Learning Manipulation Patterns

Robert Haschke

Neuroinformatics Group

February 29, 2012

Motivation

- shift to reactive motion generation ...
- ... using dynamical systems
- robustify movement skeletons

Motivation

- shift to reactive motion generation ...
- ... using dynamical systems
- robustify movement skeletons

- What are suitable motion representations?

- How can we improve by explorative learning?

Motivation

- shift to reactive motion generation ...
- ... using dynamical systems
- robustify movement skeletons

- What are suitable motion representations?
  Dynamic Movement Primitives (DMP)
- How can we improve by explorative learning?
  Policy Improvement with Path Integrals (PI^2)

Dynamic Movement Primitives

Ijspeert, Nakanishi, Schaal, ICRA'02

- motion as evolution of dynamical system
- basis: spring-damper-system

$$\tau \dot{v}_t = K(g - x_t) - D\, v_t$$
$$\tau \dot{x}_t = v$$
or
$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t)$$

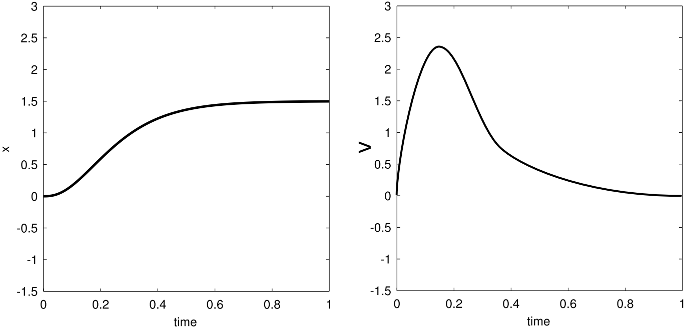
- x - current position
- x - current velocity
- g - goal of motion
- choose K and D to have critical damping

Dynamic Movement Primitives

Ijspeert, Nakanishi, Schaal, ICRA'02

- spring-damper system generates basic motion towards goal

$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t)$$

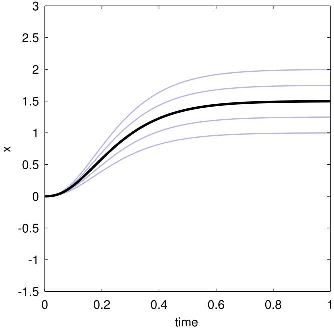


Dynamic Movement Primitives

Ijspeert, Nakanishi, Schaal, ICRA'02

- spring-damper system generates basic motion towards goal

$$\tau \ddot{x}_t = \alpha(\beta(\underset{\Uparrow}{g} - x_t) - \dot{x}_t)$$

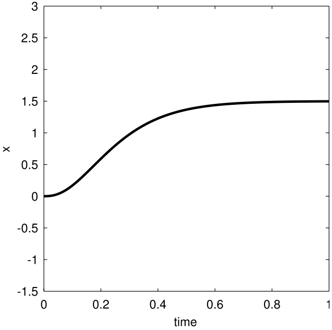


Dynamic Movement Primitives

Ijspeert, Nakanishi, Schaal, ICRA'02

- new idea: add external force to represent complex trajectory shapes

$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T \boldsymbol{\theta}$$

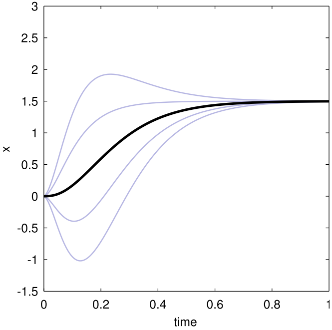


Dynamic Movement Primitives

Ijspeert, Nakanishi, Schaal, ICRA'02

- new idea: add external force to represent complex trajectory shapes

$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T \underset{\Uparrow}{\boldsymbol{\theta}}$$

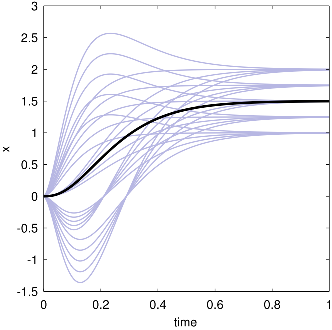


Dynamic Movement Primitives

Ijspeert, Nakanishi, Schaal, ICRA'02

- new idea: add external force to represent complex trajectory shapes

$$\tau \ddot{x}_t = \alpha(\beta(\underset{\Uparrow}{g} - x_t) - \dot{x}_t) + \mathbf{g}_t^T \underset{\Uparrow}{\boldsymbol{\theta}}$$



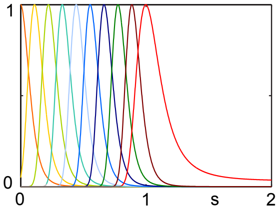
External Force

$$f(s) = \mathbf{g}_t^T \boldsymbol{\theta} = \frac{\sum_i \theta_i \psi_i(s)}{\sum_i \psi_i(s)} \cdot (g - x_0) \cdot s$$

- weighted sum of basis functions ψ_i

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

- Gaussians
- c_i logarithmically distributed in $[0 \ldots 1]$



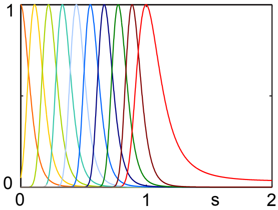
from Schaal et al, Progress Brain Research, 2007

External Force

$$f(s) = \mathbf{g}_t^T \boldsymbol{\theta} = \frac{\sum_i \theta_i \psi_i(s)}{\sum_i \psi_i(s)} \cdot (g - x_0) \cdot s$$

- weighted sum of basis functions ψ_i
- soft-max

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



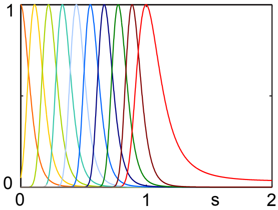
from Schaal et al, Progress Brain Research, 2007

External Force

$$f(s) = \mathbf{g}_t^T \boldsymbol{\theta} = \frac{\sum_i \theta_i \psi_i(s)}{\sum_i \psi_i(s)} \cdot (g - x_0) \cdot s$$

- weighted sum of basis functions ψ_i
- soft-max
- amplitude scaled by initial distance to goal

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



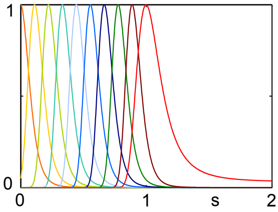
from Schaal et al, Progress Brain Research, 2007

External Force

$$f(s) = \mathbf{g}_t^T \boldsymbol{\theta} = \frac{\sum_i \theta_i \psi_i(s)}{\sum_i \psi_i(s)} \cdot (g - x_0) \cdot s$$

- weighted sum of basis functions ψ_i
- soft-max
- amplitude scaled by initial distance to goal
- influence weighted by *canonical time* $s \to 0$

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



from Schaal et al, Progress Brain Research, 2007

Canonical System

$$f(s) = \mathbf{g}_t^T \boldsymbol{\theta} = \frac{\sum_i \theta_i \psi_i(s)}{\sum_i \psi_i(s)} \cdot (g - x_0) \cdot s$$

- decouple external force from spring-damper evolution
- new phase / time variable s

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

- s initially set to 1 ...
- ... exponentially converges to 0

Canonical System

$$f(s) = \mathbf{g}_t^T \boldsymbol{\theta} = \frac{\sum_i \theta_i \psi_i(s)}{\sum_i \psi_i(s)} \cdot (g - x_0) \cdot s$$

- decouple external force from spring-damper evolution
- new phase / time variable s

$$\tau \dot{s} = -\alpha \cdot s \cdot \frac{1}{1 + \alpha_c \cdot (x_{actual} - x_{expected})^2}$$

- s initially set to 1 ...
- ... exponentially converges to 0
- pause influence of force on perturbations

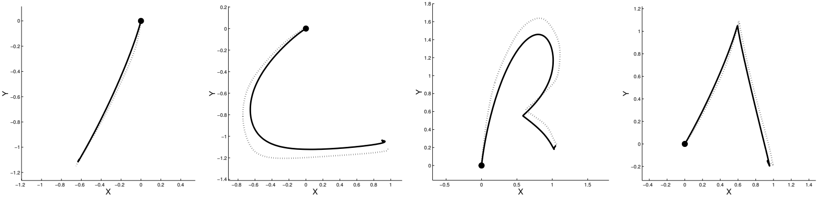
Properties

$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T \boldsymbol{\theta}$$
$$\tau \dot{s} = -\alpha \cdot s$$
$$f(s) = \mathbf{g}_t^T \boldsymbol{\theta} = \frac{\sum_i \theta_i \psi_i(s)}{\sum_i \psi_i(s)} \cdot (g - x_0) \cdot s$$

- convergence to goal g
- motions are self-similar for different goal or start point
- coupling of multiple DOF through canonical phase s
- adapt τ for temporal scaling
- robust to perturbations due to attractor dynamics
- decoupling basic goal-directed motion from task-specific trajectory “shape”
- weights θ_i can be learned with linear regression

Learning from Demonstration

- record motion $x(t), \dot{x}(t), \ddot{x}(t)$
- choose τ to match duration
- integrate canonical system $\rightarrow s(t)$
- $f_{target}(s) = \tau\ddot{x}(t) - \alpha(\beta(g - x(t)) - \dot{x}(t))$
- minimize $E = \sum_s (f_{target}(s) - f(s))^2$
  with regression



from Ijspeert, Nakanishi, Schaal, ICRA'02

Periodic Motion

- replace canonical system by limit cycle oscillator

$$\tau\dot{\phi} = 1 \mod 2\pi$$
$$f(\phi, A) = A \cdot \frac{\sum_i \theta_i \psi_i(\phi)}{\sum_i \psi_i(\phi)}$$
$$\psi_i(\phi) = \exp\left(h_i \cdot (cos(\phi - c_i) - 1)\right)$$

- ϕ - phase of oscillation
- A - amplitude of oscillation
- ψ_i - van Mises basis functions, i.e. Gaussians living on a circle

Periodic Motion - Examples

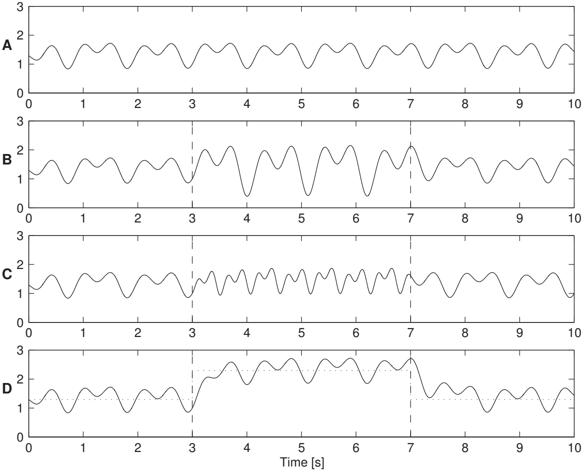


Fig. 10. Modification of the learned rhythmic drumming pattern (flexion/extension of the right elbow, R_EB). (A) Trajectory learned by the rhythmic DMP; (B) temporary modification with A’ 2A in Eq. (16); (C): temporary modification with t’ t/2 in Eqs. (9) and (15); (D): temporary modification with g’ g+ 1 in Eq. (9) (dotted line). Modified parameters were applied between t ¼ 3 s and t ¼ 7 s. Note that in all modifications, the movement patterns do not change qualitatively, and convergence to the new attractor under changed parameters is very rapid.

from Schaal et al, Progress in Brain Research, 2007

Robustifying Motions

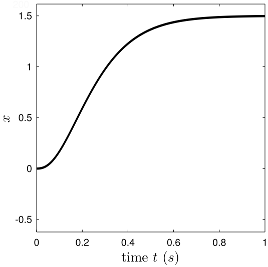
- motion from imitation learning is fragile
- robustify by self-exploration
- competing RL methods:
  - Stefan Schaal:
    PI^2– Policy Improvement with Path Integrals
  - Jan Peters:
    PoWeR – Policy Learning by Weighting Exploration with the Returns

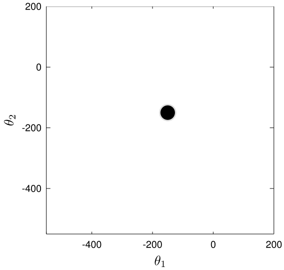
Policy Improvement with Path Integrals – PI^2

Evangelos Theodorou, PhD'11

- Optimize shape parameters $\boldsymbol{\theta}$ w.r.t. cost function J
- Use direct reinforcement learning
  - Exploration directly in policy parameter space $\boldsymbol{\theta}$
- Use Policy Improvement with Path Integrals – PI^2
  - Derived from principles of optimal control
  - Update rule based on cost-weighted averaging (next slide)

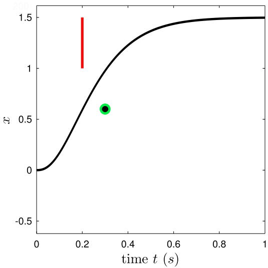
- **Input**: DMP with initial parameters $\boldsymbol{\theta}$

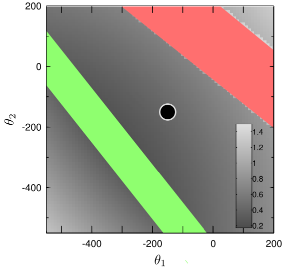




$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T \boldsymbol{\theta}$$

- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J



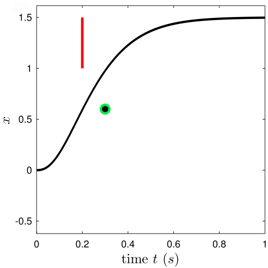


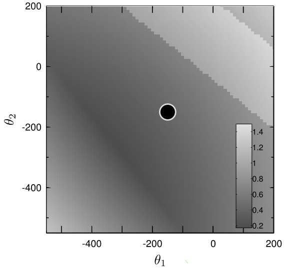
$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T \boldsymbol{\theta}$$

$J(\boldsymbol{\tau}_i)$

- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J
- While (cost not converged)

  Explore





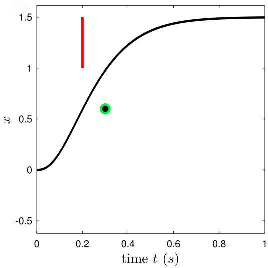
$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T \boldsymbol{\theta}$$

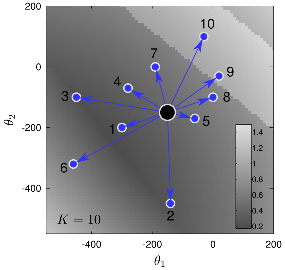
$J(\boldsymbol{\tau}_i)$

- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J
- While (cost not converged)

Explore

sample exploration vectors





$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{i,k})$$

$J(\boldsymbol{\tau}_i)$

$$\boldsymbol{\epsilon}_{i,k} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$$

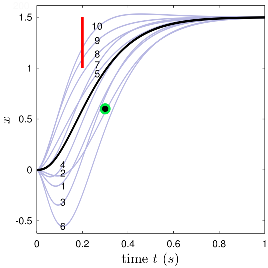
$$\boldsymbol{\theta}_k = \boldsymbol{\theta} + \boldsymbol{\epsilon}_k$$

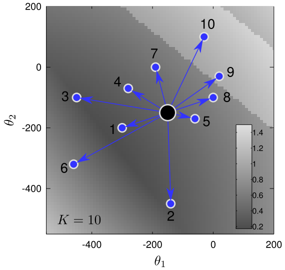
- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J
- While (cost not converged)

  Explore

  sample exploration vectors

  execute DMP





$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{i,k})$$

$$\boldsymbol{\epsilon}_{i,k} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$$

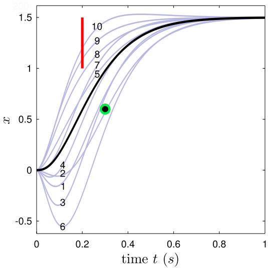
$J(\boldsymbol{\tau}_i)$

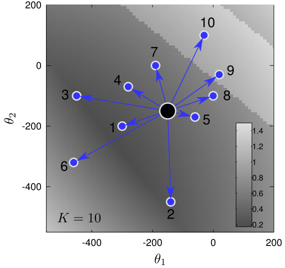
$$\boldsymbol{\theta}_k = \boldsymbol{\theta} + \boldsymbol{\epsilon}_k$$

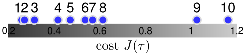
- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J
- While (cost not converged)

Explore
sample exploration vectors
execute DMP
determine cost

Update







$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{i,k})$$

$J(\boldsymbol{\tau}_i)$

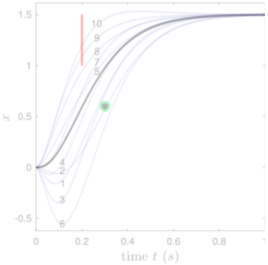
$$\boldsymbol{\epsilon}_{i,k} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$$

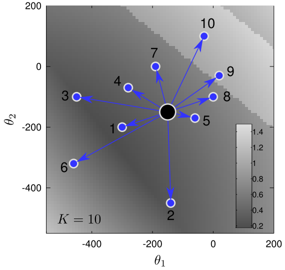
$$\boldsymbol{\theta}_k = \boldsymbol{\theta} + \boldsymbol{\epsilon}_k$$

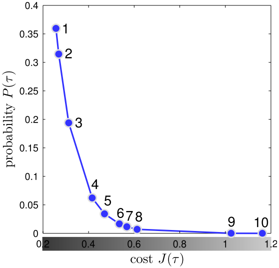
- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J
- While (cost not converged)

Explore
sample exploration vectors
execute DMP
determine cost

Update
weighted averaging
with Boltzmann dist.







$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{i,k})$$

$J(\boldsymbol{\tau}_i)$

$$\boldsymbol{\epsilon}_{i,k} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$$

$$\boldsymbol{\theta}_k = \boldsymbol{\theta} + \boldsymbol{\epsilon}_k$$

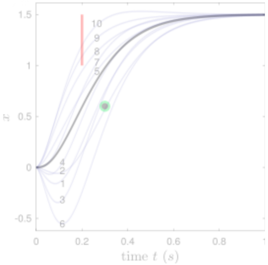
$$P(\boldsymbol{\tau}_{i,k}) = \frac{\exp(-\frac{1}{\lambda} J(\boldsymbol{\tau}_{i,k}))}{\sum_k \exp(-\frac{1}{\lambda} J(\boldsymbol{\tau}_{i,k}))}$$

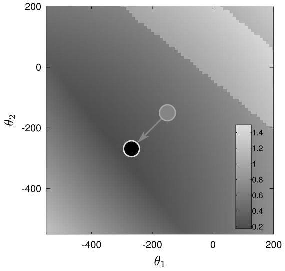
$$\Delta\boldsymbol{\theta}_{t_i} = \sum_{k=1}^{K} P(\boldsymbol{\tau}_{i,k})\boldsymbol{\epsilon}_{i,k}$$

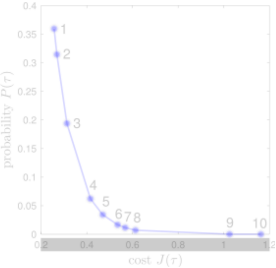
- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J
- While (cost not converged)

Explore
sample exploration vectors
execute DMP
determine cost

Update
weighted averaging
with Boltzmann dist.
parameter update







$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{i,k})$$

$$J(\boldsymbol{\tau}_i)$$

$$\boldsymbol{\epsilon}_{i,k} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$$

$$\boldsymbol{\theta}_k = \boldsymbol{\theta} + \boldsymbol{\epsilon}_k$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$$

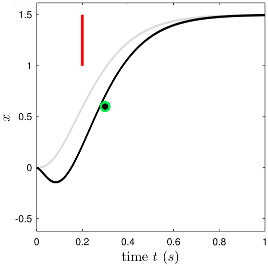
$$P(\boldsymbol{\tau}_{i,k}) = \frac{\exp(-\frac{1}{\lambda} J(\boldsymbol{\tau}_{i,k}))}{\sum_k \exp(-\frac{1}{\lambda} J(\boldsymbol{\tau}_{i,k}))}$$

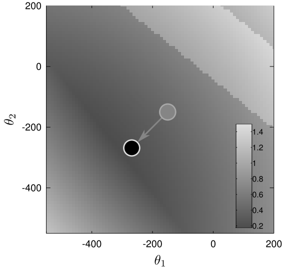
$$\Delta\boldsymbol{\theta}_{t_i} = \sum_{k=1}^{K} P(\boldsymbol{\tau}_{i,k}) \boldsymbol{\epsilon}_{i,k}$$

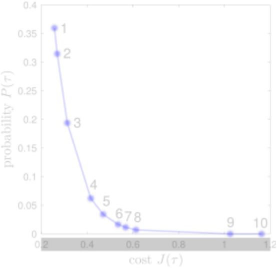
- **Input**: DMP with initial parameters $\boldsymbol{\theta}$, cost function J
- While (cost not converged)

Explore
sample exploration vectors
execute DMP
determine cost

Update
weighted averaging
with Boltzmann dist.
parameter update







$$\tau \ddot{x}_t = \alpha(\beta(g - x_t) - \dot{x}_t) + \mathbf{g}_t^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_{i,k})$$

$J(\boldsymbol{\tau}_i)$

$$\boldsymbol{\epsilon}_{i,k} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$$

$$\boldsymbol{\theta}_k = \boldsymbol{\theta} + \boldsymbol{\epsilon}_k$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$$

$$P(\boldsymbol{\tau}_{i,k}) = \frac{\exp(-\frac{1}{\lambda} J(\boldsymbol{\tau}_{i,k}))}{\sum_k \exp(-\frac{1}{\lambda} J(\boldsymbol{\tau}_{i,k}))}$$

$$\Delta\boldsymbol{\theta}_{t_i} = \sum_{k=1}^{K} P(\boldsymbol{\tau}_{i,k})\boldsymbol{\epsilon}_{i,k}$$

Some Advantages of PI^2

- Applicable to very high-dimensional spaces

  Stulp, F., Buchli, J., Theodorou, E., and Schaal, S. (2010). Reinforcement learning of full-body humanoid motor skills.

  In *10th IEEE-RAS International Conference on Humanoid Robots*. Best paper finalist.

- no gradient $\Rightarrow$ deals with discontinous noisy cost functions
- update $\Delta\boldsymbol{\theta}$ within convex hull of ϵ_k $\Rightarrow$ safe update rule