Learning by Demonstration

Eshita
2017149
eshita17149@iiitd.ac.in

Khalid Lodhi
2017160
khalid17160@iiitd.ac.in

Garvit Gupta
2018141
garvit18141@iiitd.ac.in

Lavanya Verma
2018155
lavanya18155@iiitd.ac.in

Shubhi Singhal
2018195
shubhi18195@iiitd.ac.in

Adarsh Kumar
2018213
adarsh18213@iiitd.ac.in

***Abstract*—This project proposes learning by demonstration approach based on dynamic movement primitives (DMP).To represent an observed movement, a differential equation is learned such that it reproduces this movement. Our differential equation is formulated such that generalization can be achieved simply by adapting a start and a goal parameter in the equation to the desired position values of a movement. This movement is represented by the movement of the cursor.**

***Index Terms*—DMP, movement, differential equation**

I. Introduction

Learning by demonstration, also known as imitation learning is a paradigm that enables robots to perform new tasks autonomously. Rather than requiring users to implicitly program a desired behaviour, DMP takes the view that an appropriate controller can be derived by a human's demonstration. These capabilities can then further be extended to be adapted for generalization and used by users without any programming abilities.

In this framework, any recorded movement can be represented with a set of differential equations. The equations are formulated such that adaptation to a new goal is achieved by simply changing a goal parameter. This characteristic allows generalization. Besides the end-effector position, we also need to control the orientation of the gripper and the position of the fingers. The DMP framework allows to combine the end-effector motion with any additional degree-of-freedom(DOF). In this project we have allowed 2 DOF.

Complex movements have long been thought to be composed of sets of primitive action 'building blocks' executed in sequence or in parallel, and DMPs are a proposed mathematical formalization of these primitives. The basic idea is that we take a dynamical system with well specified, stable behaviour and add another term that makes it follow some a trajectory. When we use a DMP what we are doing is planning a trajectory for your real system to follow. There are two kinds of DMPs: discrete and rhythmic. For discrete movements the base system is a point attractor, and for rhythmic movements a limit cycle is used. In this project we have used discrete DMPs.

II. Why we chose this project?

Following are the reasons we took this project:

1) We found the problem presented in the paper exciting and challenging and believe that in future this will be one of the most dominant paradigms in robotics.
2) The underlying principle behind any demonstration learning technique is to learn the nonlinear differential equation governing the system to which we all are very familiar.
3) It offers an implicit means of training a machine, such that explicit and tedious programming of a task by a human user can be minimized or eliminated.
4) The choice of demonstration learning over other robot learning methods is compelling when ideal behavior can neither be easily scripted, as done in traditional robot programming, nor be easily defined as an optimization problem, but can be demonstrated.
5) Some of the interesting applications where we can use demonstration learning in robotics are:
   - Tasks involving pick-and-place operation, water-serving task and other household tasks, which can help in situations when elder family members have to be alone at home.
   - Therapies involving robotic arms to help the patient practice hand/leg movements.

III. Related Work

The paper by Pastor et al. [1] extended the framework of dynamic movement primitives to action sequences that allow object manipulation. They have suggested several improvements of the original movement primitive framework: robust generalization to new goals, human like adaptation, and automatic obstacle avoidance.In this paper, authors have discussed the limitation of the formulation of DMPs:

- If the start and goal position of a movement is the same, then the nonlinear function cannot drive the system away from this state.
- Scaling with (g - x_0) is a bit problematic, very small difference may lead to huge acceleration, which can break the physical limit of actuators.
- Adaptation to new goal, such that (g_{new} - x_0) has different sign than (g_{old} - x), lead to mirror of generalisation, which could be unsuitable in Cartesian space.

IV. Mathematical framework

Transformation System

$$\tau \dot{v} = K(g - x) - Dv + (g - x_0)f$$

$$\tau \dot{x} = v$$

where x is the recorded trajectory, v and ˙v are numerically computed derivatives of x; x0 and g are start and end positions respectively; tau is a temporal scaling factor; K and D account for point attractor dynamics such that the system is critically damped. f is a linear combination of many nonlinear Gaussian functions. The Key is to determine the appropriate weights to approximate any complex trajectory. Linear combination of non-linear functions.

$$f(s) = \frac{\sum_i w_i \psi_i(s) s}{\sum_i \psi_i(s)}$$

where

$$\psi_i(s) = exp(-h_i(s - c_i)^2)$$

are Gaussian basis functions with mean c_i and width h_i, w_i are weights. The function f does not depend upon time directly, instead it depends upon a phase variable (s), which exponentially converges from 1 to 0 during the motion.

$$\tau \dot{s} = -\alpha s$$

alpha is a predefined constant. The last differential equation referred to the canonical system. This system has given properties:

- Asymptotic Convergence to the endpoint is ensured (for bounded weights as f(s) vanishes as the end of a movement.
- Multiplication with (g - x_0) allows the movement to be scaled as per the requirement. (Spatial Invariant)
- Time independence of f(s) allows the movement to be stretch or compressed in time as per the value of . Scaling with s ensures that the effect of f(s) decreases monotonically. (Temporal Invariant)

V. Our Implementation

Our approach is divided into 4 main parts:

1) **Recording movement**
   We recorded the mouse motion in figure framework of MATLAB. The output is a file storing the values x,y and t. To start and stop recording the motion, we have to press R, S to save the data and D to clear the record.
2) **Smoothing of recorded movement**
   Recorded movement is neither smooth nor uniformly sampled. Both of these problems can be solved by either interpolating of approximating the data points.
3) **DMP Parameter extraction**
   Main_DMP.m file performs the computations of weights and other parameters. The serialised file is saved thereby enhancing modularity.

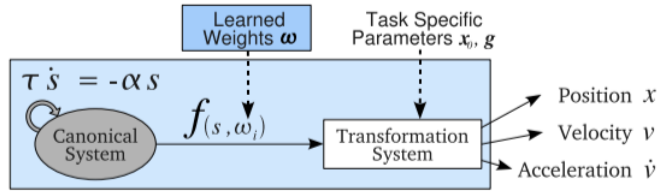

Fig. 1. Sketch of a one dimensional DMP given in reference paper

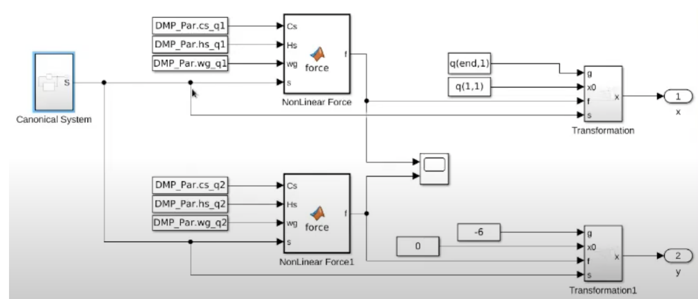

Fig. 2. Two dimensional DMP implemented by us on MATLAB

4) **Load parameter and Simulink**
   The precomputed parameters is loaded from the serialised file in required format for Simulink model. The generated trajectory is then fed into control.

Using the sketch of a one dimensional DMP given in [1], Figure 1 where the canonical system drives the nonlinear function f which perturbs the transformation system, we implemented two dimensional DMP in MATLAB as shown in Figure 2.

VI. Results

The result generated by our system can be seen in Figure 4. The demonstrated output corresponding to this output is shown in Figure 3. We also simulated these results using 2-D link robotic arm on MATLAB as shown in figures 5 and 6.

VII. Conclusion

We demonstrated the feasibility of our approach in an imitation learning setting. DMP based planner provides a viable mechanism to utilize manipulator at a quick pace however it seems to fail to complex instances. For such cases, proposed solution is to decompose the movement into submovements apply DMP on submovement.

VIII. Future work

There are several improvements of the original movement primitive framework which can be robust generalization to new goals, human like adaptation, and automatic obstacle avoidance. A few challenges need to be mastered. The correspondence problem which means that links and joints between human and robot may not match. Also we need robustness against perturbation: replaying exactly an observed movement is unrealistic in a dynamic environment, in which obstacles may appear suddenly. Another improvement is that we could generalize this task to novel situations.

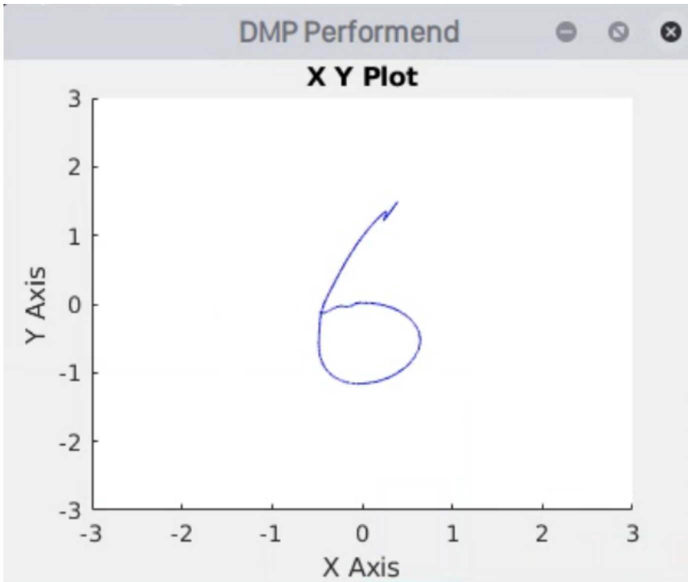

Fig. 3. Input demonstrated

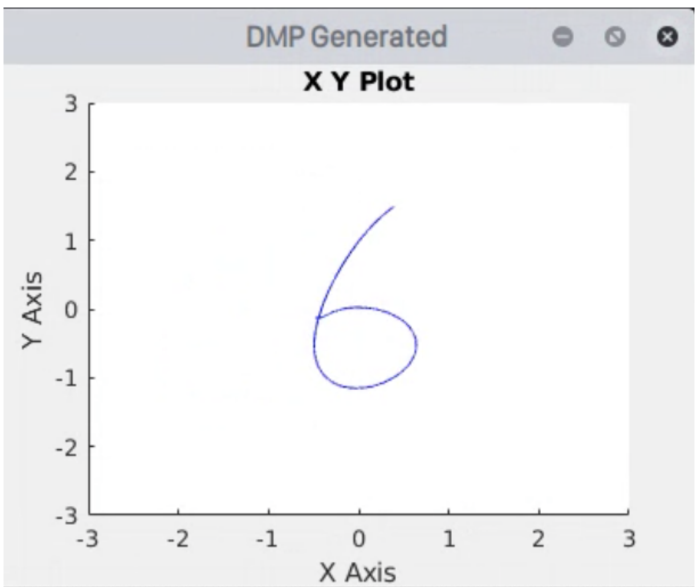

Fig. 4. Output generated

ACKNOWLEDGMENT

First and foremost, we would like to thank our professor, Dr. Sayan Basu Roy, for giving us the opportunity to do this project as a part of our robotics course. We would like to extend our sincere gratitude towards him for his constant guidance and support throughout the project. We would also like to thank our TAs for their support and help for this project.

REFERENCES

[1] Pastor, Peter & Hoffmann, Heiko & Asfour, Tamim & Schaal, Stefan. (2009). Learning and Generalization of Motor Skills by Learning from Demonstration. International Conference on Robotics and Automation (ICRA 2009). 763-768. 10.1109/ROBOT.2009.5152385.

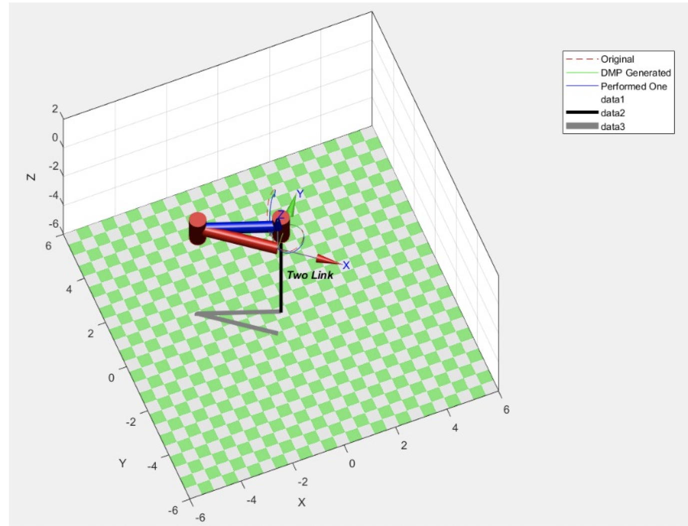

Fig. 5. 2-D link robotic arm simulation (1)

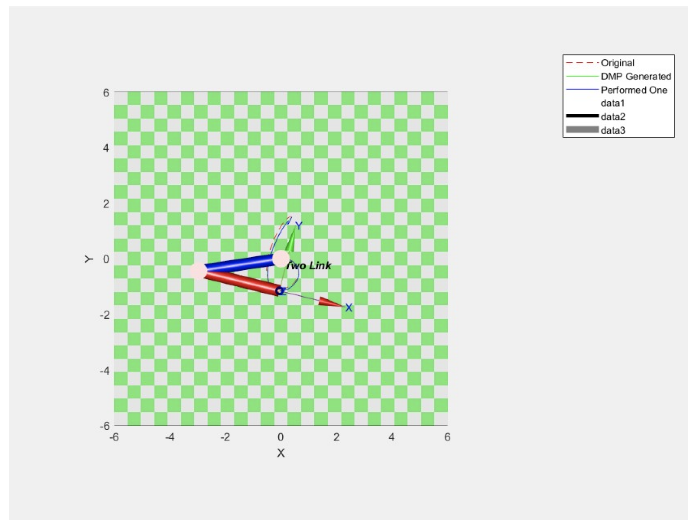

Fig. 6. 2-D link robotic arm simulation (2)

[2] Lauretti, Clemente & Cordella, Francesca & Guglielmelli, Eugenio & Zollo, Loredana. (2017). Learning by Demonstration for Planning Activities of Daily Living in Rehabilitation and Assistive Robotics. IEEE Robotics and Automation Letters. PP. 1-1. 10.1109/LRA.2017.2669369.

[3] Schaal S. (2006) Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics. In: Kimura H., Tsuchiya K., Ishiguro A., Witte H. (eds) Adaptive Motion of Animals and Machines. Springer, Tokyo. https://doi.org/10.1007/4-431-31381-8_23

[4] D. Park, H. Hoffmann, P. Pastor and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots, Daejeon, 2008, pp. 91-98, doi: 10.1109/ICHR.2008.4755937.