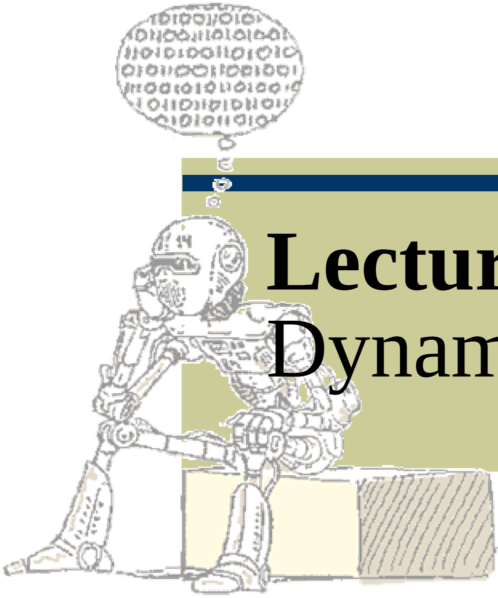


Lecture XIII
Dynamical Systems as Movement Policies

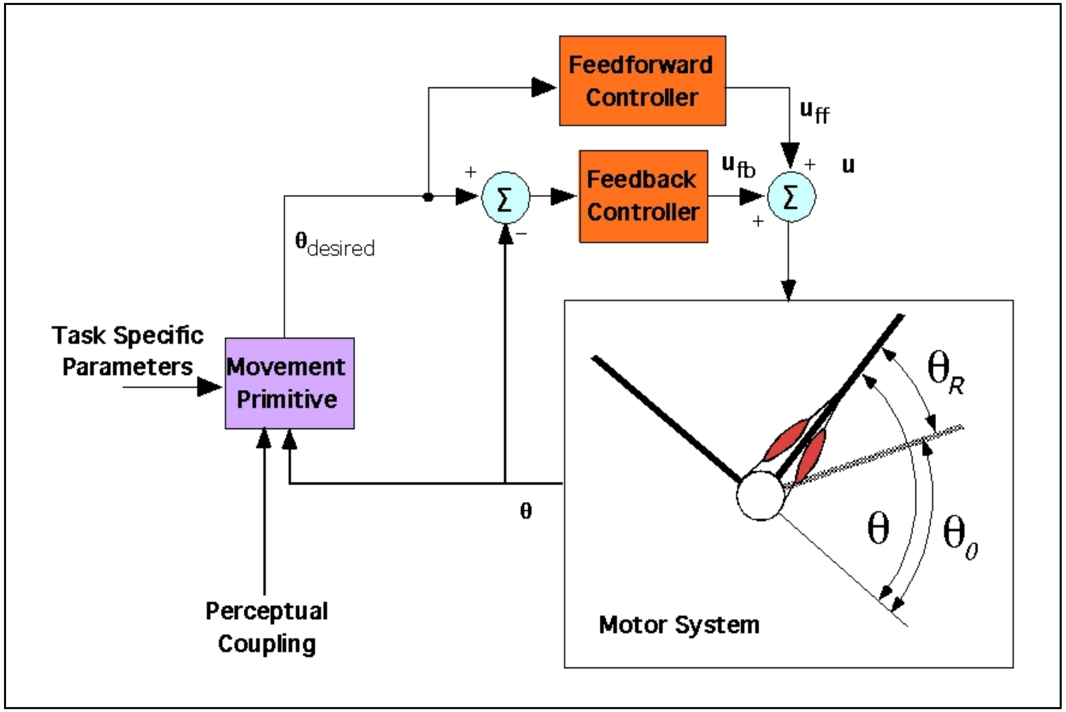
Contents:

- Differential Equation
- Force Fields, Velocity Fields
- Dynamical systems for Trajectory Plans
  - Generating plans dynamically
  - Fitting (or modifying) plans
  - Imitation based learning

Thanks to my collaborator Auke Ijspeert (EPFL) for many of the contents on the slides for this lecture.

Movement policies as Dynamical Systems

$$\tau \dot{\mathbf{x}}_{des} = f(\mathbf{x}, \mathbf{x}_{des}, goal)$$



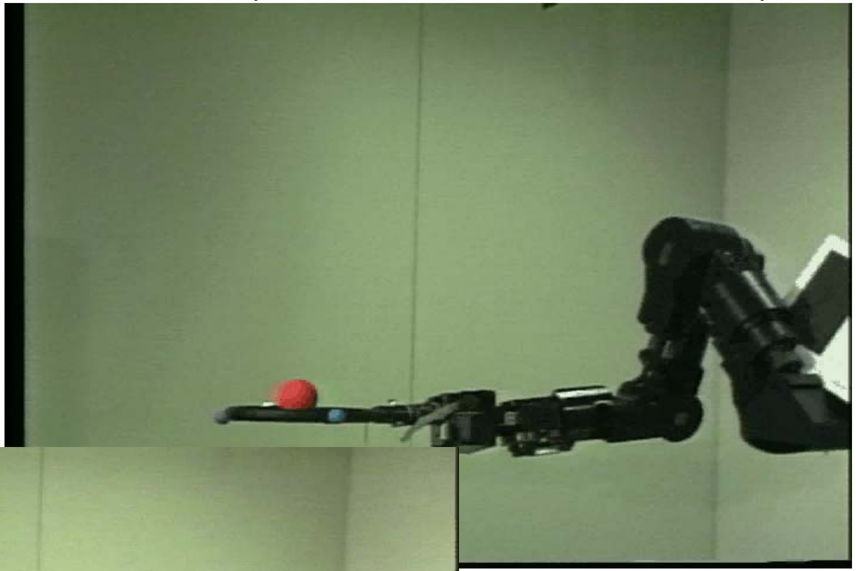
Discreet & Rhythmic Movement Primitives

- **Represent complex movements in *globally stable* attractor landscapes of nonlinear autonomous differential equations**
- **Choose kinematic representation for easy re-use in different workspace location**
- **Ensure easy temporal and spatial scaling (topological equivalence)**
- **Use local learning to modify the attractors according to teacher demonstration of teacher and self-learning**

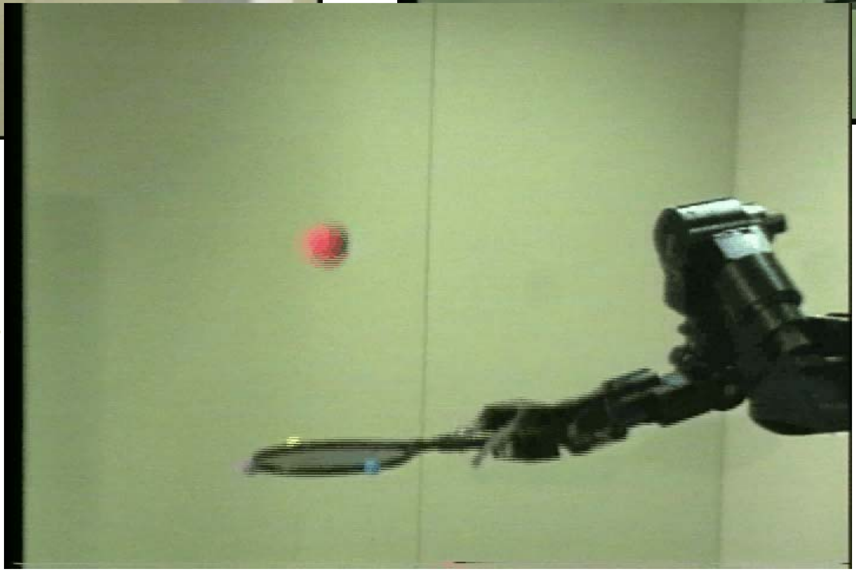
Discreet & Rhythmic Movement superposition

Open loop with oscillators

Closed loop control in horizontal plane



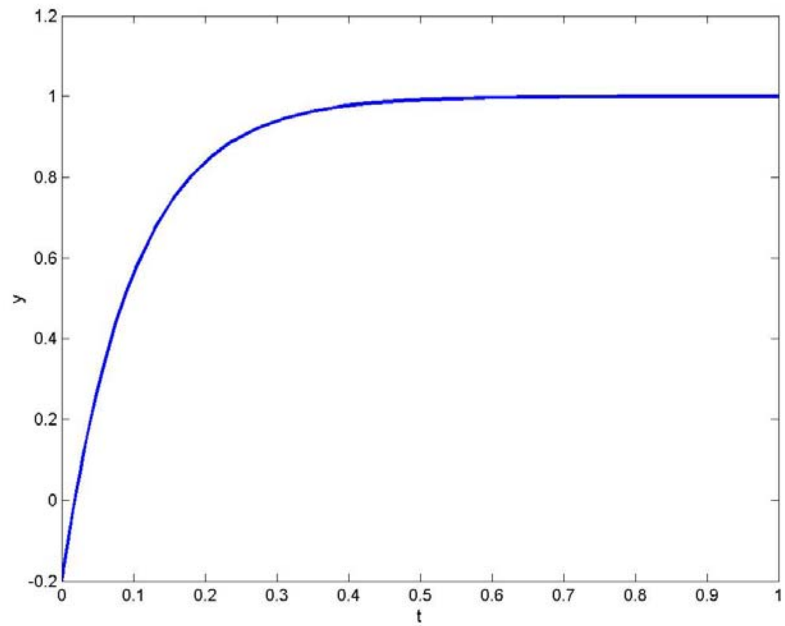
Open (vertical) +
Closed (horizontal)
loop control



What is a Differential Equation?

- ***Differential equation***: an equation that describes how state variables evolve over time, for instance:

$$\dot{y} = \alpha(c - y)$$



Some Definitions

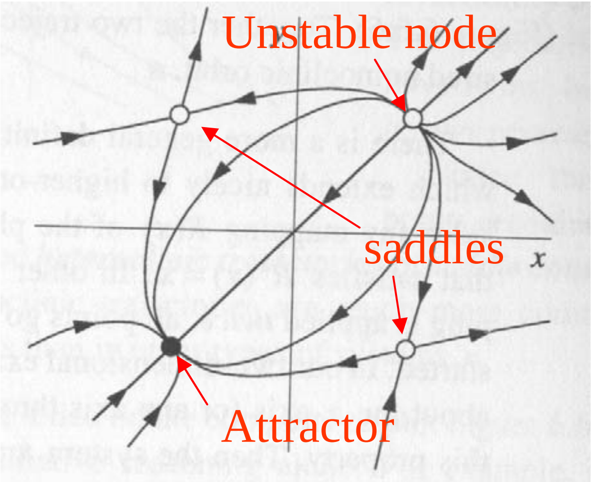
- ***Ordinary differential equation****:* differential equation that involves only ordinary derivatives (as opposed to partial derivatives)
- ***Autonomous equation****:* differential equation that does not (explicitely) depend on time

- ***Linear differential equation****:* differential equation in which the state variables only appear in linear combinations
- ***Nonlinear differential equation****:* differential equation in which some state variables appear in nonlinear combinations (e.g. products, cosine,…)

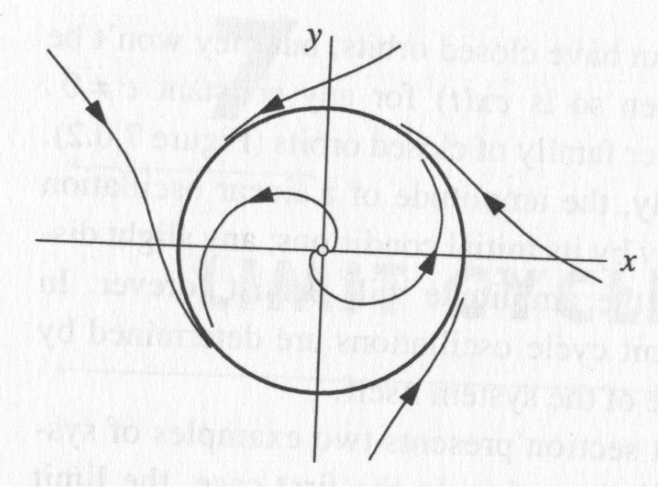
- ***Fixed point****:* point at which all derivatives are zero (can be an attractor, a repeller, or a saddle point, cf later)
- ***Limit cycle****:* periodic isolated closed trajectory (can only occur in nonlinear systems)

Interesting Regimes of Differential Equations

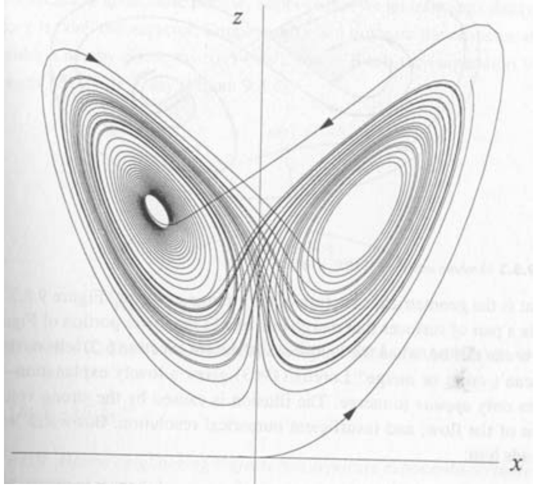
From Strogatz 1994



Attractors



Limit cycles



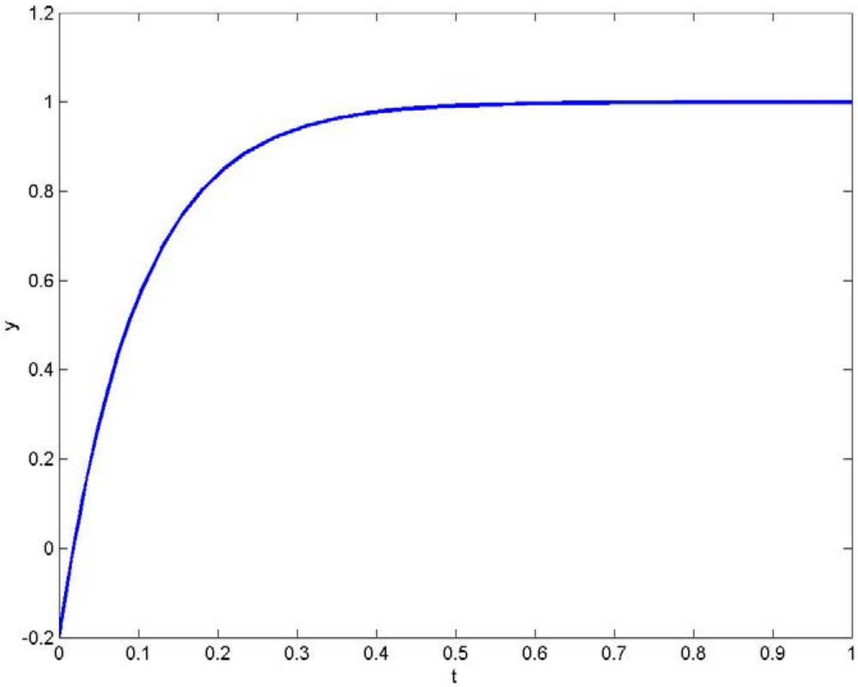
Chaos

First Order Linear Systems

- First order linear system: $\dot{y} = \alpha(c - y)$
- How to solve this equation, for a given *y(t=0)*, *c*, and α ?
- Two methods: analytical solution or numerical integration
- Analytical solution: $y(t) = (y_0 - c)\exp(-\alpha t) + c$
- Numerical integration: Euler method, Runge-Kutta,…

First Order Linear Systems

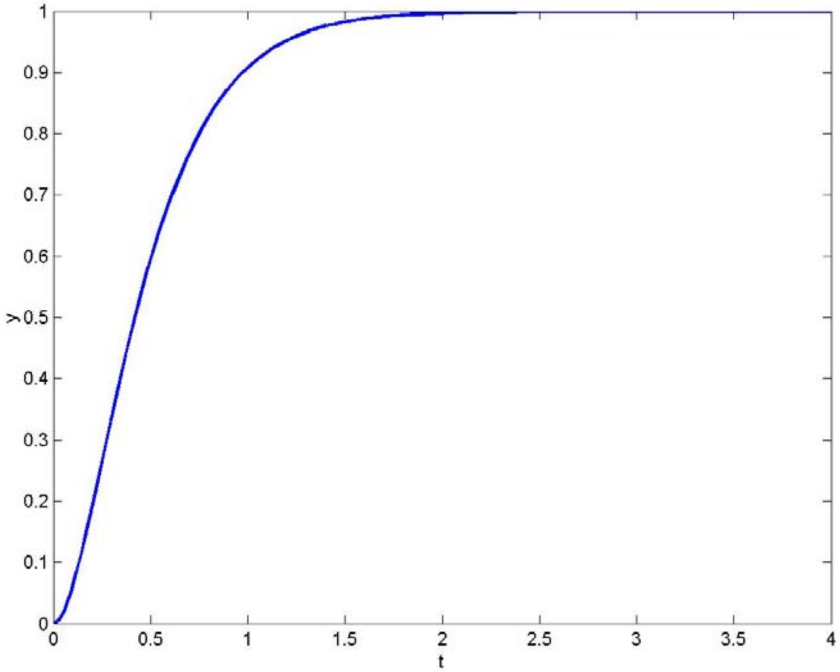
$$\dot{y} = \alpha(c - y)$$



$$y(t) = (y_0 - c)\exp(-\alpha t) + c$$

Second Order Linear Systems

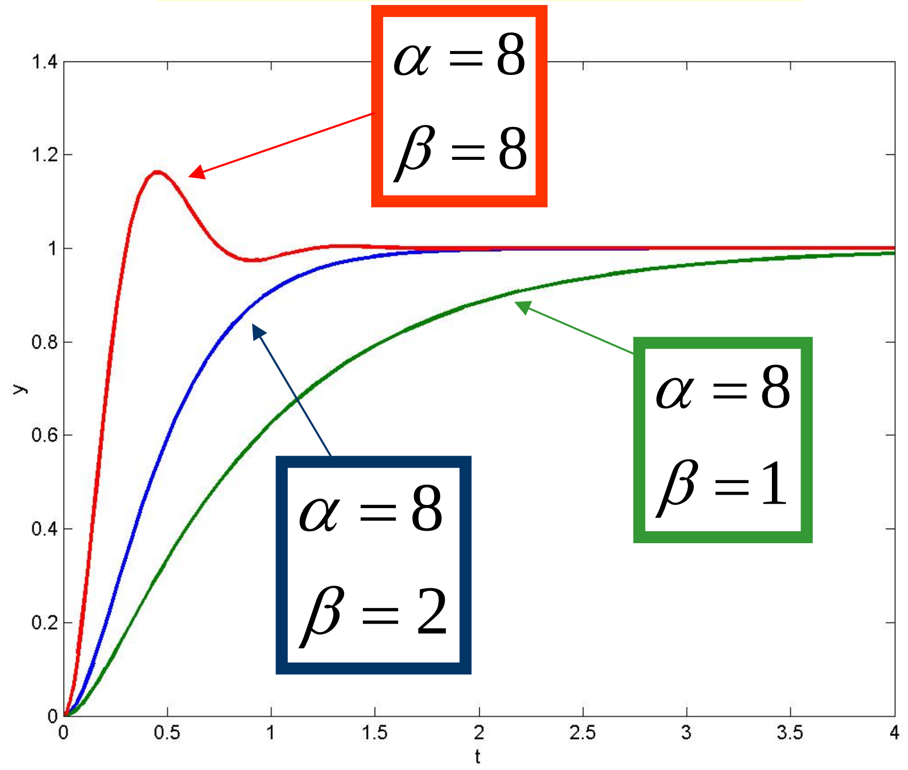
$$\dot{y} = \alpha(\beta(c - x) - y)$$
$$\dot{x} = y$$



Second Order Linear Systems

$$\dot{y} = \alpha(\beta(c - x) - y)$$

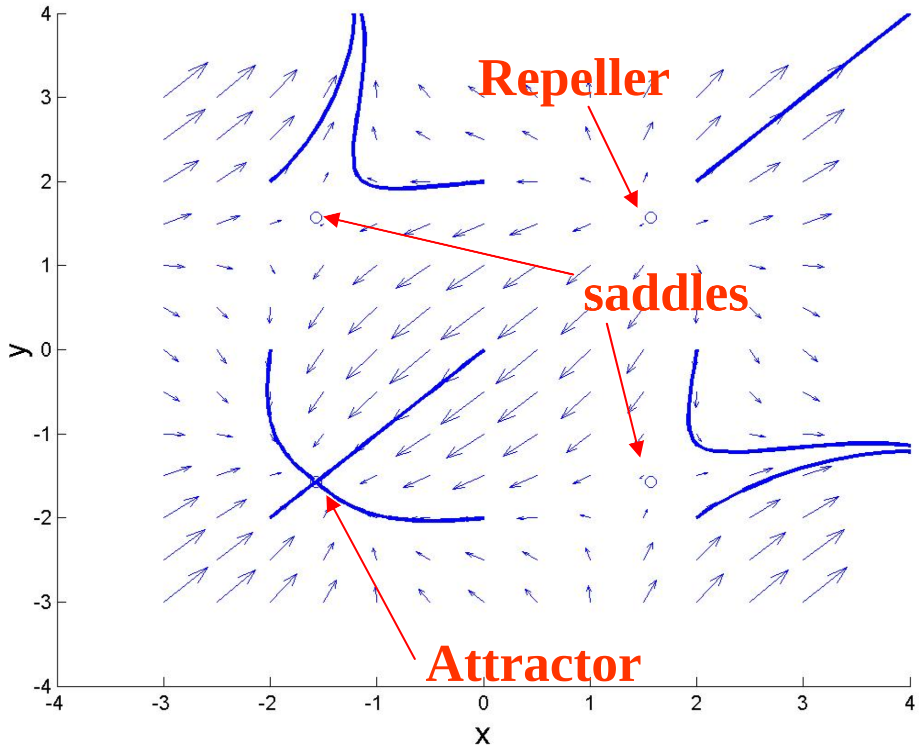
$$\dot{x} = y$$



Second Order Non-linear System

$$\dot{y} = -2\cos x - \cos y$$

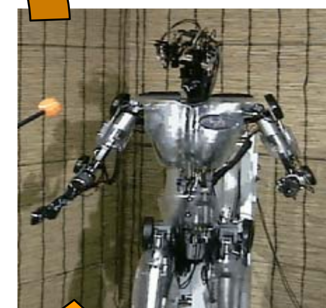
$$\dot{x} = -2\cos y - \cos x$$



Learning a movement by demonstration

Joint angles

Movement recognition → Trajectory formation system → Movement execution (Inv. Dyn.)



Task of the trajectory formation system:

- To encode demonstrated trajectories with high accuracy,
- To be able to modulate the learned trajectory when:
  - Perceptual variables are varied (e.g. timing, amplitude)
  - Perturbations occur

Encoding a trajectory

Traditionally, the problem of replaying a trajectory has been decomposed into two different issues:

- One of ***encoding*** the trajectory, and
- One of ***modifying*** the trajectory, for instance, in case the movement is perturbed, or when it requires to be modulated.

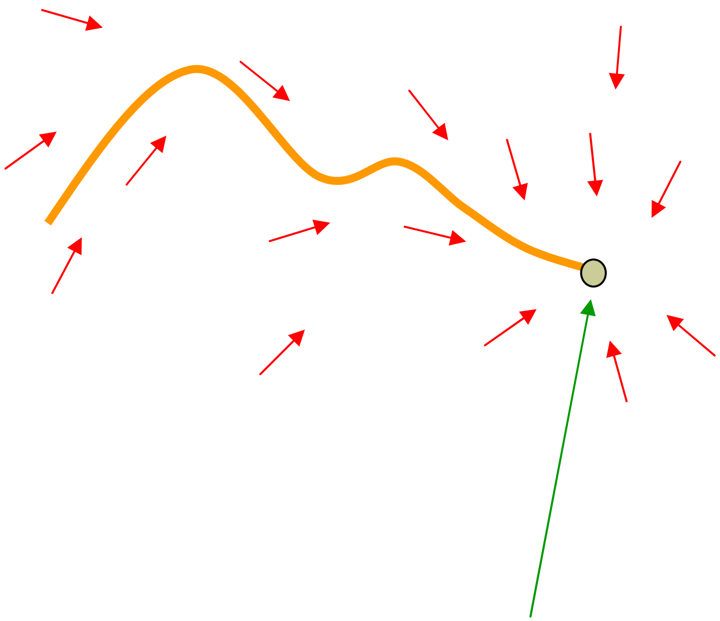
Our approach: combine both abilities in a **nonlinear dynamical system**

Aim: to encode the trajectory in a nonlinear dynamical system with well defined attractor landscape

Nonlinear Dynamical Systems Approach

Discrete movements

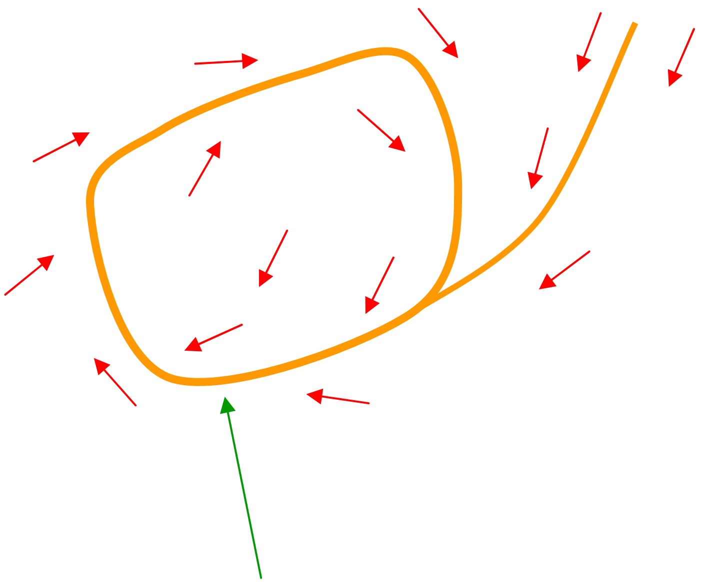
dy/dt



Single point attractor

Rhythmic movements

dy/dt

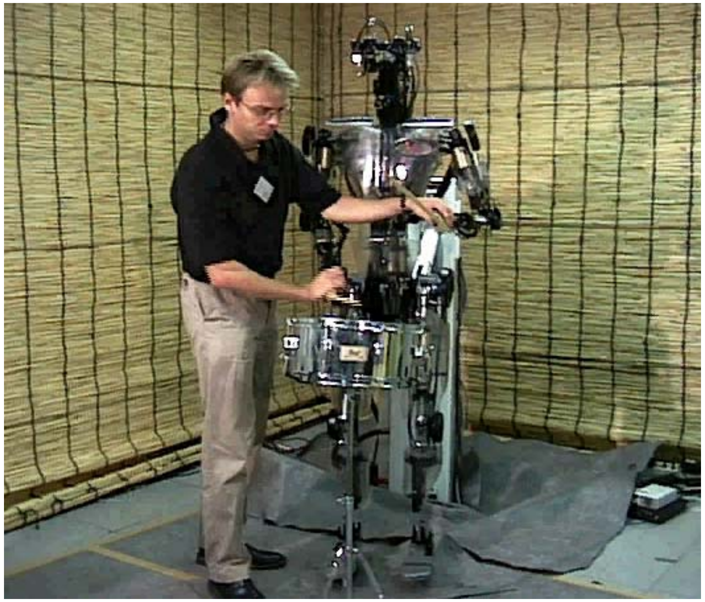


Limit cycle attractor

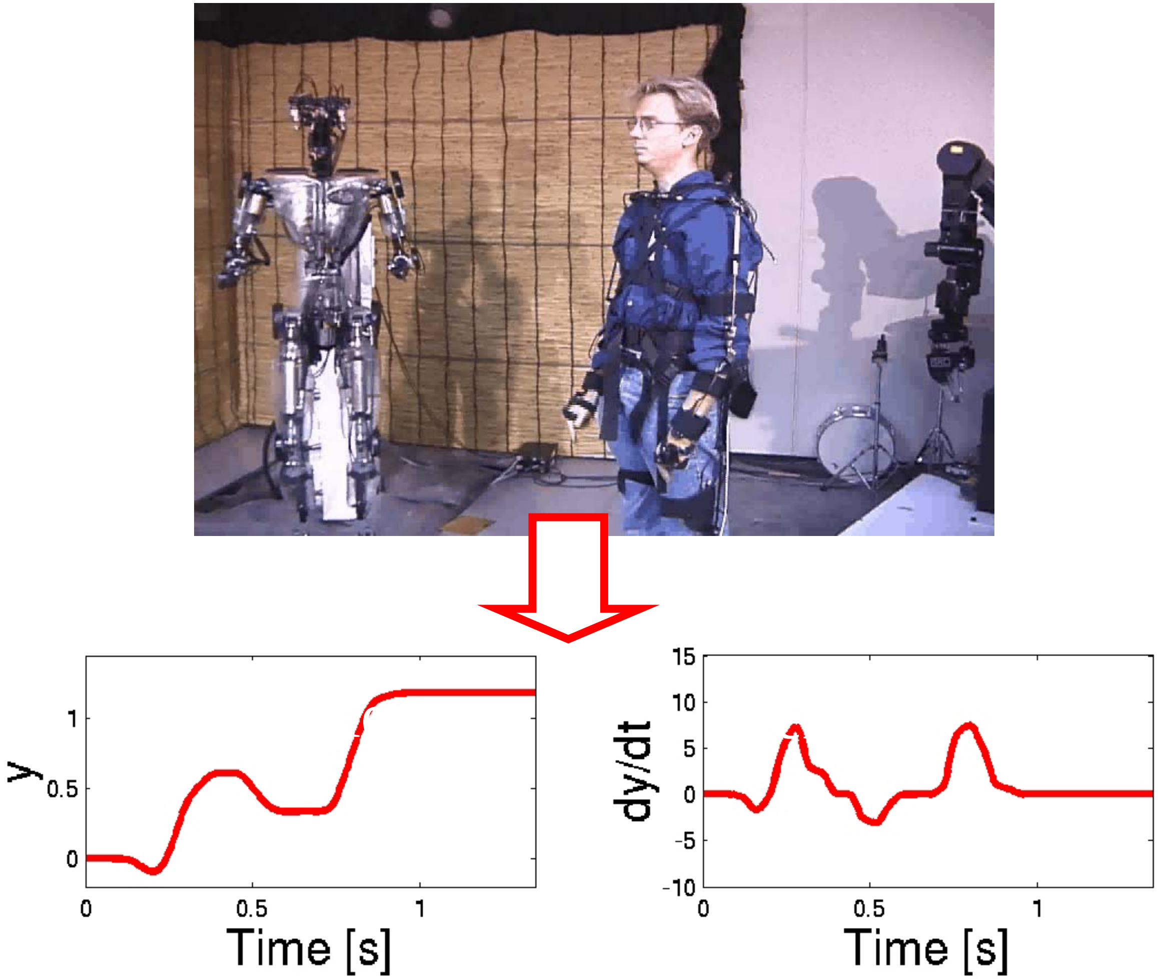
Two types of movement recordings

Sensuit

« Kinesthetic » demonstration



Pointing Demonstration

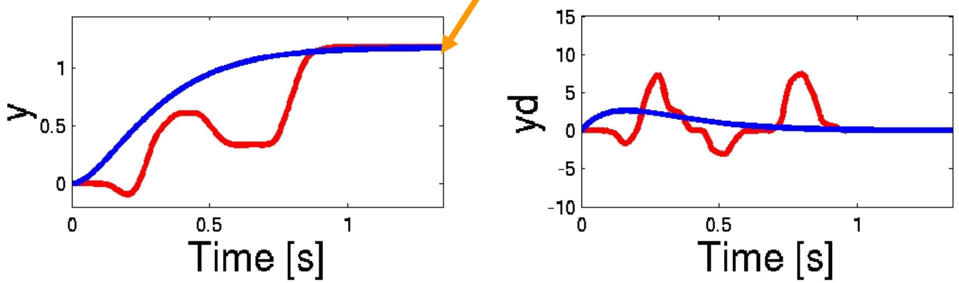


Shaping Attractor Landscapes

$$\dot{z} = \alpha_z(\beta_z(g - y) - z)$$

$$\dot{y} = z$$

Goal: g



$$\dot{z} = \alpha_z\left(\beta_z\left(g - y\right) - z\right)$$
$$\dot{y} = z$$

Can one create more complex dynamics by non-linearly modifying the dynamic system:

$$\dot{z} = \alpha_z\left(\beta_z\left(g - y\right) - z\right)$$
$$\dot{y} = \left(f\left(?\right) + z\right)$$

Discreet Control Policy

Output System

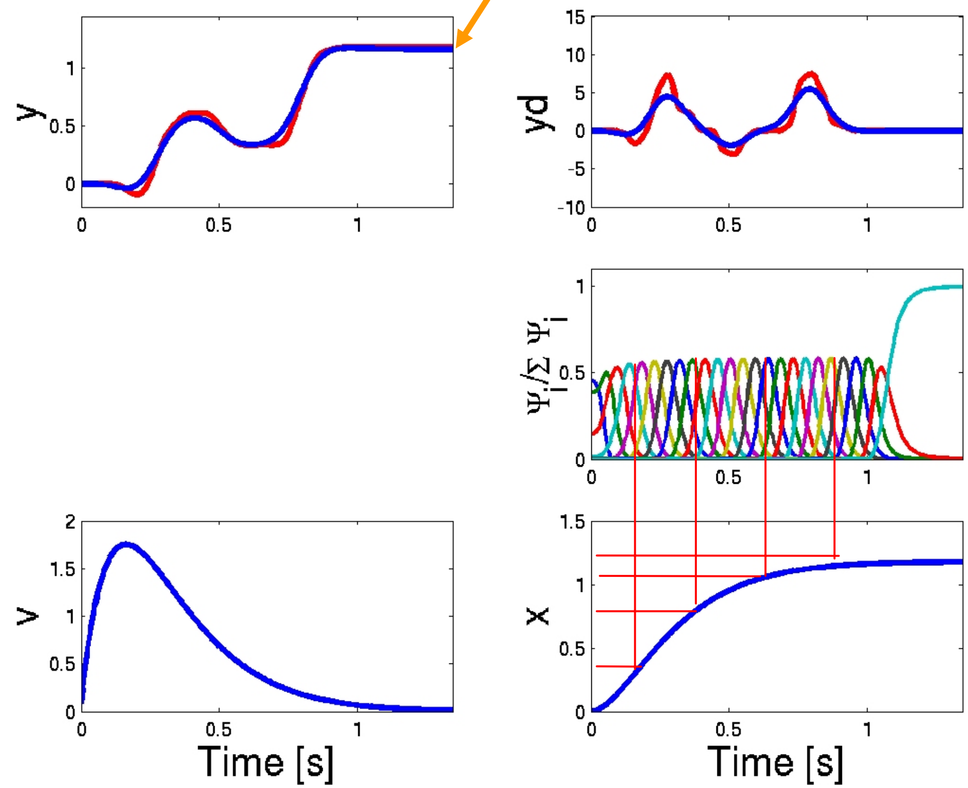
$$\dot{z} = \alpha_z(\beta_z(g - y) - z)$$
$$\dot{y} = z + \frac{\sum_{i=1}^{N} \Psi_i w_i}{\sum_{i=1}^{N} \Psi_i} v$$

$$\Psi_i = \exp\left(-\frac{1}{2\sigma_i^2}(\tilde{x} - c_i)^2\right)$$

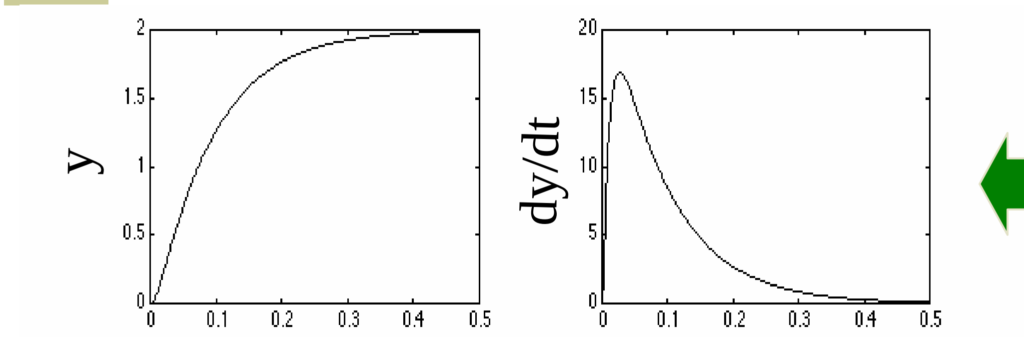
$$\dot{v} = \alpha_v(\beta_v(g - x) - v)$$
$$\dot{x} = v$$

Amplitude and phase system

Goal: g



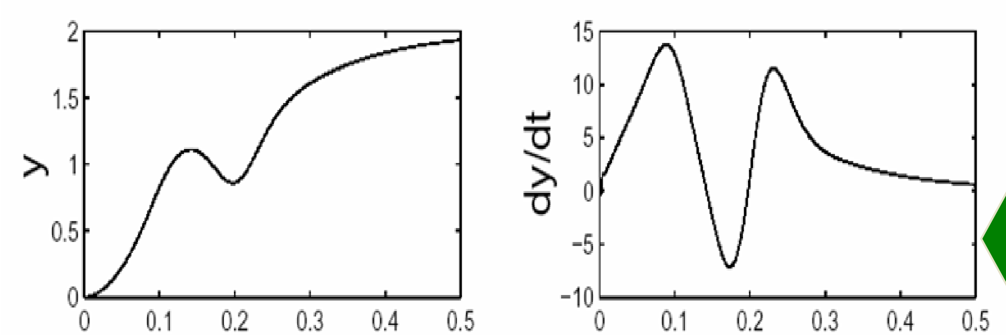
Shaping attractor landscapes



$$\dot{z} = \alpha_z\left(\beta_z\left(g-y\right)-z\right)$$
$$\dot{y} = z$$

g = goal

Can one create more complex dynamics by non-linearly modifying the above dynamic system:



$$\dot{z} = \alpha_z\left(\beta_z\left(g-y\right)-z\right)$$
$$\dot{y} = \left(f\left(?\right)+z\right)$$

g = goal

Shaping Attractor Landscapes

A globally stable learnable nonlinear point attractor:

Trajectory Plan Dynamics

$$\dot{z} = \alpha_z \left(\beta_z \left(g - y \right) - z \right)$$
$$\dot{y} = \alpha_y \left(f \left(x, v \right) + z \right)$$

where

Canonical Dynamics

$$\dot{v} = \alpha_v \left(\beta_v \left(g - x \right) - v \right)$$
$$\dot{x} = \alpha_x v$$

Local Linear Model Approx.

$$f \left(x, v \right) = \frac{\sum_{i=1}^{k} w_i b_i v}{\sum_{i=1}^{k} w_i}$$

$$w_i = \exp\left(-\frac{1}{2} d_i \left(\overline{x} - c_i \right)^2 \right) \text{ and } \overline{x} = \frac{x - x_0}{g - x_0}$$

Learning the Attractor

Given a demonstrated trajectory $y(t)_{demo}$ and a goal g

- Extract movement duration
- Adjust time constants of canonical dynamics to movement duration
- Use LWL to learn supervised problem

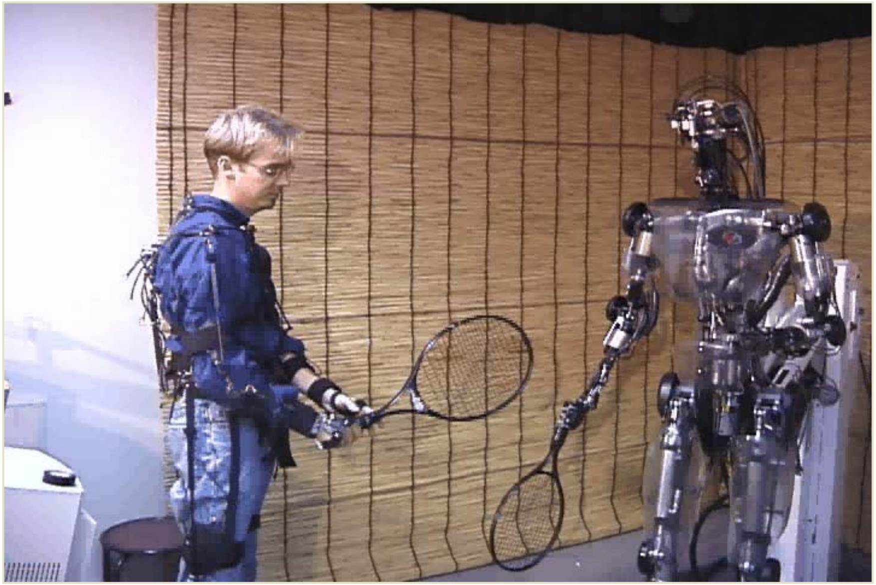
$$\dot{y}_{\text{target}} = \frac{\dot{y}_{demo}}{\alpha_y} - z = f(x, v)$$

Also extended to rhythmic primitives :

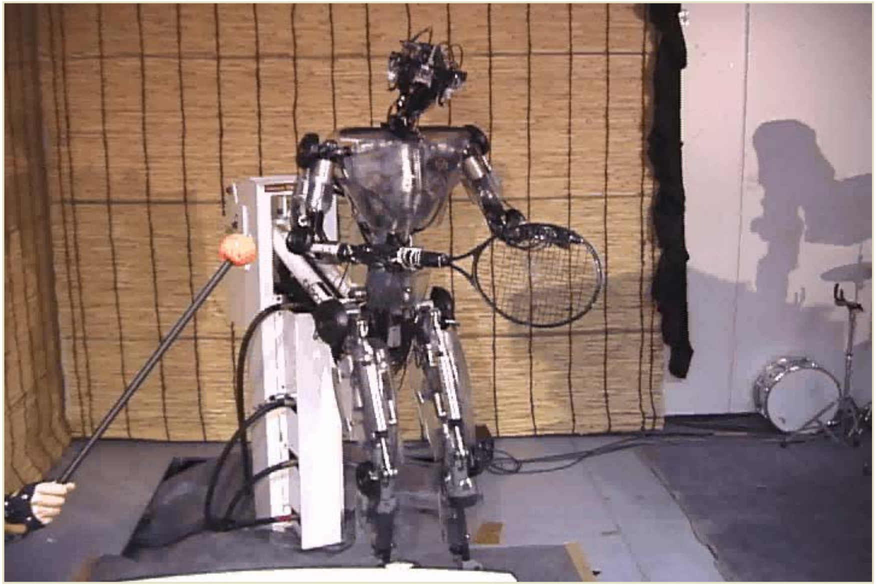
[Stefan Schaal, Sethu Vijayakumar et al, *Proc. of Intl. Symp. Rob. Res.(ISRR)* (2001)]

Trajectory following & Generalization

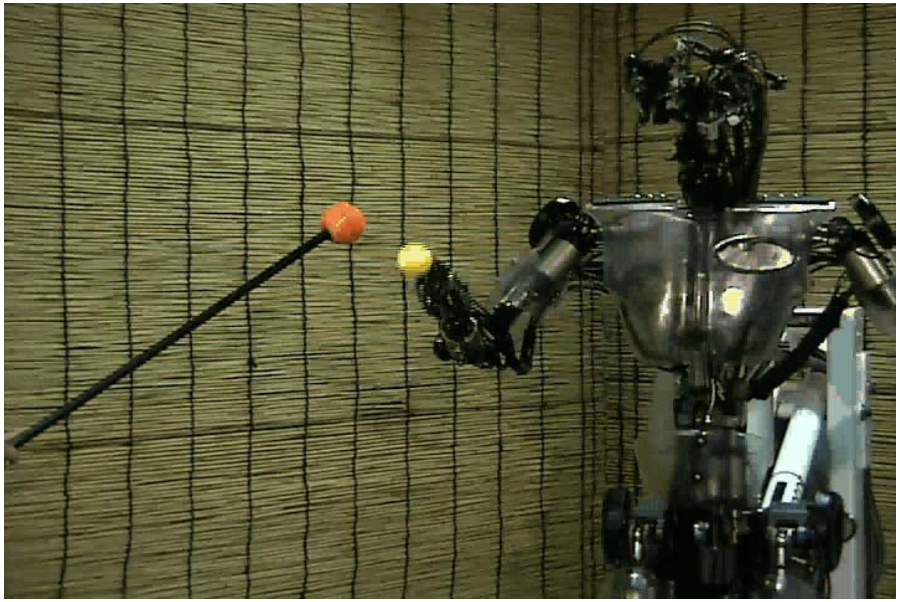
Backhand Demonstration



Backhand Reproduction

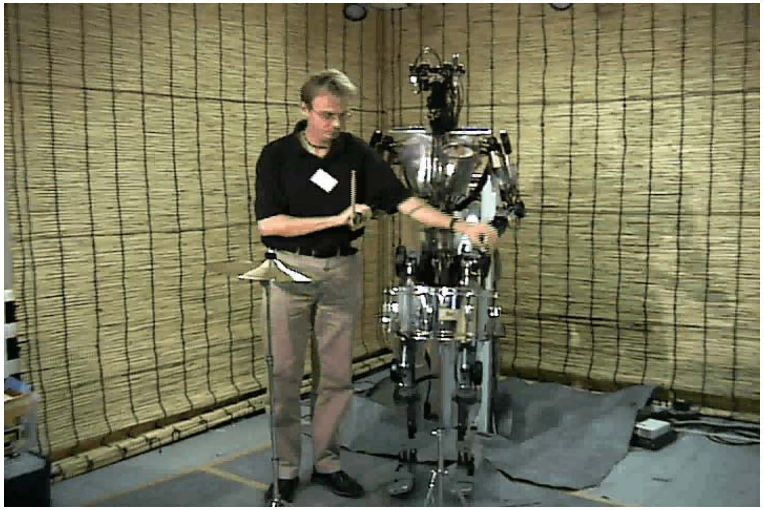


Modulation of Goal: Anchor



Drumming: Modulating Frequency

Drumming: Kinesthetic Demo



Imitation and Modulation