. In this experiment, the requisite task space trajectory and its corresponding joint space configurations for second experiment are directly recorded using Lead-Through programming approach [18]. For the recorded task space trajectory, the corresponding joint space trajectories are obtained by using the incremental inverse kinematics (Jacobian pseudoinverse) approach, and then using the geometric approach. A comparison of the two inverse kinematics approaches, along with the norm of the end-effector position error is shown in Fig 7



(a) Joint 1 (b) Joint 2

(c) Joint 3 (d) Position error-norm

Figure 7: Comparison of the two inverse kinematics approaches

The results obtained show that the joint space trajectories obtained by using incremental inverse kinematic approach are overlapping with the reference trajectories (Lead-Through approach), and it is having least variation in comparison with the geometric approach. The position error norms along the whole trajectory are also minimized for the incremental inverse kinematics approach, which indicates its effectiveness. Therefore, for real time applications, use of the incremental inverse kinematics model is justified for achieving precise imitation.

The results discussed here quantitatively support robustness of the proposed approach for precise trajectory generation and learning through direct imitation. From these results, it can be inferred that the human arm motion can be effectively used to program an industrial robot for a complex application like spray-painting operation.

4 CONCLUSIONS

In this paper, an imitation learning approach for industrial robot is presented and evaluated. To achieve imitation, the approach relies on the kinematic models of demonstrator and the imitator. The use of kinematic models for motion imitation makes this approach simple but robust, and provides an effective way to utilize natural human motion for robot programming. The experiments performed show the effectiveness of the method in generating the robot trajectories. Moreover, the adapted distance based geometric approach for generalization easily handled the variations in the demonstrated trajectories and provided a best-fit trajectory for task accomplishment. This approach utilizes all the demonstrations in the generalization process and does not require assessment of best demonstration.

REFERENCES

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. 2008. Robot programming by demonstration. In Springer handbook of robotics. Springer Berlin Heidelberg, 1371-1394.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. 2009. A survey of robot learning from demonstration. Robotics and autonomous systems, vol. 57, no. 5, 469-483.

[3] S. Schaal, A. Ijspeert and A. Billard. 2003. Computational approaches to motor learning by imitation. Philosophical Transactions of the Royal Society of London B: Biological Sciences 358, no. 1431, 537-547.

[4] A. Billard and S. Schaal. 2001. Robust learning of arm trajectories through human demonstration. In Proceedings of 2001 IEEE/RSJ Int. Conf. Intell. Robot. Syst. 2001, vol. 2, 734–739.

[5] Y. Gan and X. Dai. 2012. Human-like Manipulation Planning for Articulated Manipulator. Journal of Bionic Engineering, vol. 9, no. 4, 434-445.

[6] T. Harada, S.Taoka, T. Mori and T. Sato. 2004. Quantitative evaluation method for pose and motion similarity based on human perception. In 4th IEEE/RAS International Conference on Humanoid Robots, vol. 1, 494-512.

[7] M. Alibeigi, S. Rabiee and M.N. Ahmadabadi. 2016. Inverse Kinematics Based Human Mimicking System using Skeletal Tracking Technology. Journal of Intelligent and Robotic Systems, 1-19. DOI:10.1007/s10846-016-0384-6.

[8] J. Lei, M. Song, Z. Li, and C. Chen. 2015. Whole-body humanoid robot imitation with pose similarity evaluation. Signal Processing, vol. 108, 136–146.

[9] S. Chernova and A. L. Thomaz. 2014. Robot learning from human teachers. Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 8, no. 3, 1-121.

[10] C. L. Nehaniv and K. Dautenhahn, “Imitation and social learning in robots, humans and animals: behavioural, social and communicative dimensions.” Cambridge University Press, 2007

[11] J. Aleotti and S. Caselli, “Robust trajectory learning and approximation for robot programming by demonstration.” Robotics and Autonomous Systems, vol. 54, no. 5, pp.409-413, 2006.

[12] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi. 2012. Trajectory Learning for Robot Programming by Demonstration Using Hidden Markov Model and Dynamic Time Warping. IEEE Trans. Syst. man, Cybern. Part B, vol. 42, no. 4, 1039–1052.

[13] N. Vuković, M. Mitić and Z. Miljković. 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, 388-404.

[14] Aleotti and S. Caselli. 2005. Trajectory clustering and stochastic approximation for robot programming by demonstration. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2005, 1029-1034.l

[15] M. Mihelj. 2006. Human arm kinematics for robot based rehabilitation. Robotica, vol. 24, no. 3, 377-383.

[16] J. M. Keller, M. R. Gray and J. A. Givens. 1985. A fuzzy K-nearest neighbor algorithm," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-15, no. 4, 580-585.

[17] C. Goodall. 1991. Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society Series B (Methodological), 285-339.

[18] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish. 2012. Recent progress on programming methods for industrial robots. Robot. Comput. Integr. Manuf., vol. 28, no. 2, 87–94.