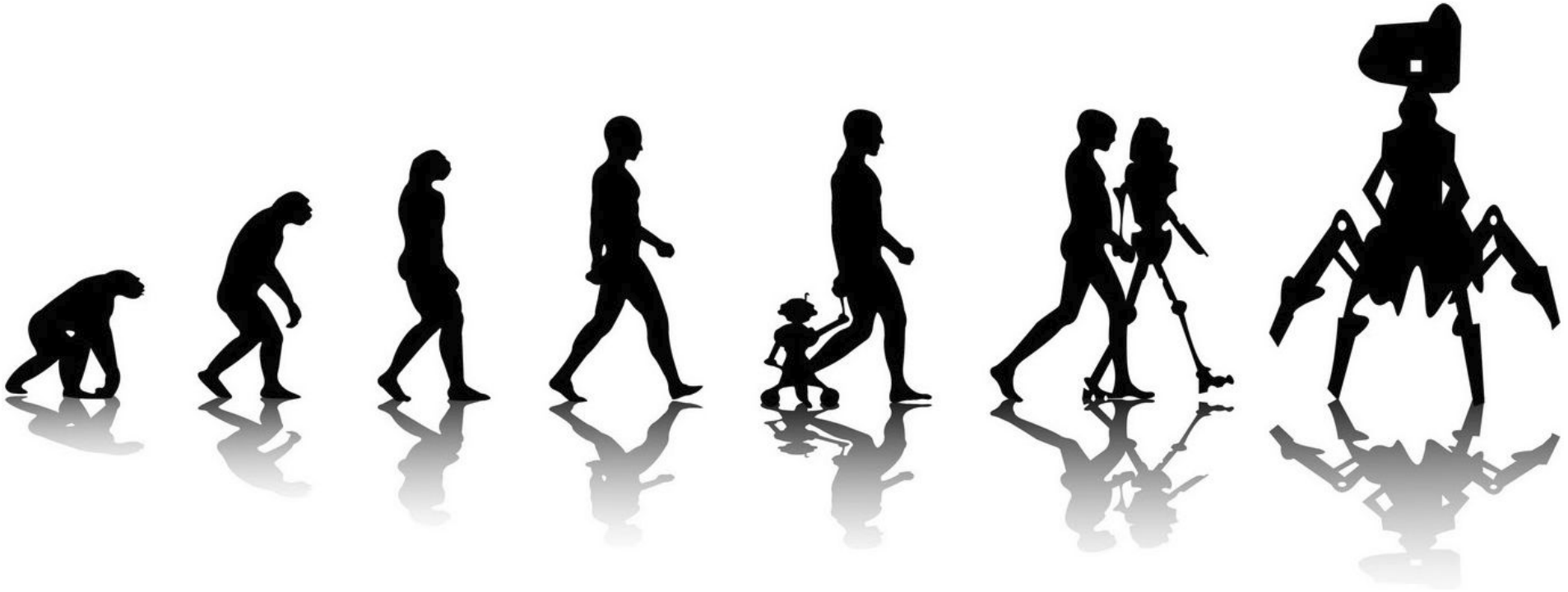


Modern Robotics: Evolutionary Robotics

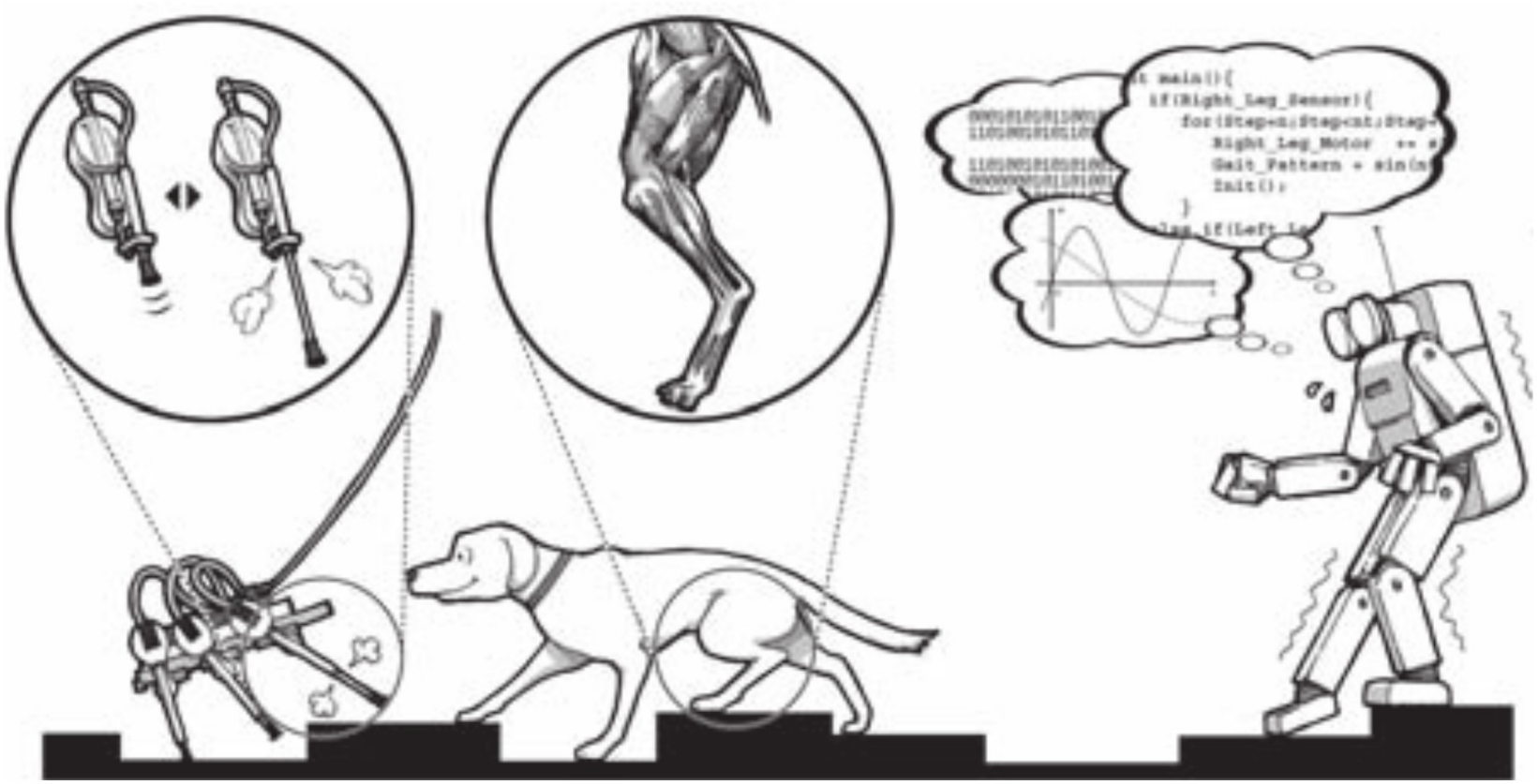
COSC 4560 / COSC 5560

Professor Cheney

2/28/18

Why Evolve Moprhologies?

[DARPA Robotics Challenge Finals 2015]



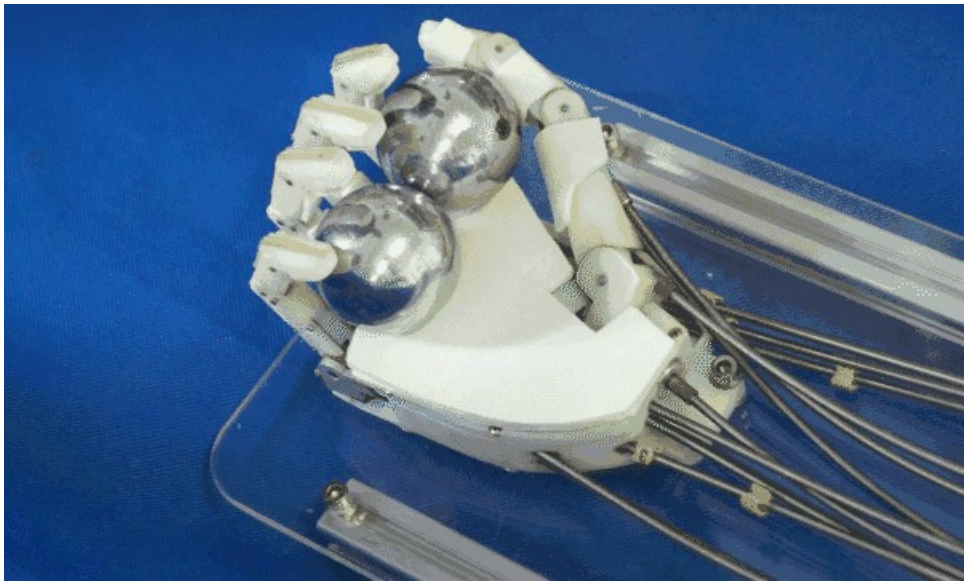
[Pfeifer and Bongard 2006]

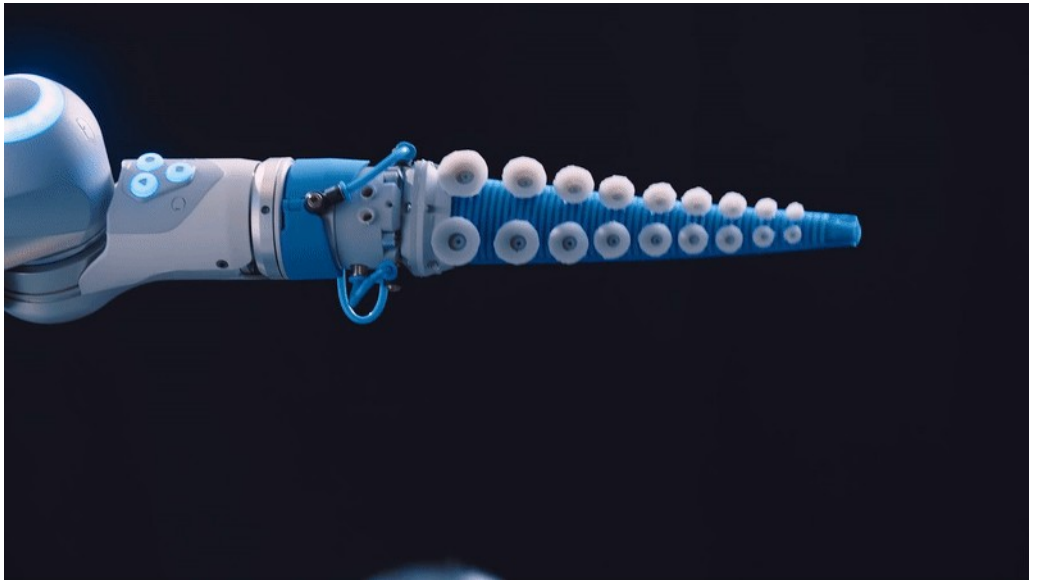
[Collins et al. 2001]

[Röthling et al. 2007]

[Amend et al. 2012]

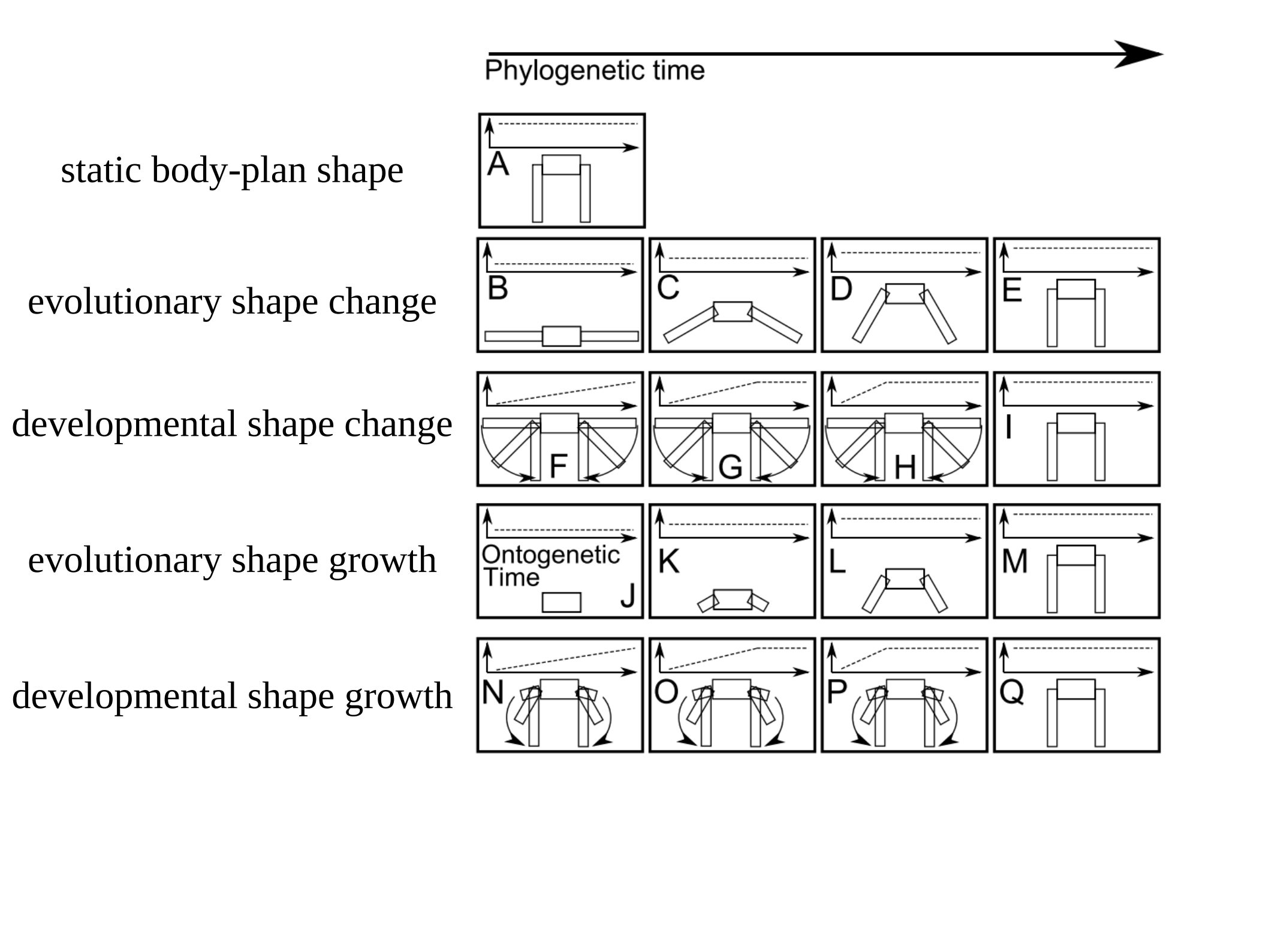
we can ask ask questions about the different types of behavioral strategies that arise, investigating how embodiment affects behaviors

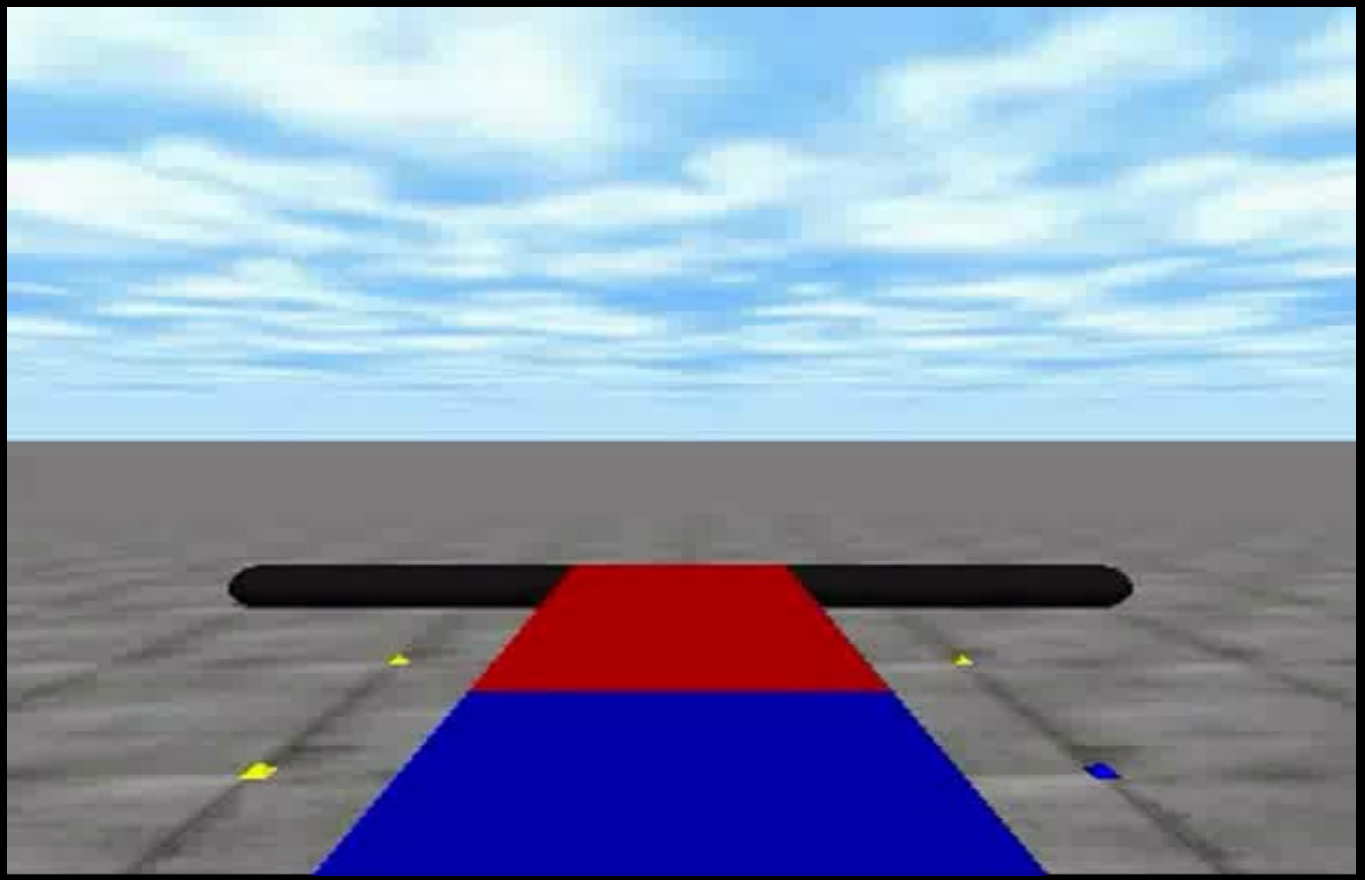


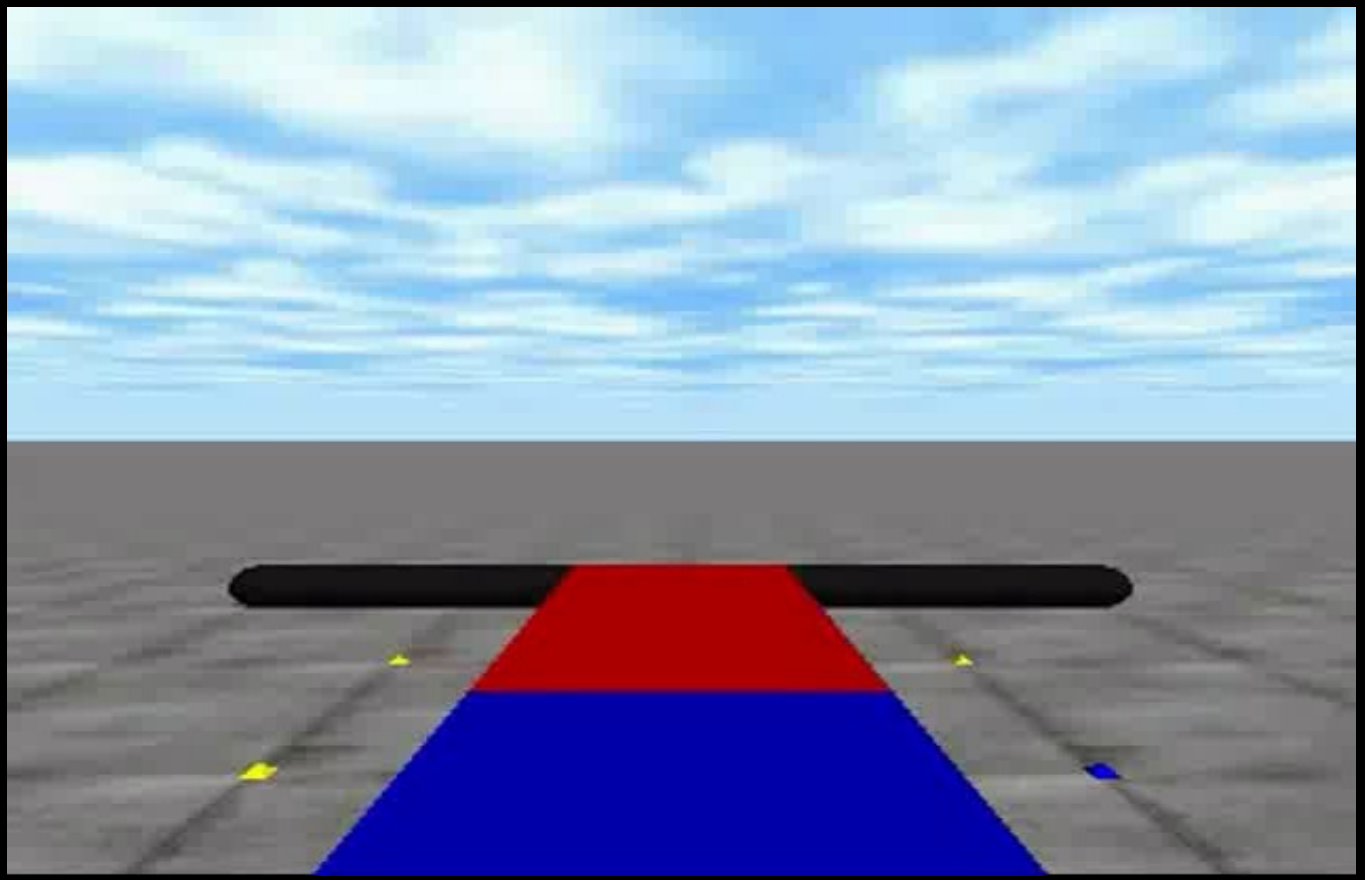


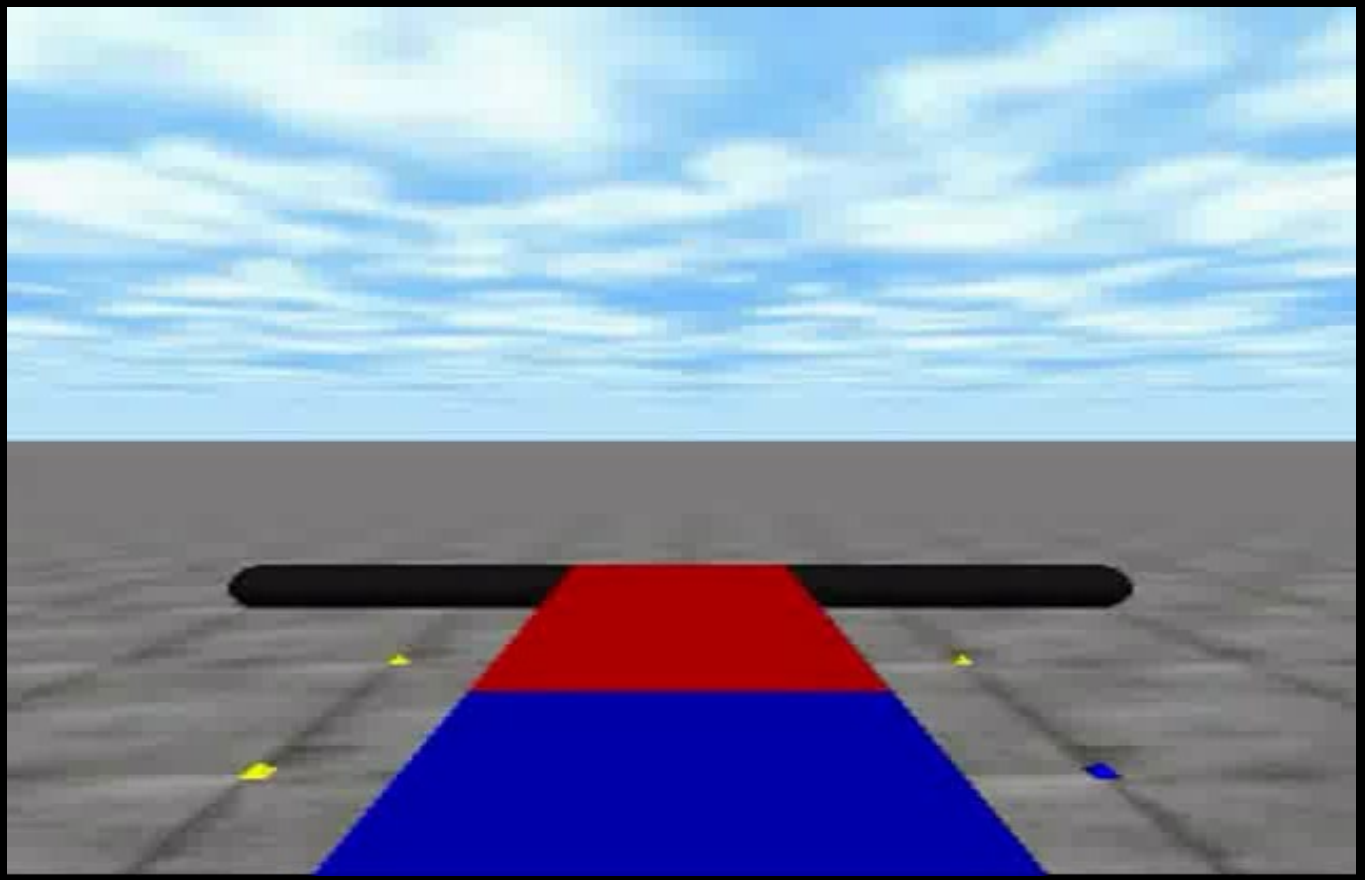
Morphological change in machines accelerates the evolution of robust behavior

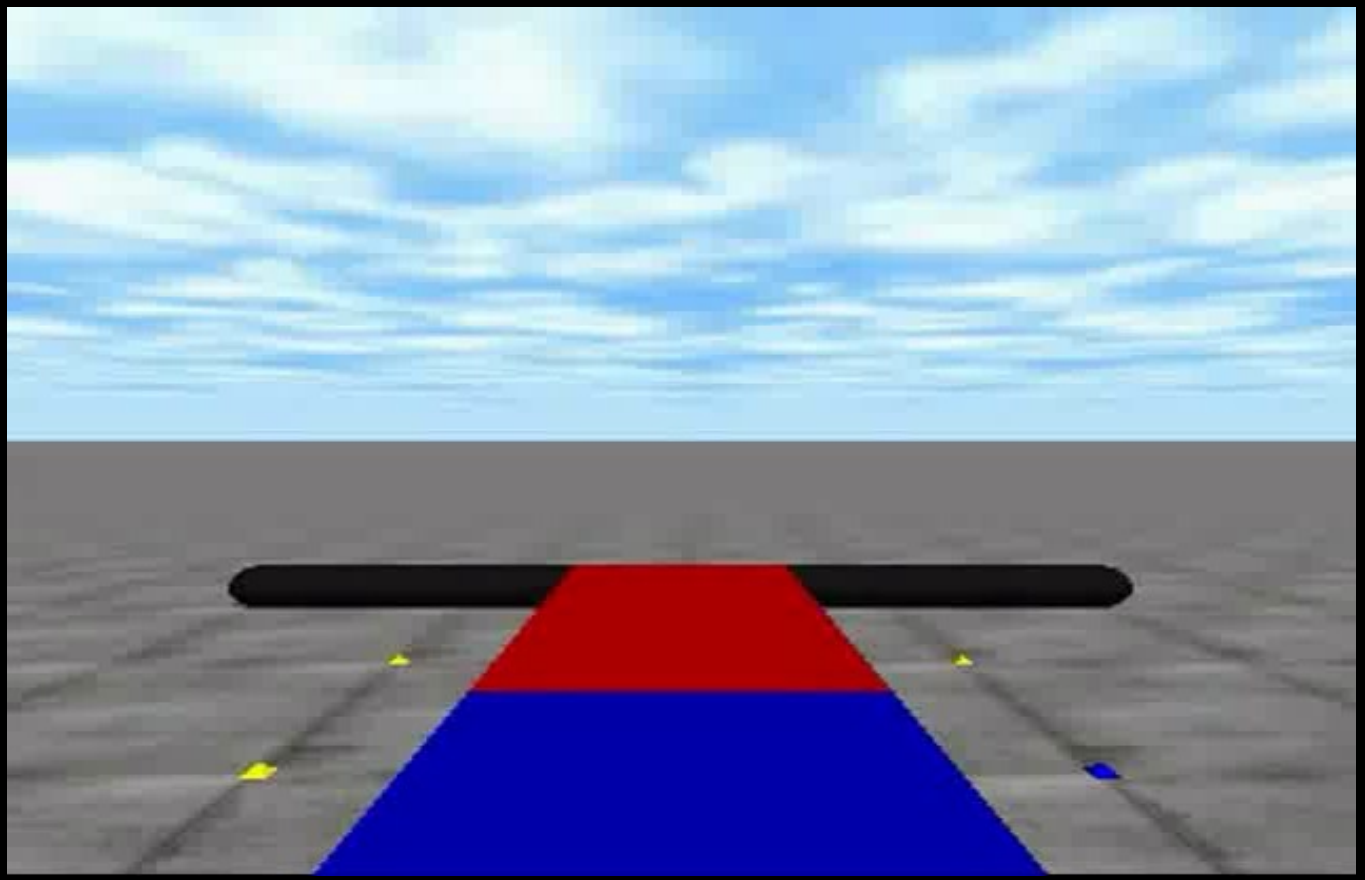
Josh Bongard[1]

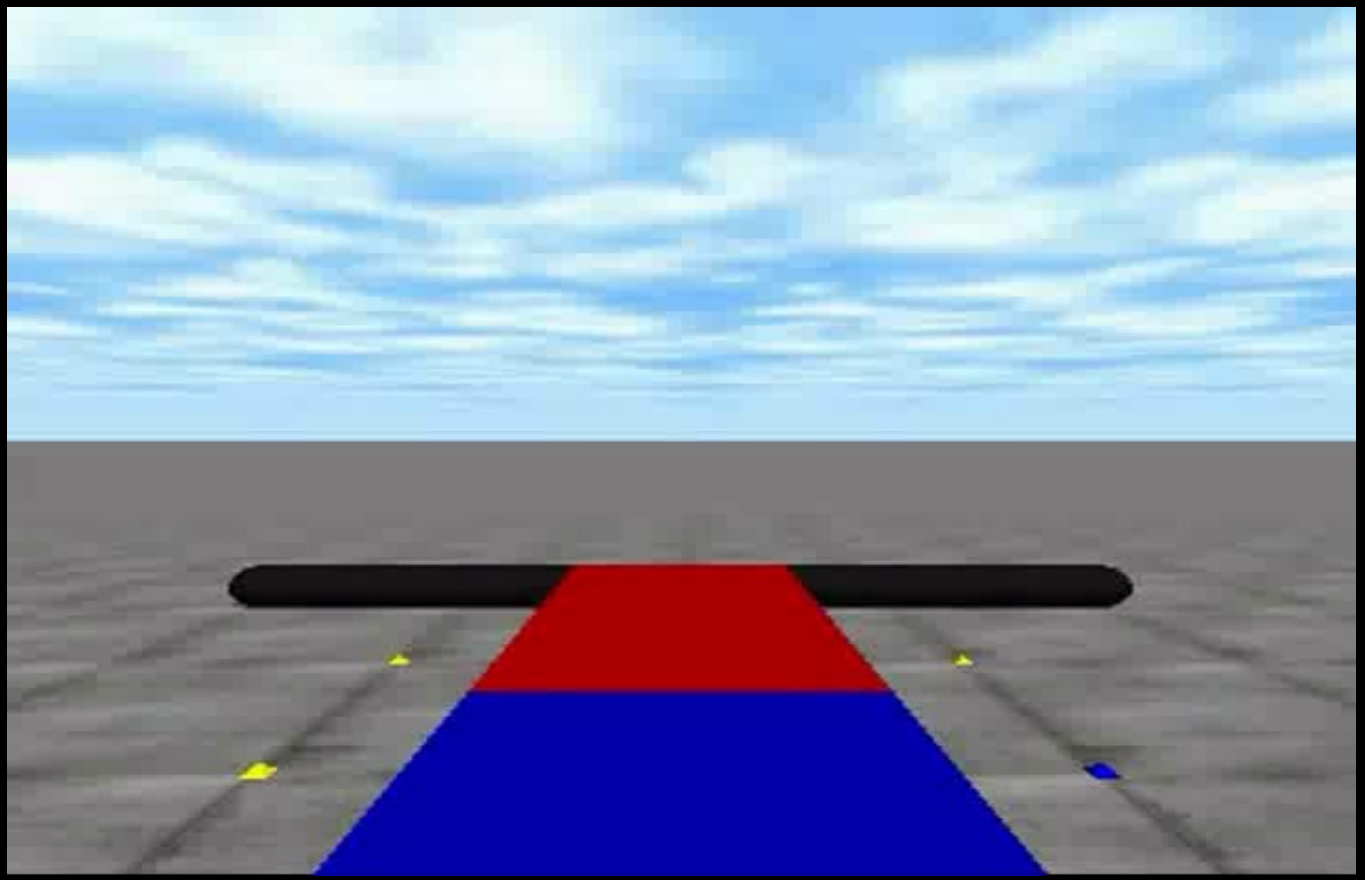

Phylogenetic time
static body-plan shape
evolutionary shape change
developmental shape change
evolutionary shape growth
developmental shape growth
A
B
C
D
E
F
G
H
I
Ontogenetic Time
J
K
L
M
N
O
P
Q

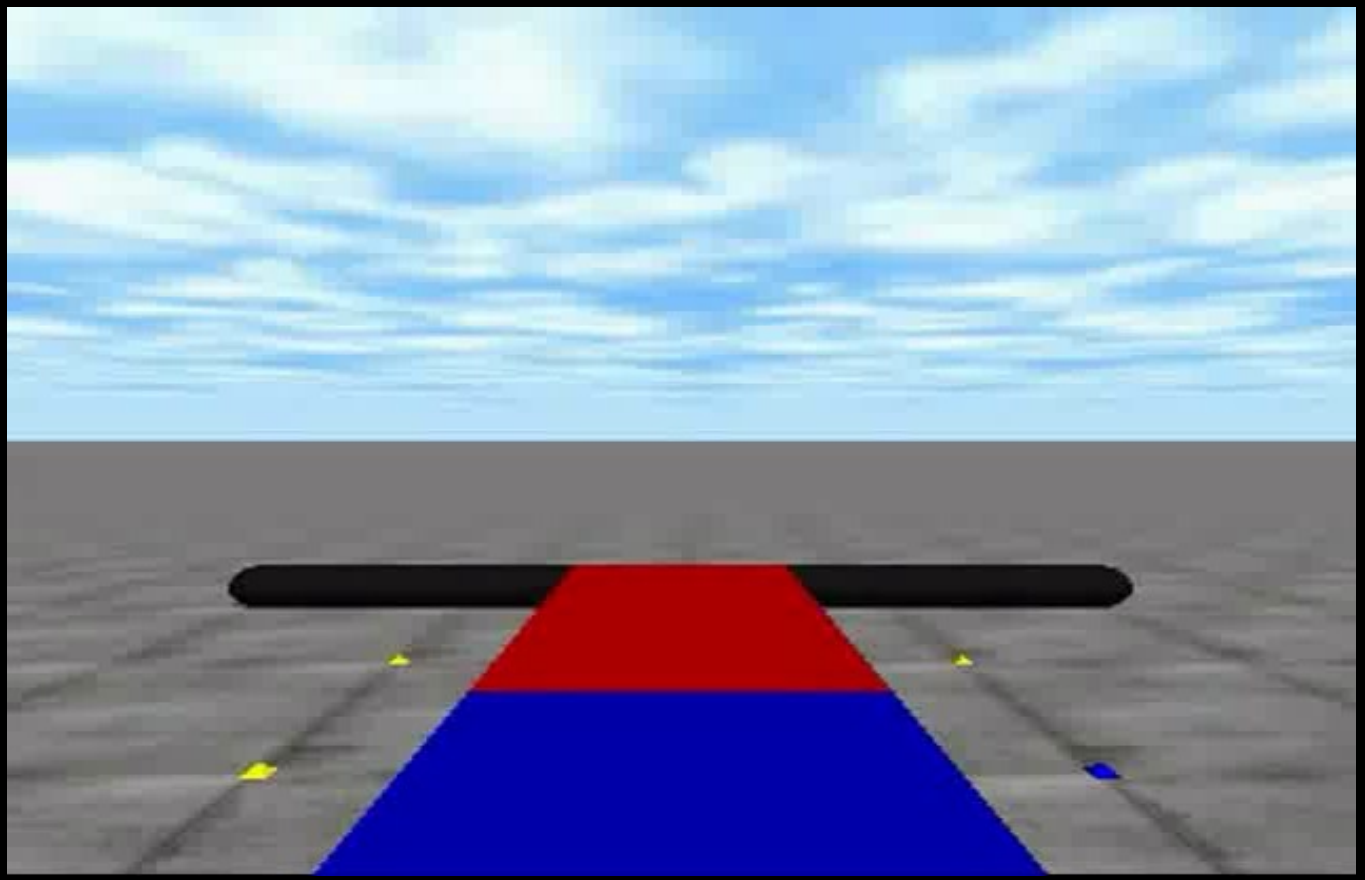


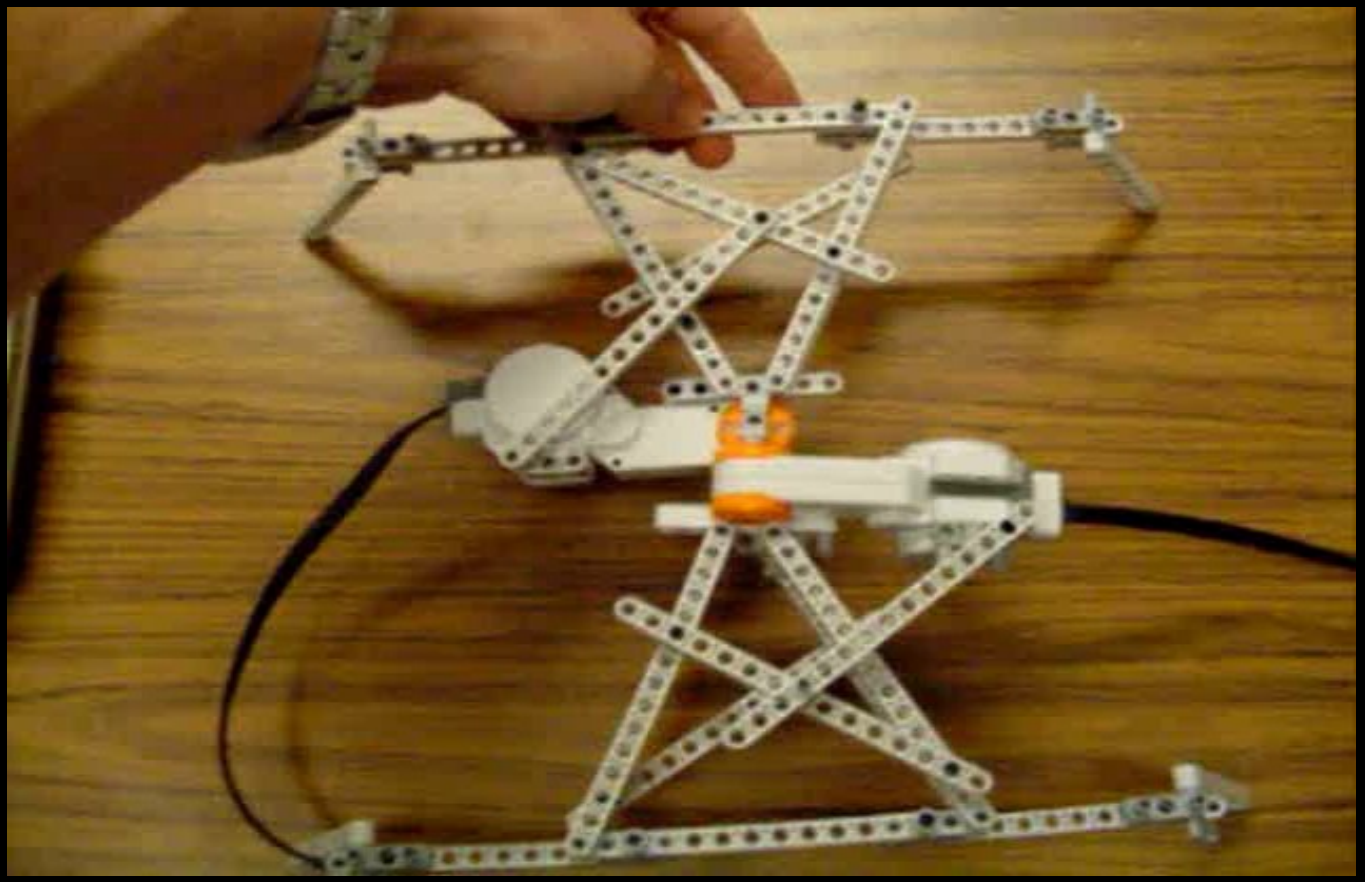












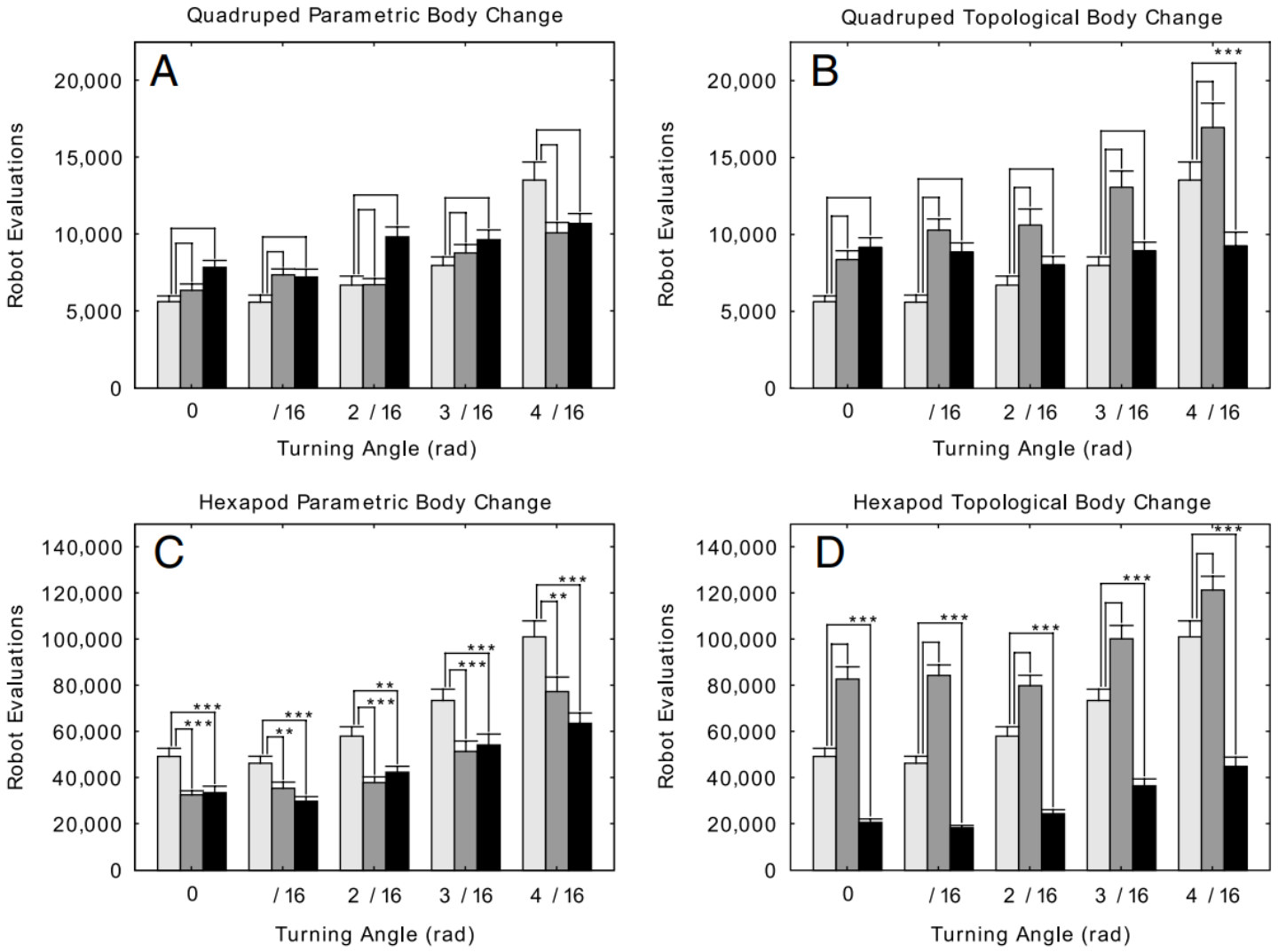


Fig. 3. How morphological change affects the time to discovery of the desired behavior in the quadruped (*A* and *B*) and hexapod (*C* and *D*) robot. Light gray bars indicate the number of controllers that had to be evaluated when no morphological change was allowed. Dark gray bars indicate the discovery time when the robots' body plans did not change during a robot's lifetime, but did change over evolutionary time. Black bars indicate the discovery time when body plans changed during each robot's lifetime, and also over evolutionary time. The dark gray and black bars in *A* and *C* report the impact of changing the robot's body plans parametrically; the dark gray and black bars in *B* and *D* report the impact of changing the robot's body plans topologically. Asterisks report statistically significant differences between no morphological change and topological, ontogenetic morphological change. Error bars report one unit of standard error of the mean.

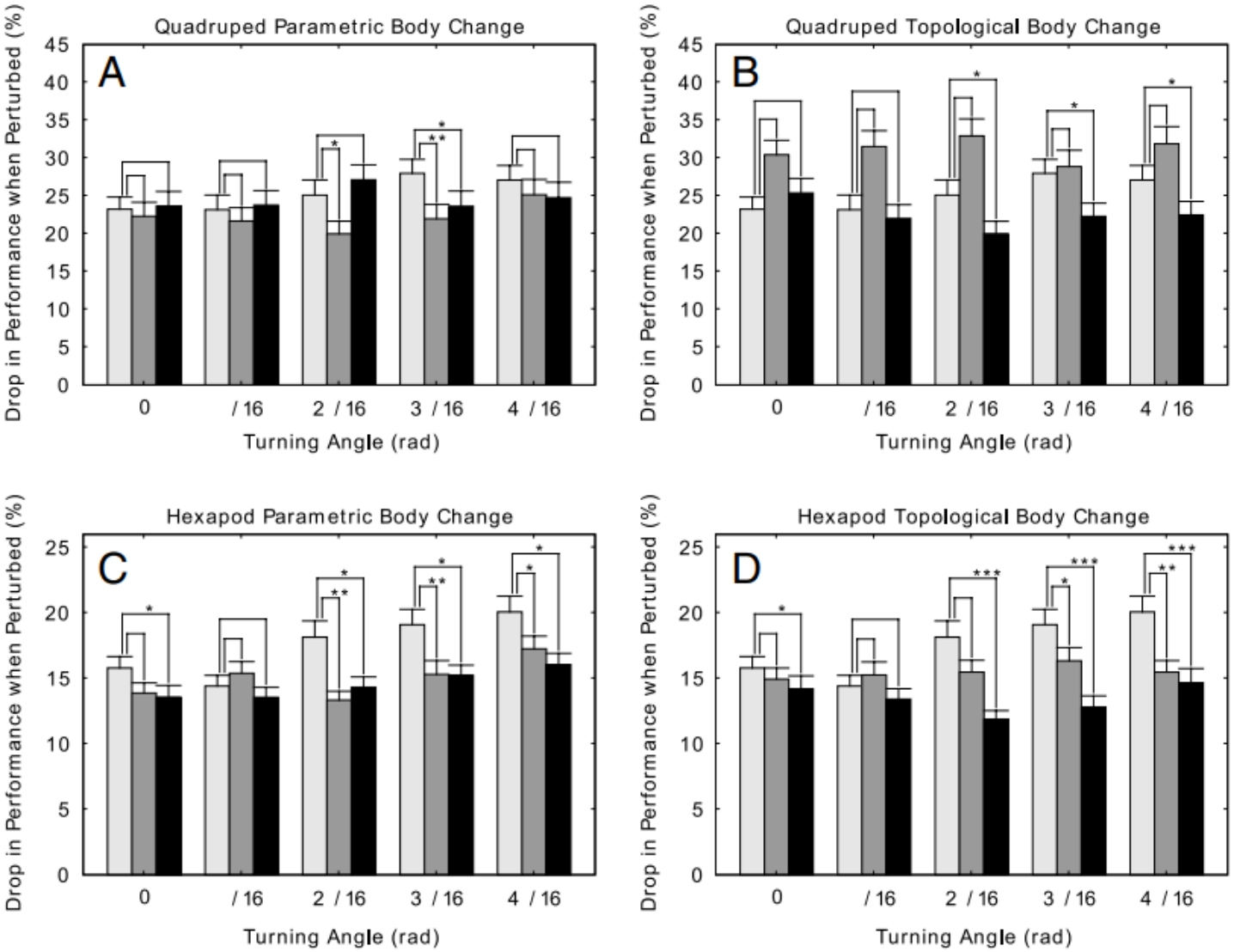


Fig. 4. How morphological change affects the robustness of the discovered behaviors. Bars report results from the same set of trials described in Fig. 3. The final robot capable of phototaxis from each independent experiment was reevaluated 100 times in the same simulated environment in which it evolved, but now exposed to small random external perturbations. The reduction in its ability to reach the light source was computed as the percent difference between the original distance it traveled and its new distance traveled during the perturbation.

this paper is great because it:

provides an (optimization) efficiency rationale
for a constraint problem in biology

uses a simple toy robotic model to zoom in on one
specific aspect of a very complex biological phenomenon

uses an established theory from psychology (shaping)
to explain the results from this toy model

uses repeated controlled experiments to show the conditions
when this phenomenon does and doesn't hold true

Towards a theoretical foundation for morphological computation with compliant bodies

Helmut Hauser · Auke J. Ijspeert · Rudolf M. Füchslin · Rolf Pfeifer · Wolfgang Maass

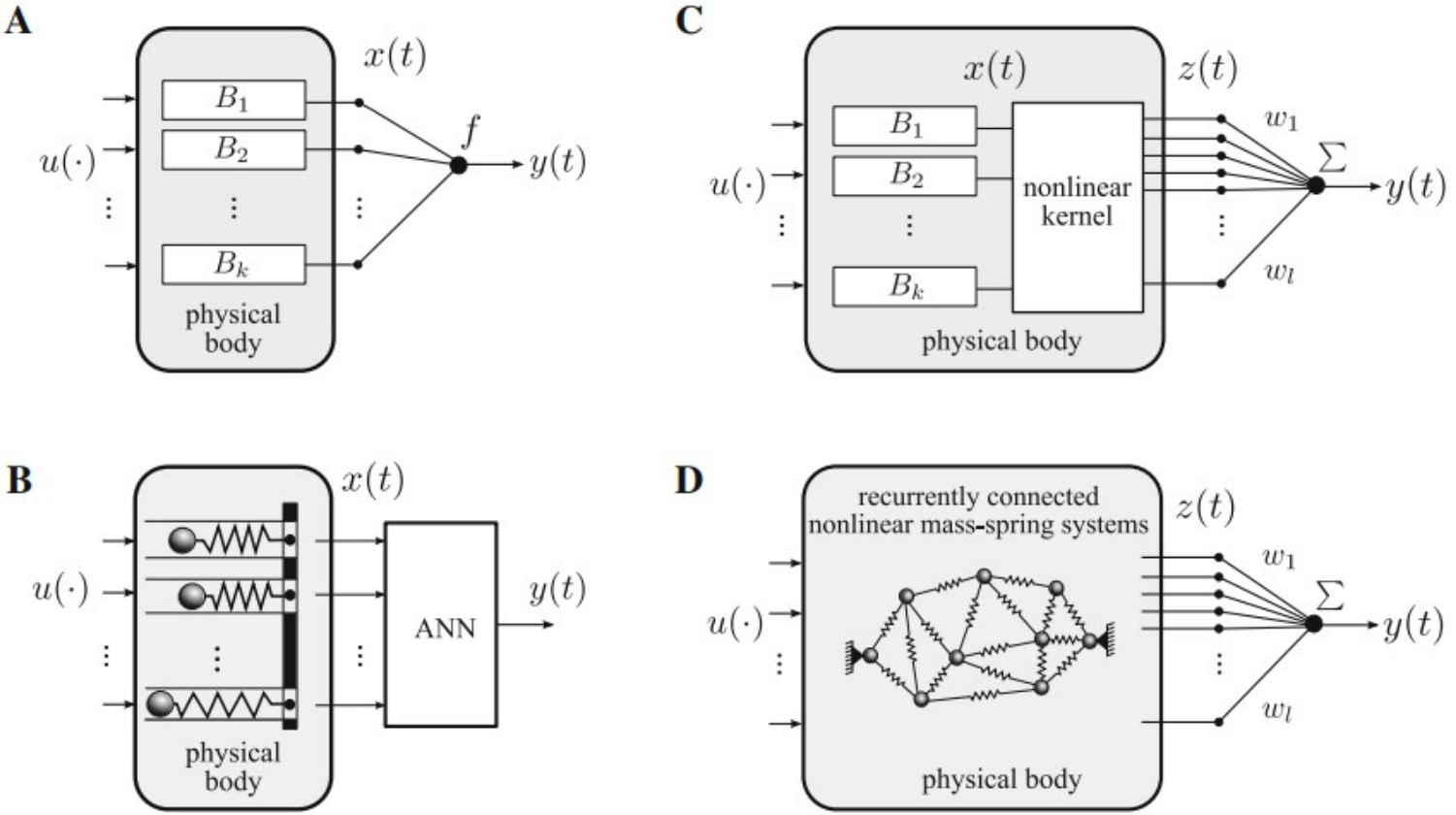
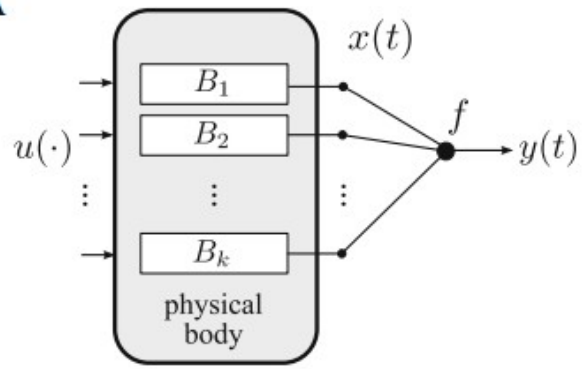


Fig. 1 From abstract theoretical models for morphological computation to real physical bodies (consisting of mass-spring systems). **a** The morphology (represented here by an array of randomly chosen, time-invariant, fading memory filters $B_1, \ldots, B_k$) contributes all temporal integration that is required to approximate a given filter $\mathcal{F}$. The readout f is here some memoryless, continuous function and provides the necessary nonlinear combination. Our theory provides evidence for a surprisingly large computational power of this simple architecture. **b** A possible implementation of (**a**) with a physical body. The filter array is built of an array of linear mass-spring systems and the readout is implemented by a feedforward artificial neural network (ANN). **c** In this architecture, the morphology contributes, in addition to the temporal integration via fading memory filters, generic nonlinear preprocessing in the form of some arbitrary kernel (i.e., nonlinear projection of $x(t)$ into a higher dimensional space). In this case, only a *linear* readout (instead of e.g., an feedforward ANN) has to be added externally. **d** A possible physical realization of (**c**). The array of filters and the kernel are both implemented by a randomly connected network of nonlinear springs and masses. In the resulting computational device, the output weights $[w_{\text{out},1}, \ldots, w_{\text{out},l}]$ are the only parameters, which are adapted in order to approximate a given complex filter $\mathcal{F}$

A



Theorem *Any time-invariant filter $\mathcal{F}$ with fading memory that maps some n-dimensional input stream $u \in U$ onto an output stream y can be approximated with any desired degree of precision by the simple computational model shown in Fig. 1a,*

1. *if there is a rich enough pool* **B** *of basis filters (time-invariant, with fading memory), from which the basis filters $B_1, \ldots, B_k$ in the filter bank can be chosen (***B** *needs to have the pointwise separation property) and*
2. *if there is a rich enough pool* **R** *from which the readout functions f can be chosen (***R** *needs to have the universal approximation property, i.e., any continuous function on a compact domain can be uniformly approximated by functions from* **R***).*

Morphological computation with feedforward mass-spring systems

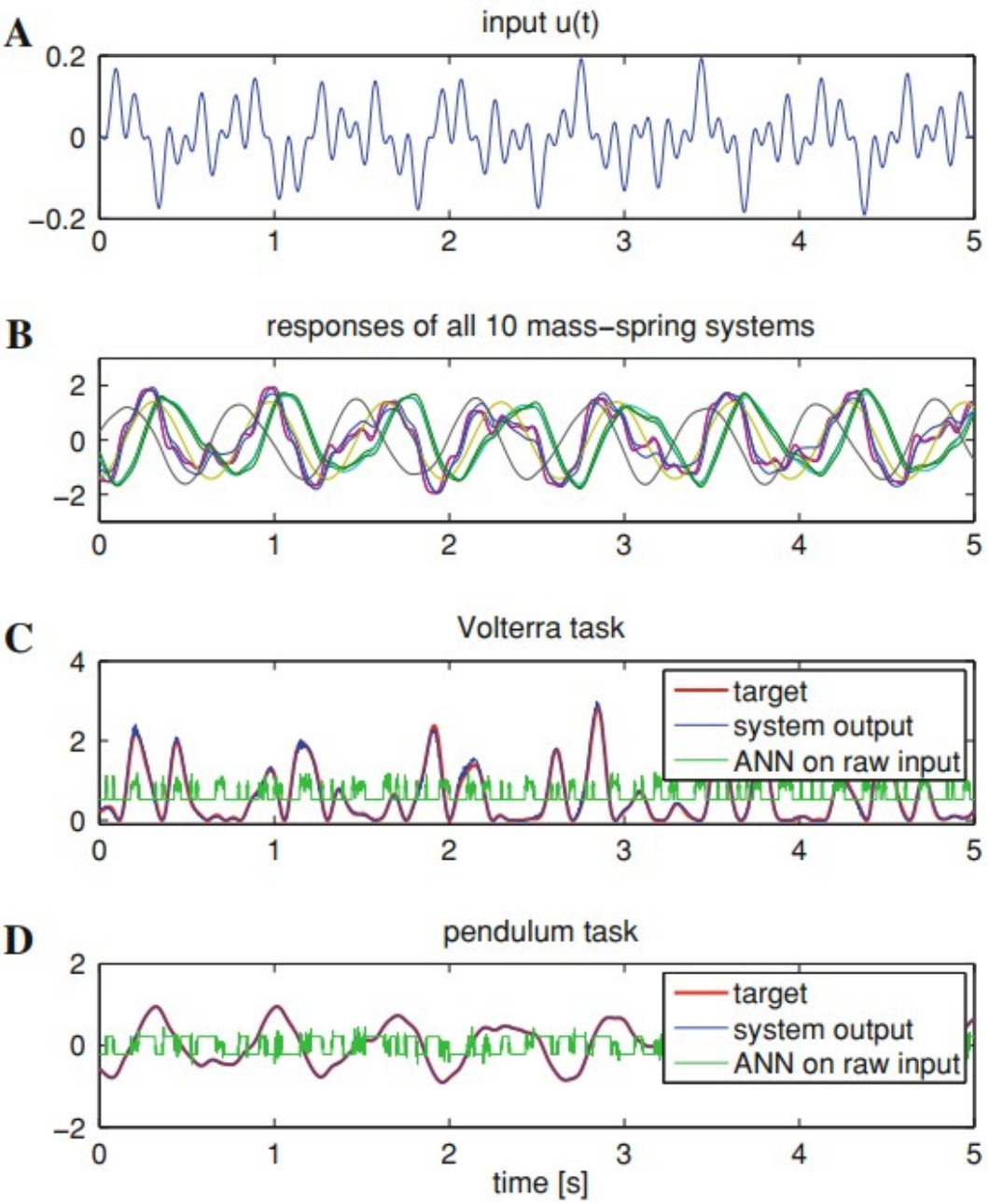


Fig. 3 Applying a feedforward morphological computation device to approximate the Volterra series operator $\mathcal{V}$ (defined by Eq. 2) and the pendulum (Eq. 3) simultaneously with one morphological structure (i.e., multitasking). **a** The used input signal $u(t)$, which consisted of a product of three different sinusoidal functions ($f_1 = 2.11$, $f_2 = 3.73$, and $f_3 = 4.33$ Hz). **b** The responses of all ten mass-spring systems to this input (for a better readability the outputs were normalized to zero mean and a standard deviation of one). **c** The performance of the proposed morphological computation device for the Volterra task. The *red line* is the target (applying the Volterra series operator to the input, i.e., $\mathcal{V}u(t)$) and the *blue line* shows the output of the morphological computation device. The *green line* shows the performance of the device, when no morphological structure was available, i.e., only the nonlinear readout of the ANN was applied to the raw input data. Clearly this approach fails, since the ANN is only a static readout and is not able to represent the necessary temporal integration, which was contributed in the previous case by the morphological structure. **d** The pendulum task: the *red line* is the target, the *blue line* the output of the morphological computation device and the *green line*, when no morphological structure was available

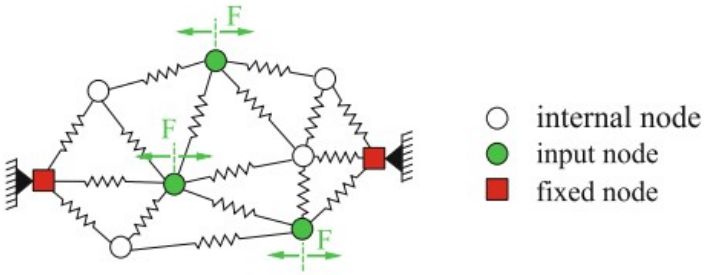


Fig. 4 Schematic example of a generic mass-spring network. The nodes (masses) are connected by nonlinear springs. The *red* nodes are fixed in order to hold the network in place. The *green* nodes are randomly chosen inputs nodes, which receive the input in form of horizontal forces scaled by randomly initiated weights

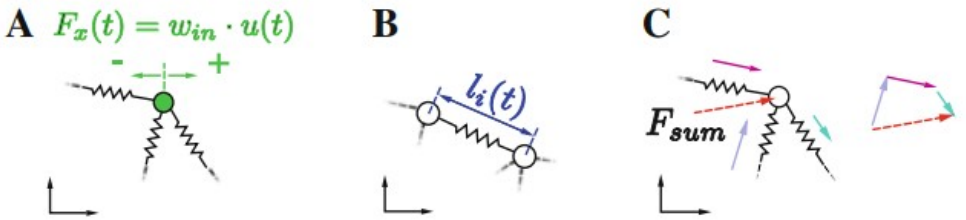


Fig. 5 Implementation of input, linear readout, and simulation of forces of the mass-spring networks. **a** The input is applied to an input node as a horizontal force F_x proportional to the input signal u (scaled by a randomly initialized weight w_{in} for this input node). **b** The readout from the network is the weighted sum of all L spring lengths $y(t) = \sum_{i=1}^{L} w_{\text{out},i} l_i(t)$. In general, the input as well as the output can be multidimensional. **c** All the spring forces act along their spring axis. The resulting force F_{sum} is the sum of all forces acting on the node and is found by the summation of the force vectors

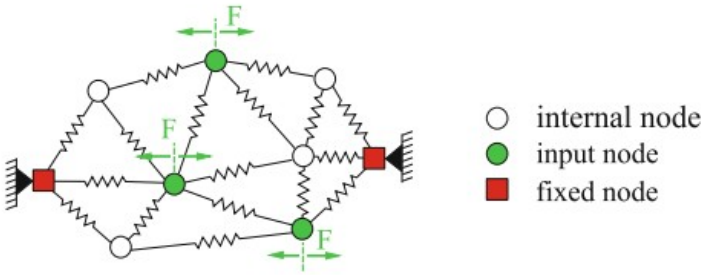


Fig. 4 Schematic example of a generic mass-spring network. The nodes (masses) are connected by nonlinear springs. The *red* nodes are fixed in order to hold the network in place. The *green* nodes are randomly chosen inputs nodes, which receive the input in form of horizontal forces scaled by randomly initiated weights

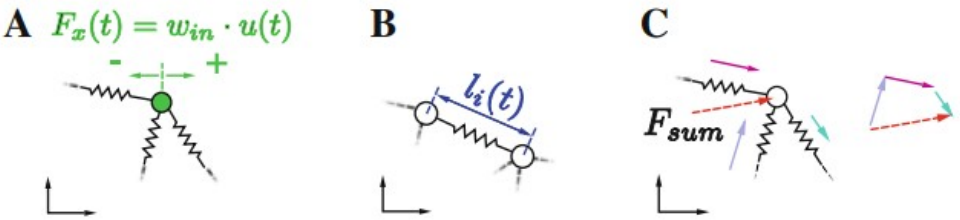


Fig. 5 Implementation of input, linear readout, and simulation of forces of the mass-spring networks. **a** The input is applied to an input node as a horizontal force F_x proportional to the input signal u (scaled by a randomly initialized weight w_{in} for this input node). **b** The readout from the network is the weighted sum of all L spring lengths $y(t) = \sum_{i=1}^{L} w_{\text{out},i} l_i(t)$. In general, the input as well as the output can be multidimensional. **c** All the spring forces act along their spring axis. The resulting force F_{sum} is the sum of all forces acting on the node and is found by the summation of the force vectors

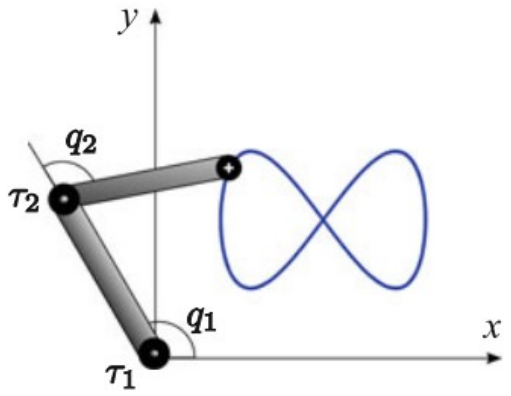


Fig. 6 Setup for the robot arm task. The *blue line* is the desired trajectory for the end-effector

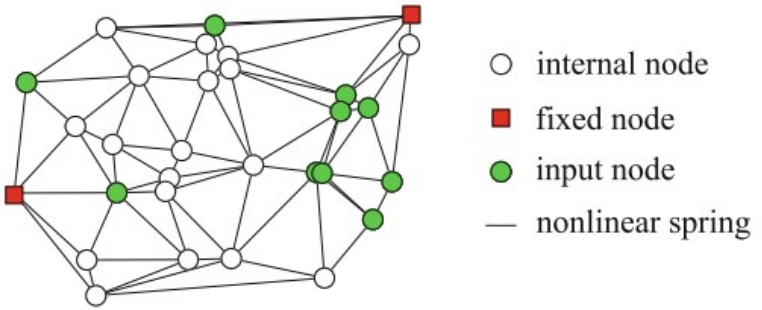


Fig. 7 Generic mass-spring network used for the robot arm task and subsequently for the multitasking task in Sect. 4.3. The *red* nodes are globally fixed and the *green* nodes are the randomly chosen input nodes. The network consisted of 30 masses and 78 nonlinear springs

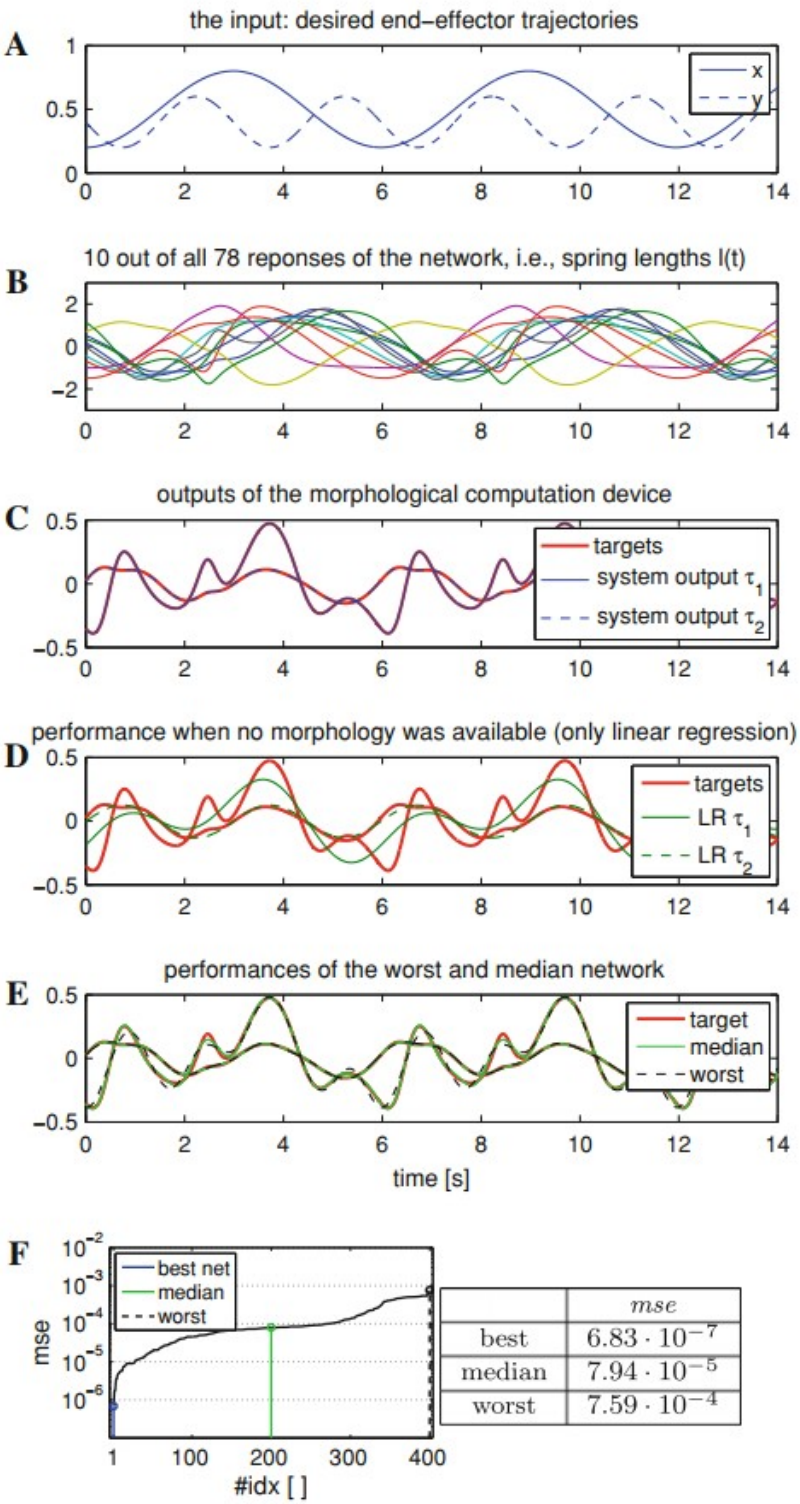
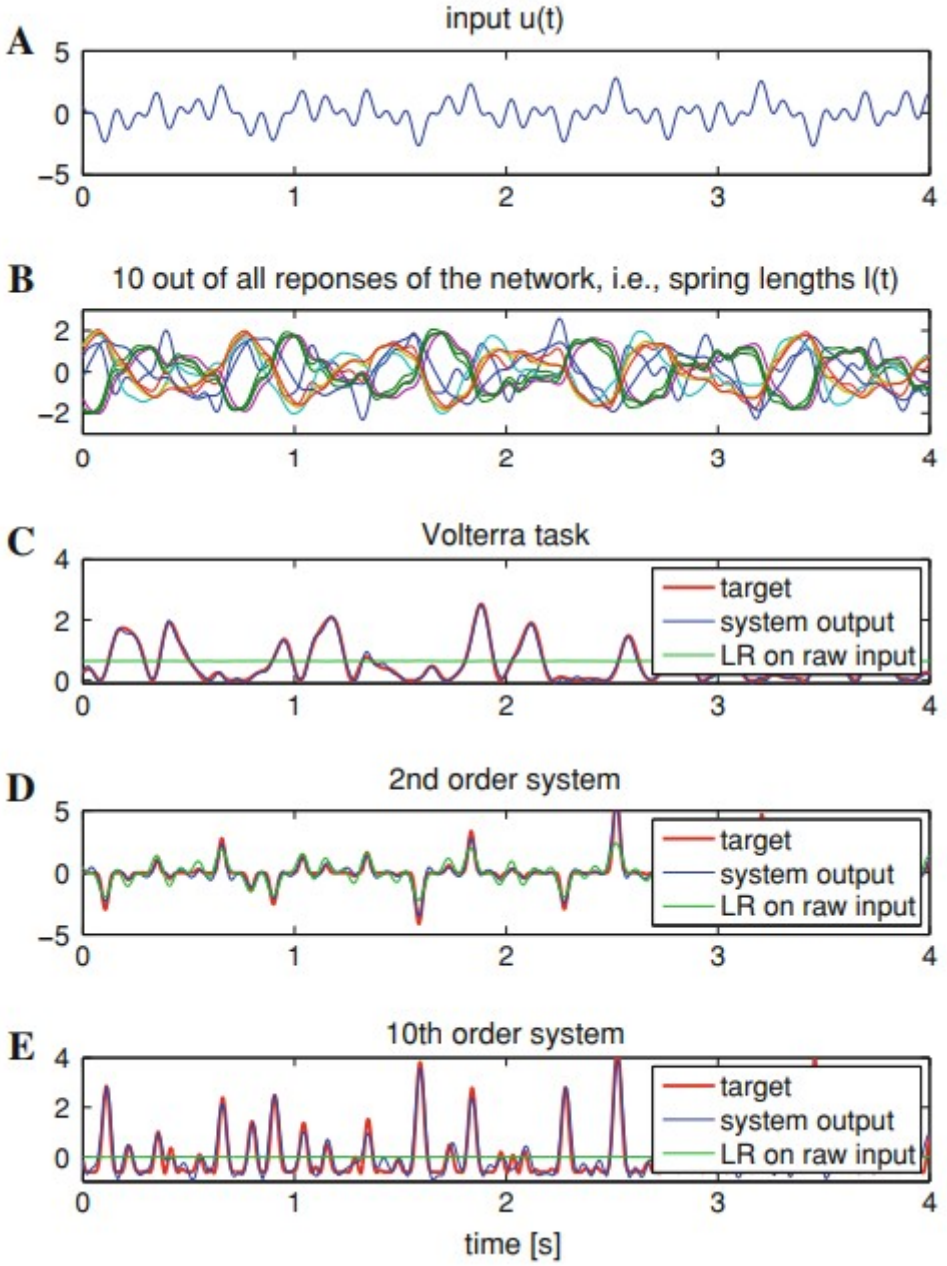


Fig. 8 Representation of the inverse dynamics of a robot arm with the help of morphological computation. **a** The desired end-effector trajectory split up in its two Cartesian coordinates x and y (i.e., inputs). **b** Ten typical responses (out of all 78) of the mass-spring network to this input. For a better readability, each signal was normalized to zero mean and a standard deviation of one. **c** The performance of the morphological computation device. The *red lines* are the target torque trajectories and the *blue lines* are the outputs of the computational device. **d** The performance when no morphological structure was available, i.e., only a LR on the actual values of the inputs remained. This approach failed to represent the dynamic and nonlinear mapping. **e**, **f** Based on the same construction parameters, we randomly generated 400 networks and sorted them by their mean squared error (*mse*) over its two outputs. The table shows the performances of the best, the worst, and the median network. The best network was used for the plot of (**c**). The performances of the worst (*black dotted line*) and the median network (*green*) are presented in (**f**)

$$y(t) = \mathcal{V}u(t)$$
$$= \int\int_{\tau_1,\tau_2 \in \mathbb{R}_0^+} h_2(\tau_1, \tau_2) u(t-\tau_1) u(t-\tau_2) d\tau_1 d\tau_2, \quad (2)$$

$$y[k+1] = 0.4y[k] + 0.4y[k]y[k-1] + 0.6u^3[k] + 0.1$$

$$y[k+1] = 0.3y[k] + 0.05y[k]\left(\sum_{i=0}^{9} y[k-i]\right) + 1.5u[k-9]u[k] + 0.1 \, .$$



Another interesting aspect of the approach is that real physical bodies provide the necessary nonlinearities and the temporal integration for *free*. The physical structure simple reacts on its *inputs*. Actually, it is not even necessary to have real physical interpretations of all the available internal signals in order to exploit them for morphological computation. Furthermore, the bodies of real biological systems are not simply computational devices, but they fulfill real functions. For example, they provide animals (and robots) with the capability to locomote and to interact with their environments. Therefore, a next step will be to apply the proposed theory to morphological structures of real robots. This would also involve the step to move from our presented abstract networks, which were be chosen to demonstrate the applicability of our presented theory, to more realistic simulations including the simulation of the interaction between a robot and its environment. In this context, one would have to investigated the impact of real-world conditions on the performance of the proposed setups. For example, typical cases for such real-word conditions are the partial loss of the state of the morphology and/or noisy readouts.