Engineering Practice Report

Visual Teleoperation of Franka Emika Panda Arm through Learning from Demonstration

29th of November 2022

Chaitanya Chawla

Student Number:	03741060
Degree Program:	Electrical Engineering and Information Technology
Module:	[EI0900] Ingenieurpraxis
Supervisors:	Prof Dr. Dongheui Lee and M.Sc. Esteve Valls

Human-Centred Assistive Robotics
Technical University Munich
School of Computation, Information and Technology

Contents:

1 Motivation, Problem definition and Goal of the project

1.1 Motivation

Making robots mimic humans as closely as possible has been one of the main areas of research in the recent decades. Our aim with this research is to create a framework that facilitates robots to learn faster and reduces human effort to teach performed demonstrations. For that purpose, we considered to use Learning from Demonstration as our starting point. Learning from demonstration's main goal is to help teach robot how to do some tasks in an easy and faster way, by learning from a human demonstration. Thus teaching robots tasks which are dangerous or difficult for humans to perform due to the prevailing conditions. It is also essential for scenarios where complete automation is impractical, and complex motions are subjected to frequent alterations. Thus, the need for updating the learned motion frequently, can be easily realized by learning from Demonstration.

In traditional robot programming, a human programmer has to manually implement the desired behavior to specify how the robot should act and respond to a certain environmental state. Whereas in the case of Learning from Demonstration, the robot can easily mimic a human user through a communication system. While there are many promising current techniques (e.g. Kinesthetic Control [1], Virtual Reality devices [2], and haptic gloves [3]), each of them suffers from some limitations in applicability, such as requiring presence of human being, expert robotics knowledge, and costly equipment.

1.2 Problem Definition

The goal of this project is to study the feasibility of robotic arm manipulation by capturing the complex human motion from a simple RGB camera/ or through pre-recorded videos and transferring to robot.

The internet contains a massive corpus of rich and diverse human hand videos. Our solution can use this data to learn various human motion trajectories and transfer this as a robot arm trajectory that is smooth, safe, and similar to the guiding demonstration.

We demonstrate that it enables previously untrained people to teleoperate a robot on various manipulation tasks.

In this paper, we present a low cost, gloves free, marker free solution, to teach tasks to Franka Emika Panda Robot through Learning from Demonstration from RGB data. Furthermore, it can be implemented by a remote, untrained operator, by using a simple RGB Camera.

We demonstrate the usability and versatility of our system through two different manipulation tasks.

2 Technical Background and Methodology

2.1 Related work

Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on YouTube [4]

The recent research most relevant to our work is the real-time teleoperation of a Robotic Arm through a RGB Camera. In this solution, they divided the complete process into two subsystems. The first subsystem uses computer vision algorithms trained to estimate 3D human poses from 2D images.The second subsystem uses a motion-retargeting algorithm to generate a robot hand-arm action that is consistent with a given human pose. The researchers trained a deep human-to-robot hand retargeter network, by taking publicly available data from the internet, to convey the motion of the whole arm and hand to the robot.

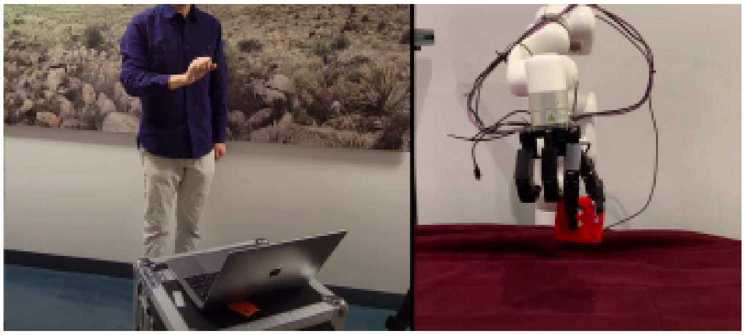


Figure 1: An operator completing dice pickup task while watching the robot through a video conference [4].

Teleoperation of Robotic Arm using Depth Sensor [5]

Another simillar approach is the teleoperation of a robotic arm using the Microsoft Kinect sensor. Kinetic is a RGB-D camera, which can be applied in robotic control as a visual device, so the robot can recognize movement from humans and produce visual interactions between humans and robots.

The researchers controlled the whole robotic arm (instead of just end-effector) by using inverse kinematics on human arm and finding the various angles and positions.

In their paper it is mentioned that the use of Microsoft Kinetic leads to false coordinates in case of occlusion of a part of the body.

2.2 Learning from Demonstration

Learning from demonstration, also known as "programming by demonstration", "imitation learning" , and "teaching by showing" received significant attention in automatic robot assembly over the last 20 years [6]. The goal was to replace the time-consuming manual programming of a robot by an automatic programming process, solely driven by showing the robot the assembly task by an expert.

Usually researchers tackle LfD by two methods, which differ on how the human teaches the demonstration to the robot. On the one hand, there is Kinesthetic teaching, i.e. force-based manipulation tasks, which require a human to physically move the end effector, and thus the motion is learned by the robot [7]. On the other hand, through training a deep network by feeding it visual data of performed motions [4].

Our solution implements Learning from Demonstration by capturing the human motion from RGB videos and transferring the end effector position to the robot. These videos can be obtained either though online recording or from existing internet videos.

3 Conception and Implementation of the Project

3.1 Methodology and Workflow

The workflow was divided into three main tasks - (i) human motion tracking, (ii) learn a trajectory from the tracked motions, and (iii) transferring this learned trajectory on the Gazebo Simulator.

For tracking human motions, MediaPipe Pose [8] was used. The coordinates of a landmark in the right hand were recorded for a short period of time, wherein the desired motion was demonstrated by the user. This process was repeated three to four times for each motion, each of which produced a slightly modified trajectory for the same motion shape.

As the second step, we used Dynamic Motion Primitives (DMPs) [9, 10] to learn a trajectory from the various input trajectories for a single motion. By using regression models, we achieved a smooth trajectory. Through the use of DMPs, one can get a learned trajectory which can be scaled, shifted and oriented as required. One can input the desired initial and end position, and from the different recorded trajectories, we get a learned trajectory scaled to the desired initial and end position.

Lastly, we executed this learned trajectory on the Frank Emika Gazebo simulator [11] by using a self-made cartesian impedance controller.

This complete operation was then repeated and tried out for various complex motions.

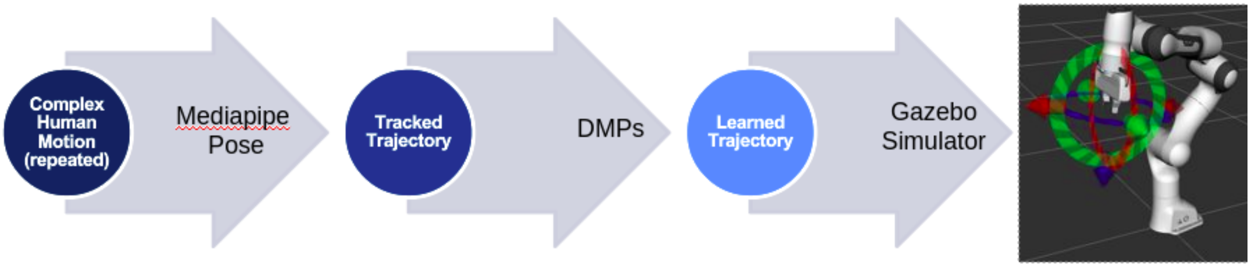


Figure 2: Workflow of the Research

3.2 MediaPipe Pose

MediaPipe Pose [8] is a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks and background segmentation mask on the whole body from RGB video frames. It utilizes a two-step detector-tracker ML pipeline, firstly locates the person/pose region-of-interest (ROI) within the frame. Secondly, it predicts the pose landmarks and segmentation mask within the ROI using the ROI-cropped frame as input.

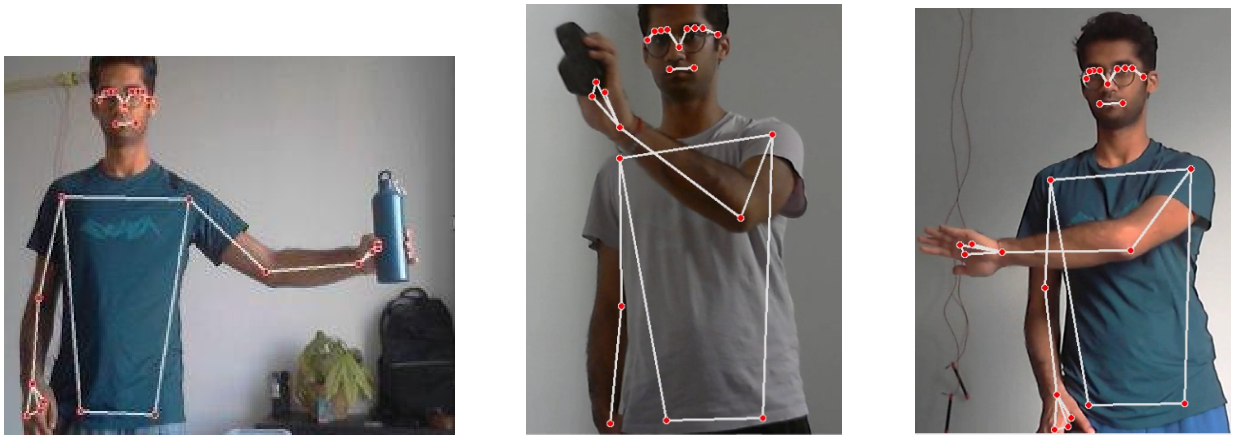


Figure 3: Human Motion Tracking Results by MediaPipe Pose

MediaPipe was chosen over other motion tracking softwares, as it gave a good approximation of the z coordinate compared to other publicly available softwares, such as AlphaPose [9]. To control the end effector of robotic arm, it was decided to use the trajectory followed by the palm of the right hand of the user, instead of using inverse Kinematics to control all joints of robotic arm separately.

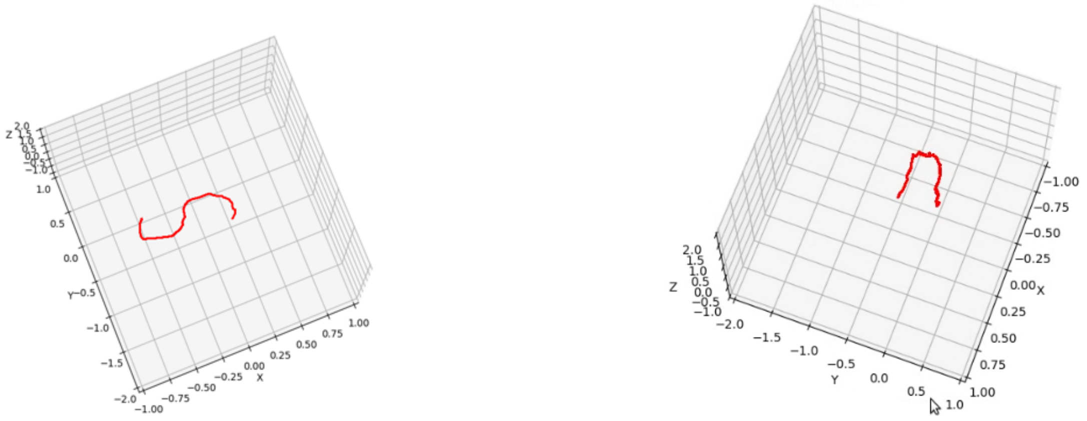


Figure 4 : Sample Trajectories recorded using MediaPipe Pose

3.3 Dynamic Motion Primitives

DMPs [10] are a proposed mathematical formalization of primitive actions, which when combined can form complex human motions. For each dimension, the activations learn the weights according to the input trajectory, and thus the trajectory can be easily scaled and shifted to coordinates while still maintaining its original form and structure.

DMPs add on a forcing term f that will let us modify the point attractor dynamical trajectory.:

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f$$

where y is our system state, g is the goal, and α and β are gain terms.
The crux of the DMP framework is an additional nonlinear system used to define the forcing function f over time, giving the problem a well defined structure that can be solved in a straight-forward way and easily generalizes. The introduced system is called the canonical dynamical system, is denoted x, and has very simple dynamics:

$$\dot{x} = -\alpha_x . x$$

The forcing function f is defined as a function of the canonical system:

$$f(x, g) = \frac{\sum_{i=1}^{N} \Psi_i \omega_i}{\sum_{i=1}^{N} \Psi_i} x(g - y_o)$$

where y_o is the initial position of the system,

$$\Psi_i = exp(-\ h_i(x - c_i)^2)$$

and ω_i is a weighting for a given basis function Ψ_i .

So our forcing function is a set of Gaussians that are ‘activated’ as the canonical system x converges to its target. Their weighted summation is normalized, and then multiplied by the $x(g - y_o)$ term, which is both a ‘diminishing’ and spatial scaling term.

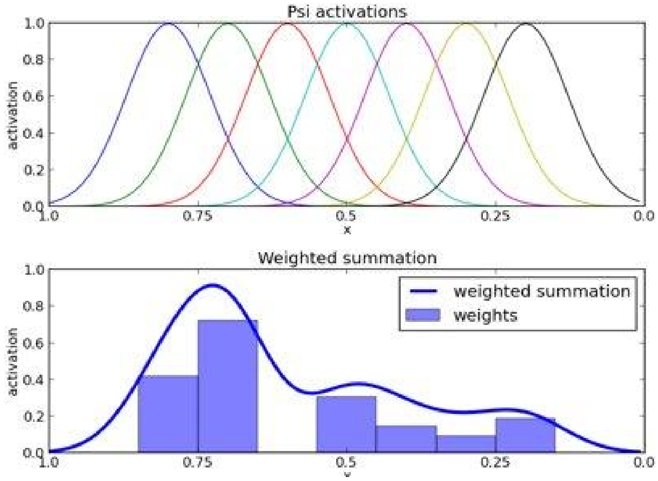


Figure 5 : Figure showing how the weights in a specific coordinate can be adjusted to get a learned trajectory

In our project, we were able to obtain a smoothened, learned trajectory through regression from the previously recorded trajectories of the same motion. This learned trajectory could adapt itself to the desired initial- and end location. For instance, in a "pick and place object" scenario we could use the object starting point as the initial position and the place location as the end position for the DMP.

During the research, we came across certain drawbacks of using DMPs to get a trajectory. For example, repetition of the same motion is required to acquire a more accurate and smoothened shape of learned trajectory. Thus, leading to real time teleoperation being a difficult and inaccurate target. Secondly, the orientation of the motion can not be ascertained if a 2D Motion is given, and thus getting a learned trajectory with the desired starting- and end point, but not the desired orientation.

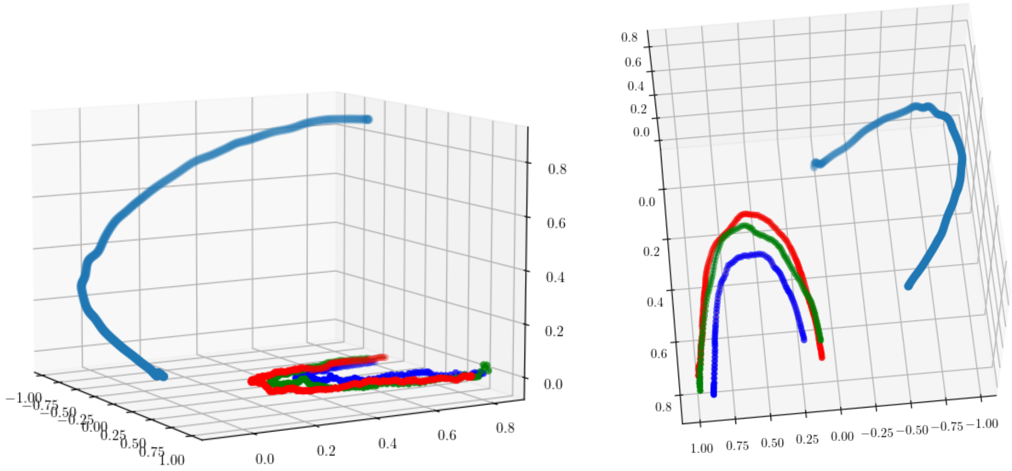


Figure 6: In Red, Green, and Dark Blue, trajectories recorded using MediaPipe Pose for a C-shaped Motion. In light Blue, learned trajectory obtained by using DMPs with initial location as (0, 0, 0) and end location as (0.5, -0.5, 0.9) - (a) Side-view and (b) Top-view

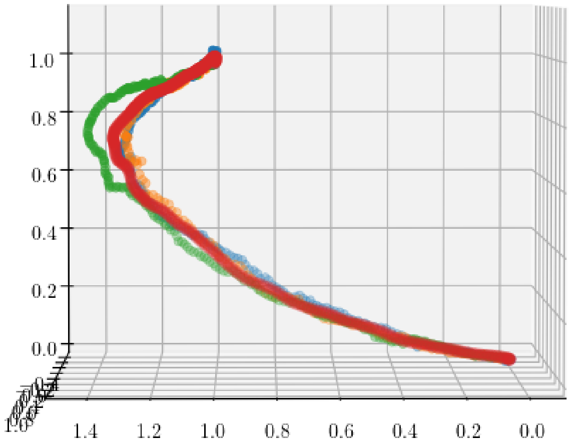


Figure 7: C-shaped motion trajectory scaled to initial location as (0,0,0) and End location as (1,1,1)

3.3 Franka Gazebo Simulator

Franka Gazebo [12] is the simulator for Franka Emika panda arm. It provides visualization for the robotic arm in Gazebo as well as RViz. It also offers various pre-defined controller implementations, such as Impedance controller, Force controller, and Velocity controller.

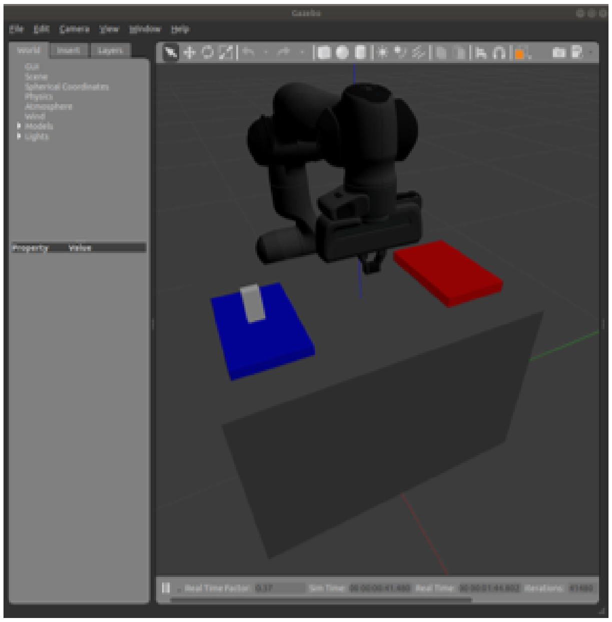
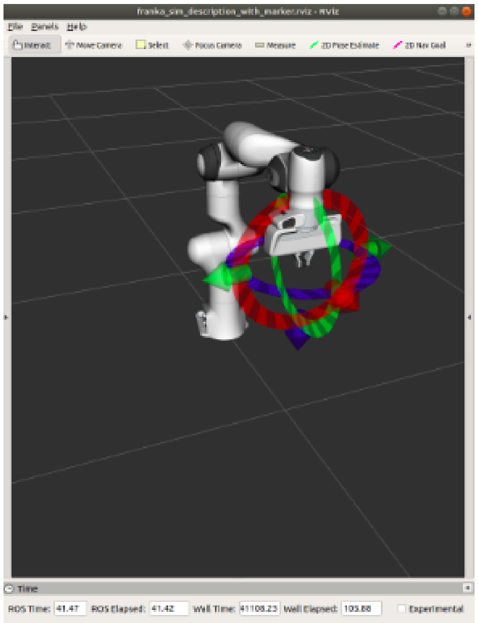


Figure 8: Franka Panda arm simulation - Gazebo



ROS Time: 41.47 ROS Elapsed: 41.42 Wall Time: 41108.23 Wall Elapsed: 103.88 □ Experimental

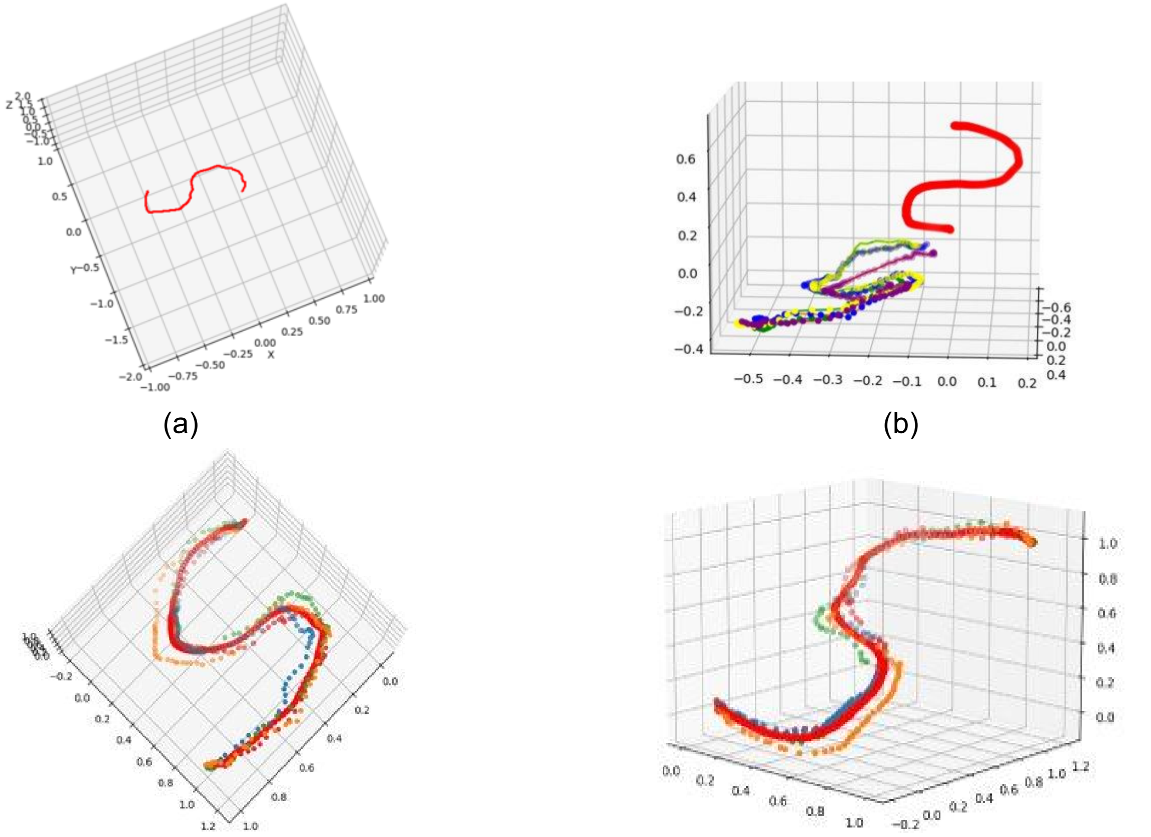
Figure 9: Franka Panda simulation in RViz

The workflow for transferring the learned trajectory to the Frank Panda arm Simulator was as follows. The first step was designing our own Impedance controller for the robotic Arm, which would follow the given trajectory. Then we tried out different parameters for the conditions, for going from one point to another, and time spent on each point in path.

4 Qualitative Results and Comparison to the original goals

4.1 Qualitative Results

We carried out the complete procedure for various motions and for different starting- and end points. Here we present the results for two such motions.
Firstly, the user performed a S-motion.



(a) (b)

(c) (d)

Figure 10: (a) S-motion trajectory as recorded by MediaPipe, (b) Smooth red - Learned trajectory with initial point - (0.0, 0.3, 0.1) and end point - (0.5, 0.0, 0.6). This trajectory was transferred to the Gazebo, (c) top view- Smooth red - Learned Trajectory obtained with help of DMPs -initial point - (0, 0, 0), end point - (1, 1, 1) Multiple dotted Trajectories - recorded trajectories, (d) side view

The links to the videos of the user performing the above motion as well as the motion being executed on the Gazebo Simulator, can be found in the supplementary material videos.

4.2 Partial Occlusion

Partial occlusion refers the scenario where part(s) of the human is occluded by objects in the environment, which could lead to the human motion tracking system returning us false coordinates.

As our solution was using a landmark from the right palm to get the trajectory, we tried holding objects of different shapes and sizes, to see the results during partial occlusion.

MediaPipe's results are not affected much if some parts of the hand are not fully visible.

As long as most of the arm is still visible it can still roughly extrapolate the location of the palm, providing us an accurate trajectory. In rare cases, even if some anomalous point crept in, the use of DMPs minimized its effects in the learned trajectory. Thus picking and placing of objects can be easily performed with this solution.

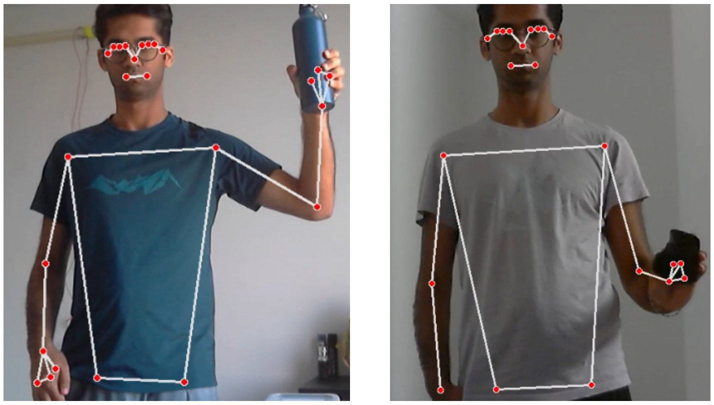
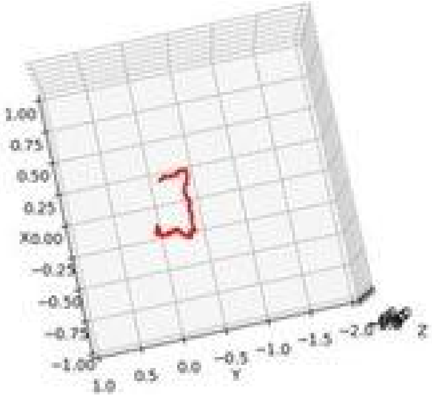
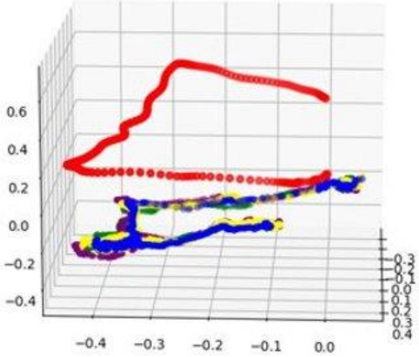


Figure 11: MediaPipe results when user is moving an object

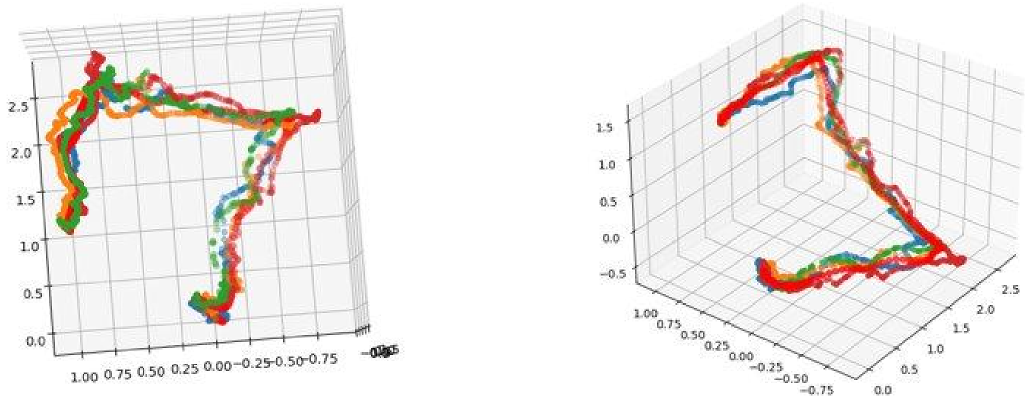
The second motion was picking and placing of a bottle.



(a)



(b)



(c) (d)

Figure 12: (a) Bottle-pick trajectory as recorded by MediaPipe, (b) Smooth red - Learned trajectory with initial point - (0, 0,0.1) and end point - (0.2, 0, 0.6). This trajectory was transferred to the Gazebo, (c) top view- Smooth red - Learned Trajectory obtained with help of DMPs -initial point - (0, 0, 0), end point - (1, 1, 1) Multiple dotted Trajectories - recorded trajectories, (d) side view

5 Conclusion

5.1 Conclusion

To summarize, the results showcased a variety that were in par with our initial targets.
Firstly, human pose estimation from MediaPipe was accurate, gave good results even in cases of partial occlusion, and can be used for manipulating robotic arms. Secondly, learned trajectories obtained using DMPs can be used for simulation, although the orientation of the motion can not be ascertained if a 2D Motion is given. As the changeable parameters are only the initial and goal location, it is tough to determine a single, fixed orientation of the entire motion. Lastly, the implementation of the trajectories on the Franka Gazebo simulator behaved as was expected and showed promising results.

5.1 Future Works

To further improve our system, we propose some developments which could make the system more accessible. To begin with, controlling opening and closing of the gripper through hand gestures of the other hand using a simple publisher script could be used to real-life objects. Using computer vision algorithms, we could detect object location and automatically select that as the start location for the learned trajectory.
Furthermore, we could improve the transferability from human to robot motion, by not only focussing on the end effector, but also the other joints of the robot so that the whole motion is similar to the arm.

6 Bibliography

[1] M. Saveriano, S. An, and D. Lee, "Incremental Kinesthetic Teaching of End-Effector and Null-Space Motion Primitives," in IEEE International Conference on Robotics and Automation (ICRA), 2015.

[2] Oculus rift. https://www.oculus.com/rift/. 1

[3] Haptx. https://haptx.com/

[4] Sivakumar, Aravind, Kenneth Shaw, and Deepak Pathak. "Robotic telekinesis: learning a robotic hand imitator by watching humans on Youtube." arXiv preprint arXiv:2202.10448 (2022).

[5] M. Syakir, E. S. Ningrum and I. Adji Sulistijono, "Teleoperation Robot Arm using Depth Sensor," 2019 International Electronics Symposium (IES), 2019, pp. 394-399, doi: 10.1109/ELECSYM.2019.8901679.

[6] Schaal, Stefan. "Learning from demonstration." *Advances in neural information processing systems* 9 (1996).

[7] Rozo, Leonel, Pablo Jiménez, and Carme Torras. "A robot learning from demonstration framework to perform force-based manipulation tasks." Intelligent service robotics 6.1 (2013): 33-51.

[8] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.

[9] Fang, Hao-Shu, et al. "AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[10] Saveriano, M., Abu-Dakka, F. J., Kramberger, A., & Peternel, L. (2021). Dynamic movement primitives in robotics: A tutorial survey. arXiv preprint arXiv:2102.03861.

[11] Ginesi, M., Github Repository https://github.com/mginesi/dmp_pp

[12] C. Jähne, F. Walch, J. R. Medina & T. Goll, Github Repository https://github.com/frankaemika/franka_ros

[13] Hu, Haiying & Li, Jiawei & Xie, Zongwu & Wang, Bin & Liu, Hong & Hirzinger, G.. (2005). A robot arm/hand teleoperation system with telepresence and shared control. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM. 2. 1312 - 1317. 10.1109/AIM.2005.1511192.