Automated Learning

Learn low level controllers
compose into high level controllers

spring-mass systems
snakes, fish, marine mammals

locomotion trials to learn low level controllers

optimize composition for high level task completion

Snake biomechanical model

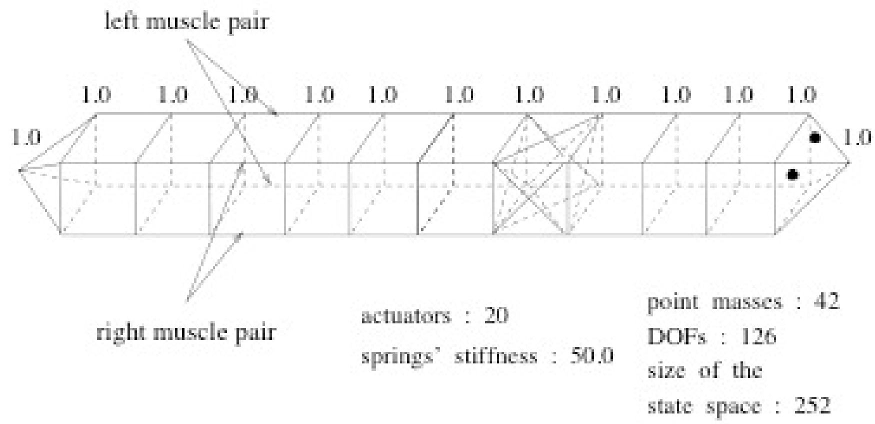


Figure 1: *The snake biomechanical model consists of nodal masses (points) and springs (lines). It has twenty independent actuators (muscle springs): ten on the left side of the body and ten on the right side. Each actuator comprises a pair of synchronous muscles. The numbers along the body indicate nodal masses in cross sectional planes. The cross-springs, shown in only one segment, maintain the structural integrity of the body.*

Low level control

Lagrange equation of motion

$$m_i \ddot{\mathbf{x}}_i + \gamma_i \dot{\mathbf{x}}_i + \sum_{j \in N_i} \mathbf{f}^s_{ij} = \mathbf{f}_i$$

Low level control

Evaluation function

$$E(\mathbf{u}(t)) = \int_{t_0}^{t_1} \left(\mu_1 E_u(\mathbf{u}(t)) + \mu_2 E_v(\mathbf{v}(t))\right) \, dt,$$

controller evaluation metric
e.g. distance traveled

trajectory evaluation metric
e.g. final distance to goal,
deviation from desired speed

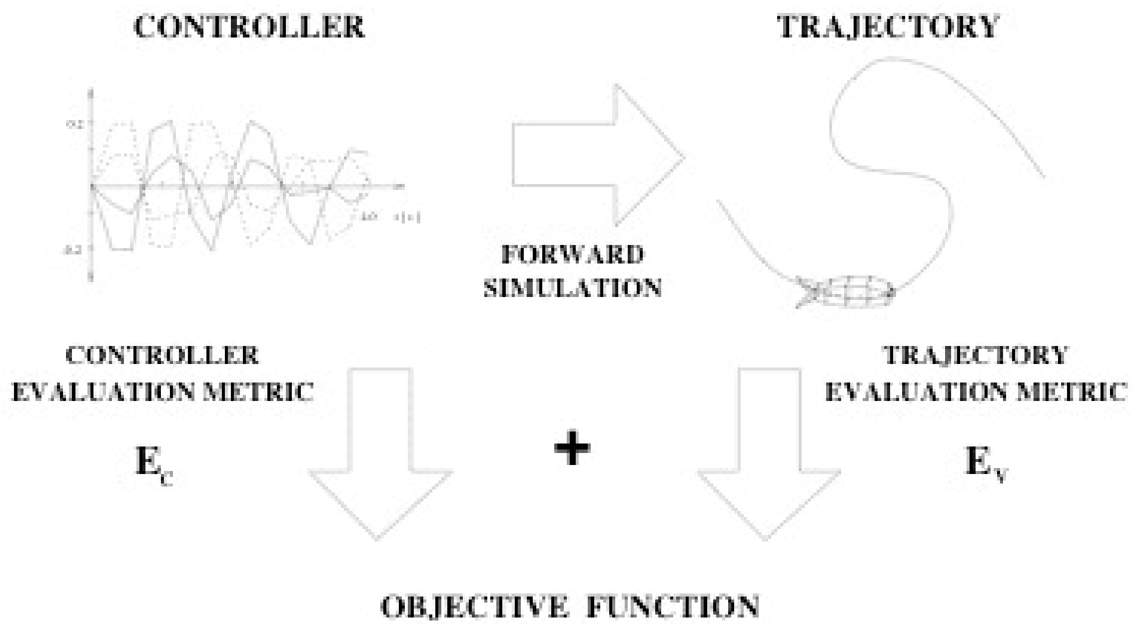


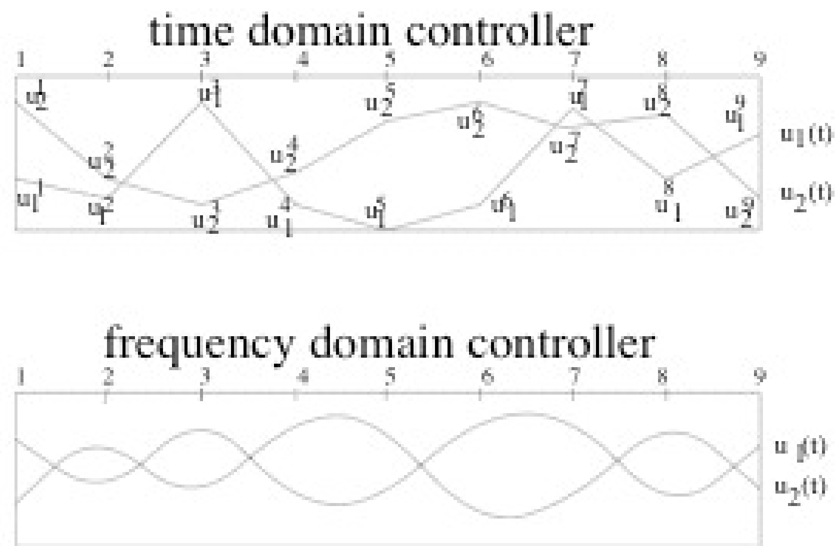
Figure 2: *The objective function guiding the optimization is a weighted sum of terms that evaluate the trajectory and the control function.*

encourage lower amplitude, smoother controllers

$$E_{\mathbf{u}} = \frac{1}{2}\left(\nu_1 \left|\frac{d\mathbf{u}}{dt}\right|^2 + \nu_2 \left|\frac{d^2\mathbf{u}}{dt^2}\right|^2\right),$$

time & frequency domain discrete controllers

$$u_i(t) = \sum_{j=1}^{M} u_i^j B^j(t),$$



$$\mathbf{u}_1(t) = \mathbf{u}_1^1\boldsymbol{B}^1 + \mathbf{u}_1^2\boldsymbol{B}^2 + \mathbf{u}_1^3\boldsymbol{B}^3 + \mathbf{u}_1^4\boldsymbol{B}^4 + \mathbf{u}_1^5\boldsymbol{B}^5 + \mathbf{u}_1^6\boldsymbol{B}^6 + \mathbf{u}_1^7\boldsymbol{B}^7 + \mathbf{u}_1^8\boldsymbol{B}^8 + \mathbf{u}_1^9\boldsymbol{B}^9$$

$$\mathbf{u}_2(t) = \mathbf{u}_2^1\boldsymbol{B}^1 + \mathbf{u}_2^2\boldsymbol{B}^2 + \mathbf{u}_2^3\boldsymbol{B}^3 + \mathbf{u}_2^4\boldsymbol{B}^4 + \mathbf{u}_2^5\boldsymbol{B}^5 + \mathbf{u}_2^6\boldsymbol{B}^6 + \mathbf{u}_2^7\boldsymbol{B}^7 + \mathbf{u}_2^8\boldsymbol{B}^8 + \mathbf{u}_2^9\boldsymbol{B}^9$$

Figure 3: *Simple time domain controller (top) with two control functions* $\mathbf{u}_1(t)$ *and* $\mathbf{u}_2(t)$. *Each function is a piecewise linear polynomial generated by 9 control points. Simple frequency domain controller (bottom) with two control functions, each a sum of 9 sinusoidal basis functions* $\boldsymbol{B}^j(t) = \cos(\omega^j t + \phi^j)$.

optimization of discrete object function

N control functions, M basis functions

NM parameters

$$E([u_1^1, ..., u_N^M])$$

optimize parameters

optimization of discrete object function

<u>Simulated annealing</u>
global
no gradient
for large DoF problems

<u>Simplex method</u>
local
fast

Plate 1: *Locomotion pattern learned by the artificial snake*.

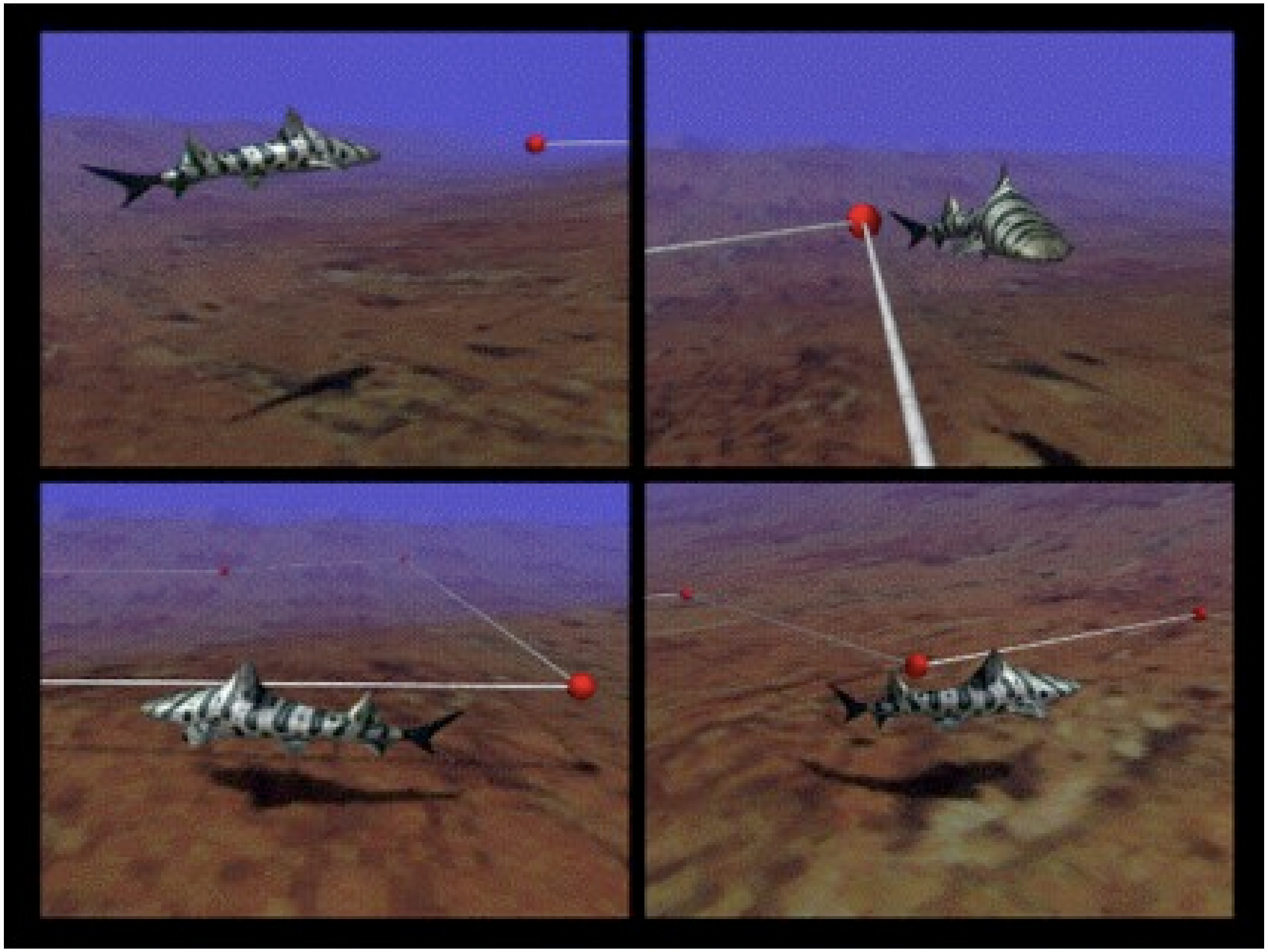


Plate 4: *Target tracking using abstracted controllers (see text).*

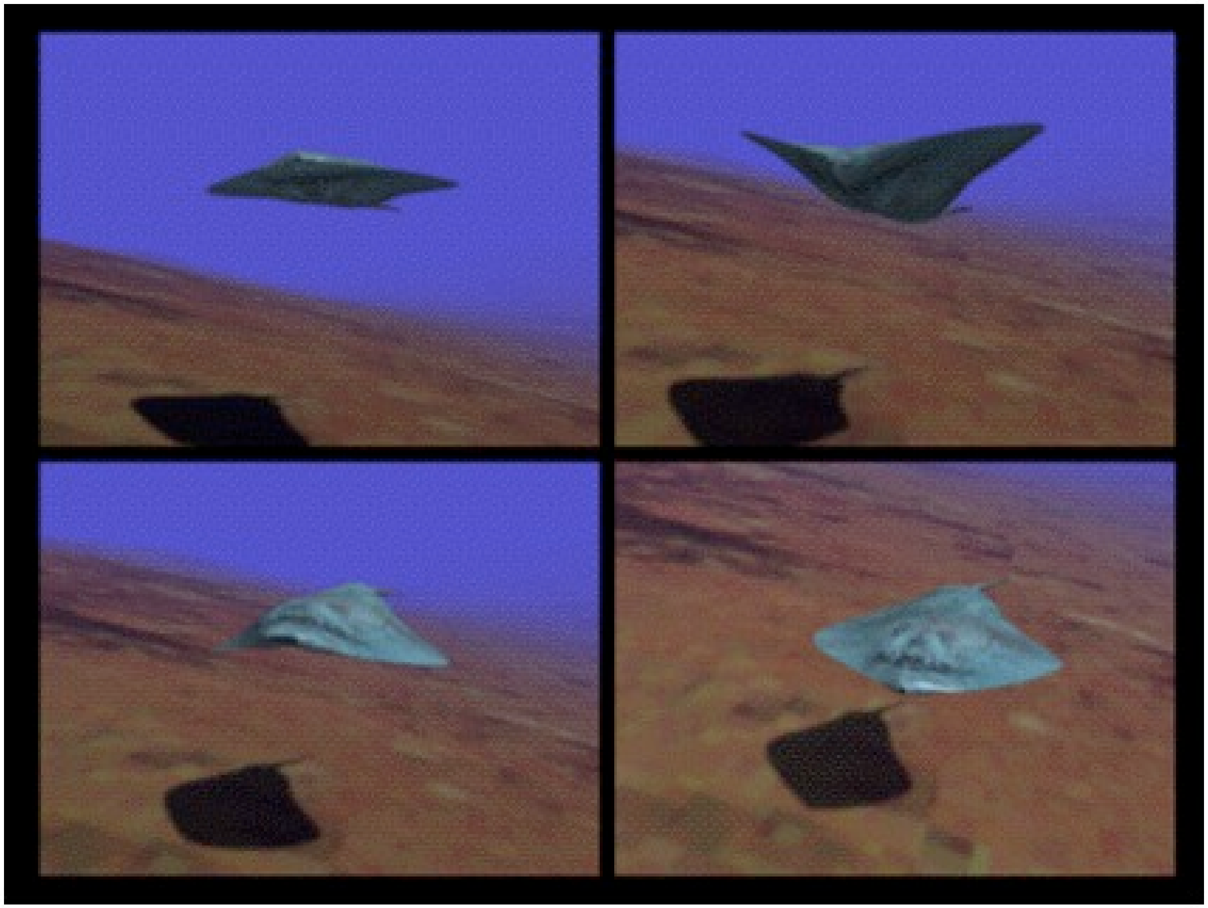


Plate 2: *Locomotion pattern learned by the artificial ray.*

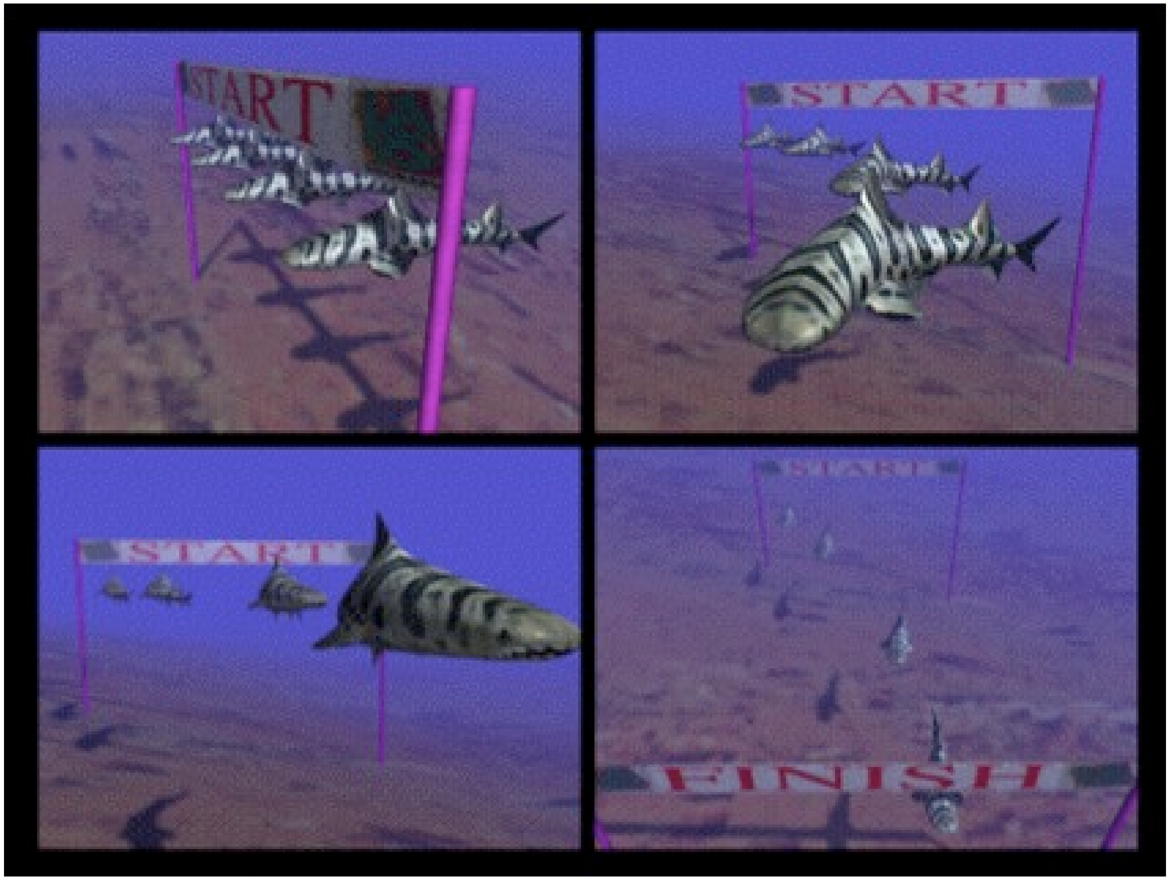


Plate 3: *Shark race illustrates the progress of learning (see text).*

Abstracting high level control

dimensionally reducing change of representation

to avoid complexity, must abstract compact higher level controllers from low level learned controllers

reuse low level controllers for different tasks

natural, steady state locomotion is quasi-periodic
use frequency domain controller
apply FFT to time domain controller
suppress those with small amplitudes

“basic training” to develop low level controllers

concatenate in sequence, with blended overlap for higher level tasks

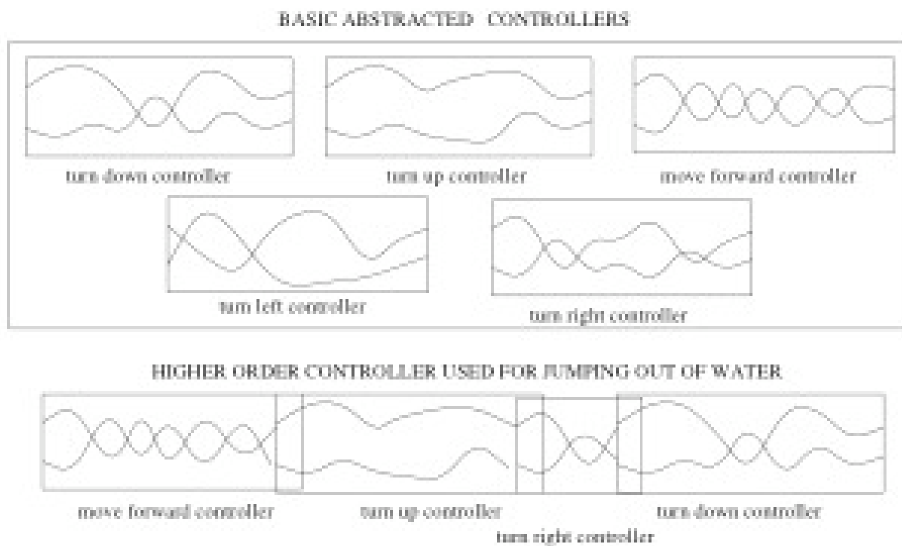


Figure 4: *Higher level controller for jumping out of water is constructed from a set of abstracted basic controllers.*

abstract controllers in 2 ways

greedy algorithm of compositing low level controllers

fails for higher level tasks (e.g. planning)

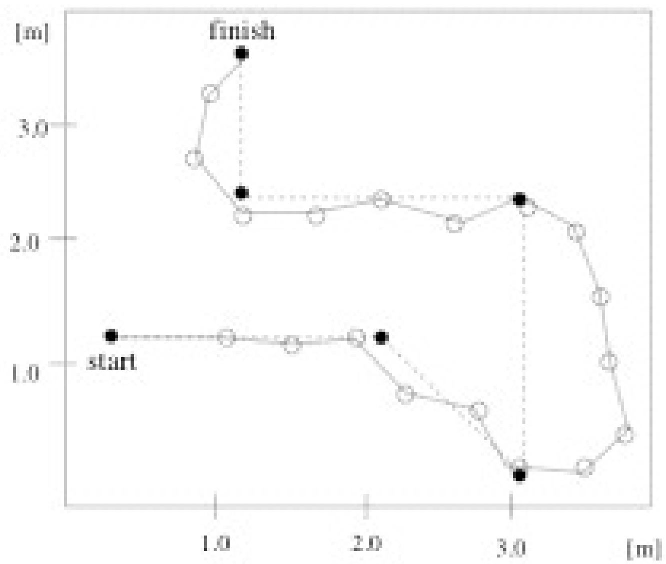


Figure 5: *The solid curve indicates the path of the fish tracking the goal. Black dots mark consecutive positions of the goal and white dots mark the starting point of a new controller.*

optimize sequence of controllers

simulated annealing

optimizes over selection, ordering, duration of controllers composing macro controllers

train on 5 basic tasks:
turn-up, turn-left, turn-right, move-forward

optimize on jumping out of water

add 'style' terms for height, body alignment, etc.

simulated annealing not great for this
doesn't retain partial solutions
maybe use genetic algorithms in future

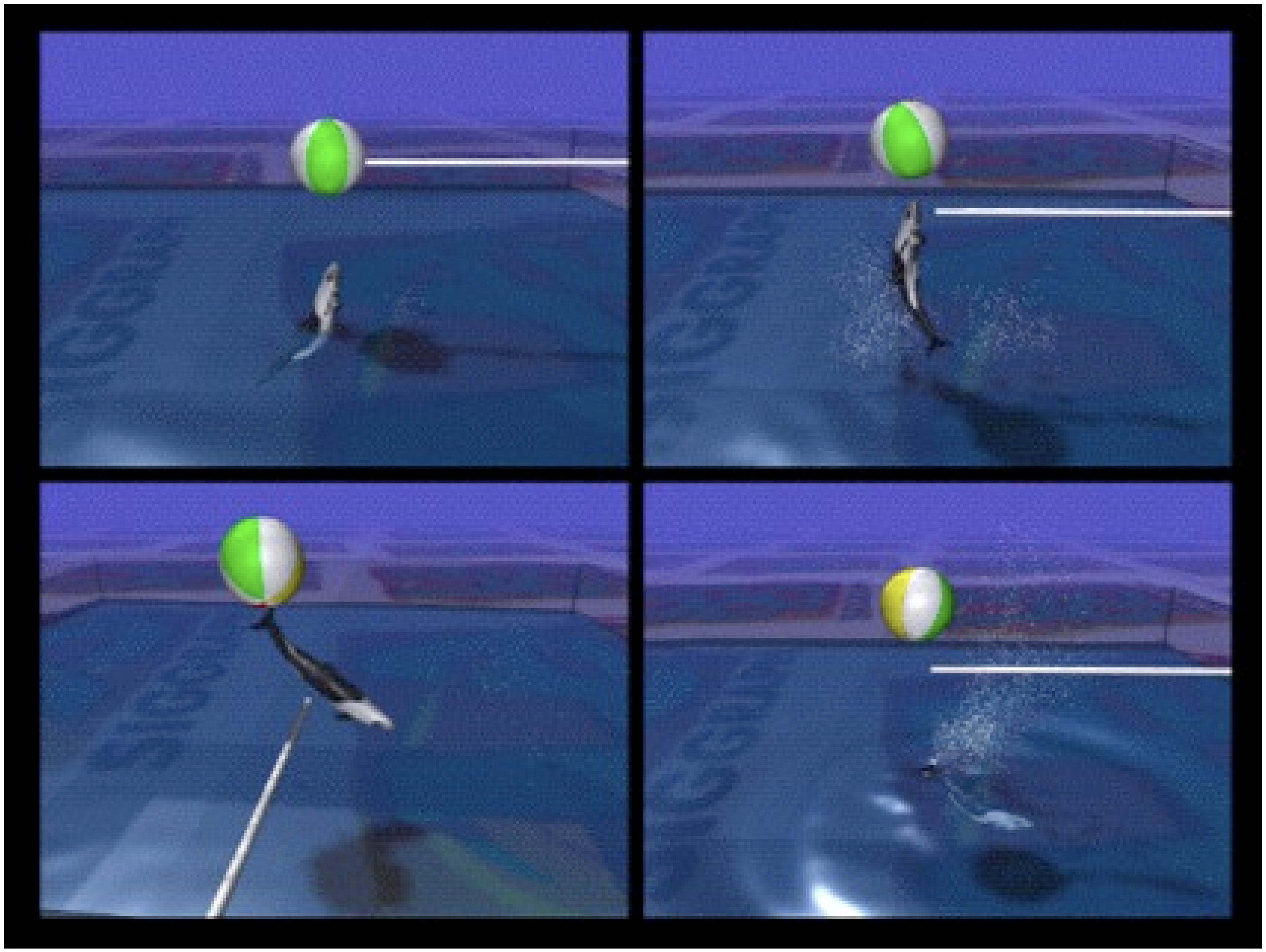


Plate 5: *SeaWorld tricks learned by the artificial dolphin*.

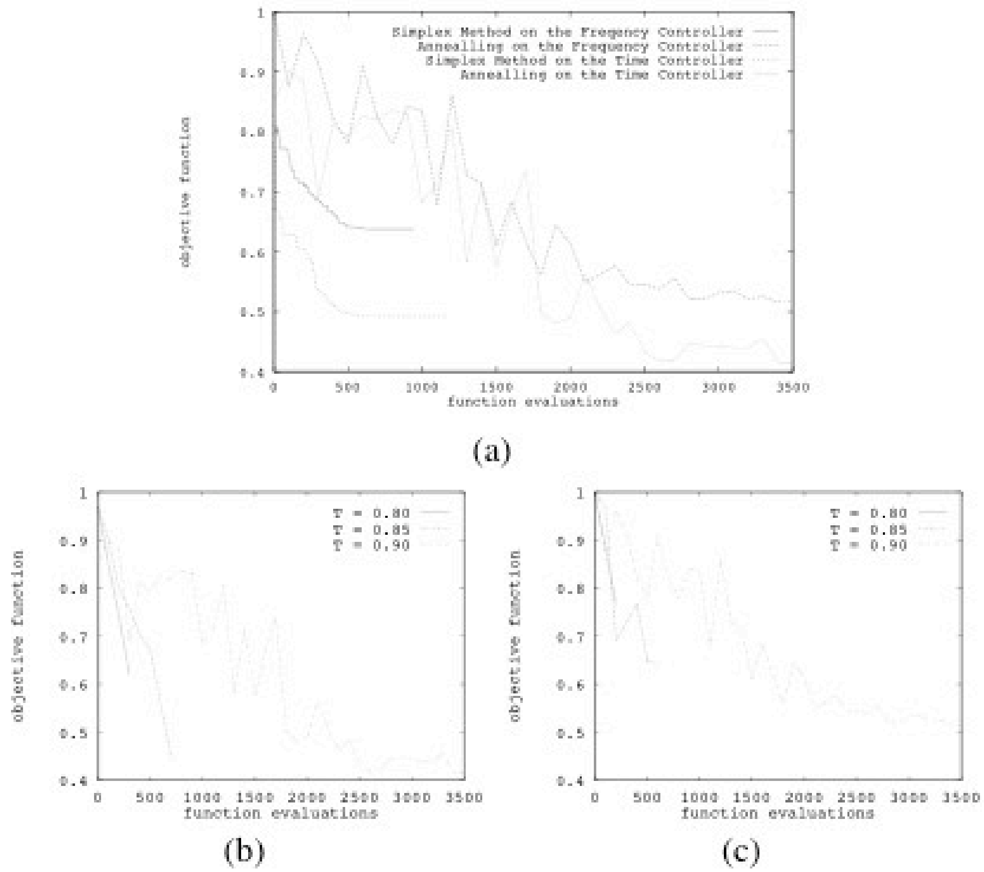
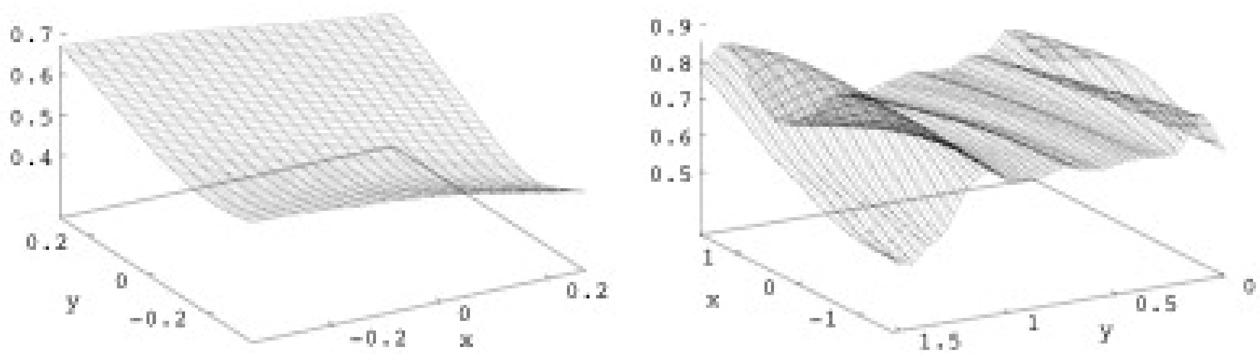
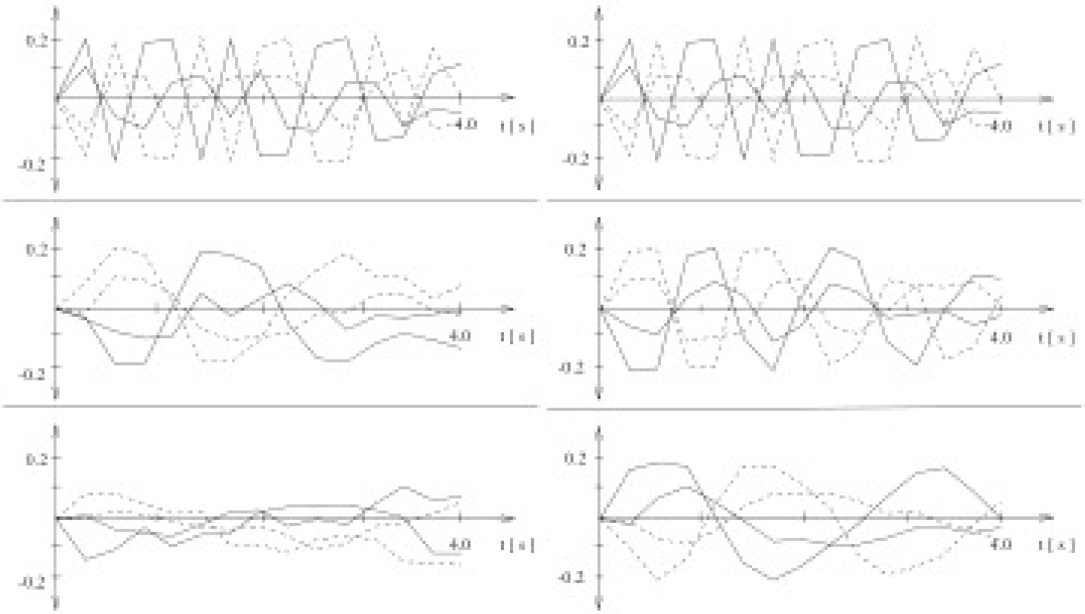


Figure 6: *(a) Performance comparison of the simplex method and of simulated annealing. Convergence rate of simulated annealing on the time domain controller (b) and on the frequency controller (c) with cooling rates:* $T_0 = 0.8$, $T_1 = 0.85$, *and* $T_2 = 0.9$.



(a) (b)

Figure 7: *Topography of objective function (in 2 dimensions) of time domain representation (a) and frequency domain representation (b).*



(a) (b)

Figure 8: *Influence of E_u on controller: (a)* $\nu_2 = 0$*; (b)* $\nu_1 = 0$.

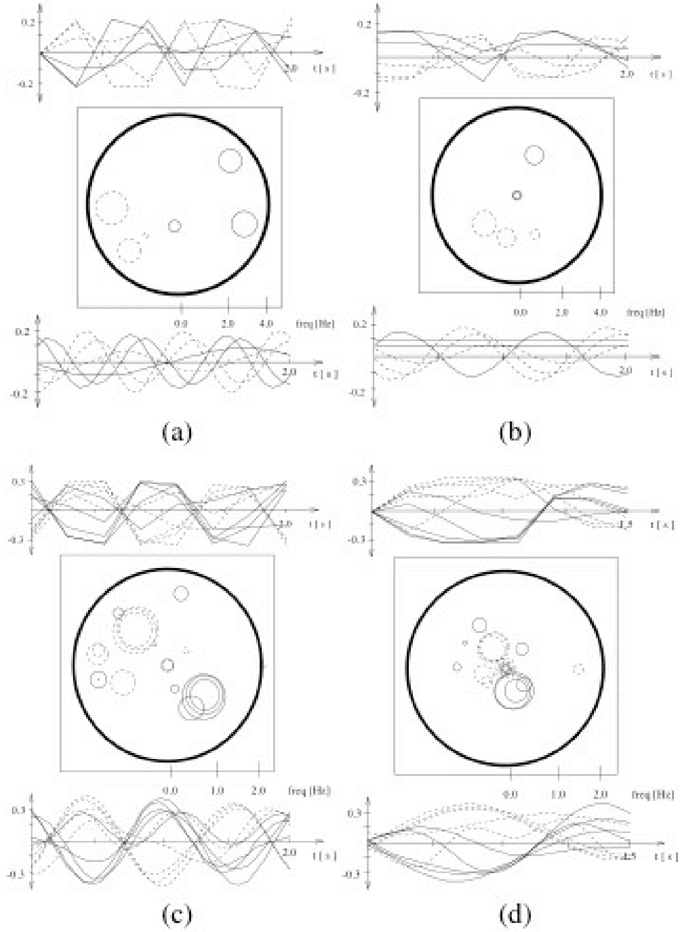


Figure 9: *Learned controller for the swim straight (a) and the left turn (b) for the shark. Learned controller for the straight motion (c) and the left turn (d) for the snake. For each part: (top) learned time domain controller (dotted lines indicate actuator functions for left side of body, solid lines indicate actuator functions for right side); (center) primary modes of controller FFT (radius of circles indicates mode amplitudes, radial distances from center of surrounding circle indicate frequencies, angular positions within surrounding circle indicate phases); (bottom) abstracted controller obtained by retaining primary modes.*

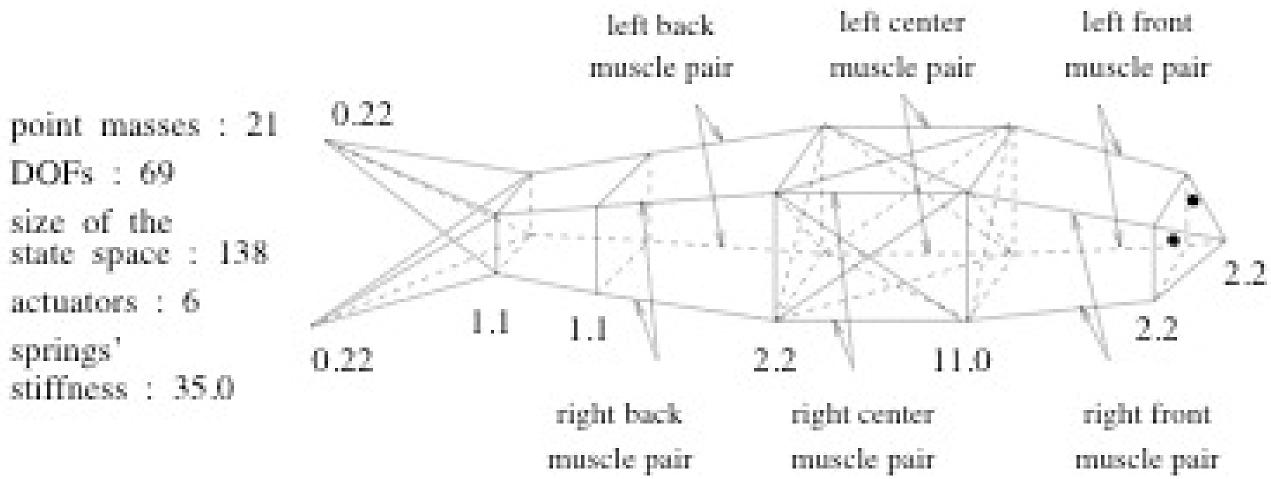


Figure 10: *The shark biomechanical model has six actuators consisting of a pair of muscles that share the same activation function. The numbers along the body indicate the mass of each point in the corresponding cross sectional plane. The cross-springs that maintain the structural integrity of the body are indicated in one of the segments only.*

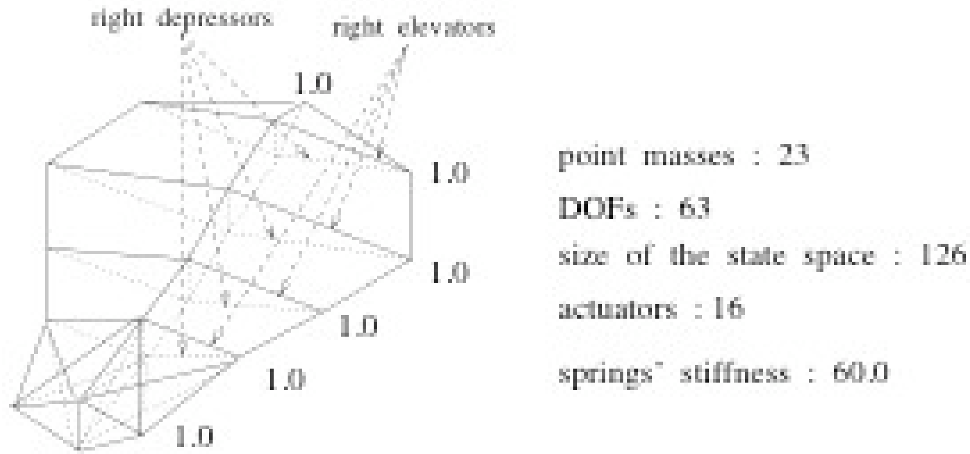


Figure 11: *The ray biomechanical model has four sets of actuators: left and right depressors and left and right elevators. The numbers along the body indicate the mass of each point in the corresponding cross sectional plane. The cross-springs that maintain the structural integrity of the body are indicated in one of the segments only.*