Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Bonn-Aachen
International Center for
Information Technology

Dynamic Motion Primitives

Research and Development Project

November 21, 2018

Abhishek Padalkar

Motivation

- Humans **learn** variety of motions and use them in similar situations.
- Human motions consist of motion primitives.
- Concept of motion primitives can be adopted for robots.
- Learned motion primitives can be combined to do complex task.
- Several approaches are available for learning motion primitives.

Advantages of DMP

- It is a model free learning approach.
- Any arbitrary trajectory can be learned in end-effector space as well as in joint space.
- Here learning is linear regression, so it does not need large dataset. One trajectory is sufficient ideally.
- Trajectories can be scaled in space as well as in time.
- Trajectory evolves as robot actually moves along the trajectory. Hence on-line modifications in the trajectory are possible.

Formulation of DMP

$$\tau\dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x) \tag{1}$$

$$\tau\dot{y} = z \tag{2}$$

$$f(x) = \frac{\sum_{i=1}^{N}\psi_i(x)w_i}{\sum_{i=1}^{N}\psi_i(x)}x(g - y_0) \tag{3}$$

where,

$$\psi_i = \exp(-\frac{1}{2\sigma_i^2}(x - c_i)^2) \tag{4}$$

and,

$$\tau\dot{x} = -\alpha_x x \tag{5}$$

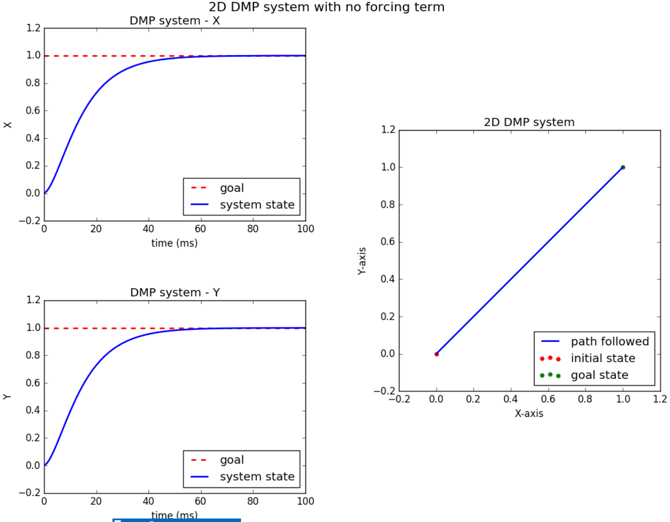
1

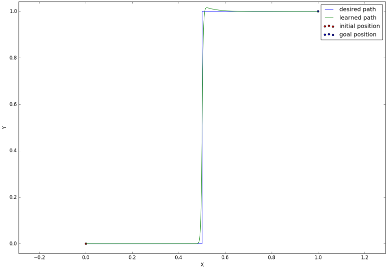
[1] Formulation of DMP taken from [1]

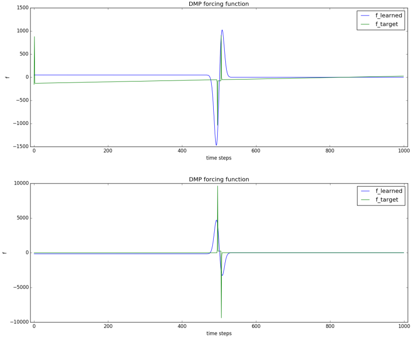
- Second order differential equation representing **damped mass spring system.**
- Non-linear term *f(x)* modifies the acceleration and hence characterizes the motion.
- *f(x)* is normalized weighted sum of equally spaced Gaussian functions.
- Learning a motion primitive means learning the wights w_i.
- Phase variable **x** ensures the synchronization between multiple degrees of freedom.

Working of DMP

2D DMP system with no forcing term

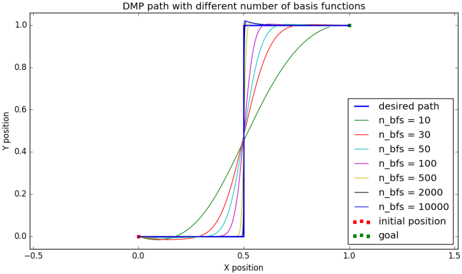



desired path
learned path
initial position
goal position
X
Y


DMP forcing function
f_learned
f_target
time steps
f
DMP forcing function
f_learned
f_target
time steps
f

Analysis of the effects of the parameters used in DMP

Effect of the number of basis functions on the trajectory approximation



$$f(x) = \frac{\sum_{i=1}^{N} \psi_i(x) w_i}{\sum_{i=1}^{N} \psi_i(x)} x(g - y_0) \qquad (6)$$

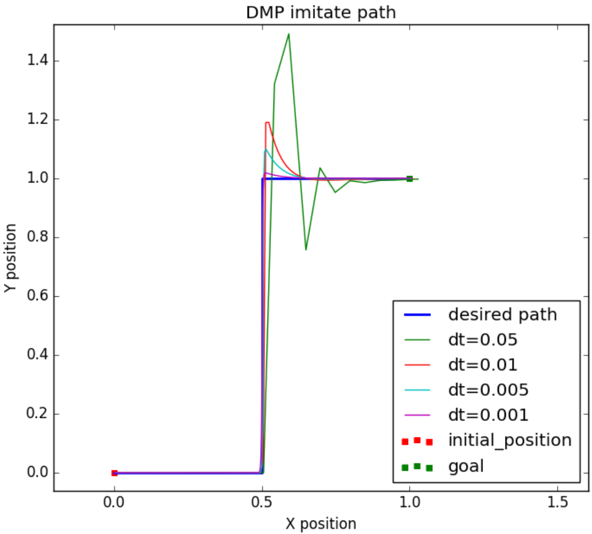
Error in mimicking the trajectory

n_bfs	10	30	50	100	500	2000	10000
Error	0.093	0.038	0.021	0.009	0.002	0.001	0.001

Table 1: Error in mimicking the trajectory

- Higher number of basis functions results in better approximation
- More number of basis functions learn more noise

Effect of the time step on the trajectory approximation

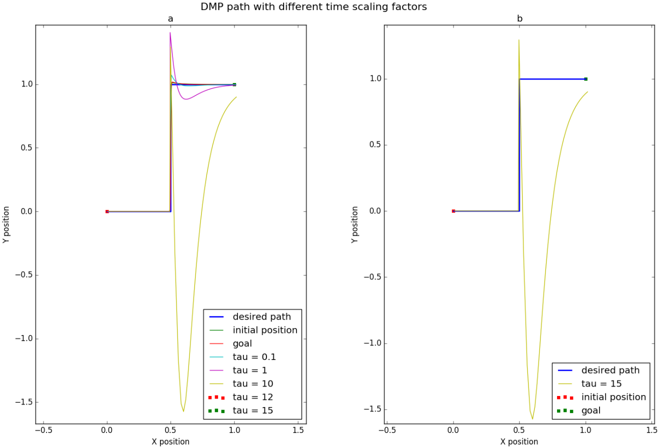


Effect of the time step on the trajectory approximation

Time step size	0.05	0.01	0.005	0.001
Error	0.062	0.017	0.011	0.007

Table 2: Error in mimicking the trajectory

Effect of the time scaling factor on the trajectory approximation



$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x) \quad (7)$$

$$\tau \dot{y} = z \quad (8)$$

Effect of the time scaling factor on the trajectory approximation

Time scaling factor (tau)	0.1	1	10	12	15
Error	0.007	0.007	0.010	0.036	0.292

Table 3: Error in mimicking the trajectory

Inverse Kinematic Solver

- Motion is learned in Cartesian space.
- Need of inverse kinematic solver to convert Cartesian velocity commands to joint velocity commands.
- Limitations of manipulators due to 5 degrees of freedom.
- Weighted Damped Least Square method for calculating joint velocities.

Whole Body Motion Control

$$m_{cap} = \frac{(\sigma_{min} - \sigma_l)}{(\sigma_h - \sigma_l)} \tag{9}$$

$$b_{cap} = \frac{(d - d_l)}{(d_h - d_l)} \tag{10}$$

$$v_{ee} = \frac{m_{cap}}{m_{cap} + b_{cap}}.v \tag{11}$$

$$v_b = \frac{b_{cap}}{m_{cap} + b_{cap}}.v \tag{12}$$

Where,

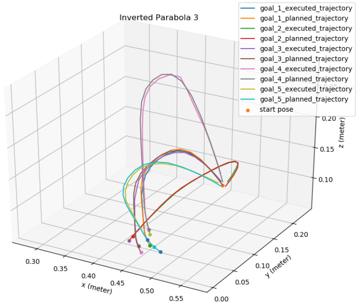
σ_{min} is the smallest sigma value,
σ_l is the lower limit on σ_{min},
σ_h is the upper limit on σ_{min},
d is the distance of the obstacle from base,
d_l is the lower limit on d,
d_h is the upper limit on d,
v is the desired velocity of end-effector in global frame of reference,
v_{ee} is the velocity command for end-effector of the manipulator in global frame of reference,
v_b is the velocity command for mobile base in global frame of reference.

Experiments

- Experiments were conducted on Kuka YouBot and Toyota HSR to evaluate :
  - ***Learning from Demonstration*** framework
  - ***Whole body motion control***
- Experiments on Kuka YouBot :
  - 3 inverse parabolic trajectories
  - 1 step function like trajectory
  - 1 square function trajectory
  - 1 inverse parabolic trajectory (whole body motion)
  - 2 square wave trajectories (whole body motion)
- Experiments on Toyota HSR:
  - Sequencing 2 DMPs for pick and place task
  - Grasping an object

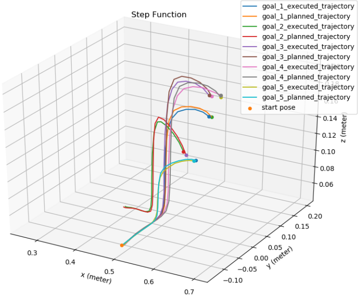
Results

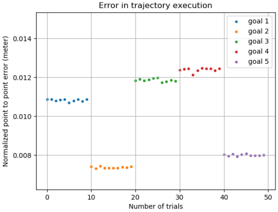
Inverse Parabolic Trajectory



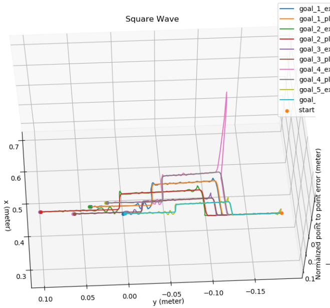
Results

Step Function Trajectory



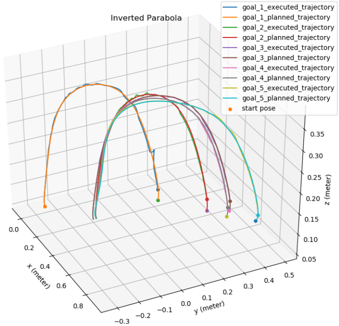


Results


Square Wave
goal_1_executed_trajectory
goal_1_planned_trajectory
goal_2_executed_trajectory
goal_2_planned_trajectory
goal_3_executed_trajectory
goal_3_planned_trajectory
goal_4_executed_trajectory
goal_4_planned_trajectory
goal_5_executed_trajectory
goal_
start
x (meter)
y (meter)
0.7
0.6
0.5
0.4
0.3
0.10
0.05
0.00
−0.05
−0.10
−0.15


Error in trajectory execution
Normalized point to point error (meter)
Number of trials
goal 1
goal 2
goal 3
goal 4
goal 5
0.004
0.002
0.000
−0.002
0
10
20
30
40
50

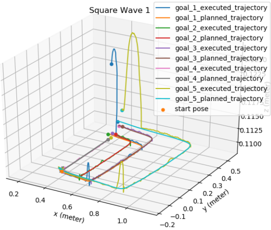
Results



Inverted Parabola
goal_1_executed_trajectory
goal_1_planned_trajectory
goal_2_executed_trajectory
goal_2_planned_trajectory
goal_3_executed_trajectory
goal_3_planned_trajectory
goal_4_executed_trajectory
goal_4_planned_trajectory
goal_5_executed_trajectory
goal_5_planned_trajectory
start pose
x (meter)
y (meter)
z (meter)

Error in trajectory execution
goal 1
goal 2
goal 3
goal 4
goal 5
Normalized point to point error (meter)
Number of trials

Results

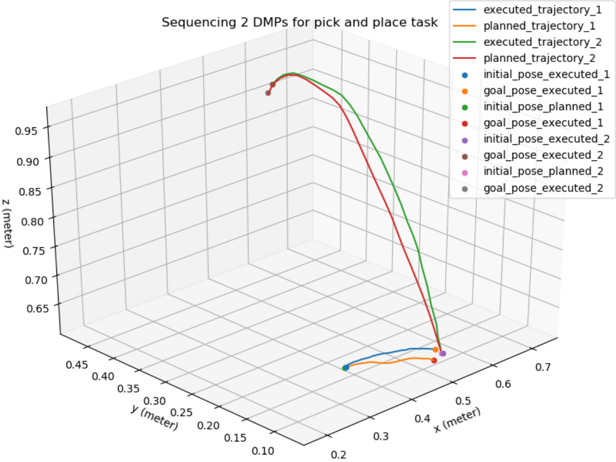


Square Wave 1
goal_1_executed_trajectory
goal_1_planned_trajectory
goal_2_executed_trajectory
goal_2_planned_trajectory
goal_3_executed_trajectory
goal_3_planned_trajectory
goal_4_executed_trajectory
goal_4_planned_trajectory
goal_5_executed_trajectory
goal_5_planned_trajectory
start pose
x (meter)
y (meter)
z (meter)

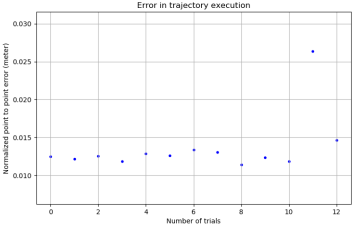
Error in trajectory execution
goal 1
goal 2
goal 3
goal 4
goal 5
Normalized point to point error (meter)
Number of trials

Results

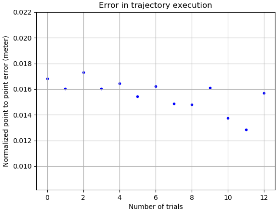
Sequencing Two DMPs for object pick and place



Results



Error in trajectory execution
Normalized point to point error (meter)
0.030
0.025
0.020
0.015
0.010
0
2
4
6
8
10
12
Number of trials



Error in trajectory execution
Normalized point to point error (meter)
0.022
0.020
0.018
0.016
0.014
0.012
0.010
0
2
4
6
8
10
12
Number of trials

Conclusion

- Development of *Learning from Demonstration* framework for robot programming using DMPs.
- Analysis of the effects of the parameters used in DMPs on trajectory approximation.
- Evaluation of *Learning from Demonstration* framework on KUKA YouBot and Toyota HSR.
- Development and evaluation of whole body motion control architecture.
- Gathering insights of working of DMPs as well as their capabilities and limitations.
- Integration into current software solution for manipulation for Toyota HSR (replacing MoveIt!).

Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal.
Dynamical movement primitives: learning attractor models for motor behaviors.
Neural computation, 25(2):328–373, 2013.