CS 188 Robotics Week 6

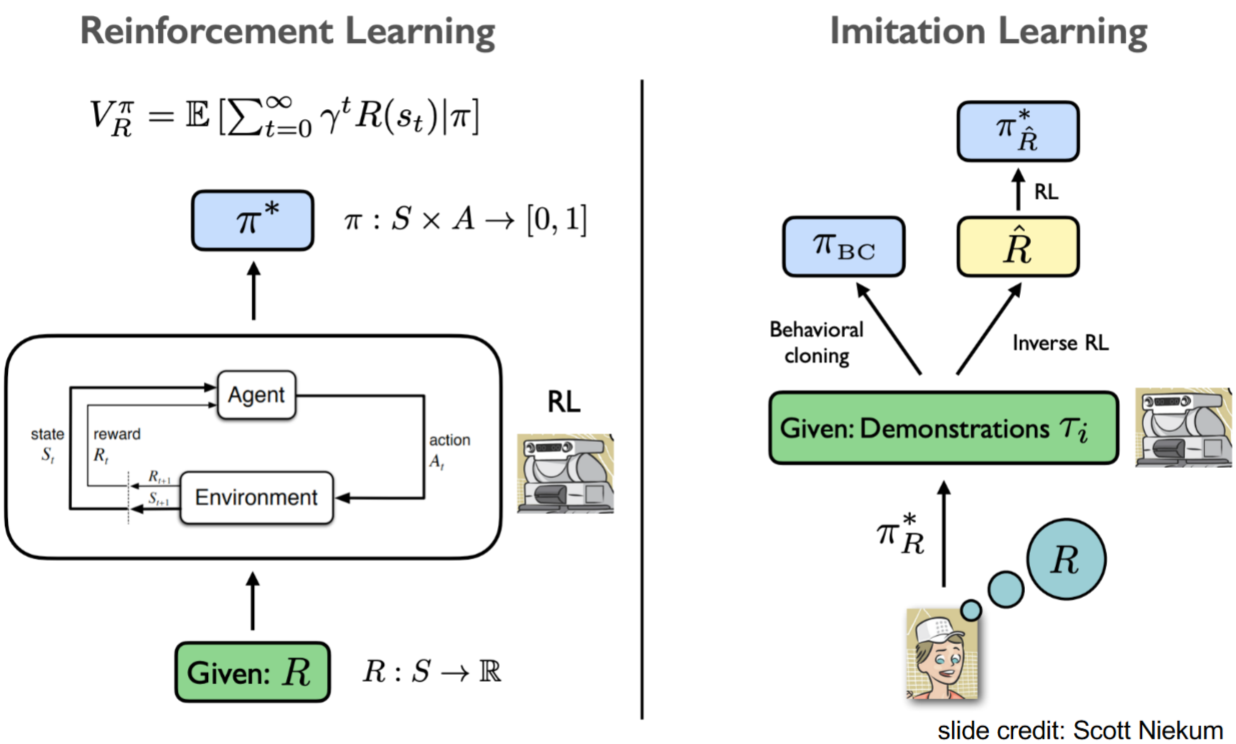
Aidan Jan

May 6, 2025

Imitation Learning

Specifying reward for RL is hard...

- **Reward hacking:** AI system learns to exploit loopholds or unintended behaviors in its reward function to achieve high rewards without actually accomplishing the intended task



Why learn from demonstrations?

- Natural and expressive
- No expert knowledge required
- Valuable human intuition
- Program new tasks as-needed

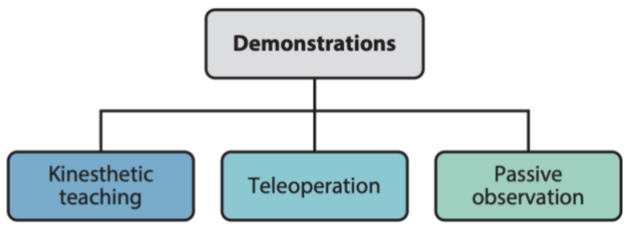
Human babies imitate adults when to learn.

How to Imitate?

Demonstrations to Autonomous Behavior

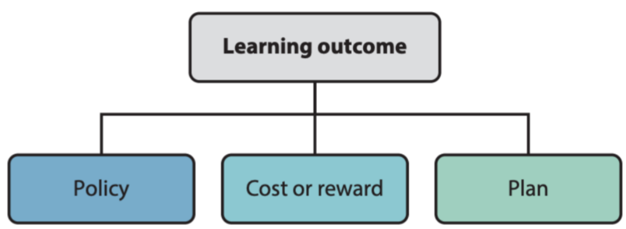
- Dynamic Movement Primitives (DMP): replay the motion
- **Behavior Cloning (BC)**: supervised learning of behavior
  - This is what everyone (as in, robotics companies) is trying to do
- Inverse Reinforcement Learning (IRL): inferring the underlying intent

Types of Demonstrations



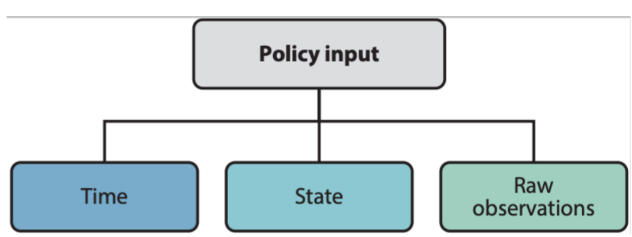
Demonstration	Ease of demonstration	High DOFs	Ease of mapping
Kinesthetic teaching	✓		✓
Teleoperation		✓	✓
Passive observation	✓	✓	

Learning Outcomes



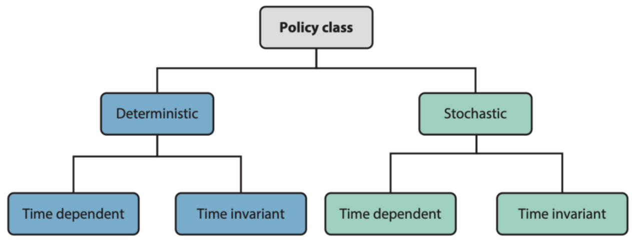
Learning outcome	Low-level control	Action space continuity	Compact representation	Long-horizon planning	Multistep tasks
Policy	✓	✓	✓		
Cost or reward	✓	✓		✓	
Plan			✓	✓	✓

Policy Parameterization



Policy input	Ease of design	Performance guarantees	Robustness to perturbations	Task Variety	Algorithmic efficiency
Time	✓	✓			✓
State		✓	✓		✓
Raw observations	✓		✓	✓	

Policy Class

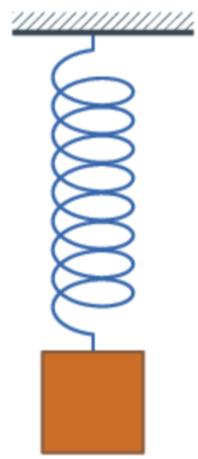


Policy class	**Temporal context**	**Robustness to temporal perturbations**	**Repeatability**	**Multimodal behavior**
Deterministic and time dependent	✓		✓	
Deterministic and time invariant		✓	✓	
Stochastic and time dependent	✓			✓
Stochastic and time invariant		✓		✓

Dynamic Movement Primitives (DMP)

$$\begin{aligned}\tau\dot{v} &= K(\overset{\text{goal}}{\underline{g}} - x) - Dv + (\overset{\text{goal}}{\underline{g}} - \overset{\text{initial state}}{\underline{x_0}})\, f \\ \tau\dot{x} &= v\end{aligned}$$

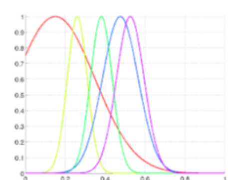
K: spring constant
D: damping term



Non-linear force function:

$$f(s) = \frac{\sum_i w_i \psi_i(s) s}{\sum_i \psi_i(s)} \qquad \psi_i(s) = \exp(-h_i(s-c_i)^2)$$

Gaussian basis functions



canonical system: $\tau\dot{s} = -\alpha\, s$

s: phase variable

Learning:

$$f_{\text{target}}(s) = \frac{-K(g-x) + Dv + \tau\dot{v}}{g - x_0} \qquad f(s) = \frac{\sum_i w_i \psi_i(s) s}{\sum_i \psi_i(s)}$$

$$J = \sum_s \left(f_{\text{target}}(s) - f(s)\right)^2$$

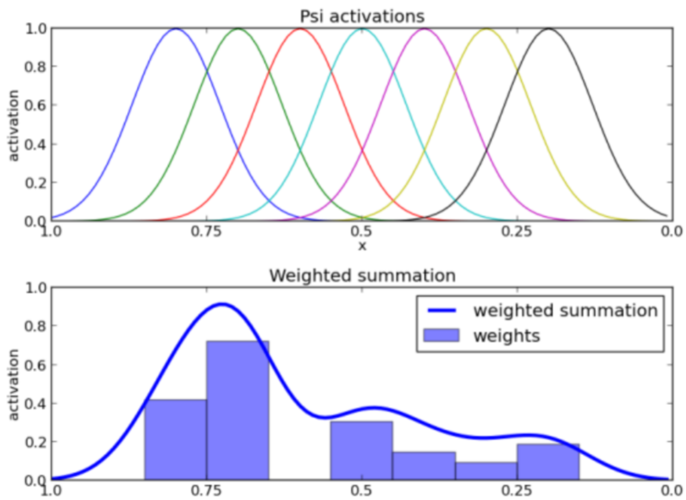
Linear regression

Pastor, Peter, et al. "Learning and generalization of motor skills by learning from demonstration." *2009 IEEE international conference on robotics and automation*. IEEE, 2009.

Characteristics of DMPs

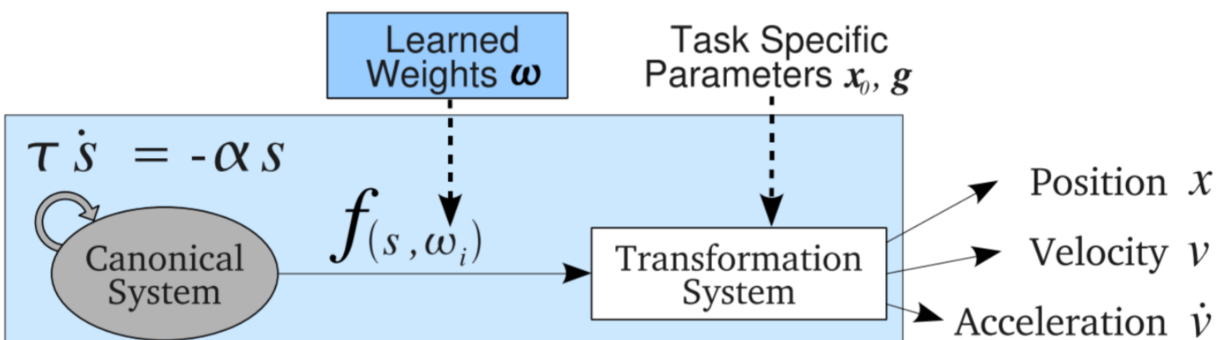
- Convergence to the goal g is guaranteed (for bounded weights) since $f(s)$ vanishes at the end of a movement
- The weights w_i can be learned to generate any desired *smooth* trajectory.
- The equations are spatial and temporal invariant, i.e., movements are self-similar for a change in goal, start point, and temporal scaling without a need to change the weights w_i
- The formulation generates movements which are robust against perturbation due to the inherent attractor dynamics of the equations.

Weighted Sum of Gaussian Basis



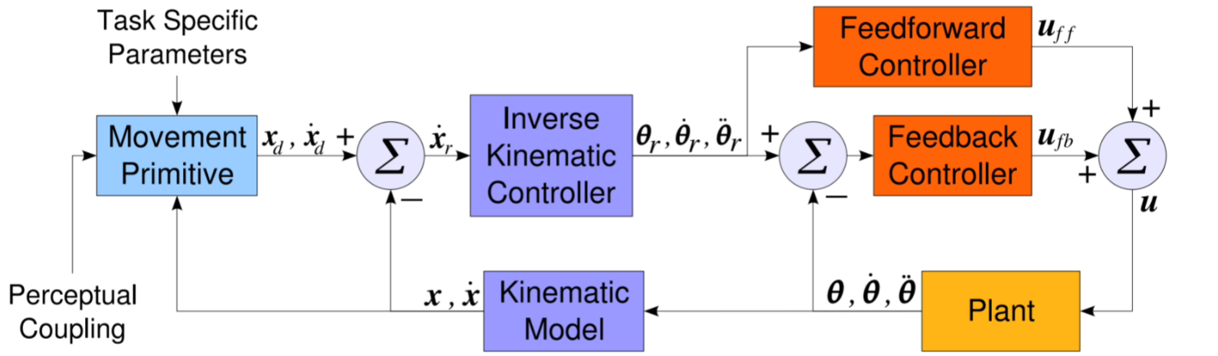
https://studywolf.wordpress.com/2013/11/16/dynamic-movement-primitives-part-1-the-basics/

Dynamic Movement Primitives



Sketch of a one dimensional DMP: the canonical system drives the nonlinear function f which perturbs the transformation system.

Pastor, Peter, et al. "Learning and generalization of motor skills by learning from demonstration." *2009 IEEE international conference on robotics and automation*. IEEE, 2009.

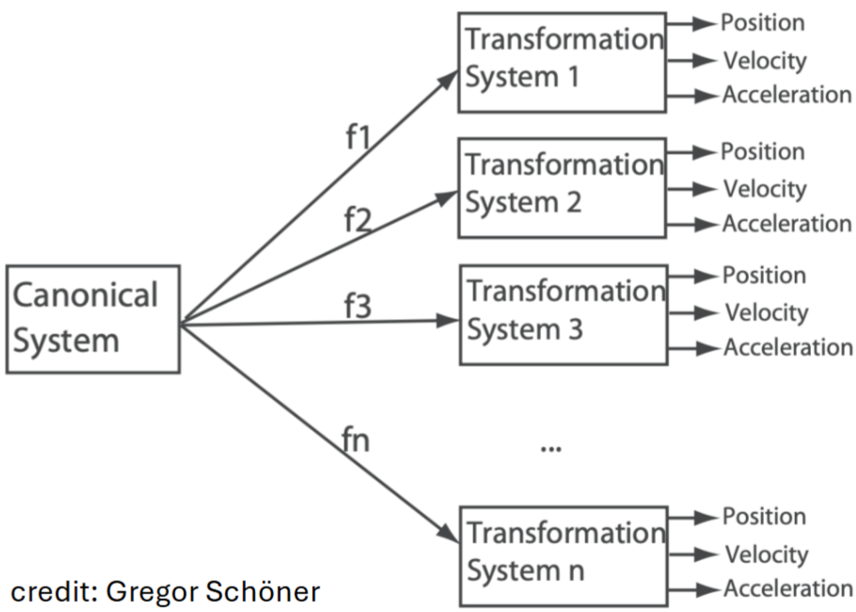


DMP control diagram: the desired task space positions and velocities are xd, x ˙ d, the reference task space velocity commands are x ˙ r , the reference joint positions, joint velocities, and joint accelerations are θr, θ ˙ r, and θ ¨ r.

Pastor, Peter, et al. "Learning and generalization of motor skills by learning from demonstration." *2009 IEEE international conference on robotics and automation*. IEEE, 2009.

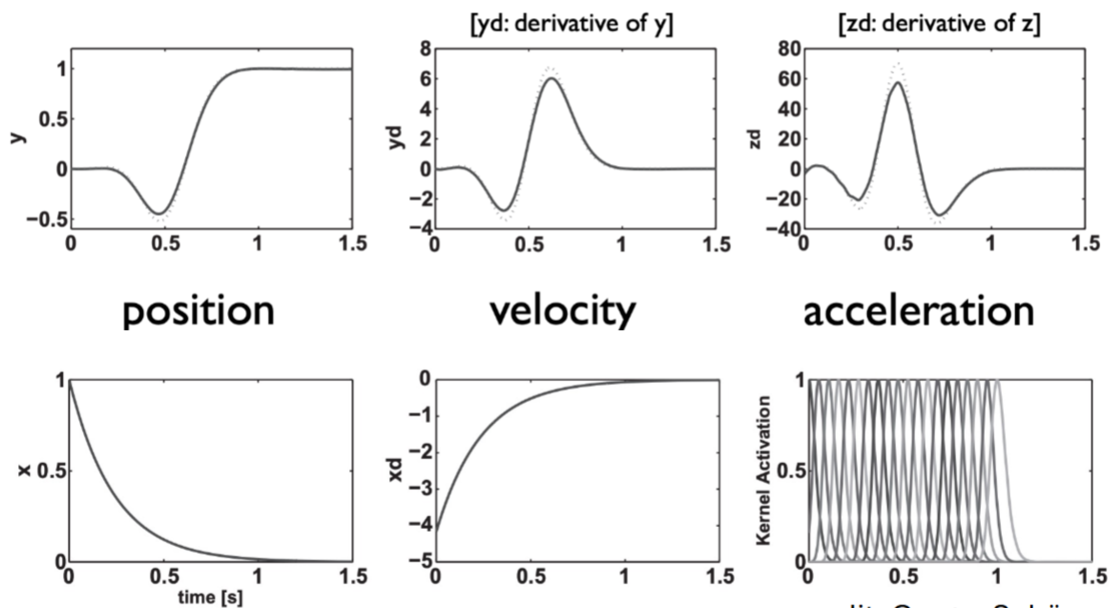
Multidimensional

- one central harmonic oscillator
- multiple transformations



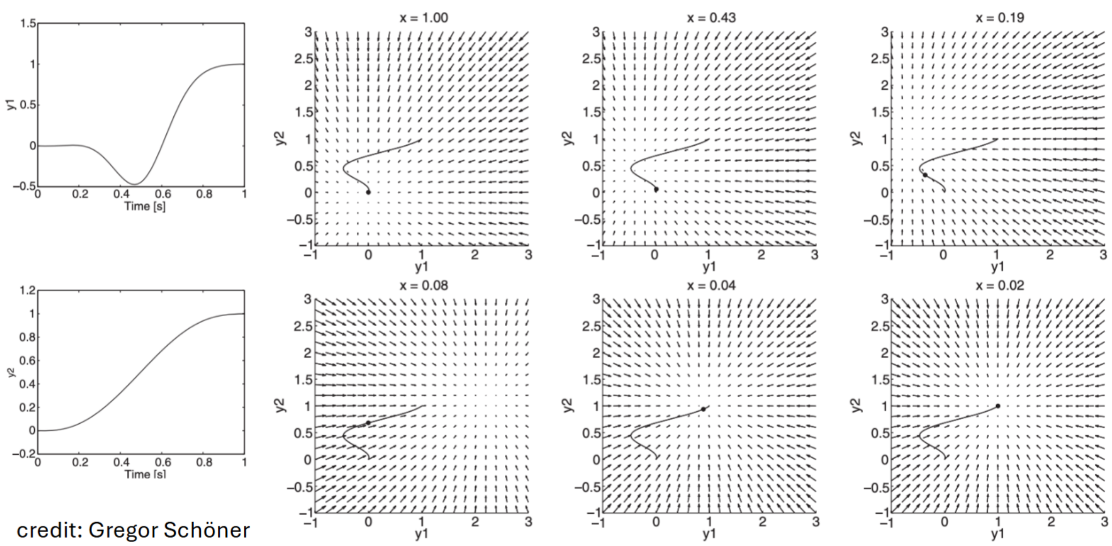
Examples:

- 1 Dimensional:



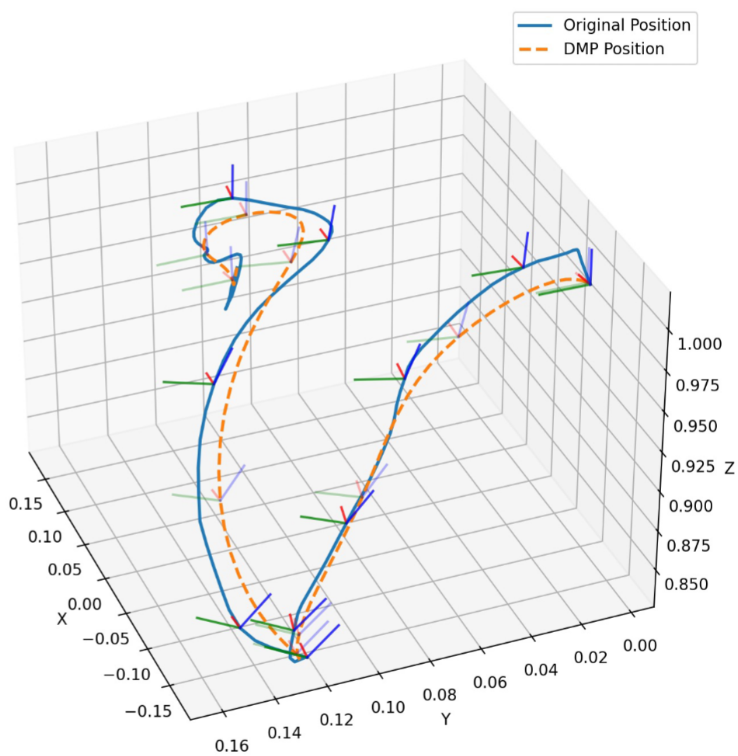
credit: Gregor Schöner

- 2 Dimensional:

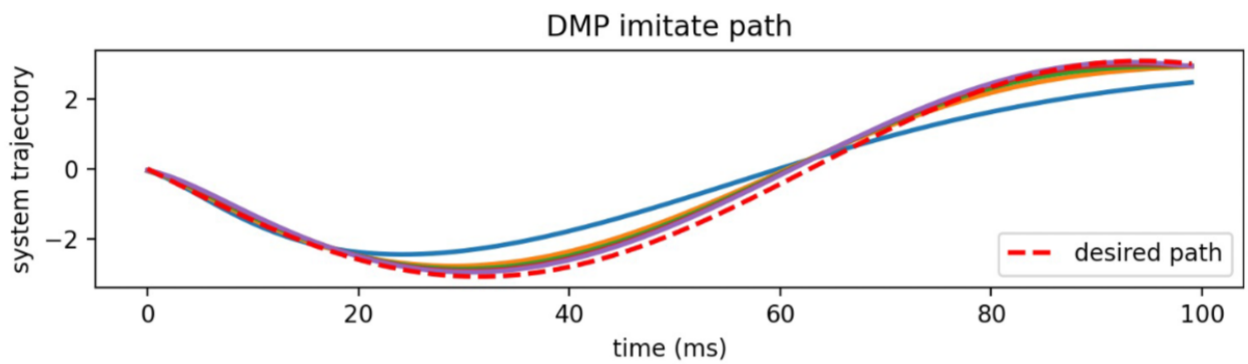


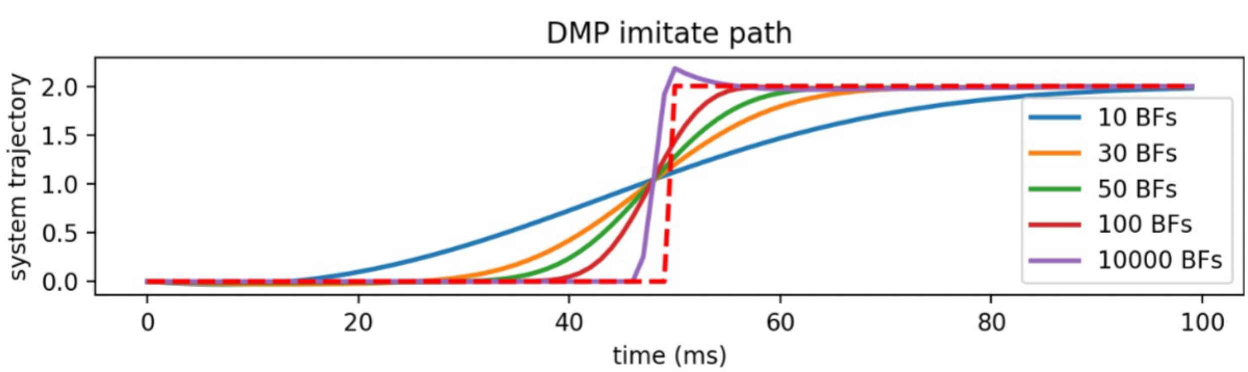
credit: Gregor Schöner

- 3 Dimensional / 6 Dimensional:



Limitations of DMPs





Summary

- DMP enable learning "movement styles" while enabling generalization to new movement targets
- DMP is a purely kinematic account $\Rightarrow$ DMP is not addressing control in that respect, analogy to force-fields is misleading
- DMP addresses timing, but account of coordination is limited
- DMP for different tasks and their combination…?