Short Review Article: Generalisation from Demonstrations Dynamic Movement Primitives

Lavanya Verma

November 2020

Abstract

Robot manipulators provide a sustainable solution to repeated, precise, and dangerous jobs. Traditional design procedure takes a significantly long time in their deployment and addition to that; the designer also needs to consider an exhaustive list of possibilities that may face. Robotic systems lose their stability once they are induced to an environment different than what they were designed for. The learning by demonstration method addresses this issue. Dynamic Movement Primitives is one such effective model of learning from demonstration, which provides a compliant and robust design to environmental perturbations.

1 Introduction

The use of robotic manipulators has recently increased in the manufacturing industry. They can adhere to the manufacturing process's specifics in speed, automation of repetitive tasks, reduction of margins of error of nominal rates, 24/7 operation, and cost-efficiency. They allow humans to focus more on high-level tasks, thereby enhancing their capability to serve better in statistical measures. Automated factories provide robots very high assurance of predictability of the environment for time-critical and kinematic operations. This higher predictability is exploited in the design of the control policy of factory tailored robots. Robots to be leveraged in other human endeavors such as rehabilitation, assistive social care, and activities of daily living(ADLs), they have to show certain improvements

- Deployed faster: Traditional robot programming requires every to be explicitly fed into the controller concerning every possible use case, no matter how unlikely it could be.
- Compliant and Robust against environmental perturbations.
- Able to learn user's training chore.

Robots can assure high repeatability of movement and intensity of treatment, active roles of patients, an enhancement of their level of independence, patient-tailored planning strategies enabling the user to relearn personal motion style. The benefits of performing a user's personal-motion type in rehabilitation sessions are not well-established in literature yet. Still, the adoption of similar

strategies showed encouraging results, such as mirror therapy in stroke.
According to (Billard and Grollman 2013), Learning from Demonstration/ Programming from Demonstration (LfD/PfD) paradigm provides complementary solutions to the addressed issues. A human operator shows a robot how to perform a specific task, and the robot will decompose that into a set of basic units to perform that task without programming. If any failure occurs, the end-user needs to provide more demonstrations rather than changes made in the robot's software. Learning from Demonstration is not "Record and Replay"; learning and generalizing is its property.
One LfD nonlinear differential equation-based technique is Dynamic movement primitives (DMPs), with a well-defined attractor landscape. It allows attractors to replicate the nonlinear trajectory through the weighted sum of equally spaced Gaussian kernels; weights are computed from demonstrated movements, in a single iteration, with a locally weighted regression technique. (Schaal 2006) DMPs overcomes the performance of other LfD approaches such as Time-dependent Gaussian Mixture Regression, Locally Weighted Regression, and Locally weighted projection regression in terms of robustness to accuracy, spatial perturbations, and computation time. Formulation of DMPs carried over from neurobiology is related to motor pattern generators(MPG). MPGs are neural circuits with only limited modifiable properties (Park et al. 2008). However, to allow for the immense flexibility of human limb control, the MPGs concept needs to augment a component that can be adjusted task-specifically, leading to Dynamic Movement Primitives.

The Scope of Review paper provides an introduction to the mathematics foundation of discrete DMPs.

2 Body

In paper (Schaal 2006), the Authors presented the idea of dynamic movement primitives; they started by viewing control policy design. Control policy was defined as a rather general function dependent upon the mechanical system's state, environmental parameters, and time. Reinforcement learning techniques can learn this function for given cost criteria. However, it is still computationally intractable for modern computers, too (Bellman 1957). Proportional-derivate point attractor dynamics generate trajectories that work fine for industrial application, but they lack flexibility. The dynamic environment requires a control policy to generalize from previously acquired knowledge. From the viewpoint of statistical learning, the desired trajectory generation constitutes a nonlinear function approximation problem. Nonlinear dynamical systems exhibit two elementary behaviors, point attractive and limit cycle, parallel by discrete and rhythmic movement respectively in motor control. It can generate discrete and rhythmic movements. Discrete movements are a change of rest position of the system whereas rhythmic movements encompass the oscillatory behaviors such as the beating of heart, walking, etc. Dual behavior permits us to exploit both spaces: (1.) joint space which is most efficient to generate rhythmic movements (2.) Cartesian space is relevant for the desired reference tasks.

DMPs have two components, a point attractor system and a limit system. Point attractor ensures stability and robustness, whereas a nonlinear function approximates the nonlinear behavior of the desired trajectory. A one-dimensional trajectory is generated by integrating the set of differential equations.

Transformation System

$$\begin{aligned} \tau\dot{v} &= K(g-x) - Dv + (g-x_0)f \\ \tau\dot{x} &= v \end{aligned} \tag{1}$$

where x is the recorded trajectory, v and $\dot{v}$ are numerically computed derivatives of x; x_0 and g are start and end positions respectively; τ 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.

$$f(s) = \frac{\sum_{i=1}^{N} w_i \psi_i(s) s}{\sum_{i=1}^{N} \psi_i(s)} \tag{2}$$

where $\psi_i(s) = e^{-h_i(s-c_i)^2}$ 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.

$$\tau\dot{s} = -\alpha s \tag{3}$$

α 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 τ. Scaling with s ensures that the effect of $f(s)$ decreases monotonically. (Temporal Invariant)

In paper (Ijspeert et al. 2013), authors have demonstrated how the same system can be transformed to accommodate periodic movements.

To learn a movement from demonstration, first, a movement is observed, the observed perceptual data is mapped to joint space. Using the corresponding points(x) and their derivatives are computed for each time step $t = 0 \ldots T$. Second, the canonical system is integrated with an appropriate τ. Using these $f_{target}(s)$ is computed as

$$f_{target}(s) = \frac{-K(g-x) + Dv + \tau\dot{v}}{g - x_0} \tag{4}$$

x_0 and g are first and last value of x[n] respectively. Finding weights w_i is associated with some minimization criterion, $J = \sum_{s=s_0}^{s_T} (f_{target}(s) - f(s))^2$

which can be solved by regression analysis. Learned movement can be generated on the fly, by using the appropriate weights, current state of system x_0 and goal state g.
In paper (Pastor et al. 2009), authors have discussed the limitation of the above 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 $(g_{new} - x0)$ has different sign than $(g_{old} - x)$, lead to mirror of generalisation, which could be unsuitable in Cartesian space.

To deal with the above problems, the authors have presented a different formulation,
Transformation System

$$\begin{aligned} \tau\dot{v} &= K(g - x) - Dv - K(g - x_0)s + Kf(s) \\ \tau\dot{x} &= v \end{aligned} \tag{5}$$

The rest of the things left the same as before, canonical system, the nonlinear function $f(s)$. $K(g - x_0)$ is added for smooth propagation. For the Multivariate case, a multiple transformation system can be stacked together taking phase input from a common canonical system.
The paper (Park et al. 2008) published in IEEE-RAS presented the notion of dynamic potential fields and how DMPs provides a natural extension to obstacle avoidance; minor adjustment in transformation equations can permit those enhancements.

$$\begin{aligned} \tau\dot{v} &= K(g - x) - Dv - K(g - x_0)s + Kf(s) + \phi_(x, v) \\ \tau\dot{x} &= v \end{aligned}$$

The dependence of ϕ on v makes it dynamic. By design, it has the following properties

- Magnitude of potential decreases with the relative distance of x with respect to obstacle.
- Magnitude of the potential field increases with velocity(v) and is zero when the speed of x is zero.
- Magnitude of potential varies with the angle between the current velocity direction of x and the direction towards the obstacle. Magnitude is zero when the angle is over $\pi/2$ (away from the obstacle).

$$U_{dyn}(x, v) = \begin{cases} \lambda(-\cos(\theta))^{\beta} \frac{\|v\|}{p(x)}, \frac{\pi}{2} \leq \theta \leq \pi \\ 0, 0 \leq \theta \leq \pi/2 \end{cases}$$

where λ is constant, accounting for the strength factor of the entire field, β is a constant. The angle θ is taken between current velocity v and the current-position x relative to the position of the obstacle, p(x) is the current position of the end-effector.

$$\cos(\theta) = \frac{v.x}{\|v\| p(x)}$$

In the experiments performed in the original paper (Park et al. 2008), θ was ranging from 0 to π. This might be problematic as for real-world case where θ can vary from $-\pi$ to π.
The obstacle force is the negative gradient of the potential function as

$$\begin{aligned}\phi(x, v) &= -\nabla_x U_{dyn}(x, v) \\ &= \lambda(-cos)^{\beta-1} \frac{\|v\|}{p}(\beta \nabla_x \cos(\theta) - \frac{\cos(\theta)}{p} \nabla_x p)\end{aligned}$$

Above force formulation will respond in the movement of end-effector towards the obstacle, but to prevent the collision of obstacle and intermediary links (Maciejewski and Klein 1985) approach had been discussed in the paper.

Although (Schaal 2006) and (Pastor et al. 2009) presented the notion of DMP, experimentally verified its viability on manipulators, their specific choice of kernel distribution was not mentioned. Optimized kernel distribution outperforms equally spaced kernel distribution, (Lauretti et al. 2017).

3 Discussions

DMPs provide an elegant way to manipulate robotics arms, specifically enabled them to be utilized by non-expert humans. Future efforts will be oriented toward building a library of movements. To reconstruct a movement, only weights are required to be stored, known as the basis functions. Build a framework to decompose higher-level tasks such as pouring water into smaller subtasks, then choosing appropriate weights and sequencing the movement to form the desired behavior.

References

Bellman, Richard (1957). *Dynamic Programming*. Dover Publications. ISBN: 9780486428093.

Billard, A. and D. Grollman (2013). "Robot learning by demonstration". In: *Scholarpedia* 8.12. revision #138061, p. 3824. DOI: 10.4249/scholarpedia.3824.

Ijspeert, Auke Jan et al. (2013). "Dynamical movement primitives: learning attractor models for motor behaviors". In: *Neural computation* 25.2, pp. 328–373.

Lauretti, Clemente et al. (2017). "Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics". In: *IEEE Robotics and Automation Letters* 2.3, pp. 1375–1382.

Maciejewski, Anthony A and Charles A Klein (1985). "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments". In: *The international journal of robotics research* 4.3, pp. 109–117.

Park, Dae-Hyung et al. (2008). "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields". In: *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 91–98.

Pastor, Peter et al. (2009). “Learning and generalization of motor skills by learning from demonstration”. In: *2009 IEEE International Conference on Robotics and Automation*. IEEE, pp. 763–768.

Schaal, Stefan (2006). “Dynamic movement primitives-a framework for motor control in humans and humanoid robotics”. In: *Adaptive motion of animals and machines*. Springer, pp. 261–280.