Dynamic Movement Primitives (DMP)

DMP in a nutshell

- A method of trajectory control/planning[3]
- Complex movements are considered as composed of sets of primitive action 'building blocks'
  - Executed in sequence *and/or* in parallel
  - DMPs are a proposed mathematical formalization of these primitives.
- It can represent *nonlinear* motion with a set of differential equations
- These equations can be adapted to generate any movement trajectory

[3] Schaal, Stefan. "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics." Adaptive Motion of Animals and Machines. Springer. 261-280.

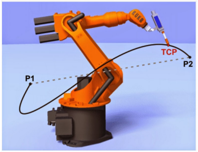
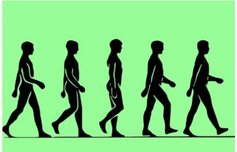
Formulation of DMP

DMP is for generating a control signal to guide the real system

Underlying Idea[4]

1. Take a dynamical system with well specified stable behaviour
2. Add another term that makes it follow some interesting trajectory

There are two kinds of DMP:

Discrete DMP	Rhythmic DMP
	
Point to Point Motion	Human Walking

[4] Ijspeert, A. J., J. Nakanishi, and S. Schaal. "Learning control policies for movement imitation and movement recognition." Neural information processing system. Vol. 15. 2003.

Formulation of DMP

Let's start with point attractor dynamics[5]

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) \tag{1}$$

where:

- y is system state and g is goal state
- α and β are gain terms

Now add a forcing term f on eq(1) that will let us to modify this trajectory

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f \tag{2}$$

f is nonlinear function defined over time.

* The introduced system in eq(2) is called the *canonical system.*
* It is denoted by x as $\dot{x} = -\alpha_x x$

[5] Ijspeert, Auke Jan, Jun Nakanishi, and Stefan Schaal. "Movement imitation with nonlinear dynamical systems in humanoid robots." Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. Vol. 2. IEEE, 2002.

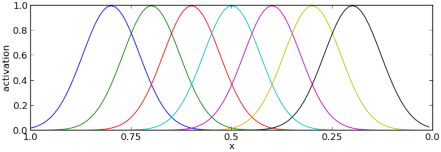
Formulation of DMP

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

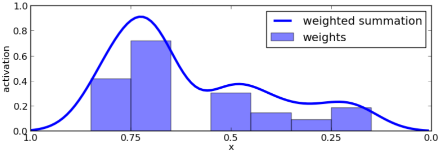
$$f(x, g) = \frac{\sum_{i=1}^{N} \psi_i w_i}{\sum_{i=1}^{N} \psi_i} x(g - y_0) \tag{3}$$

where:

- y_0 is the initial state of the system
- w_i is a weighting for a given basis function[6] ψ_i
- $\psi_i = \exp\left(-h_i (x - c_i)^2\right)$ is Gaussian with mean c_i and variance h_i



ψ Activation



Weighted Summation

[6] Bishop, C. M. (2006). "Pattern recognition and machine learning". New York: Springer.

Imitating a desired path

The forcing term f which affects the system acceleration, can be re-written as

$$\mathbf{f}_d = \ddot{\mathbf{y}}_d - \alpha_y(\beta_y(g - \mathbf{y}) - \dot{\mathbf{y}}) \tag{4}$$

where

- $\mathbf{y}_d$ is desired trajectory, given by $\ddot{\mathbf{y}}_d = \frac{\partial}{\partial t}\dot{\mathbf{y}}_d = \frac{\partial}{\partial t}\frac{\partial}{\partial t}\mathbf{y}_d$

Forcing term

- Comprised of weighted summation of basis functions
- We can use optimization technique like LWR[7]
  - To choose the weights over our basis functions
  - Minimize

$$\Sigma_t \psi_i(t)(f_d(t) - w_i(x(t)(g - y_0)))^2 \tag{5}$$

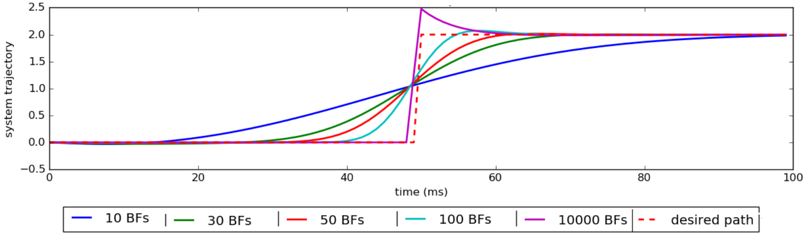
[7] Cleveland, William S. "Robust locally weighted regression and smoothing scatterplots." Journal of the American statistical association 74.368 (1979): 829-836.

Imitating a desired path

The solution[8] of eq(5) $w_i = \frac{\mathbf{s}^T \boldsymbol{\psi}_i \mathbf{f}_d}{\mathbf{s}^T \boldsymbol{\psi}_i \mathbf{s}}$

where

$$\mathbf{s} = \begin{pmatrix} x_{t_0}(g - y_0) \\ \vdots \\ x_{t_N}(g - y_0) \end{pmatrix} \text{ and } \boldsymbol{\psi}_i = \begin{pmatrix} \psi_i(t_0) & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \psi_i(t_n) \end{pmatrix}$$



[8] Schaal, Stefan, Christopher G. Atkeson, and Sethu Vijayakumar. "Scalable techniques from nonparametric statistics for real time robot learning." Applied Intelligence 17.1 (2002): 49-60.