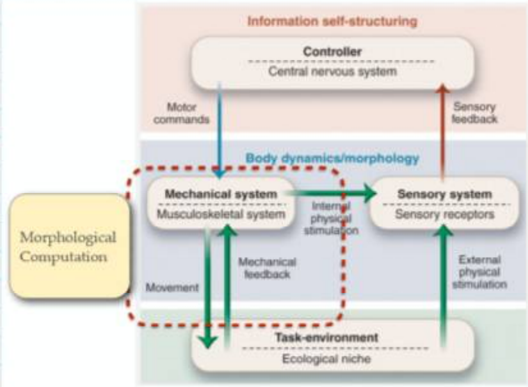
Morphological Computation

23 May 2017 15:16

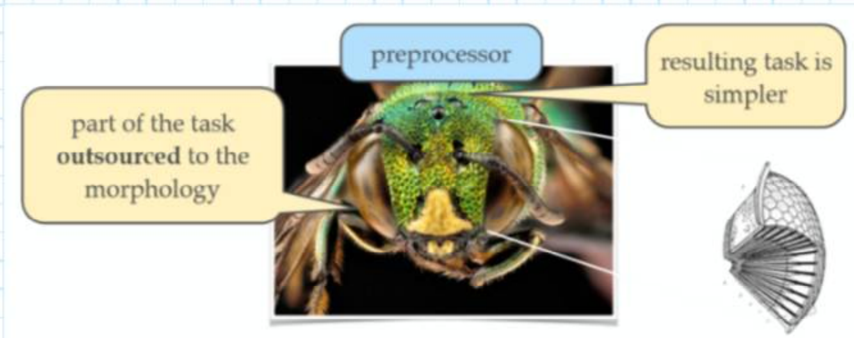
Embodiment

This school of thought claims that intelligent embodiment is an important aspect of AIs that are put in physical environments.



Morphological computation (MC) is a term, which captures conceptually the observation that biological systems take advantage of their morphology to conduct computations needed for sucessful interaction with their environments.

Prototypical Example: Insect Eyes



"Design choices":

- Compound eye
- Non-homogeneous arrangement (more dense towards front of the eye) in order to get better motion estimates.

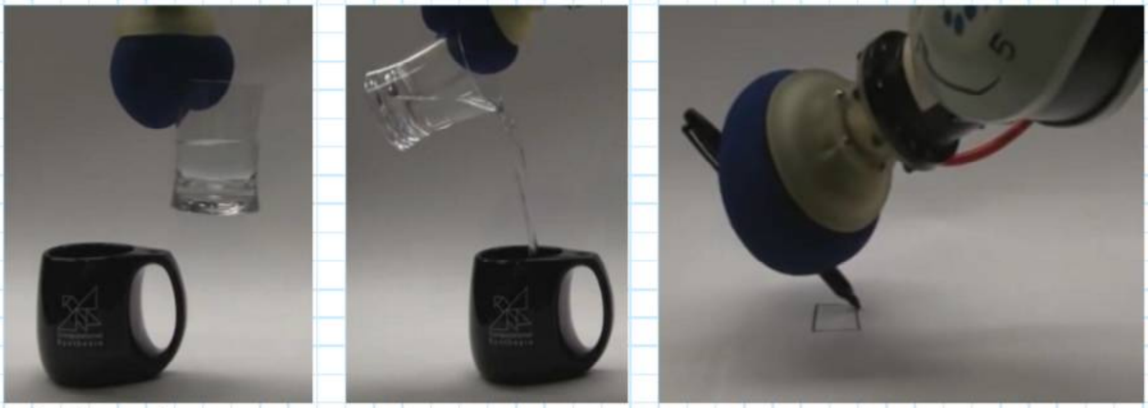
Trout Morphology:

Source: Lauder Lab, Havard

Despite this trout being dead, it morphology reacts to a flow of water with swimming motions. This makes the animal remarkably energy efficient.

Robotic Examples

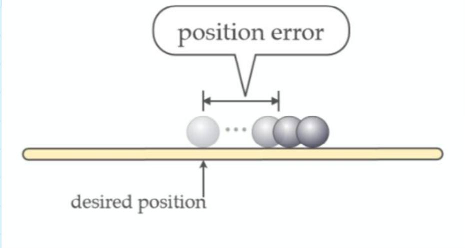
- Slinkies have an ability to robustly locomote in a variety of environments with no centralised control or intelligence.
- Universal ballon grippers



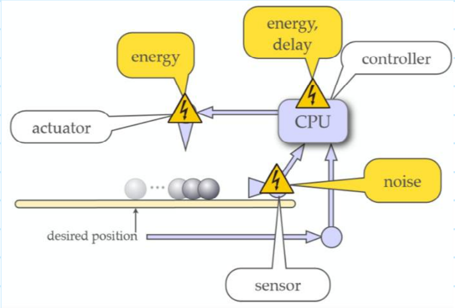
- Adaptive
- Fine control

Simple Thought Experiment

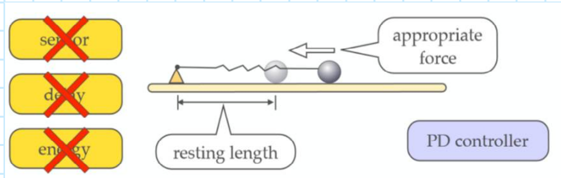
Suppose a ball is on a frictionless surface, and the goal is to keep the ball in a desired position in the face of perturbations,



Classical robotics approach:



Morphological robotics approach:

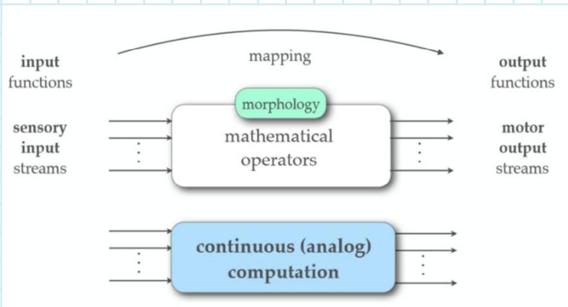


We will use this spring analogy for mechanical computation when designing / analysing soft robotics.

Theoretical Model of MC:

The trade-off between MC and centralised controllers comes down to having the morphology handle robust interaction with the environment, and providing a reduced dimensionality set of parameters that the controller can manipulate to cause different emergent behaviours.

Our model of MC will be constructed,



Thereby we can implement
- Stable nonlinear ODE with singular equilibria
- Limit cycles
- Bifurcation

- Limit cycles
- Bifurcation
- Analog finite state machines
- Volterra series:

Volterra series

From Wikipedia, the free encyclopedia

The **Volterra series** is a model for non-linear behavior similar to the Taylor series. It differs from the Taylor series in its ability to capture 'memory' effects. The Taylor series can be used for approximating the response of a nonlinear system to a given input if the output of this system depends strictly on the input at that particular time. In the Volterra series the output of the nonlinear system depends on the input to the system at *all* other times. This provides the ability to capture the 'memory' effect of devices like capacitors and inductors.

Computation in Mechanical Systems

We model soft bodies as attachment points latticed together with springs obeying Hooke's law.

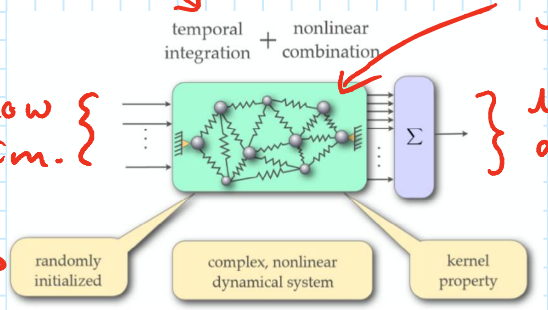
Internal state of the lattice

high dim internally

Any opt. technique can be used

low dim.

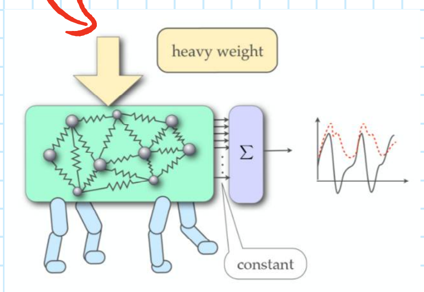
low dim.

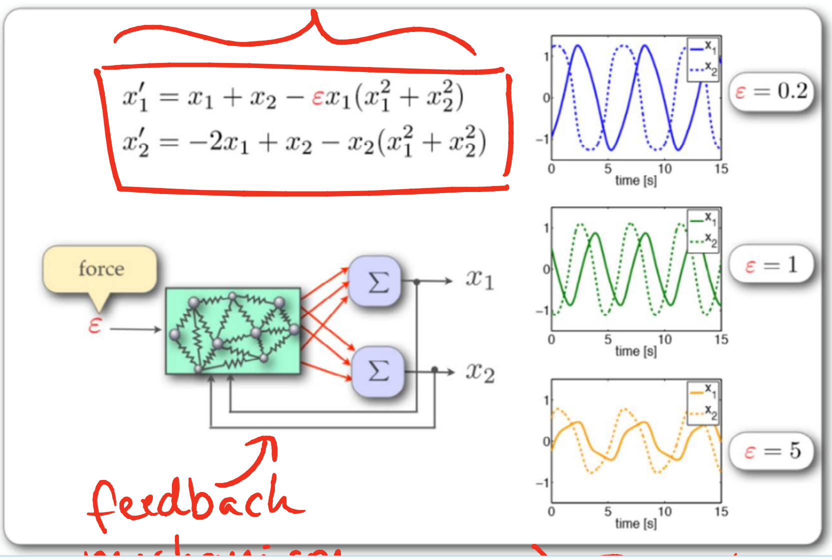


Example: Load bearing system:

This could also be controller input

dynamical eqns





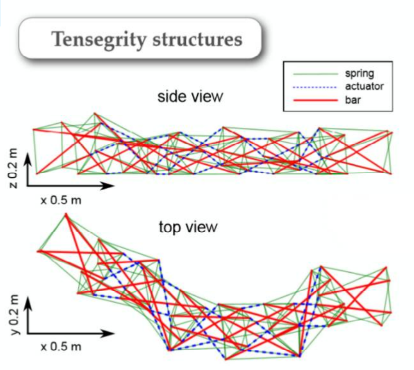
feedback

feedback mechanism

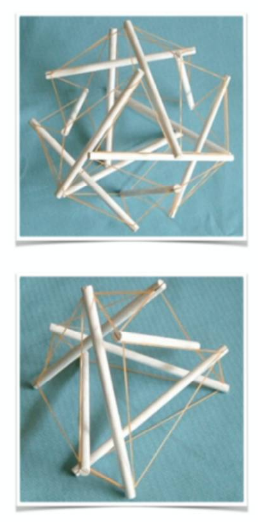
different behaviours

Physical Example: Tensegrity Robots

Rigid rods combined with elastic connections.



Actuator params can be learned to cause the structure to locomote / sense the environment.



Conclusive Comparison

classical		*embodied*
centralized control	→	interplay of brain, body, and environment
top-down control	→	guided self-organization
algorithm	→	dynamical system
abstract symbol processing	→	sensory-motor coordination
top-down design	→	design for emergence
fixed morphology	→	morpho-functional machines