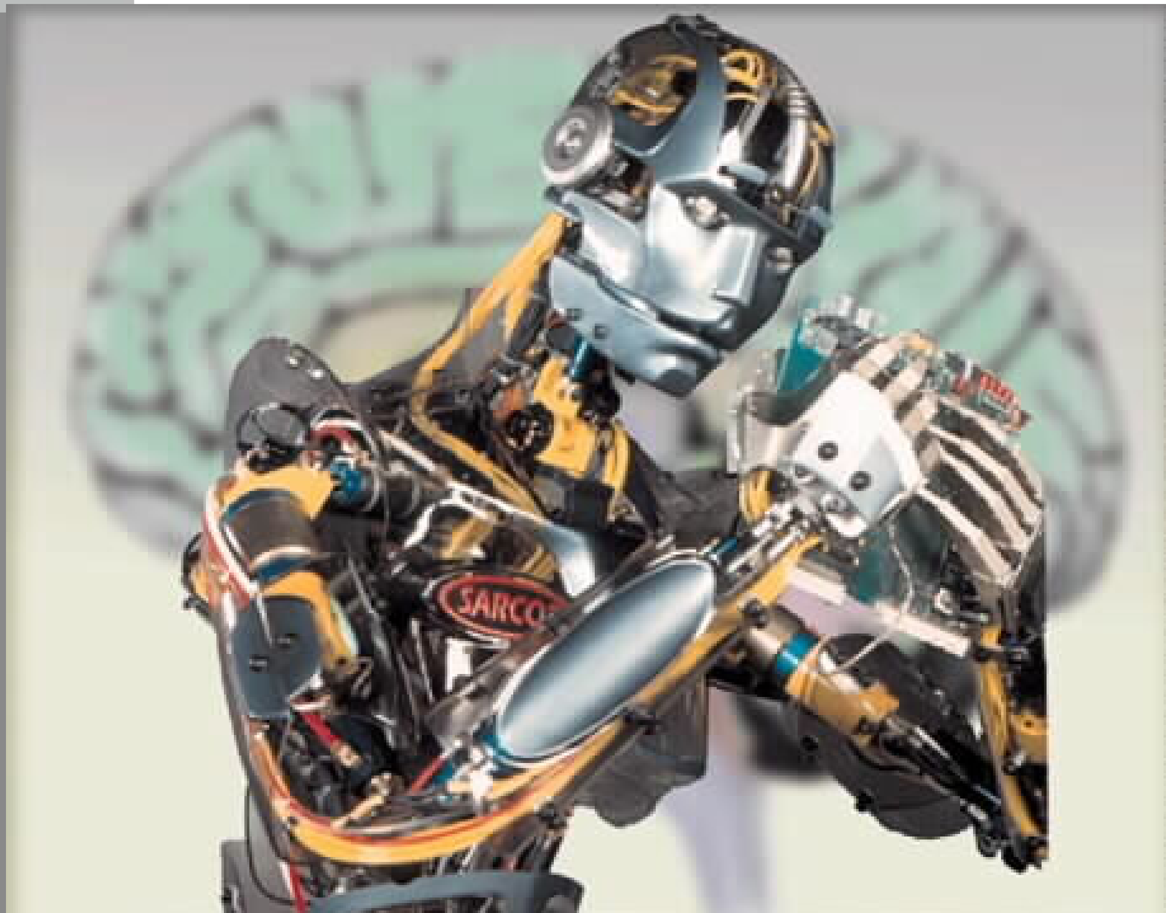
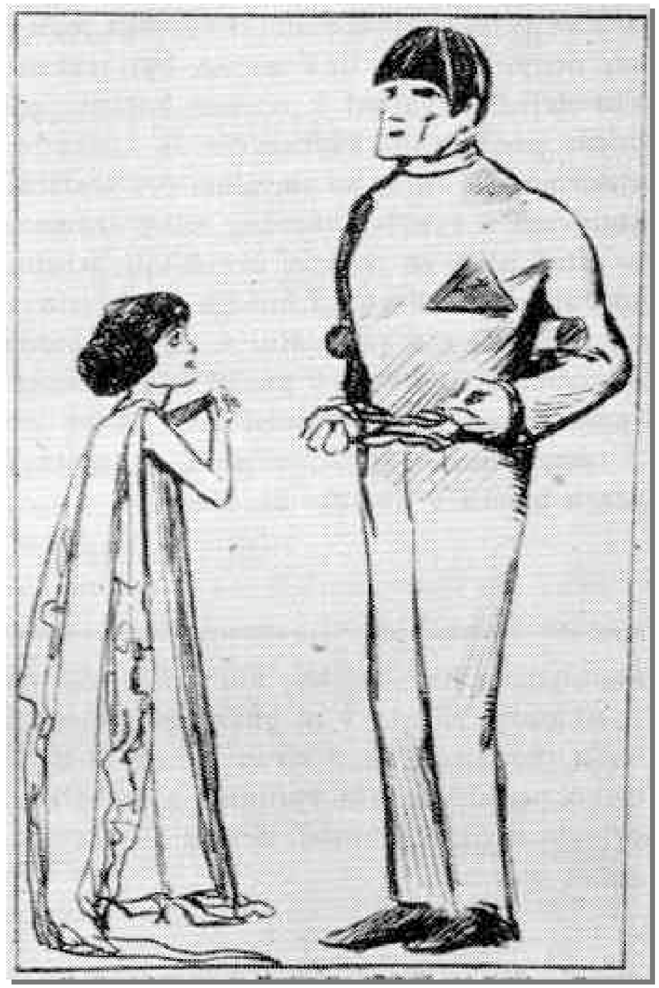


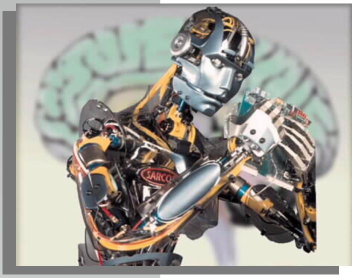
Stefan Schaal
Computer Science & Neuroscience
University of Southern California
and
ATR Human Information
Sciences Laboratory
sschaal@usc.edu
http://www-clmc.usc.edu

Robots That Learn

Robots–The Original Vision

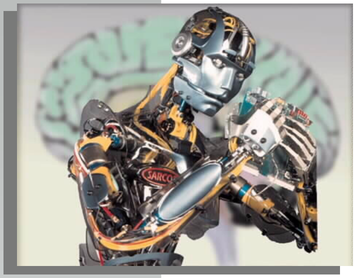
Karel Capek
1920



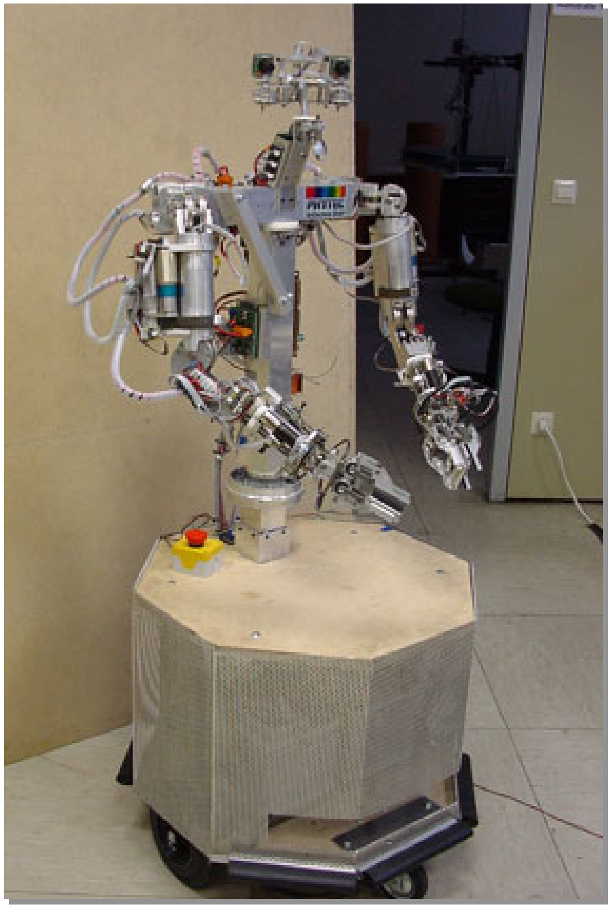


Robots–The Reality

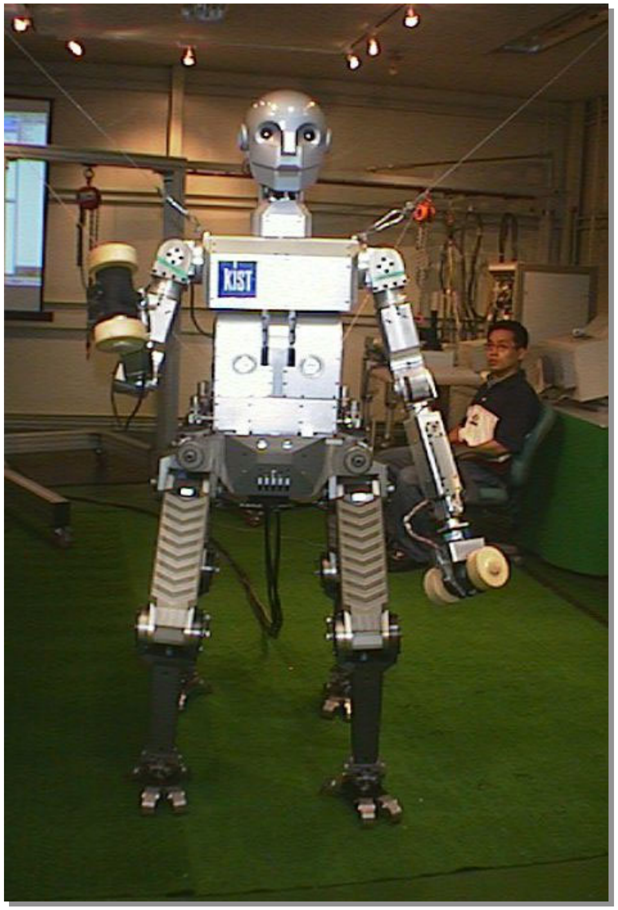
Robots–What We Might Want



Robots—A New Wave: The Humanoids Are Coming …

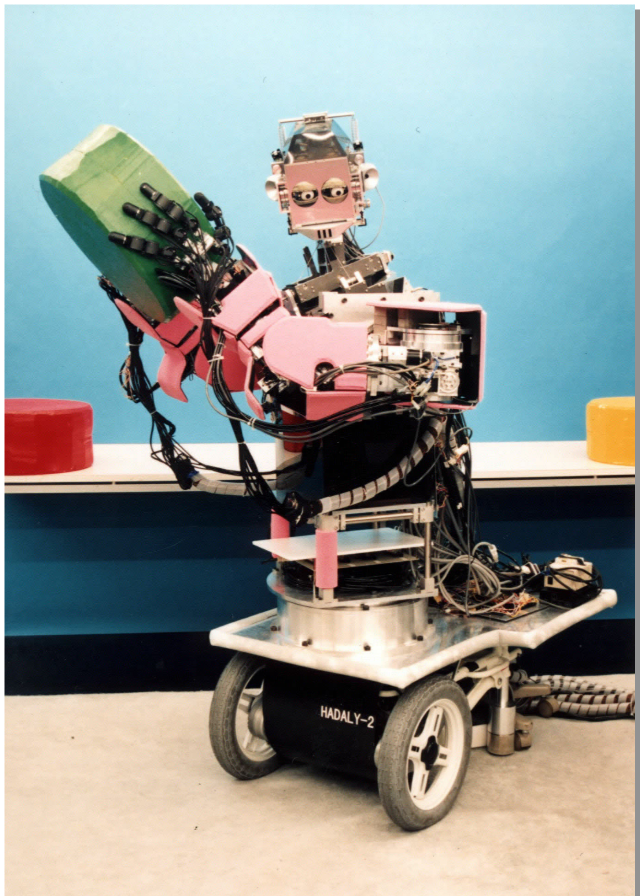


Amar–FZI, Karlsruhe



Centaur–KIST, Korea

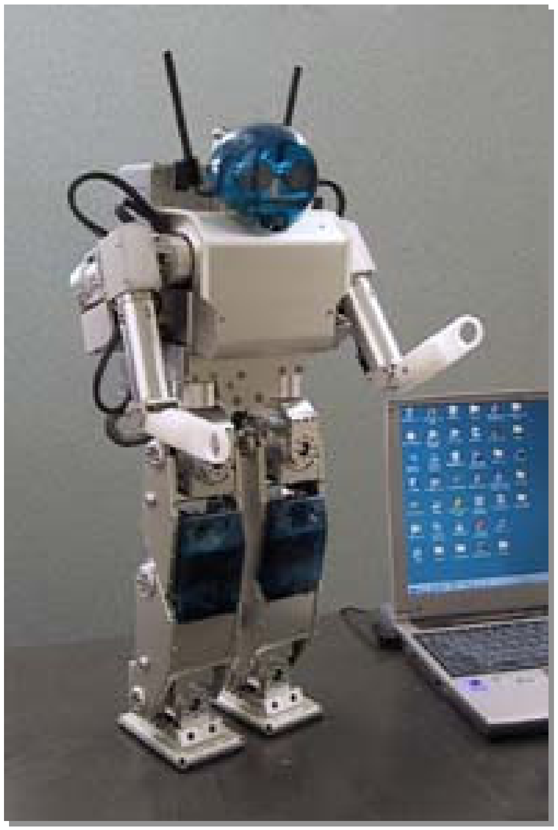
Robots—A New Wave: The Humanoids Are Coming …

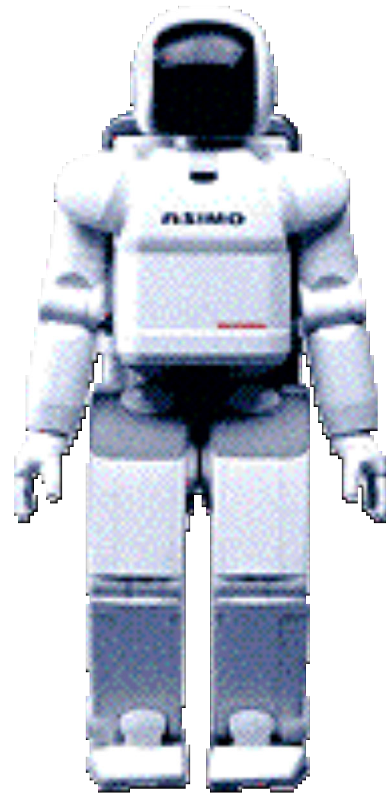


Hadaly–SuganoLab, Waseda

Hermes–BWH, Munich

Robots—A New Wave: The Humanoids Are Coming …





Hoap–Fujitsu, Japan

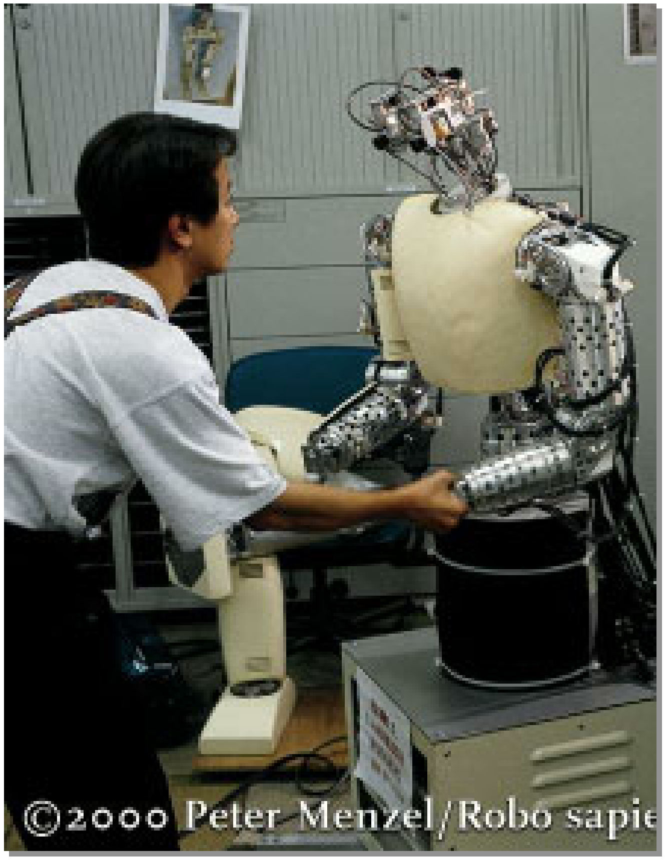
Asimo–Honda, Japan

Robots—A New Wave: The Humanoids Are Coming …

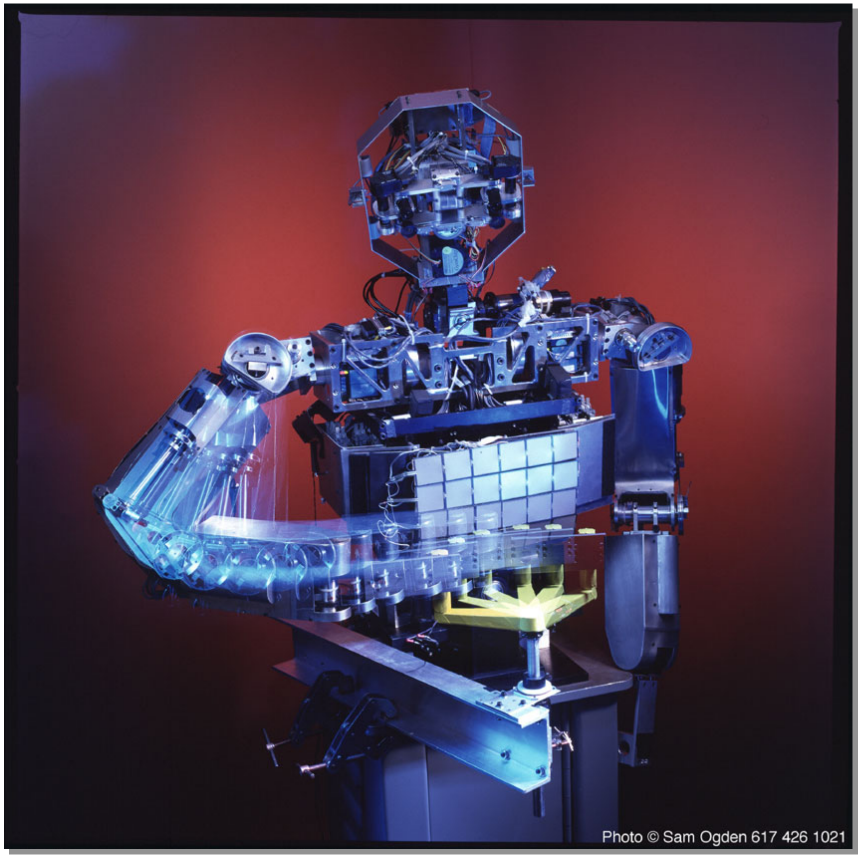
HRP-2P–Kawada, Japan

Isamu–Kawada, Japan

Robots—A New Wave: The Humanoids Are Coming …

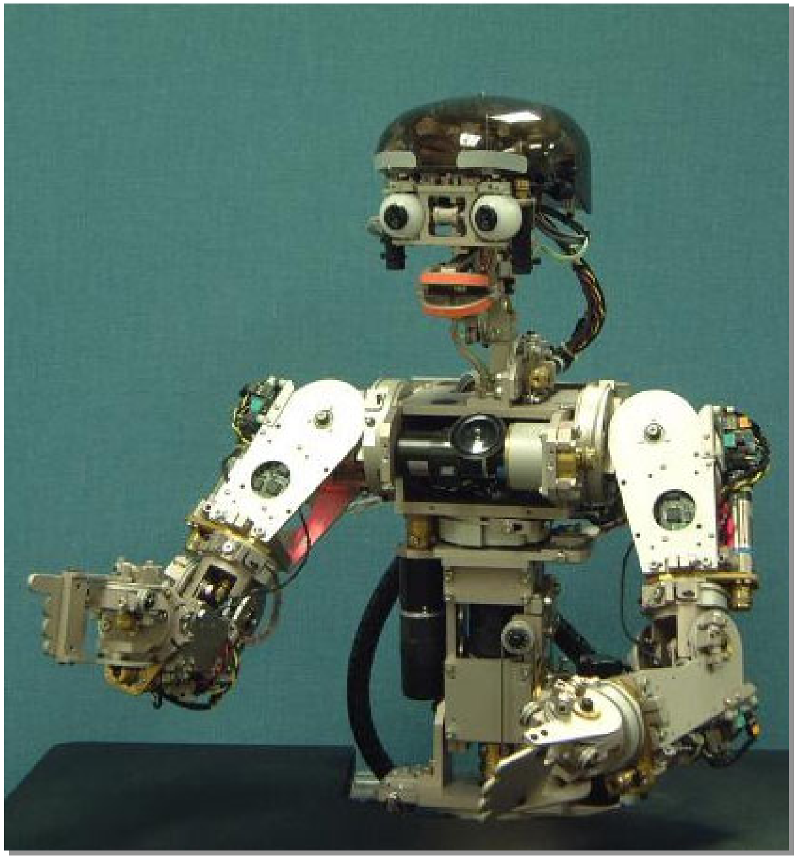


Jack–ETL, Japan



Cog–MIT

Robots—A New Wave: The Humanoids Are Coming …



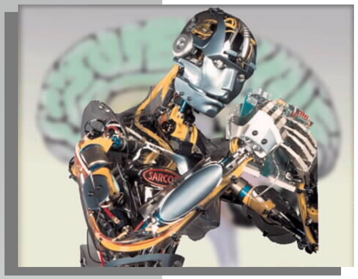
Infanoid–CRL/Kozima, Japan

Robotnaut–NASA

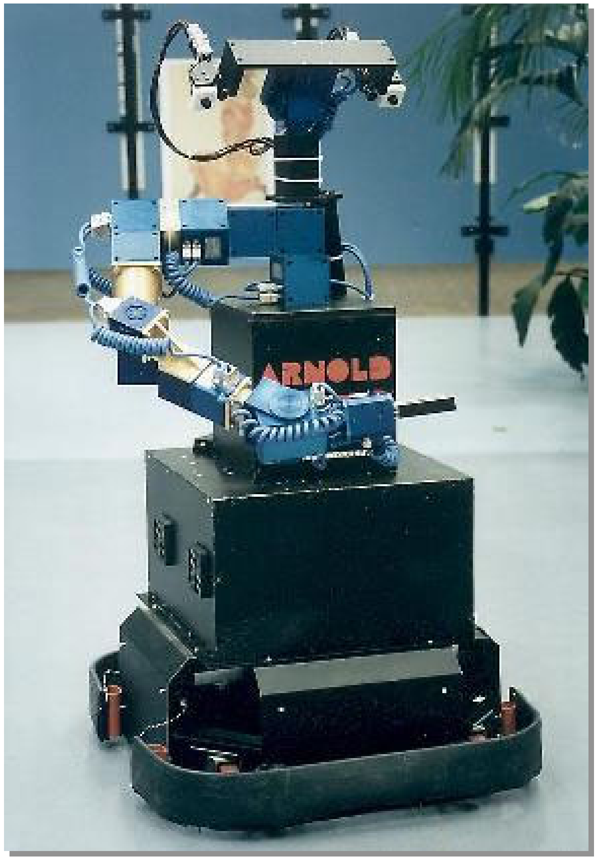
Robots—A New Wave: The Humanoids Are Coming …

Morph3–Kitano, Japan

JSK-H7–Tokyo University



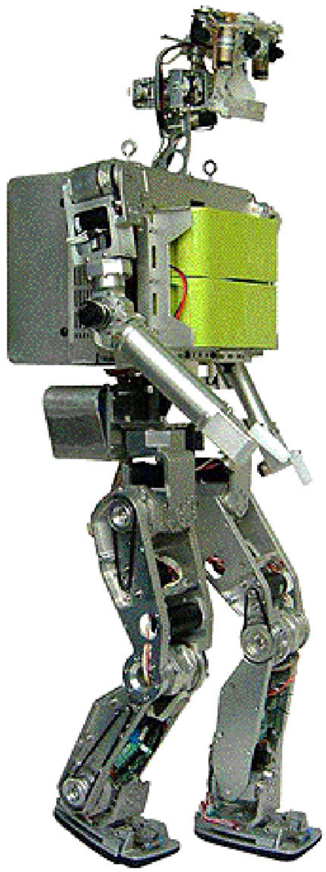
Robots—A New Wave: The Humanoids Are Coming …



Arnold—INI/Bochum, Germany,

Pino—Kitano, Japan

Robots—A New Wave:
The Humanoids Are Coming …

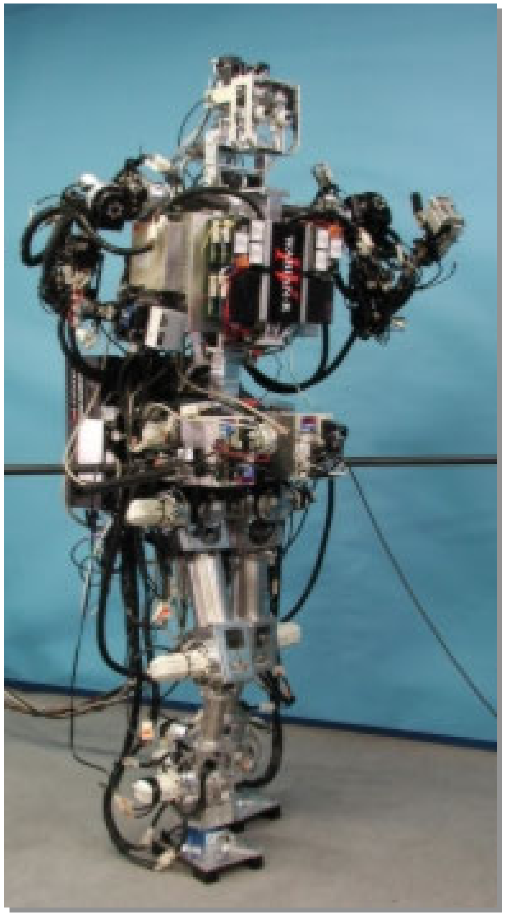


Robos—Kozoh, Japan

Sony Robot

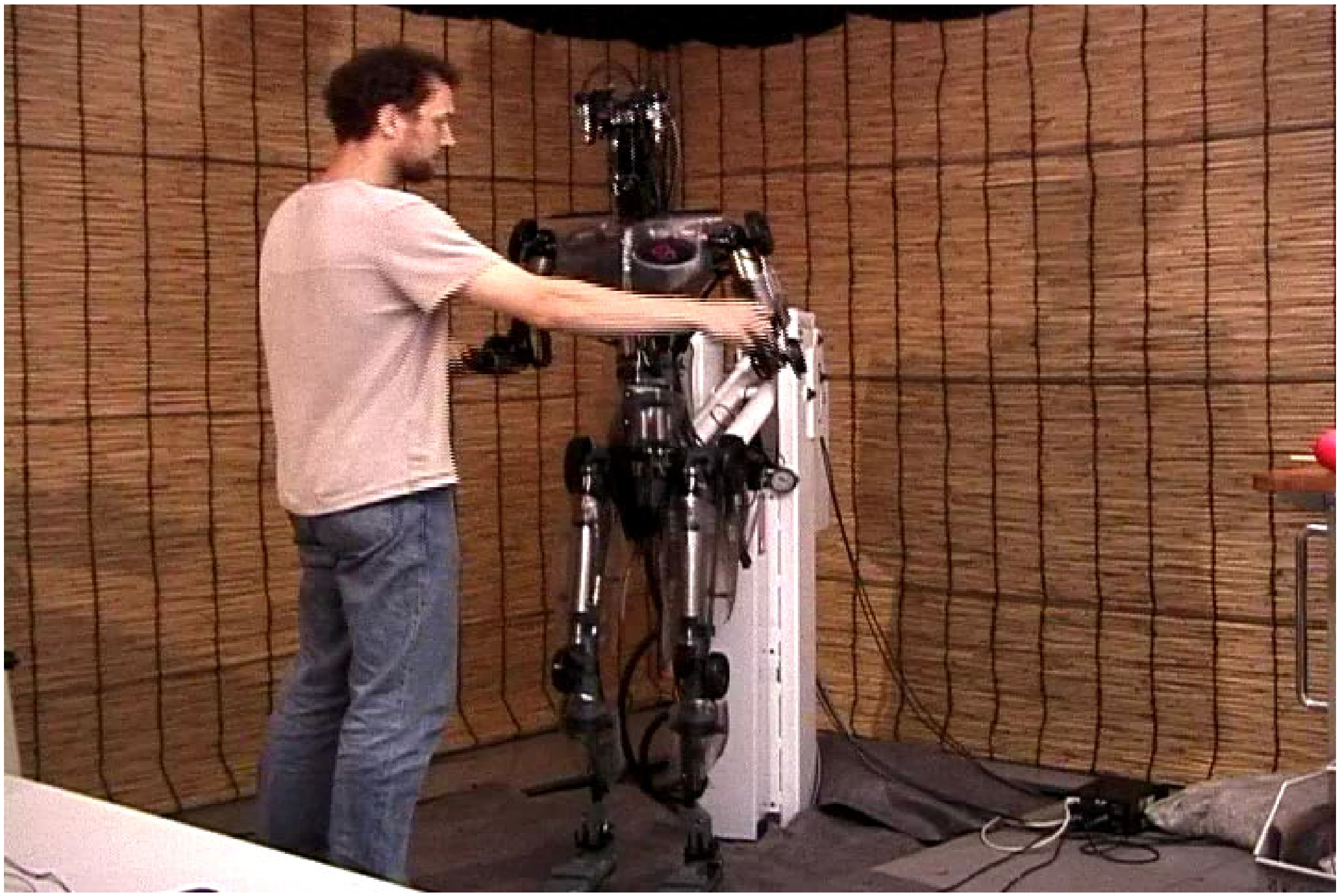
Robots—A New Wave: The Humanoids Are Coming …

Robotic Surrogate—RHD, USA

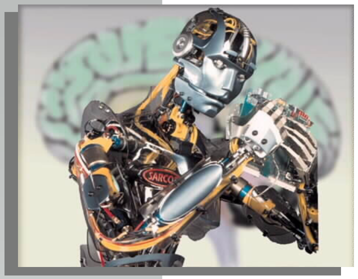


Wabian–Waseda/Takanishi, Japan

Robots—A New Wave: The Humanoids Are Coming …

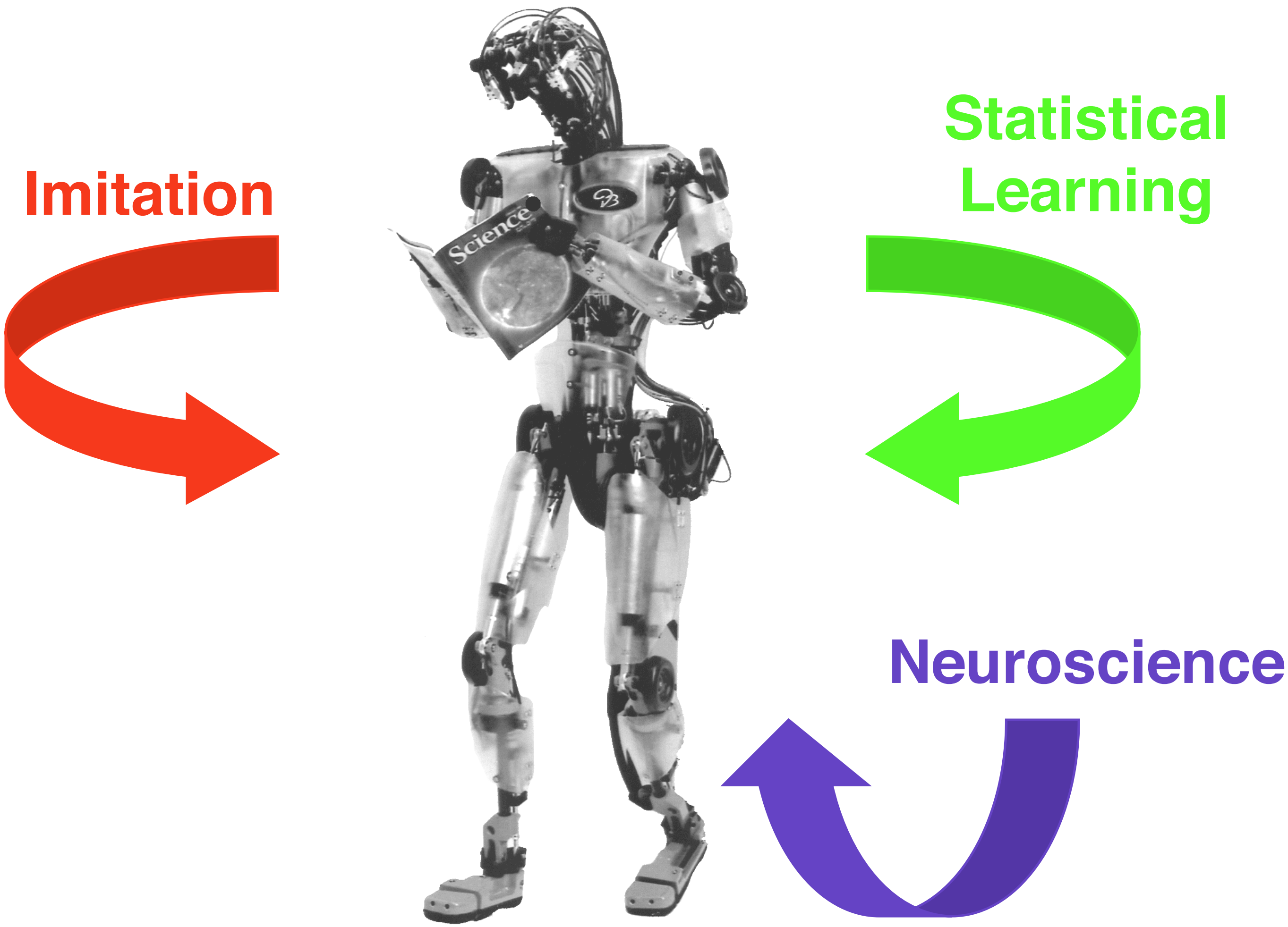


Sarcos Humanoid—ATR, Japan



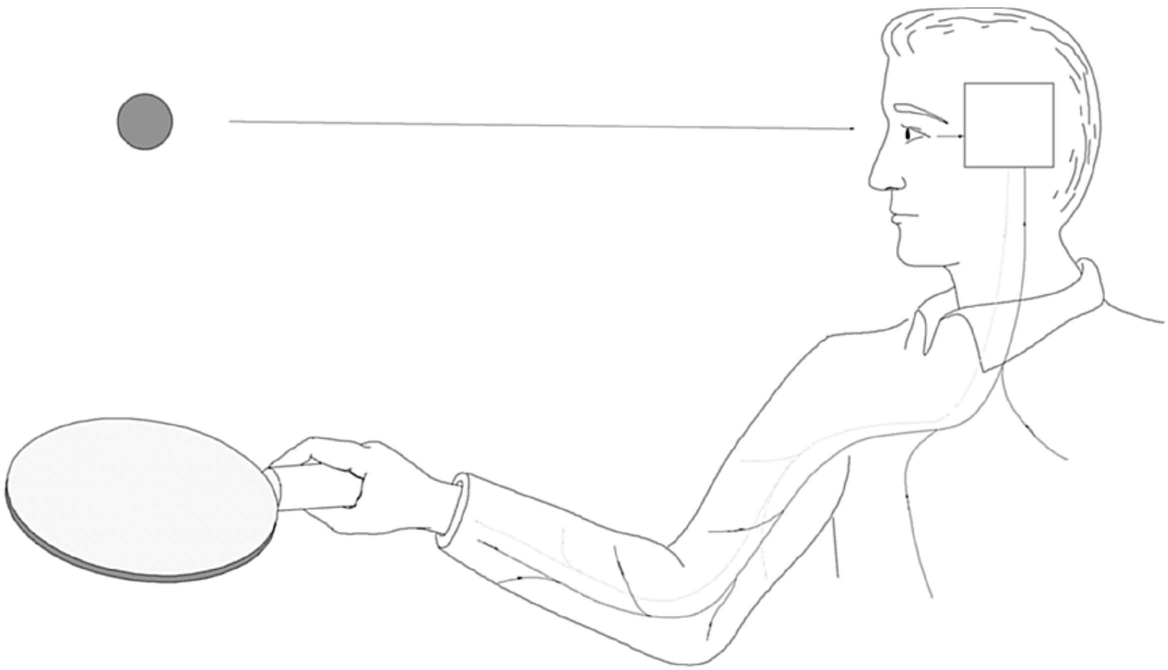
Robots–What We Might Want
But Can We Program Them?

Learning—A Key Element in Future Robots



Example 1: Statistical Learning for Motor Control

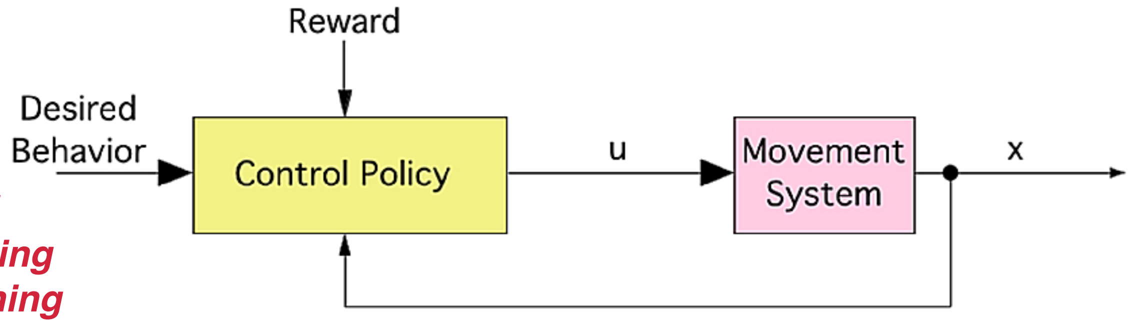
Policy: $$\mathbf{u}(t) = p(\mathbf{x}(t), t, \alpha)$$



Internal & External State: $\mathbf{x}(t) \longrightarrow$ Action: $\mathbf{u}(t)$

Many Elements of Learning Involve Function Approximation

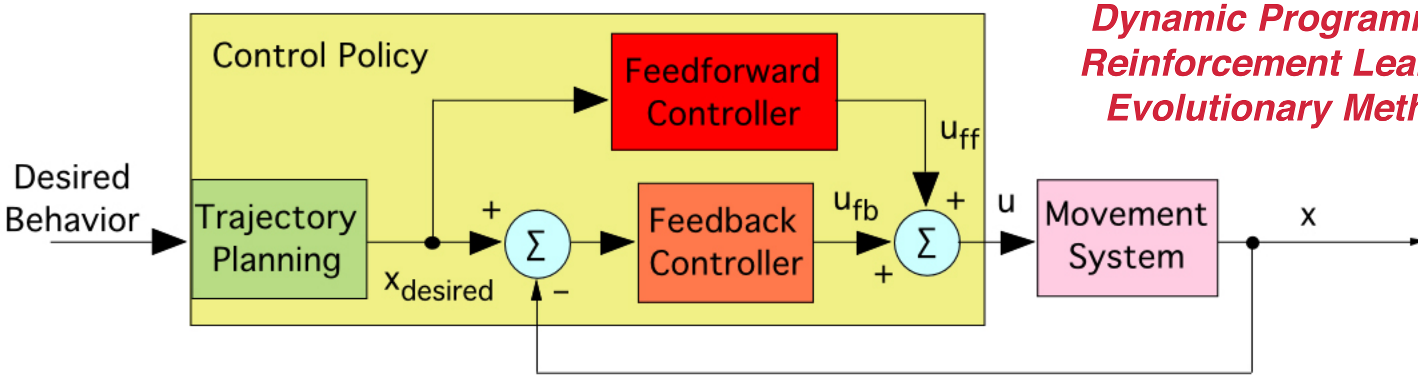
- Direct Control (Model-free)

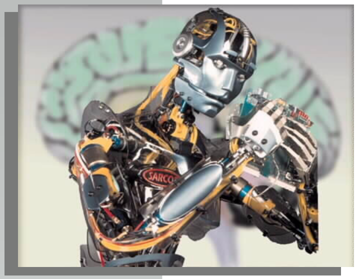


Adaptive Control
Dynamic Programming
Reinforcement Learning
Evolutionary Methods

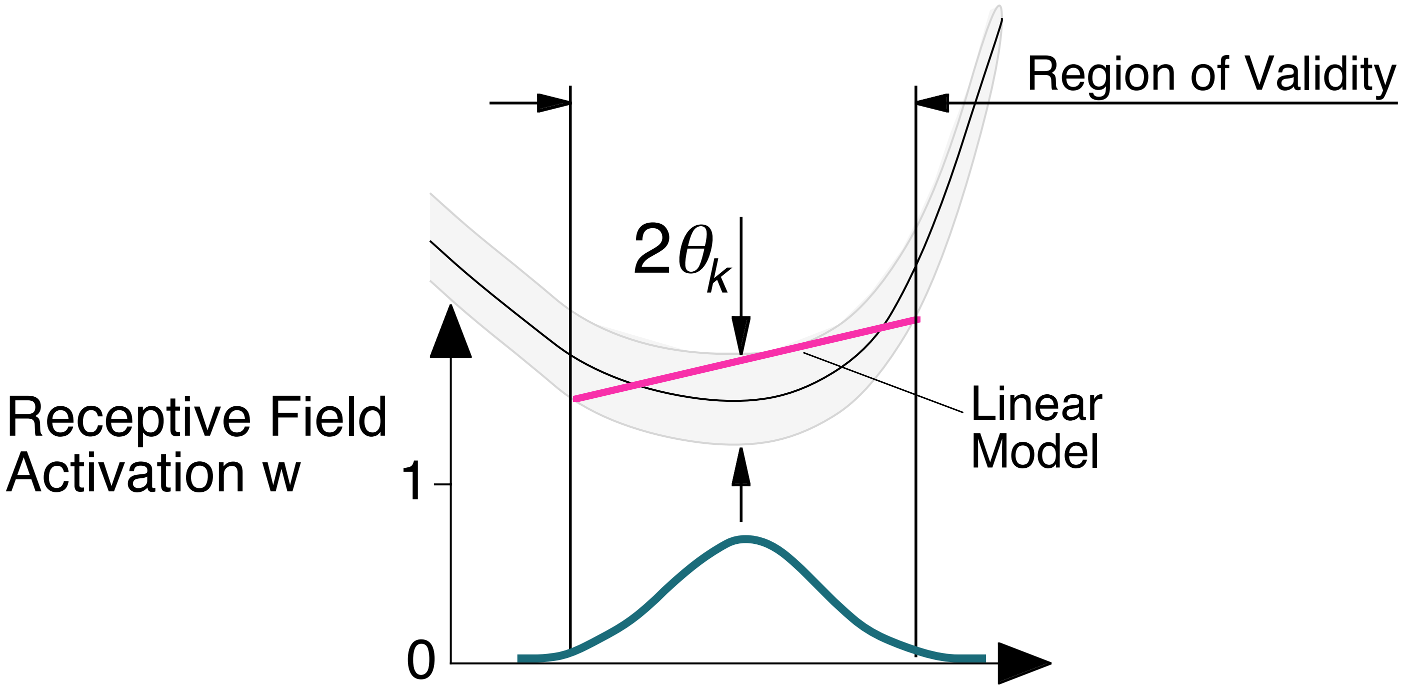
- Indirect Control (Model-based)

Supervised Learning
Adaptive Control
Dynamic Programming
Reinforcement Learning
Evolutionary Methods



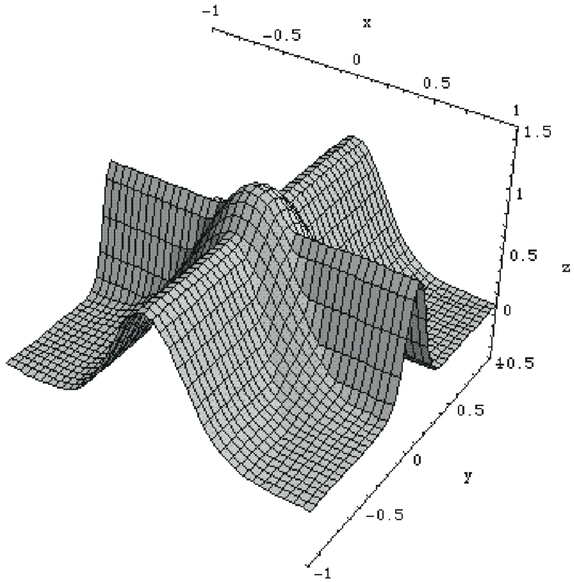


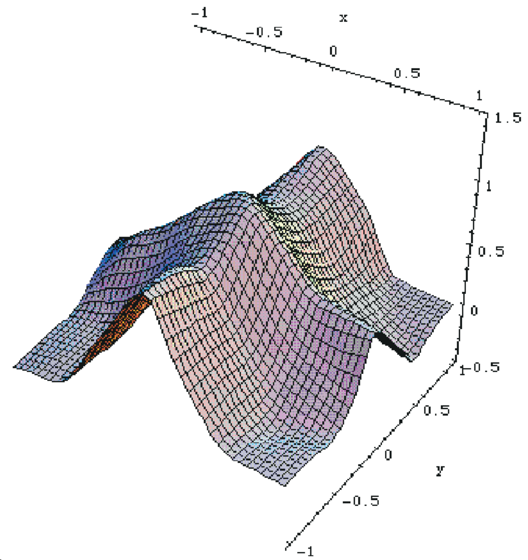
Function Approximation with Locally Linear Models

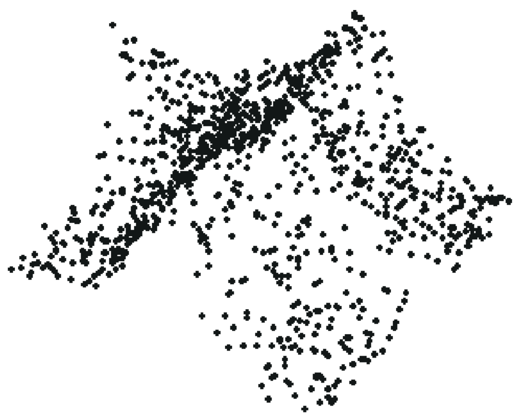


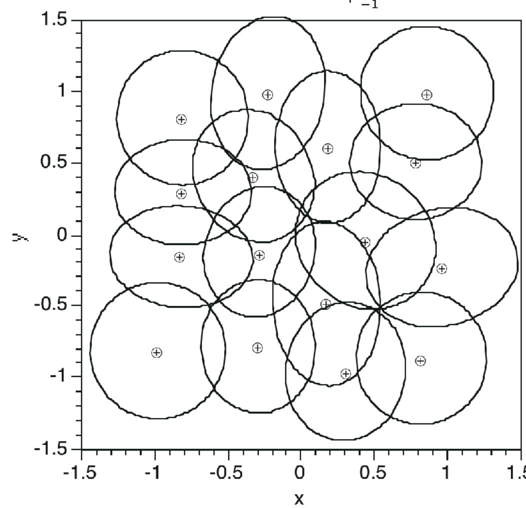
If we can find the tangent (plane) and the region of validity from only local data, the function approximation problem can be solved efficiently

Example: 2D Fitting



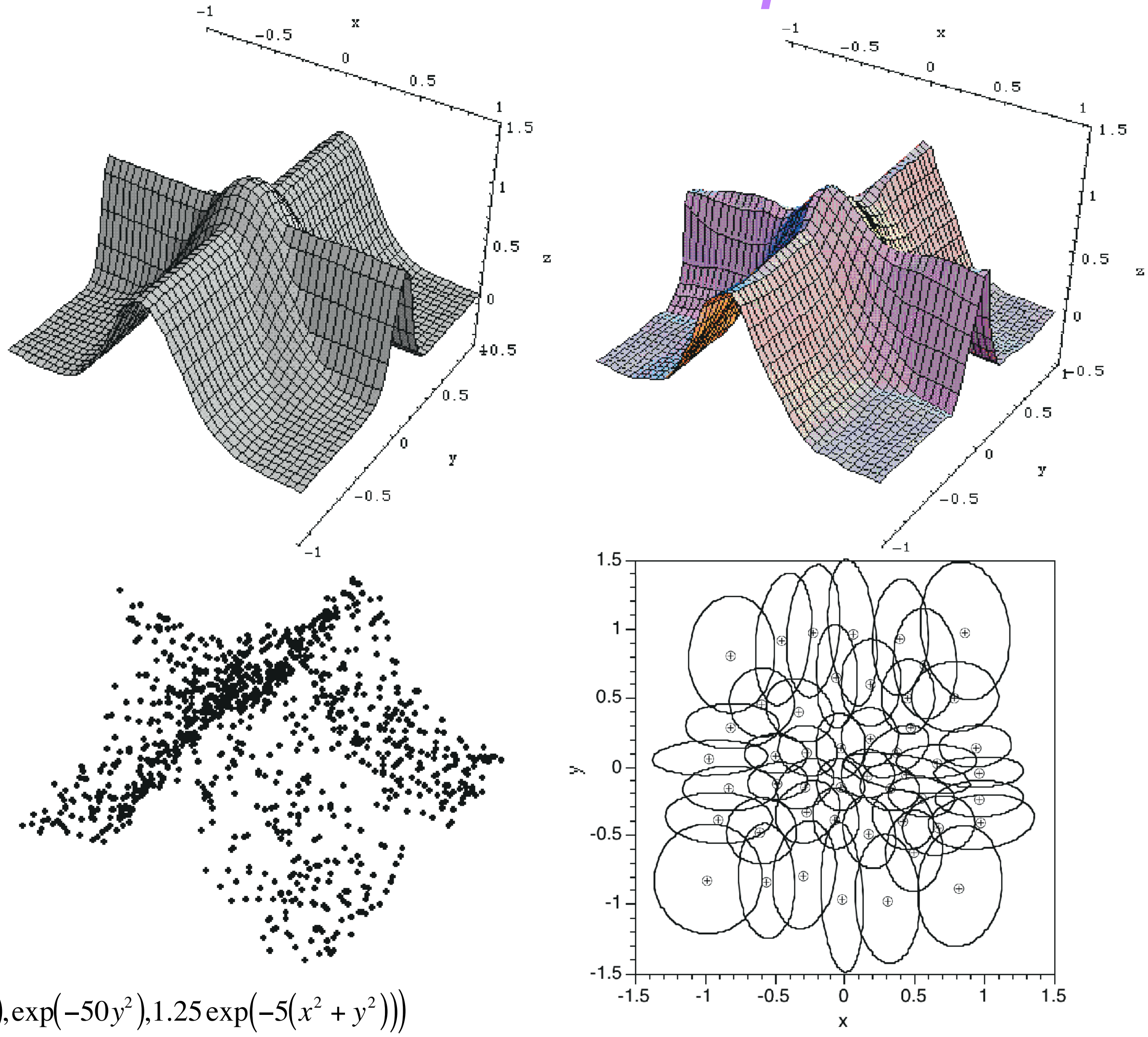






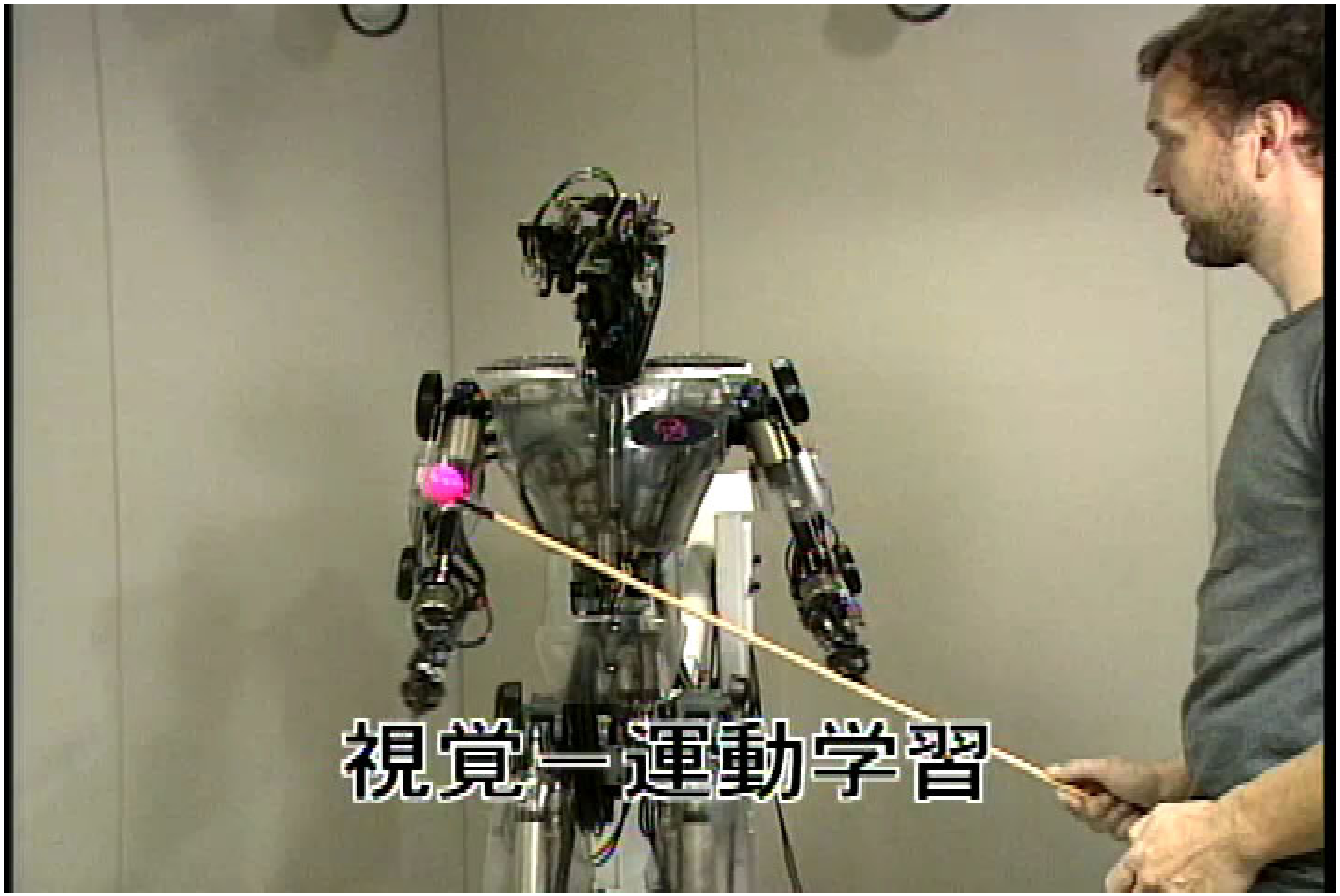
$$z = \max\left(\exp\left(-10x^2\right), \exp\left(-50y^2\right), 1.25\exp\left(-5\left(x^2+y^2\right)\right)\right)$$

Example: 2D Fitting



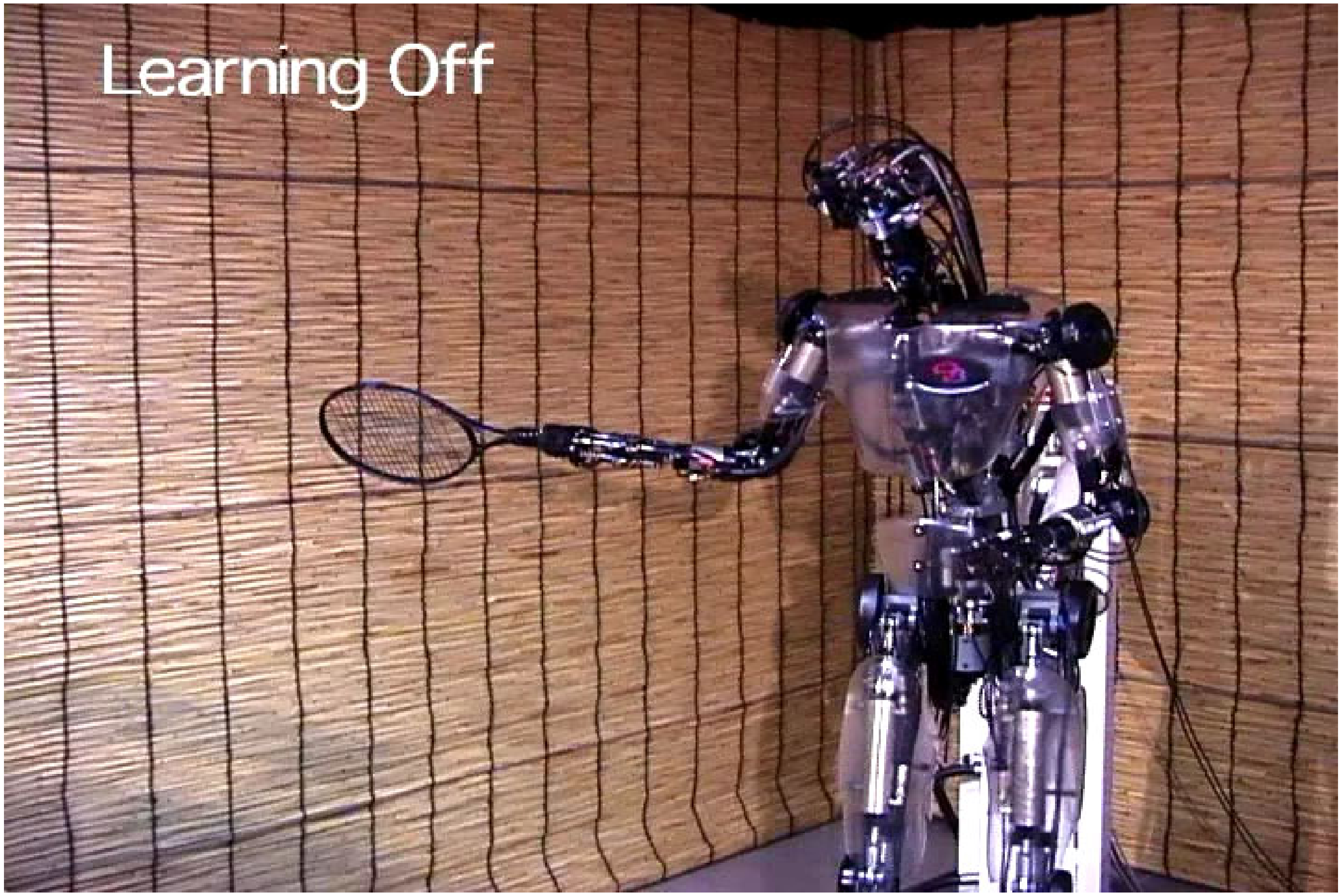
$$z = \max\left(\exp\left(-10x^2\right), \exp\left(-50y^2\right), 1.25\exp\left(-5\left(x^2+y^2\right)\right)\right)$$

RMRC Inverse Kinematics Learning



Trick: Learn direct kinematics and additionally local inverse in each local model

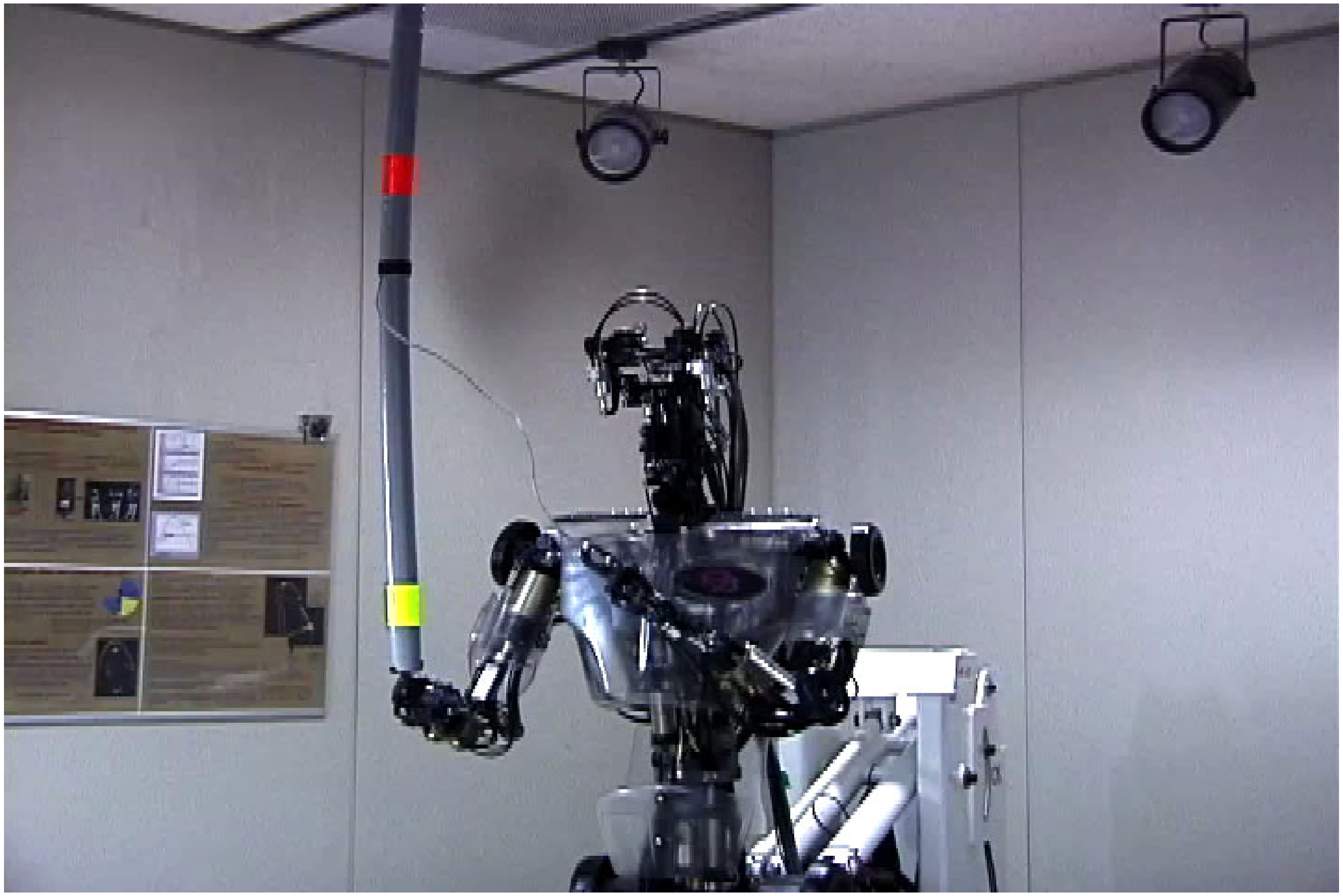
Inverse Dynamics Learning



On-line 90->30 Dim. Mapping in Computed Torque Controller

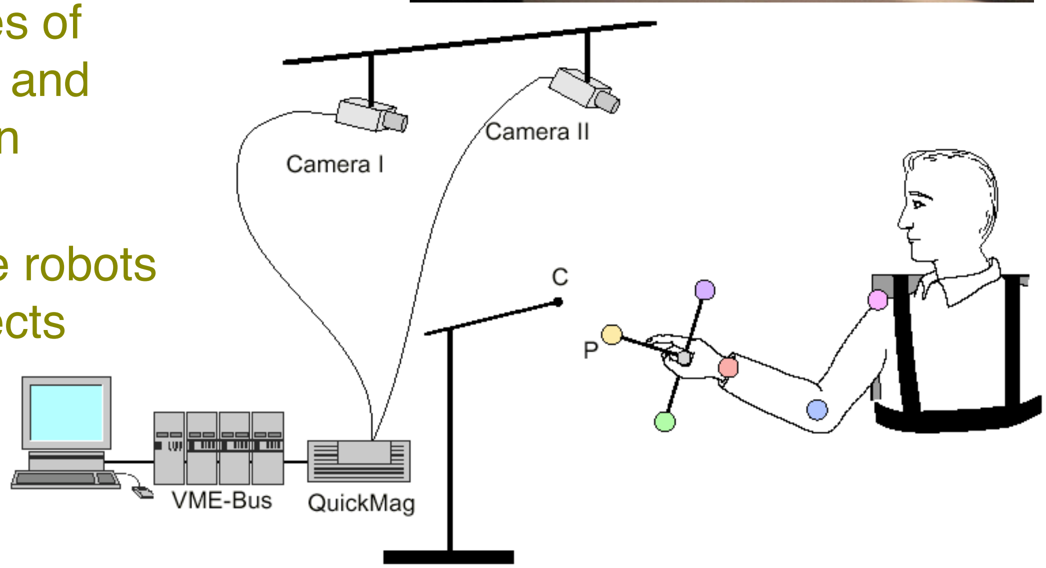
Skill Learning

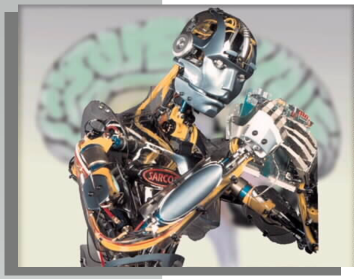
(requires accurate dynamics model)



Example: Behavioral and Computational Neuroscience

- Measure human movement
- Measure brain activity
- Analyze data to extract principles of learning and control in humans
- Also use robots as subjects

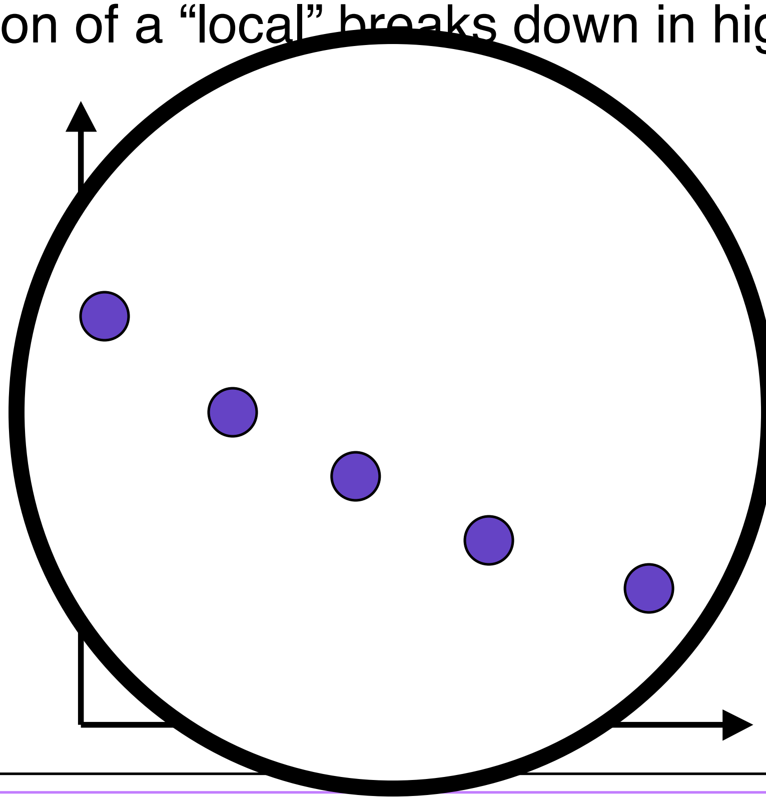




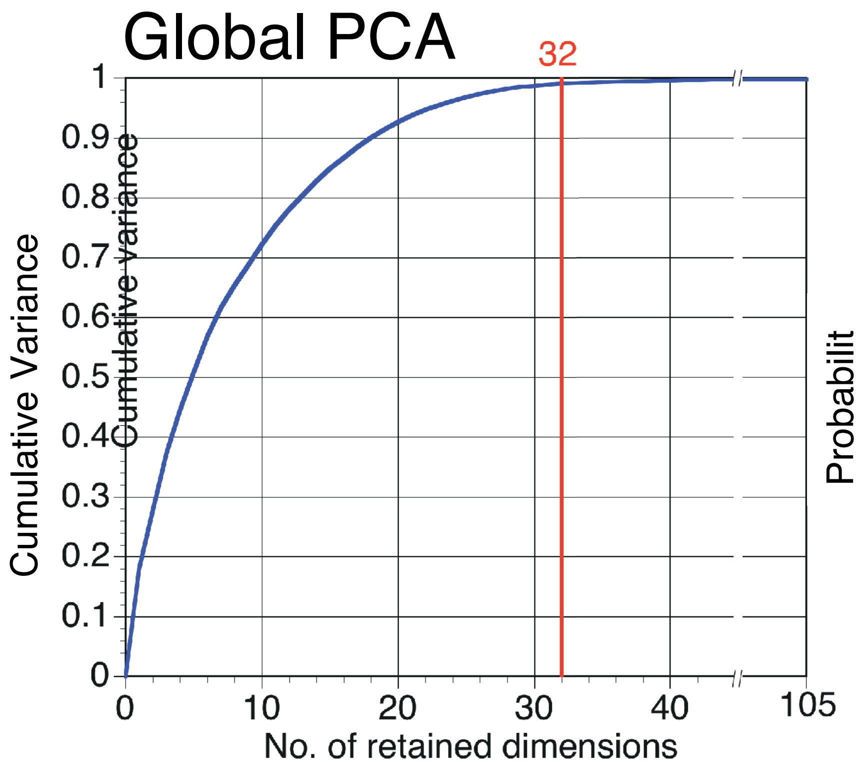
Motor Learning in High Dimensional Spaces

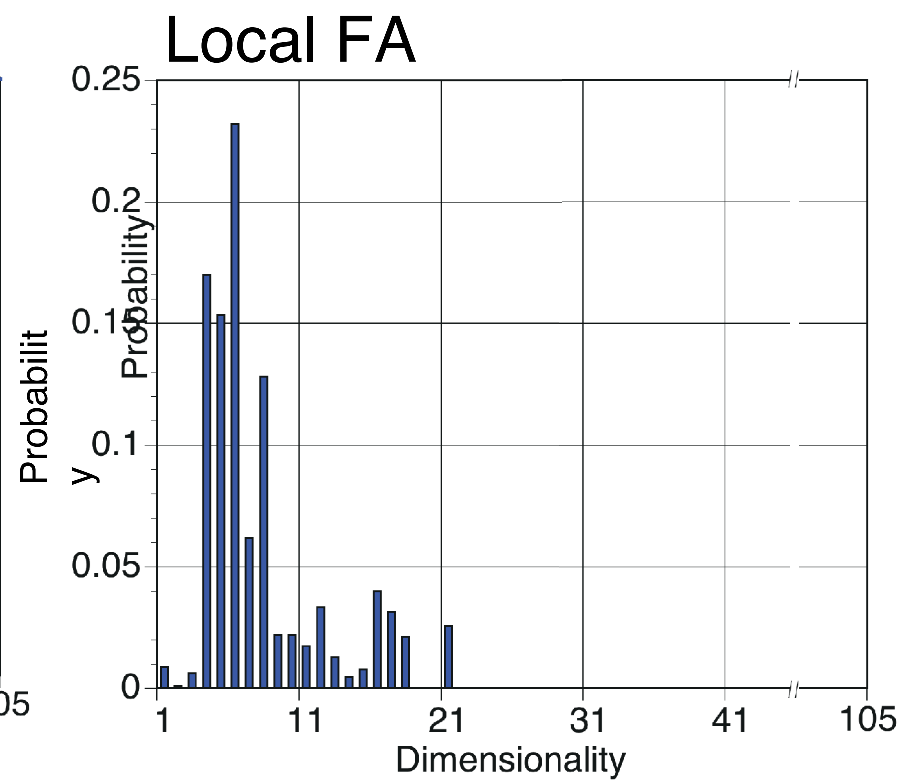
– The Curse of Dimensionality

- The power of local learning comes from exploiting the discriminative power of local neighborhood relations, but the notion of a “local” breaks down in high dim. spaces



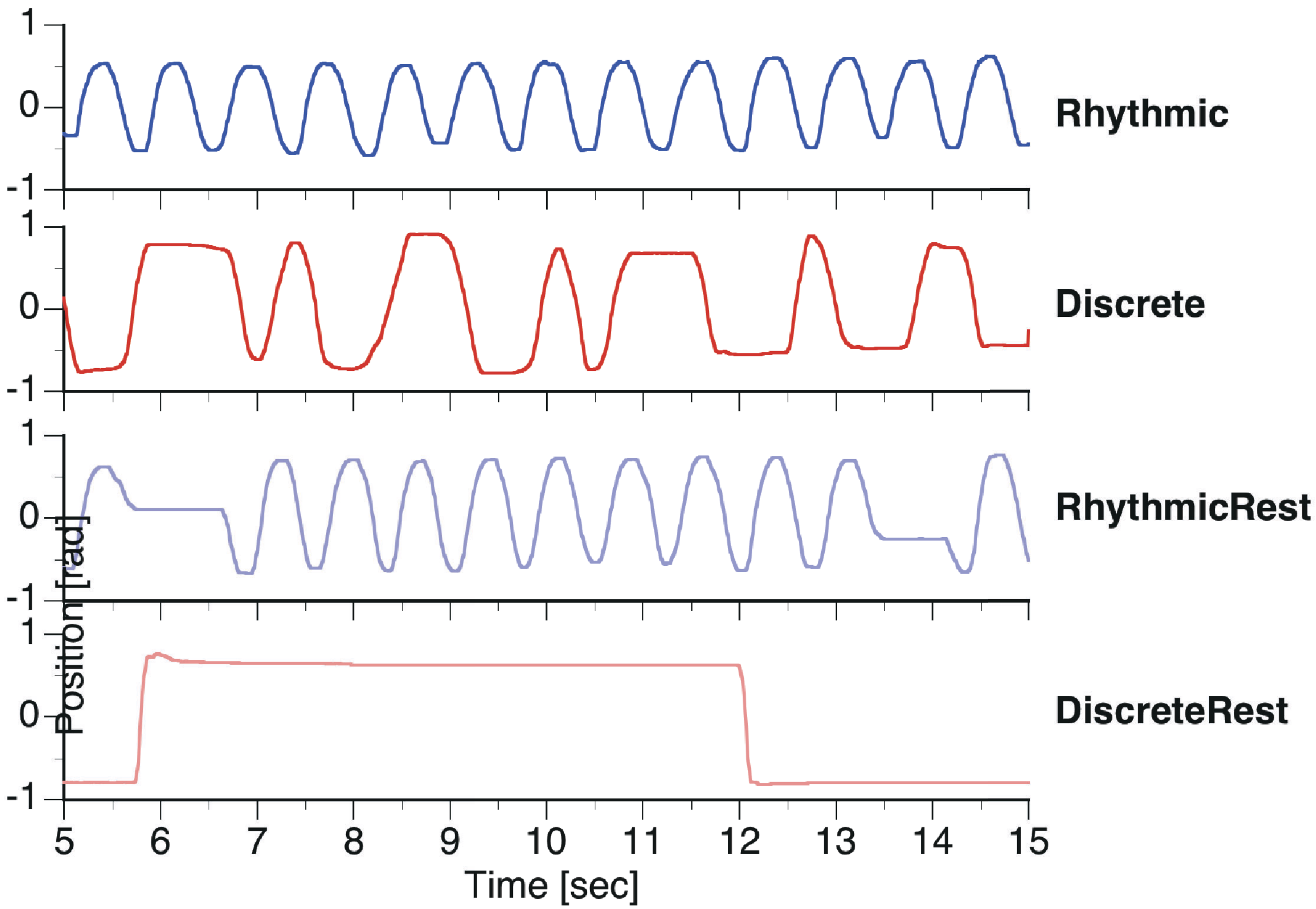
Dimensionality of Full Body Motion



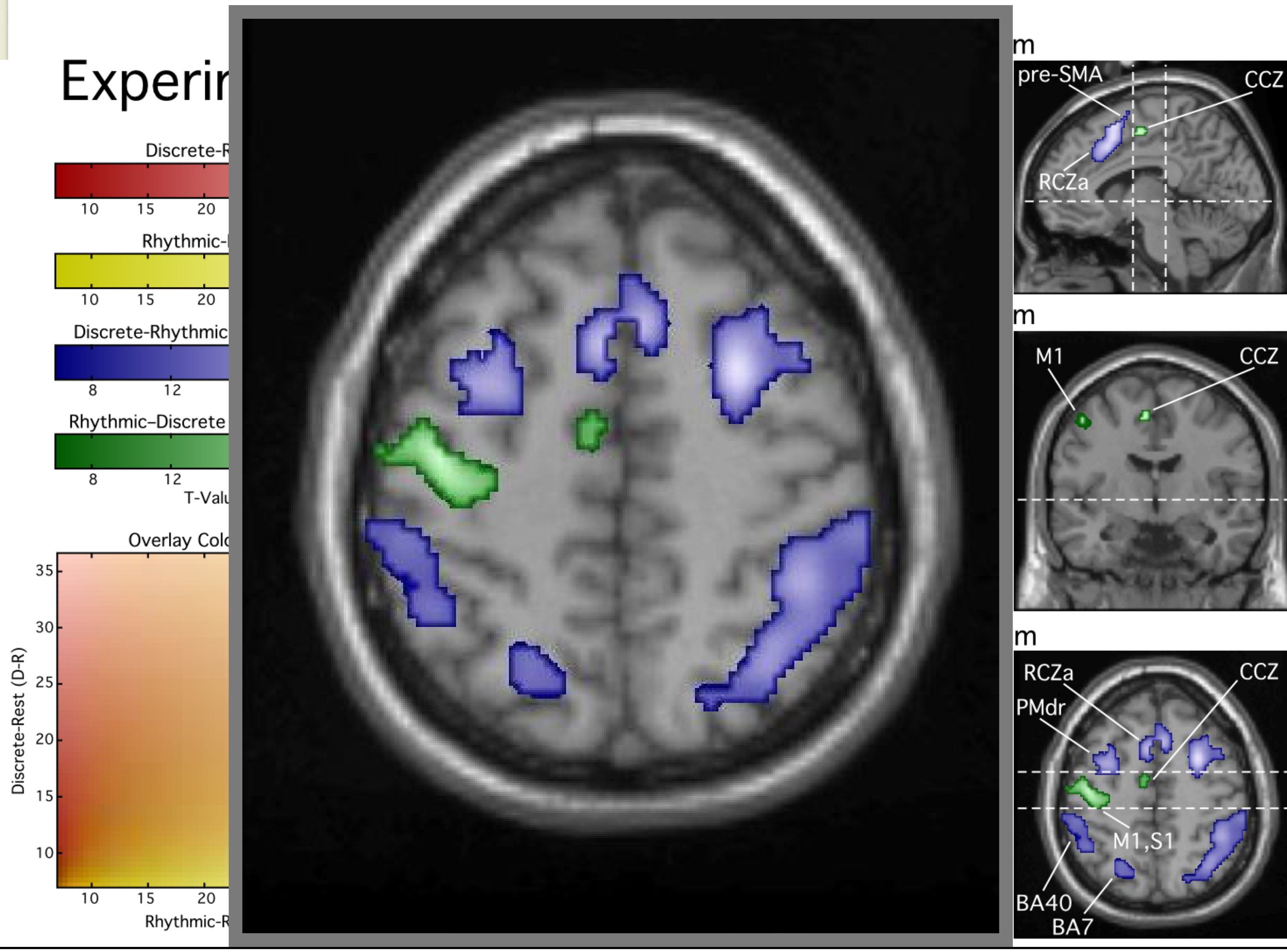


About 8 dimensions in the space formed by joint positions, velocities, and accelerations are needed to model an inverse dynamics model

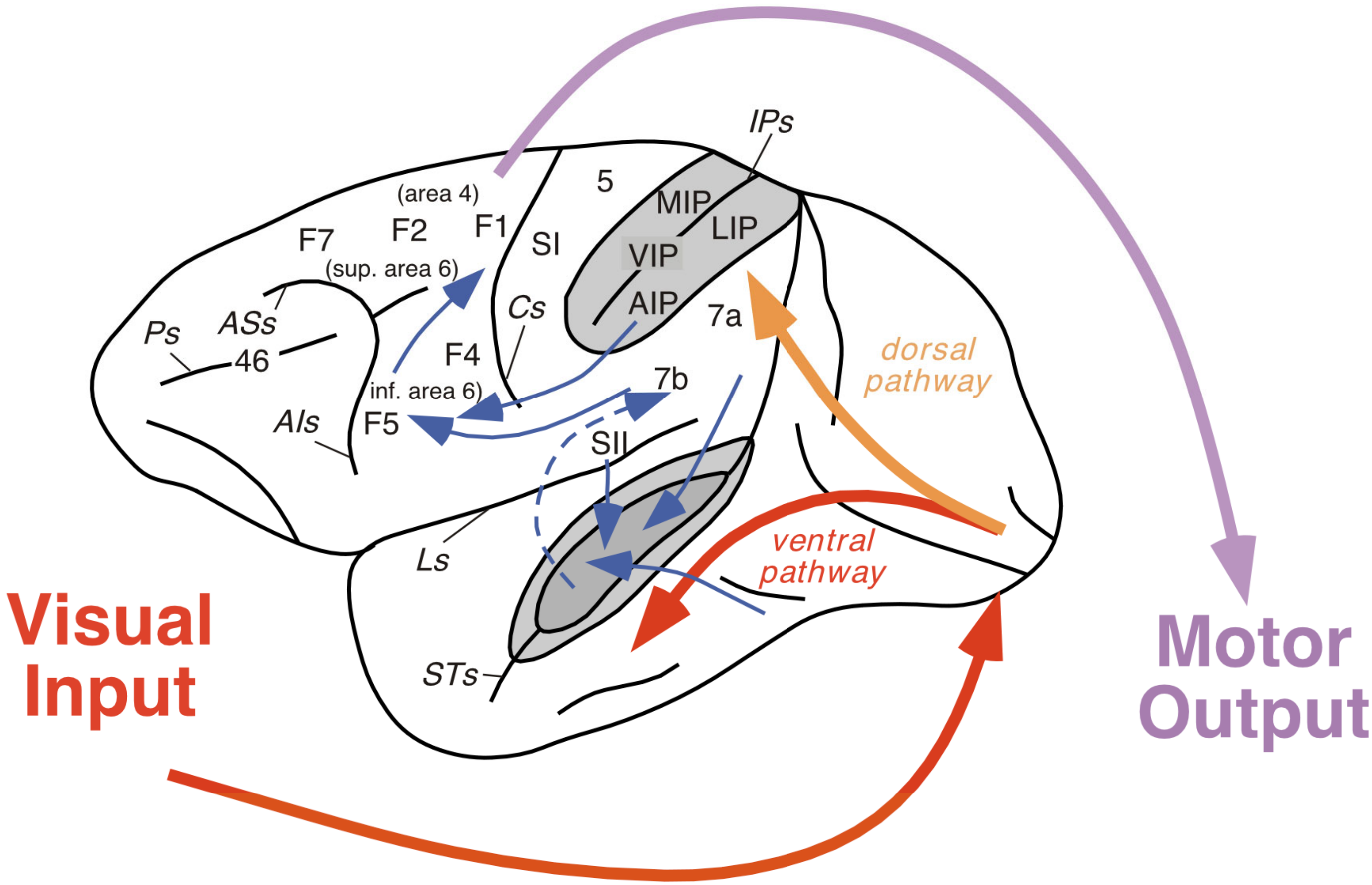
Example: Wrist Trajectories



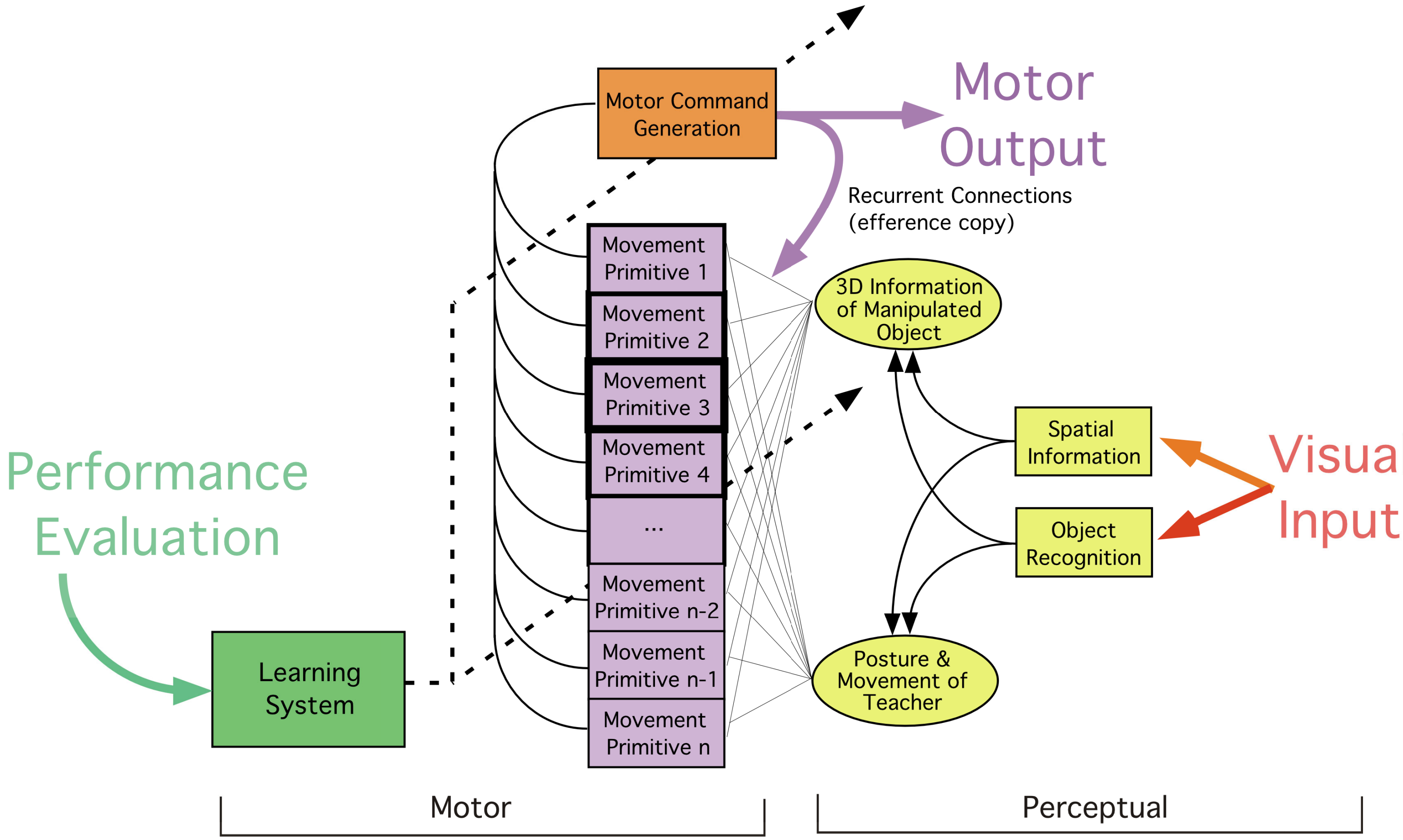
fMRI Summary Data



Example: Imitation Learning



A Concept of Imitation



Dynamic Systems Primitives: *Implicit Desired Trajectories*

- What is a dynamic system primitive?
  - A dynamical system (differential equation) with a particular behavior (a.k.a. pattern generator)
    - E.g.: Reaching movement can be interpreted as a point attractive behavior:

$$\dot{\theta}_d = \alpha\left(\theta_f - \theta_d\right)$$

Speed Target

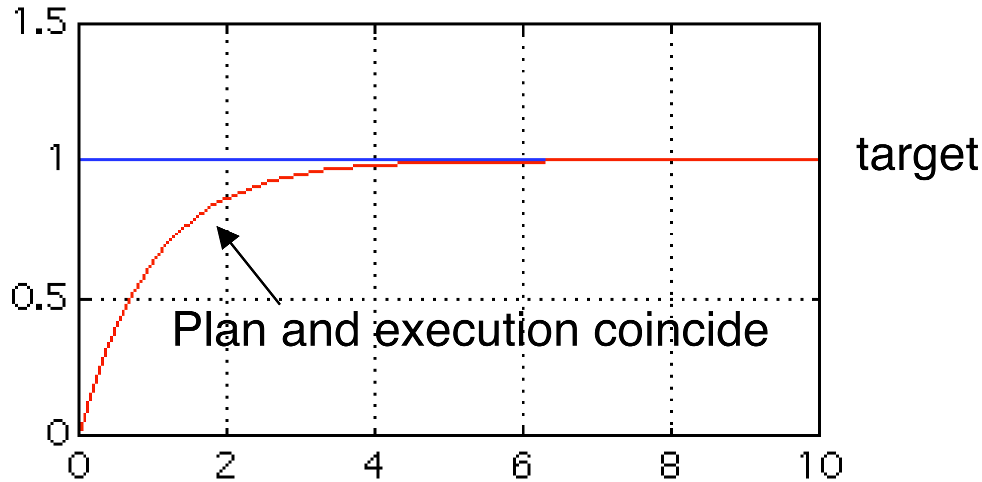
- What is the advantage of dynamic system primitives?
  - Independent of initial conditions
  - Online planning
  - Online modification through additional “coupling” terms. i.e., planning can react to sensory input

$$\dot{\theta}_d = \alpha\left(\theta_f - \theta_d\right) + \beta\left(\theta - \theta_d\right)$$

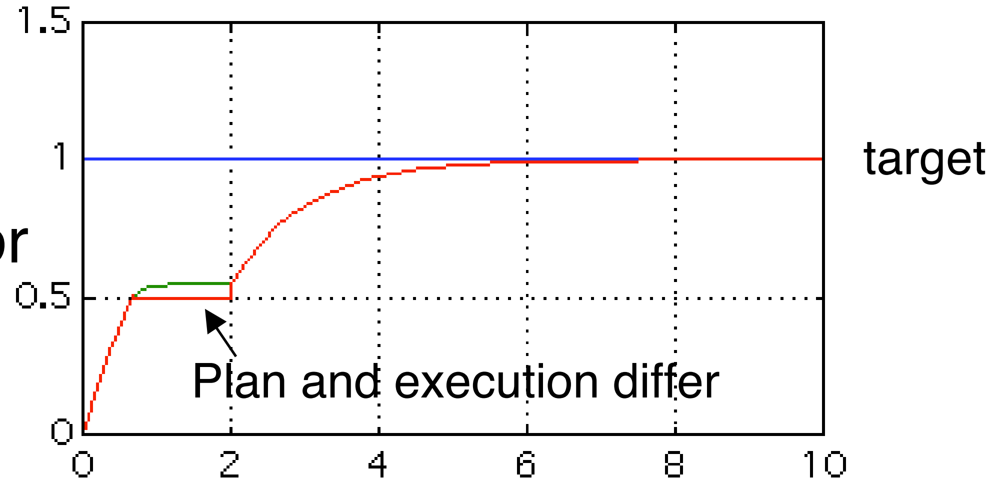
Dynamic Systems Primitives:

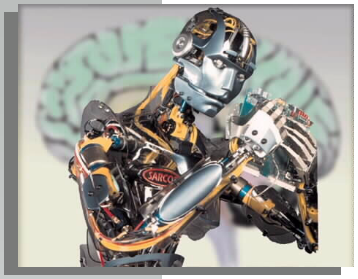
On-line Modification of Trajectory Planning

Unperturbed Behavior

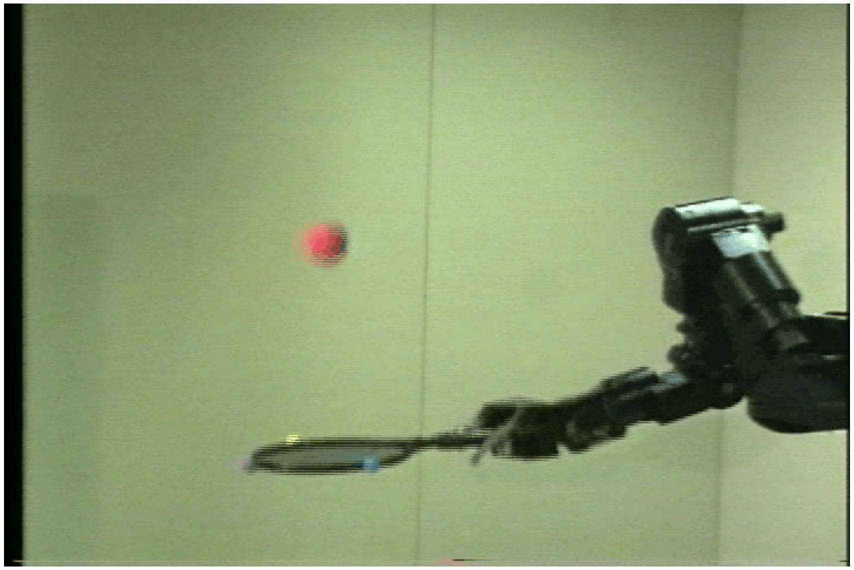


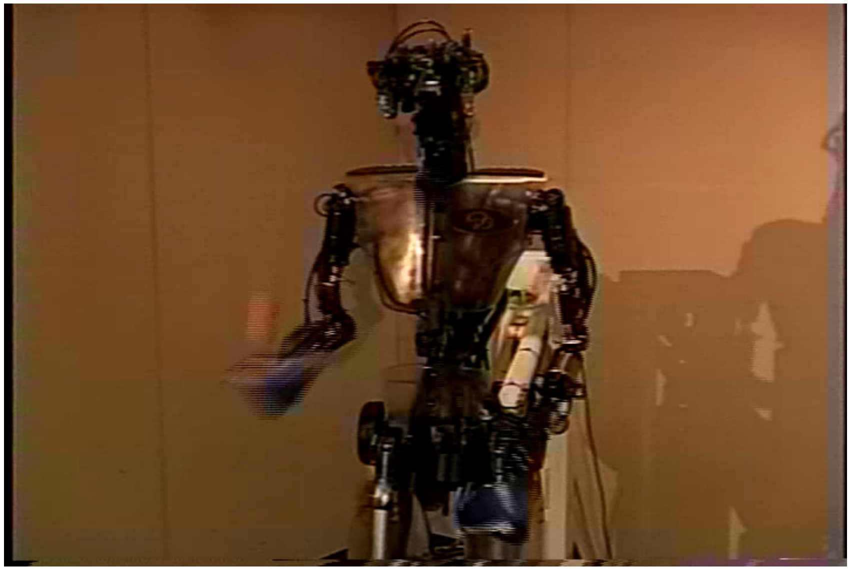
Temporarily Perturbed Behavior

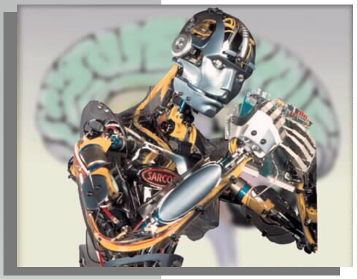




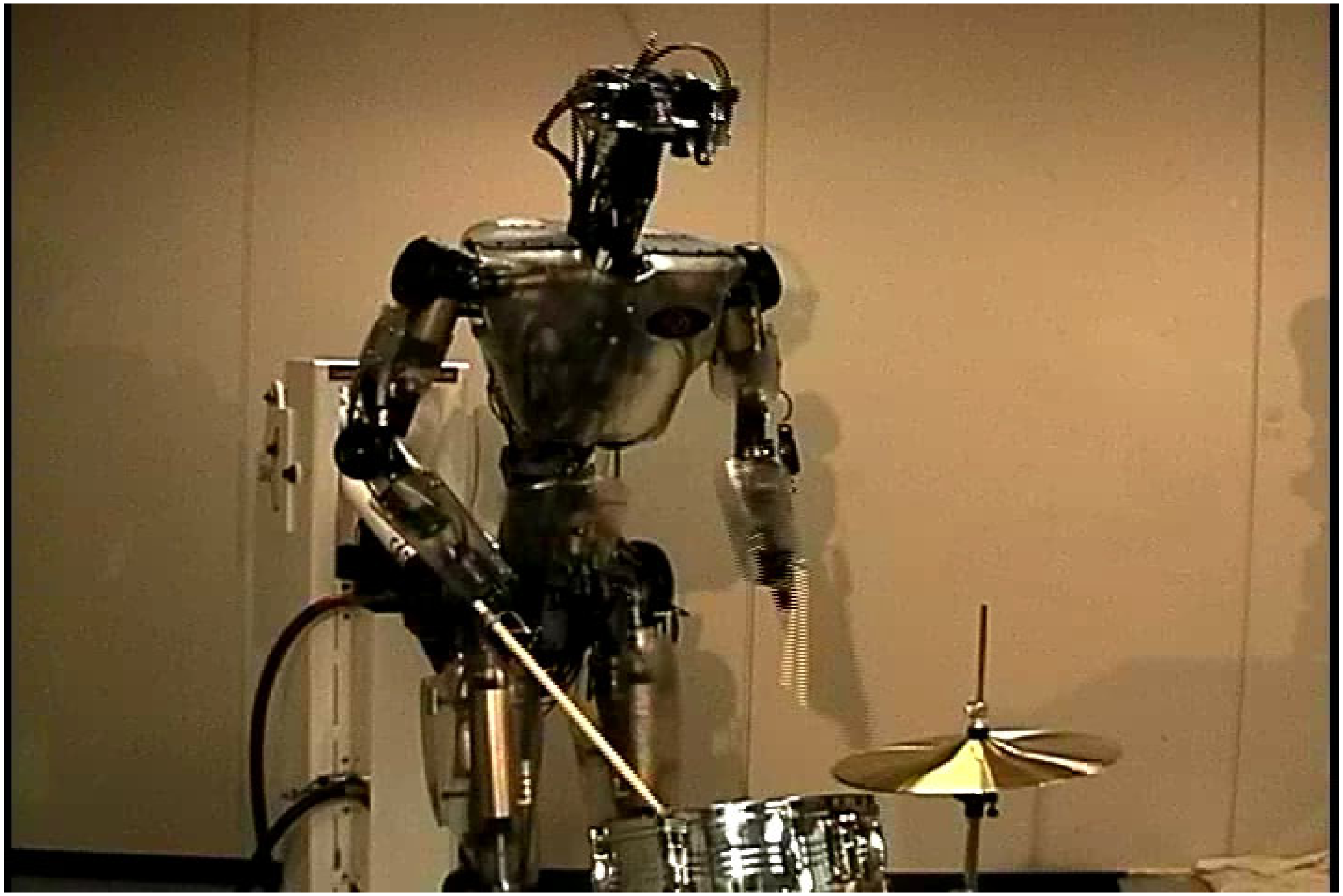
Dynamic Systems As Motor Primitives



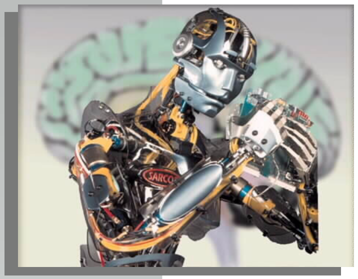




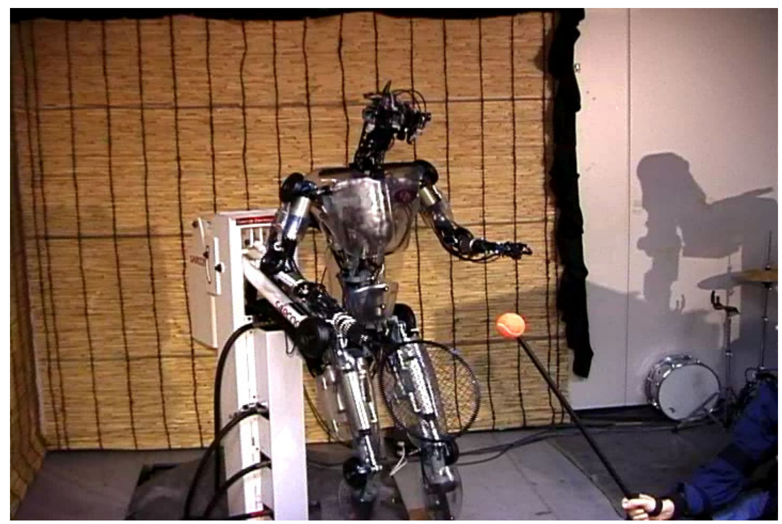
Dynamic Systems As Motor Primitives



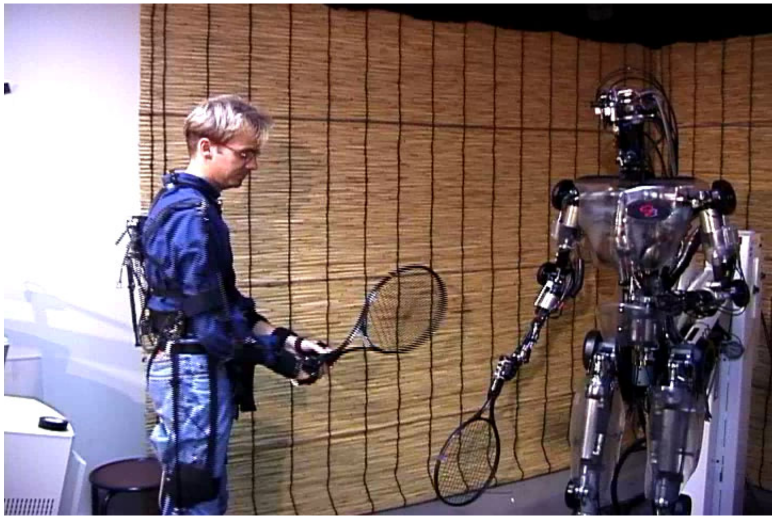
Dynamic Systems As Motor Primitives

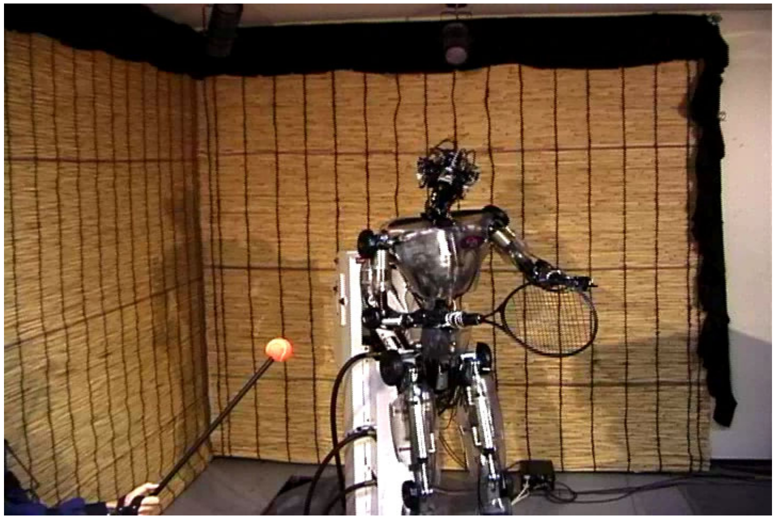


Imitation Learning With Dynamic Primitives

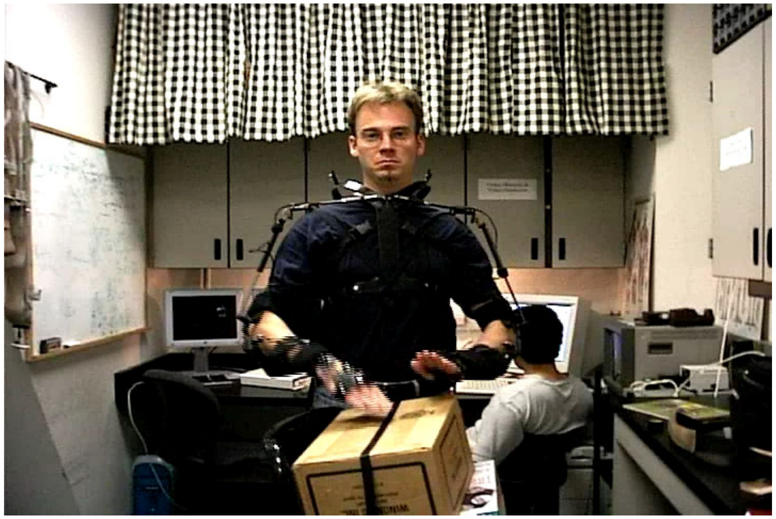


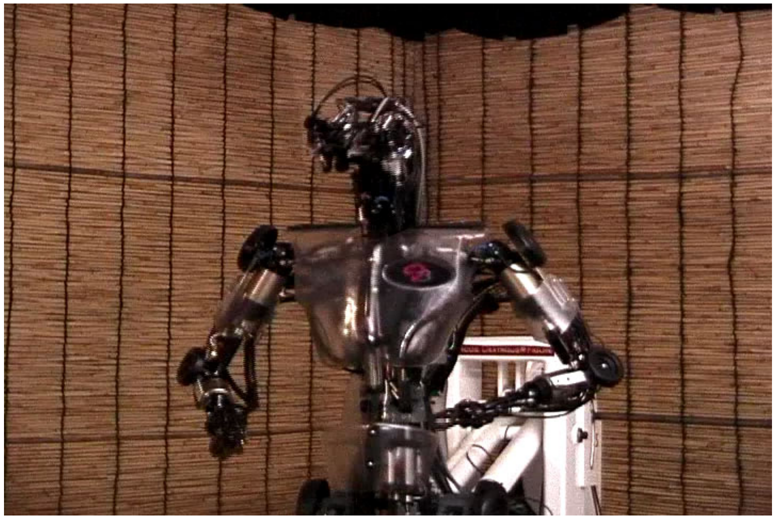
Imitation Learning With Dynamic Primitives



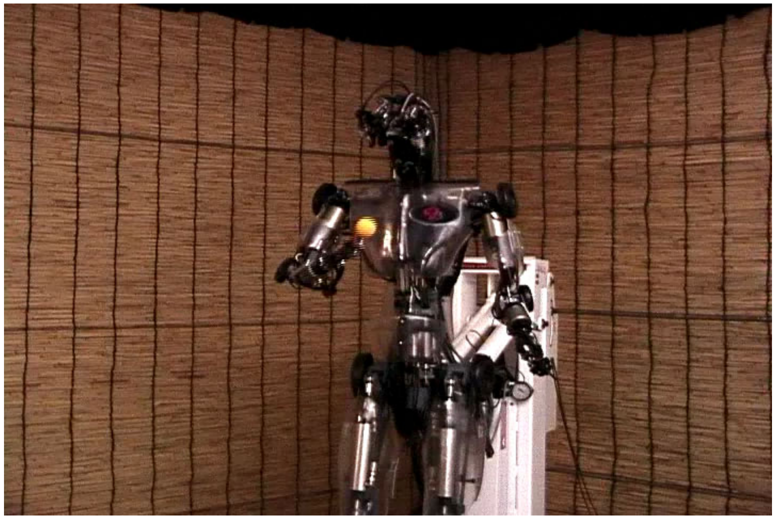


Imitation Learning With Dynamic Primitives

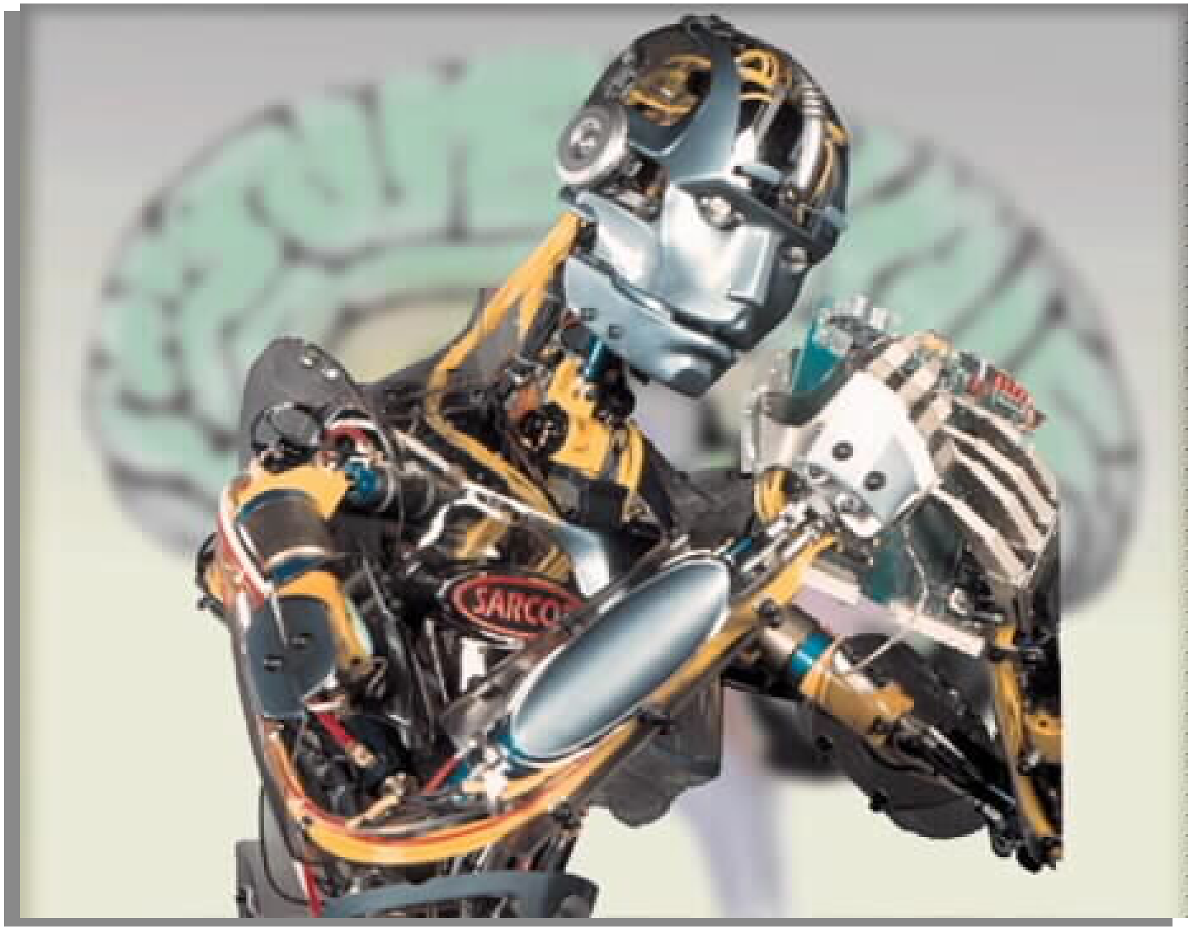




Imitation Learning With Dynamic Primitives



More Information…



http://www-clmc.usc.edu