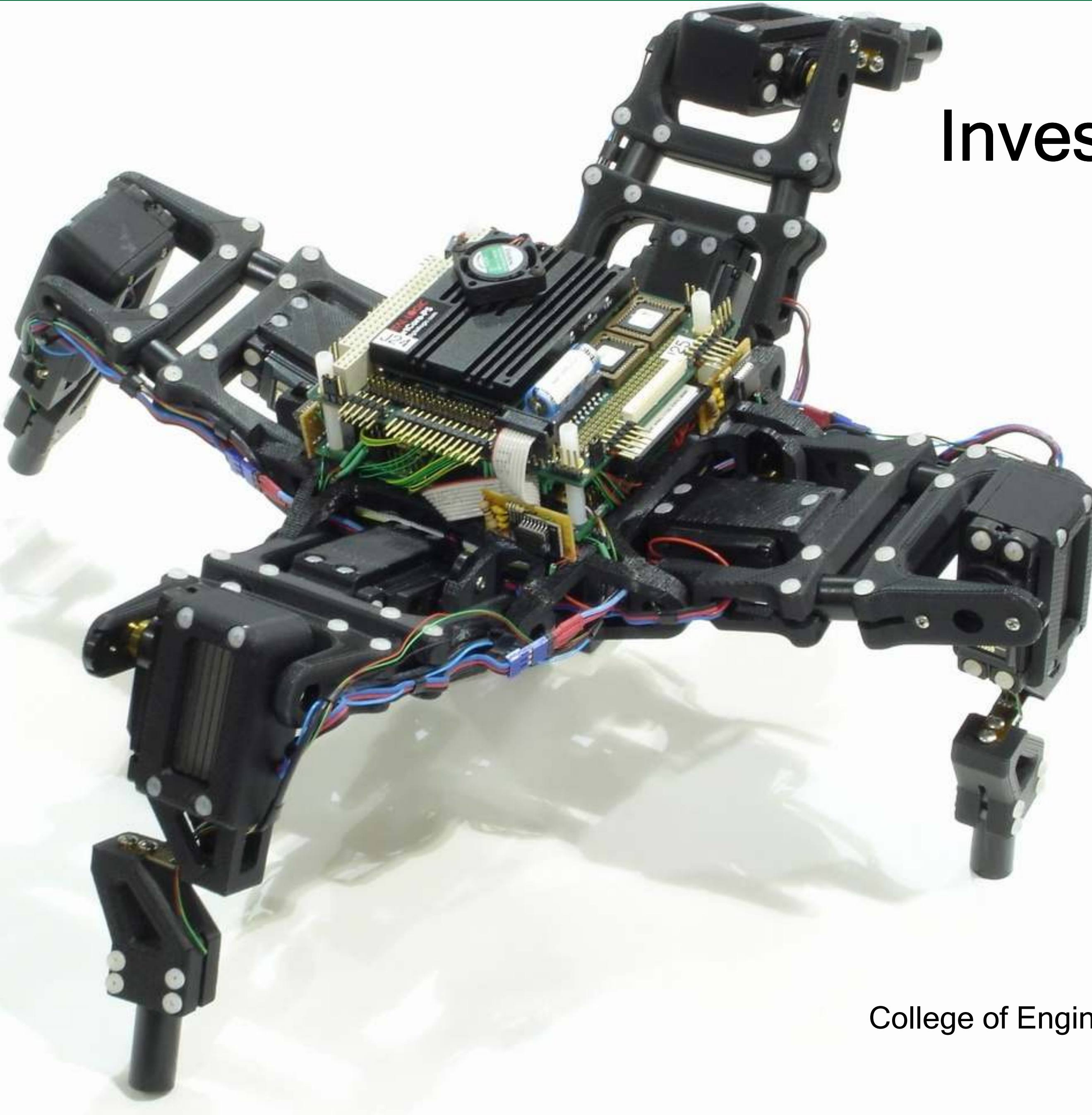
July 8, 2008

Investigations at the interface of morphology, evolution and cognition



Josh Bongard
Department of Computer Science
College of Engineering and Mathematical Sciences
University of Vermont
josh.bongard@uvm.edu

The Tool Set

MorphEngine
MorphEngine `Unplugged’
Physical Simulation

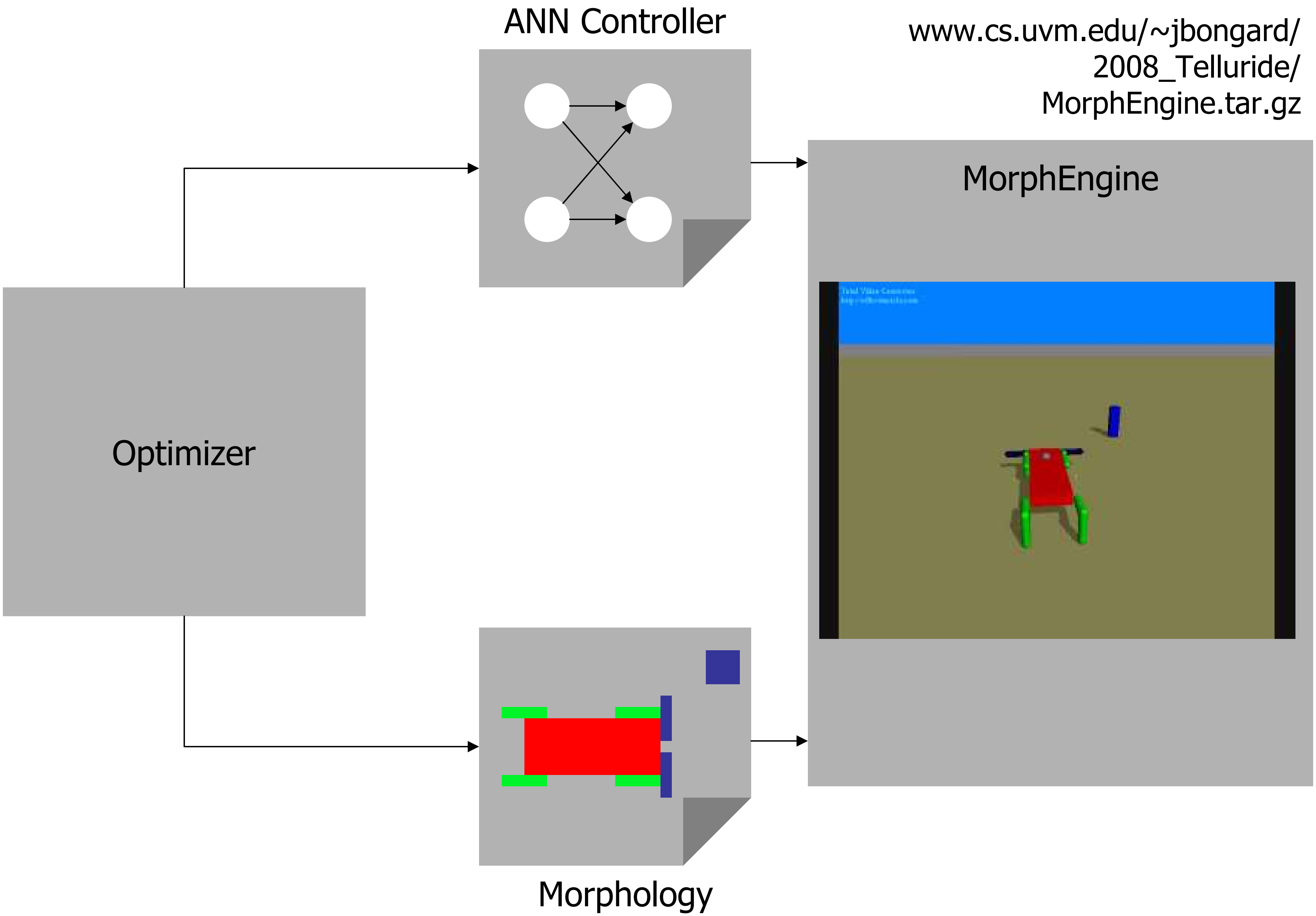
Example Investigations

Evolving Robot Morphologies and Controllers Together
Evolving Self-Models
Evolving Coupled, Nonlinear Models
Evolving Robots Capable of Multiple Behaviors

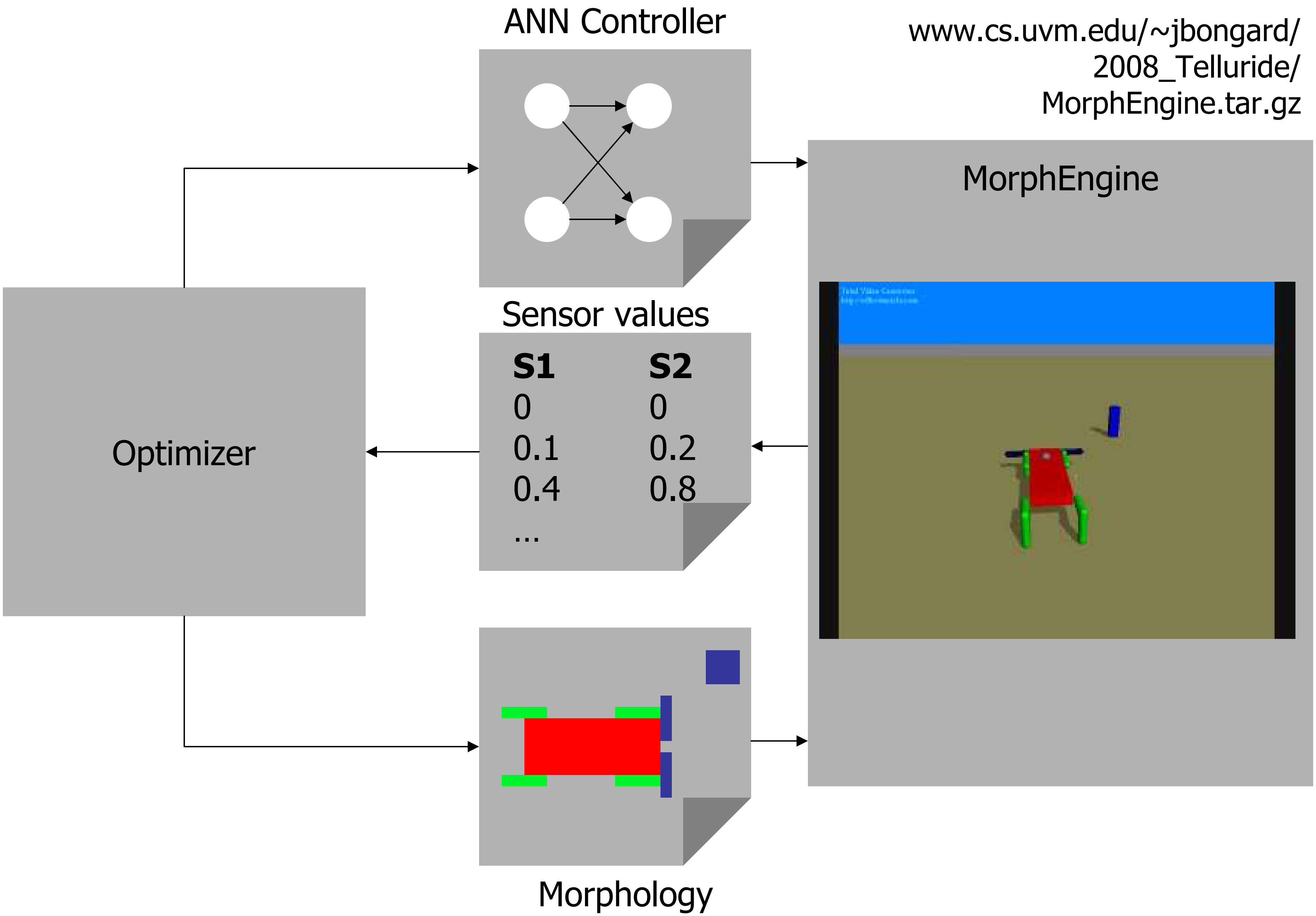
Conclusions

Proximate and Ultimate Mechanisms of Cognition
Summary

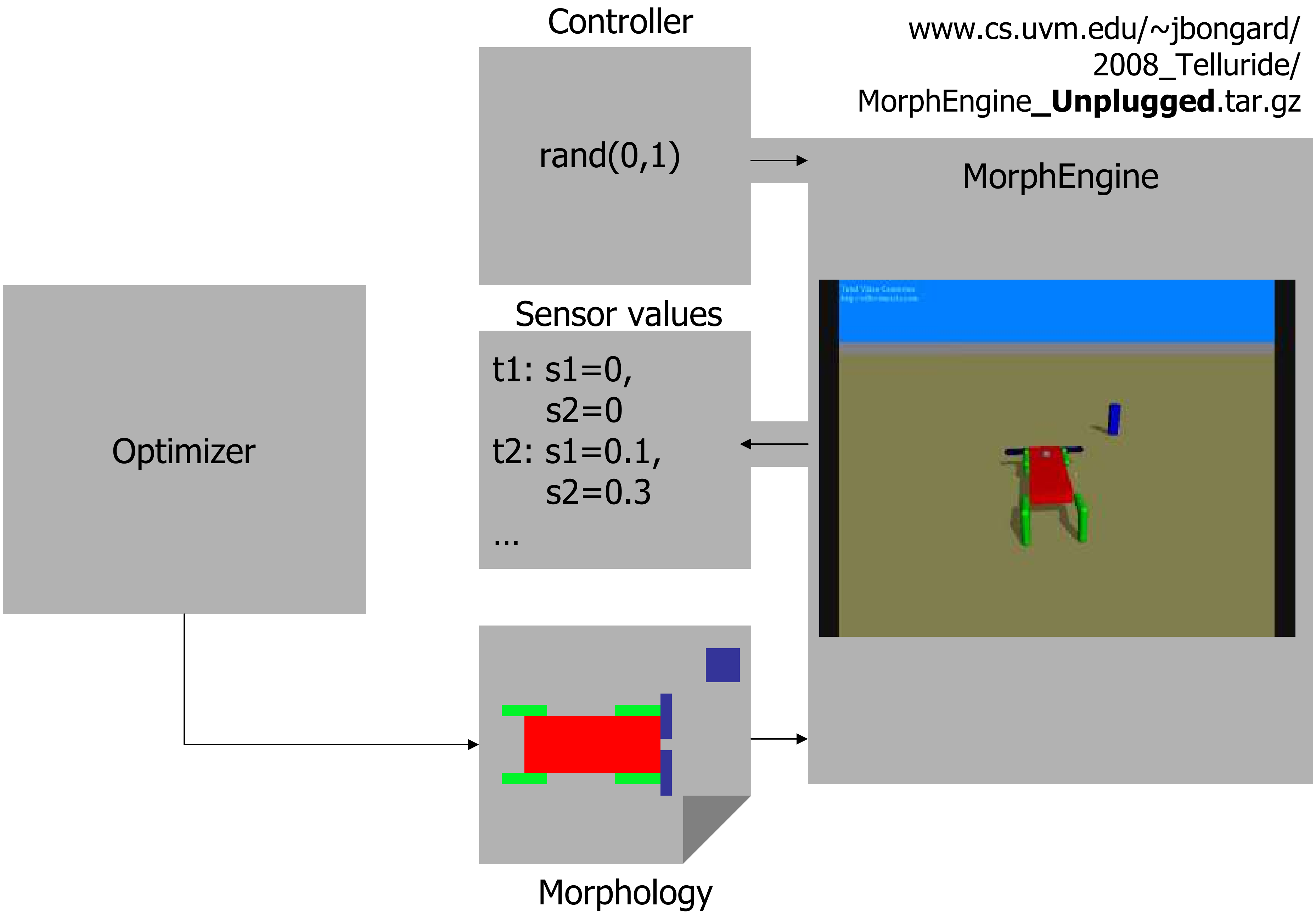
www.cs.uvm.edu/~jbongard/
2008_Telluride/
MorphEngine.tar.gz



www.cs.uvm.edu/~jbongard/
2008_Telluride/
MorphEngine.tar.gz

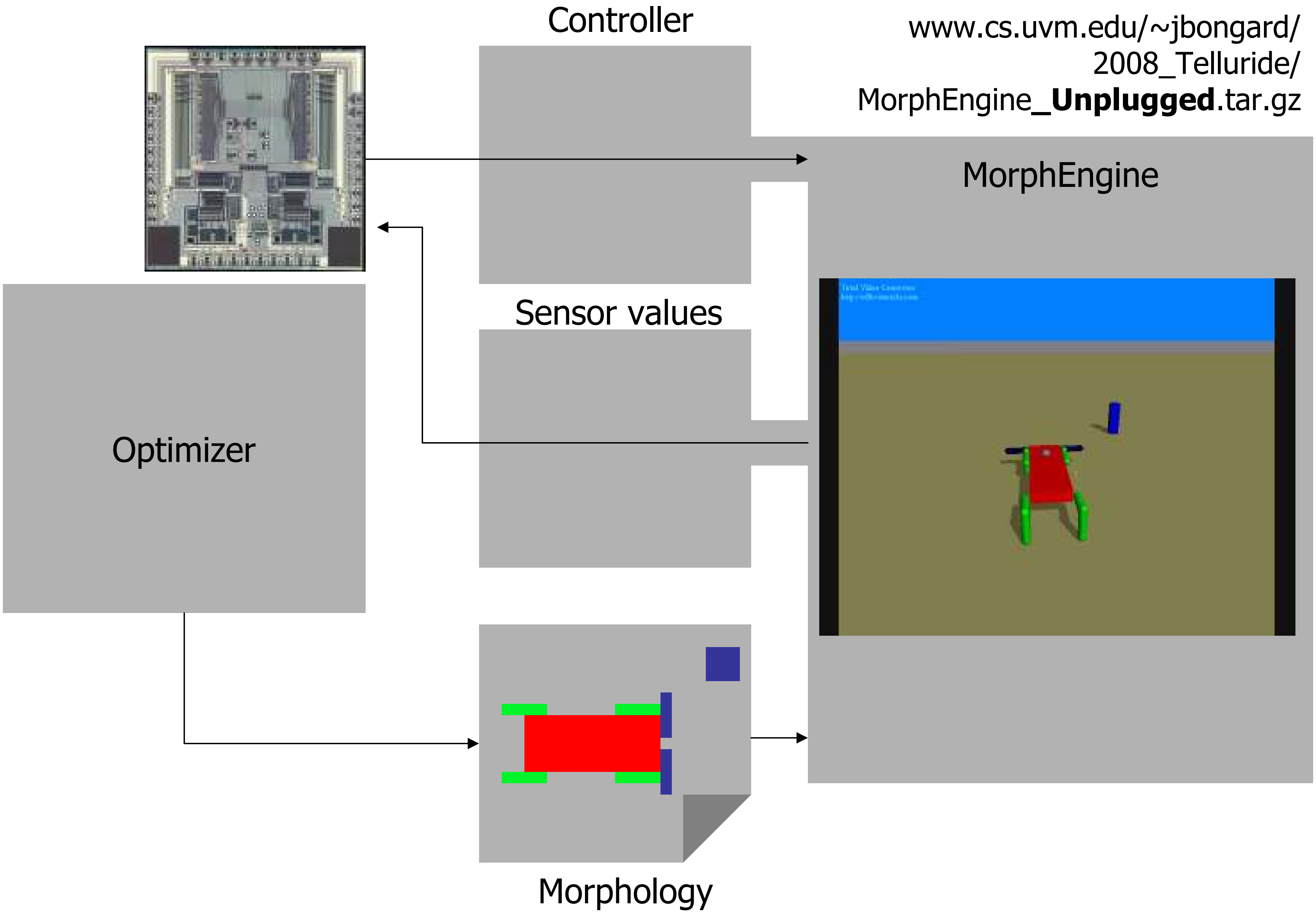


www.cs.uvm.edu/~jbongard/
2008_Telluride/
MorphEngine_**Unplugged**.tar.gz



MorphEngine Unplugged — Tool set

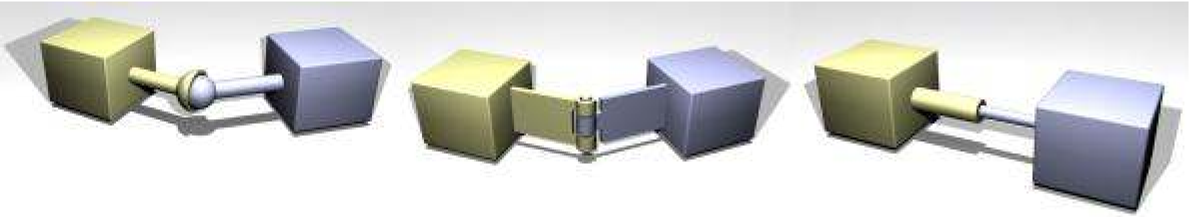
www.cs.uvm.edu/~jbongard/
2008_Telluride/
MorphEngine_**Unplugged**.tar.gz



Three-dimensional

For each time step

Internal, external **forces** are calculated for each object in the simulation
Positions, orientations and velocities of each **object** are updated



From ODE documentation

Collisions between objects are detected, and resolved

Simulated sensors
Real-time sensor data → Input sensor values → neural network

Simulated motors
Output neuron values → desired joint angles → Torque

The Tool Set

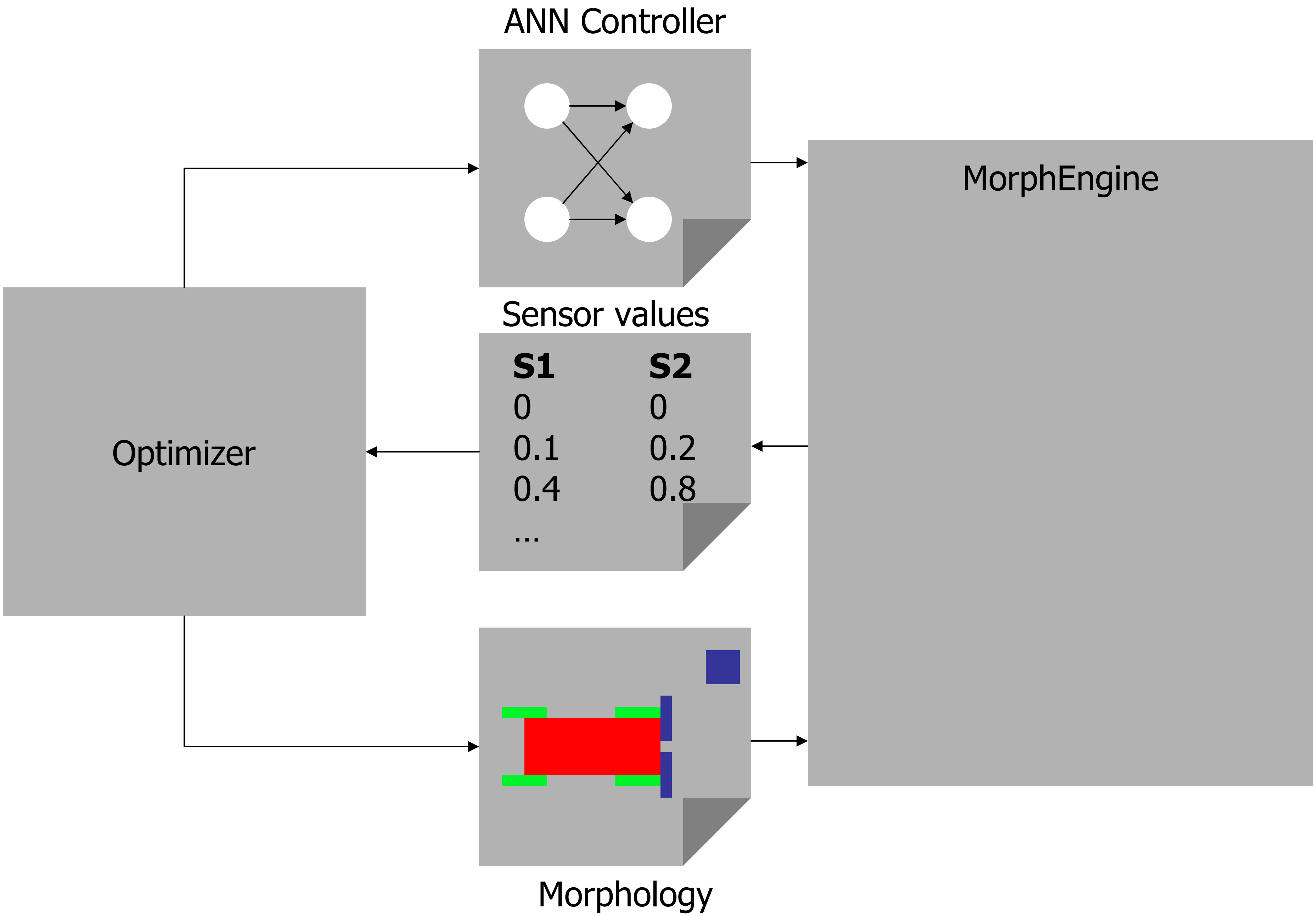
MorphEngine
MorphEngine `Unplugged'
Physical Simulation

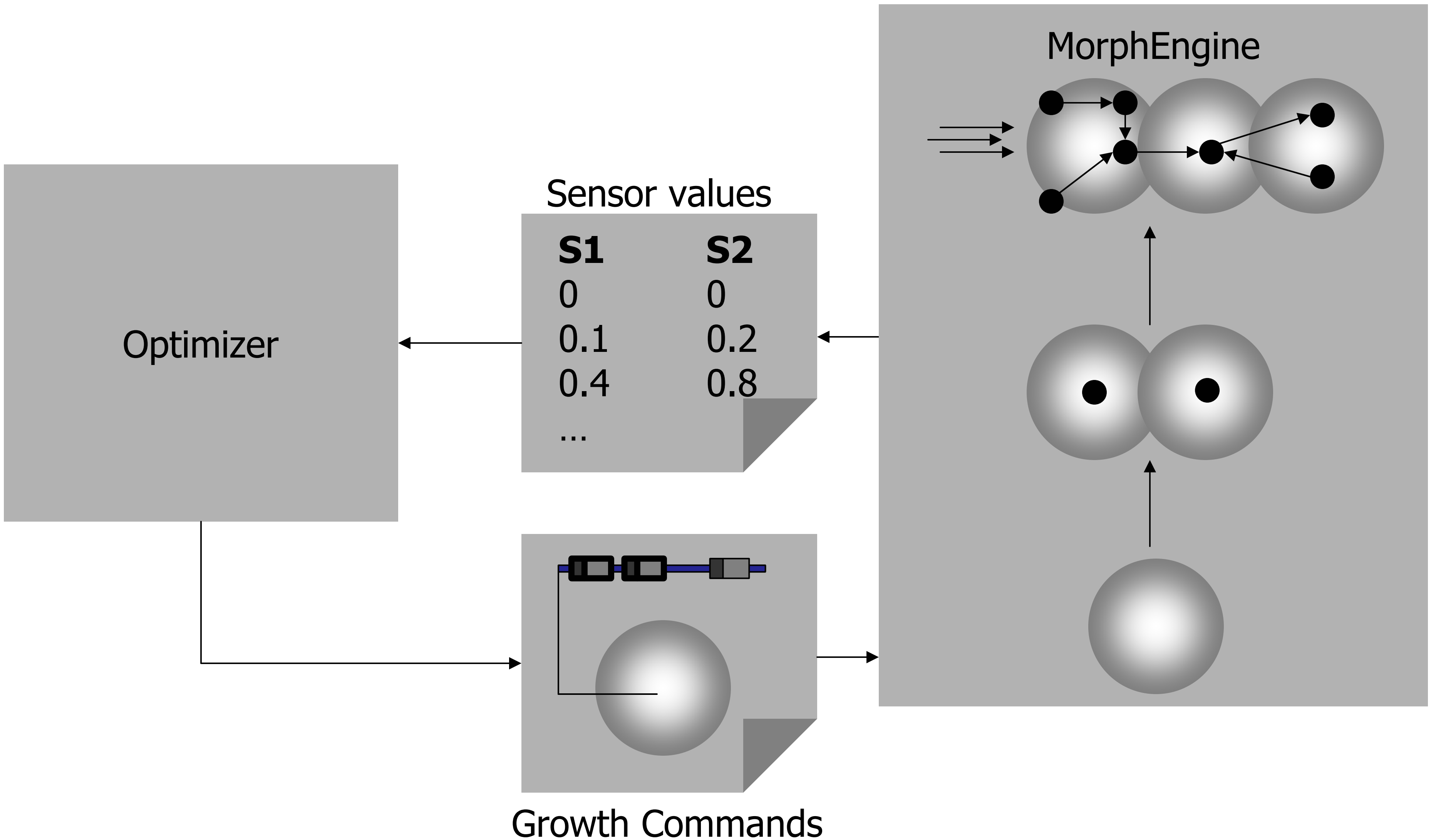
Example Investigations

Evolving Robot Morphologies and Controllers Together
Evolving Self-Models
Evolving Coupled, Nonlinear Models
Evolving Robots Capable of Multiple Behaviors

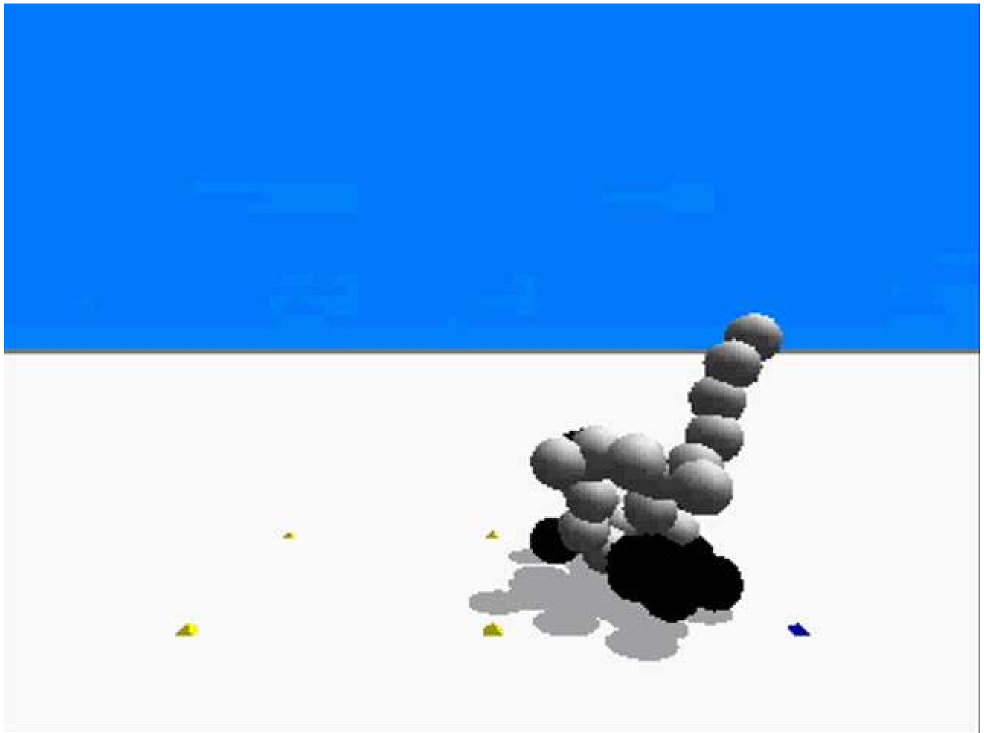
Conclusions

Proximate and Ultimate Mechanisms of Cognition
Summary

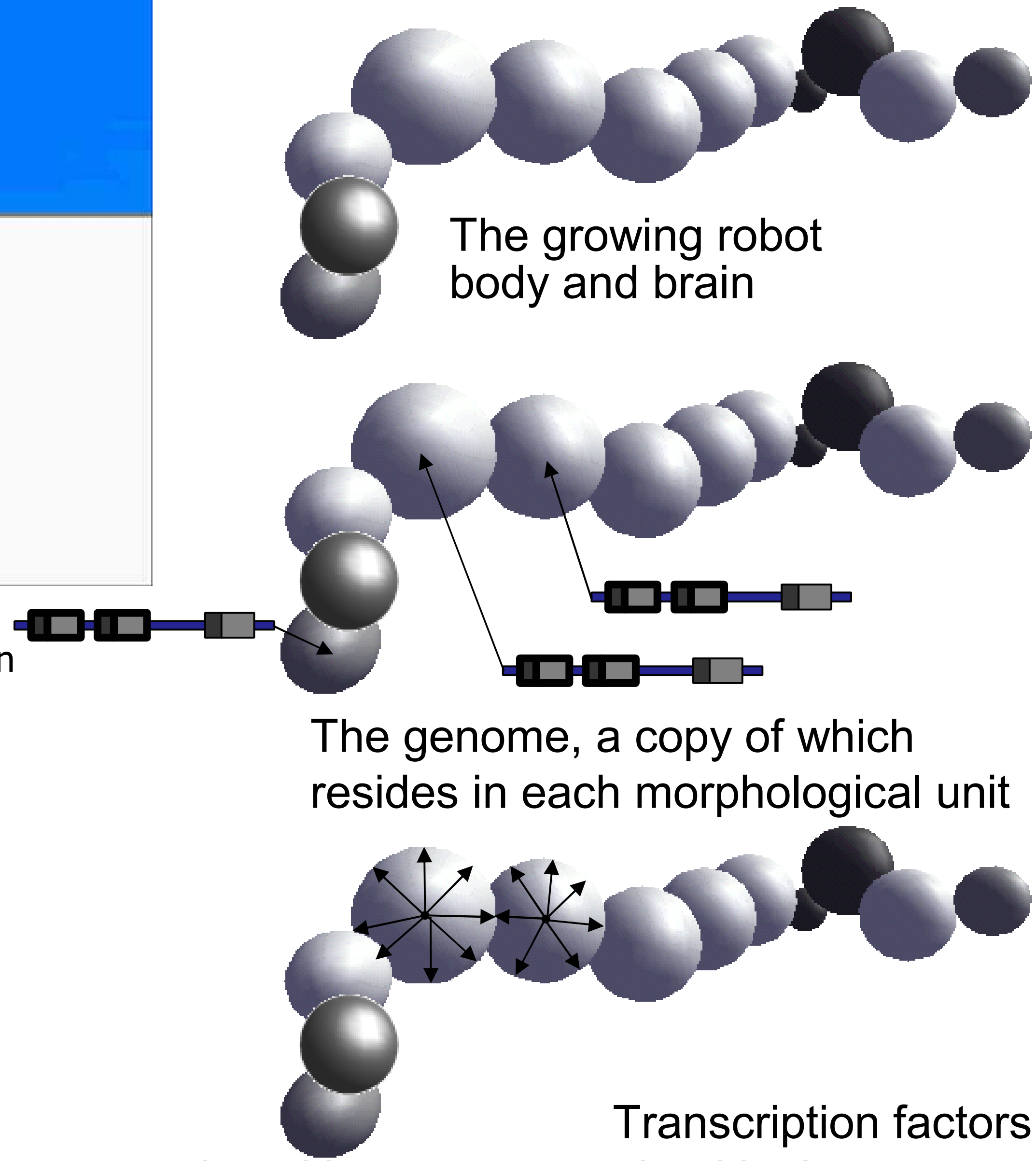

ANN Controller
MorphEngine
Sensor values
S1 S2
0 0
0.1 0.2
0.4 0.8
...
Optimizer
Morphology


MorphEngine
Sensor values
S1 S2
0 0
0.1 0.2
0.4 0.8
...
Optimizer
Growth Commands

Automating Robot Design

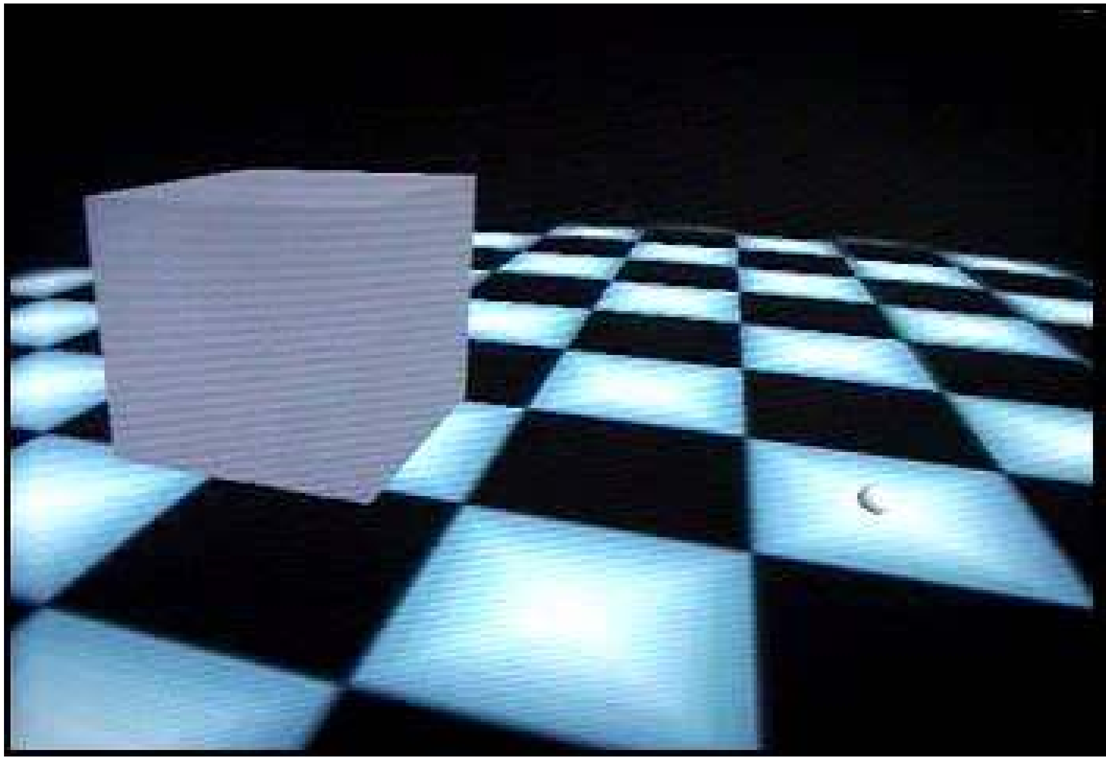


Bongard, J. C. (2002)
Evolving Modular Genetic Regulatory Networks, in *Proceedings of the IEEE 2002 Congress on Evolutionary Computation (CEC2002)*, pp. 1872-1877.

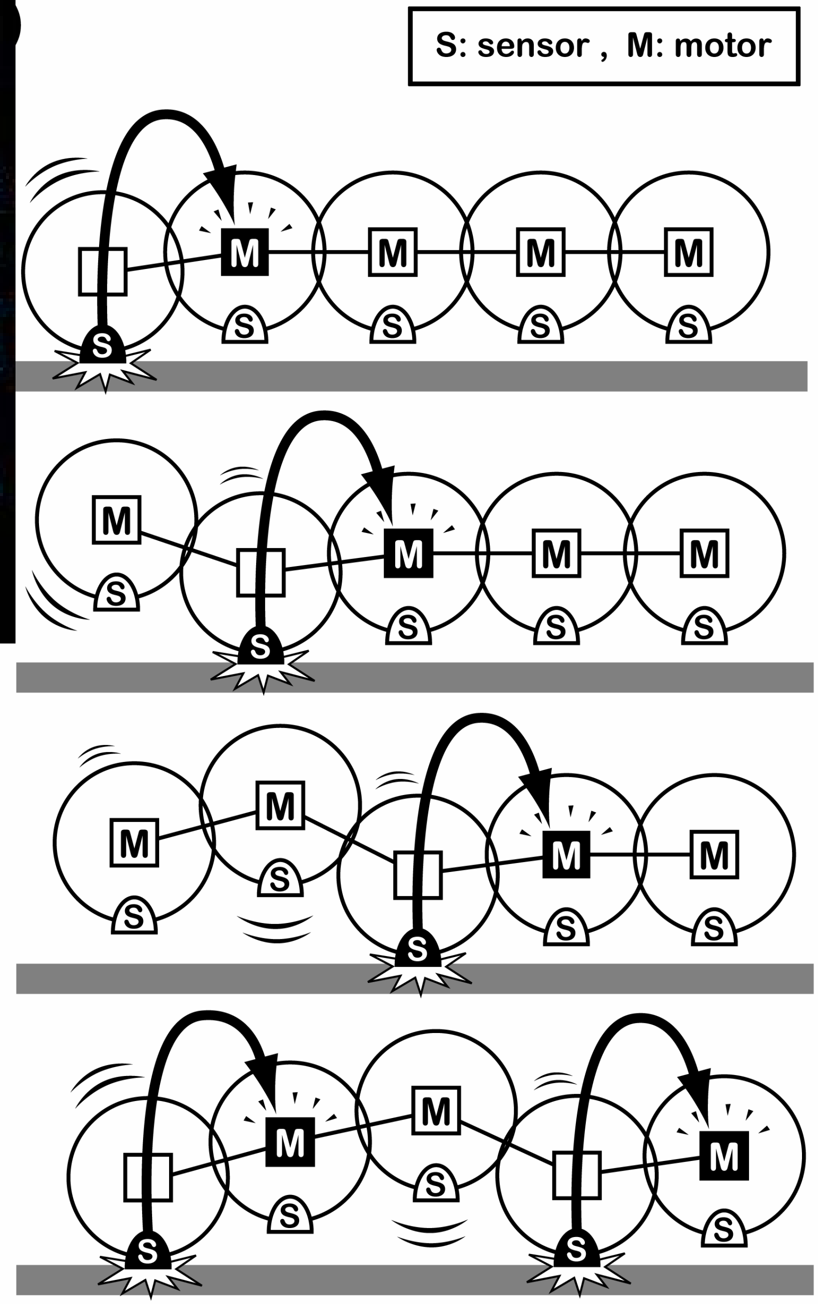


Transcription factors,
produced by genes contained in the genome,
diffuse through the robot's body, causing phenotypic change

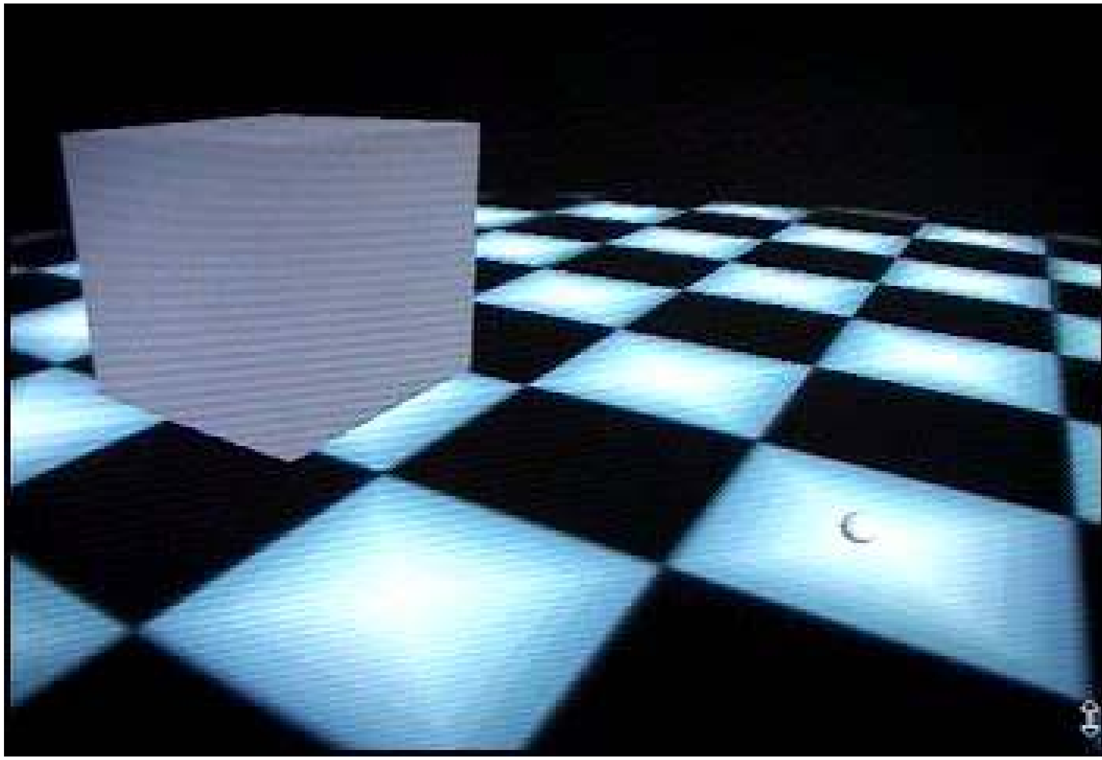
Automating Robot Design



Bongard, J. C. and R. Pfeifer (2001)
Repeated Structure and Dissociation of Genotypic and Phenotypic Complexity in Artificial Ontogeny, in *Proceedings of The Genetic and Evolutionary Computation Conference*,
pp. 829-836.

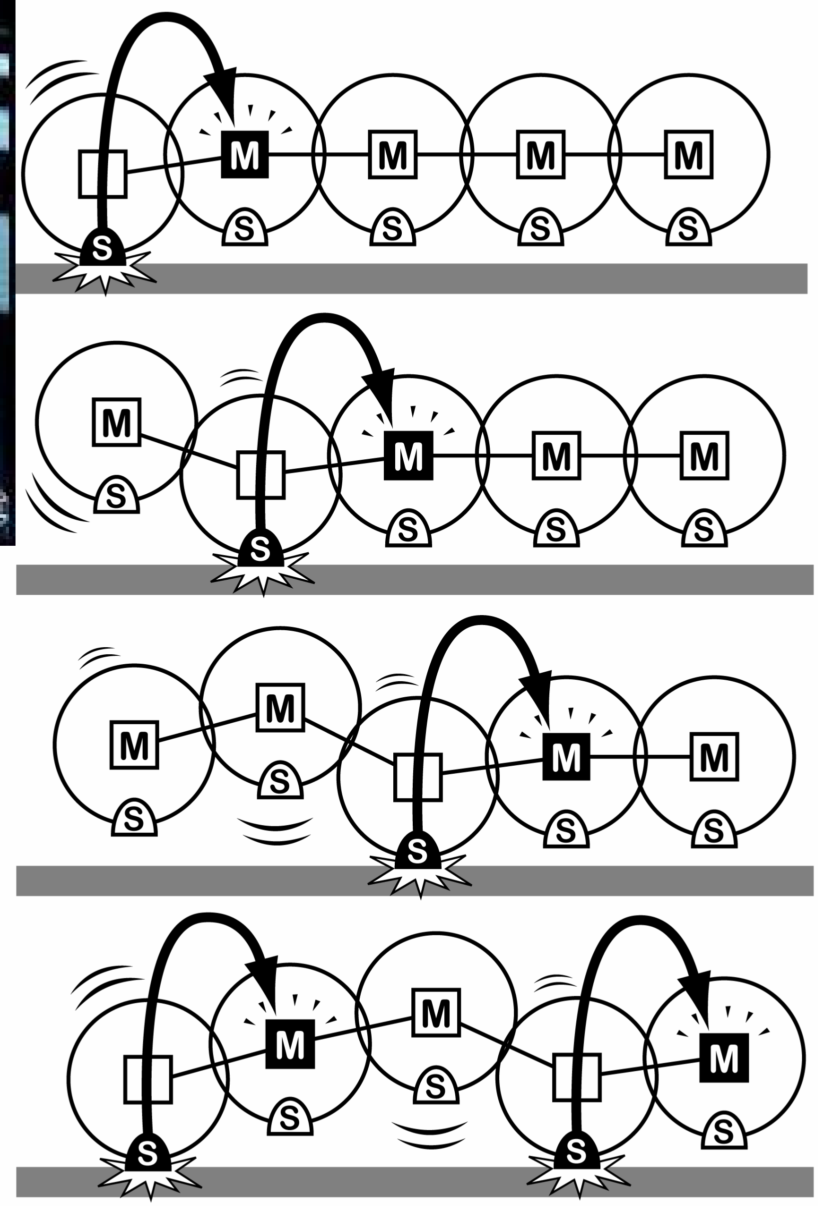


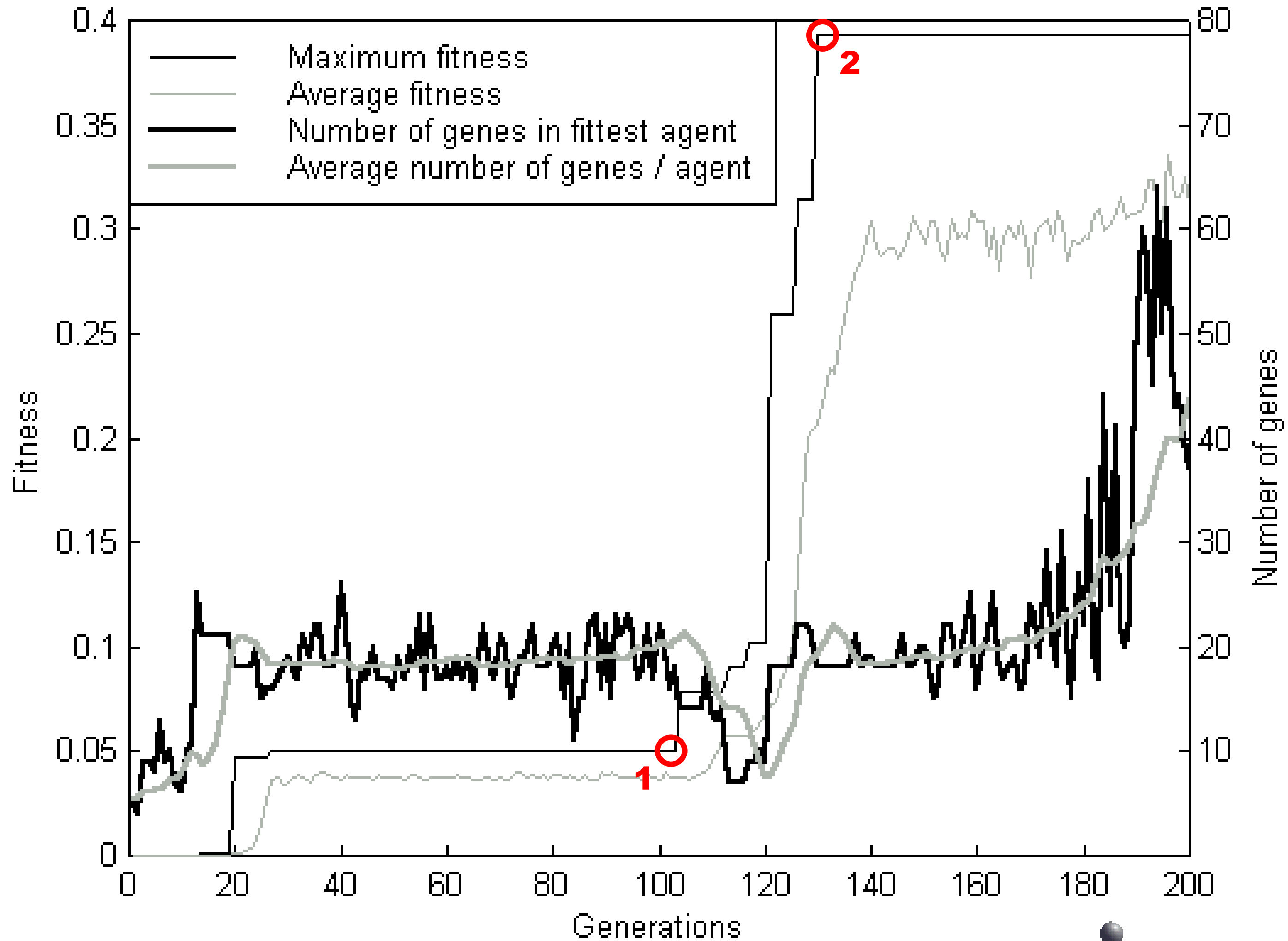
Automating Robot Design

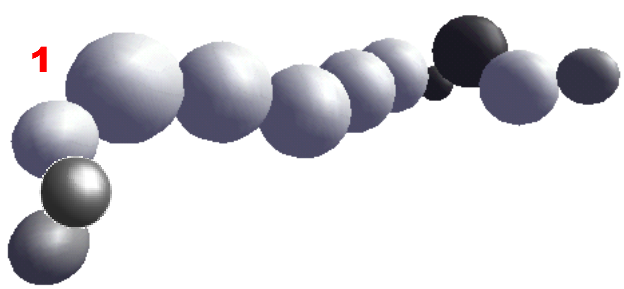


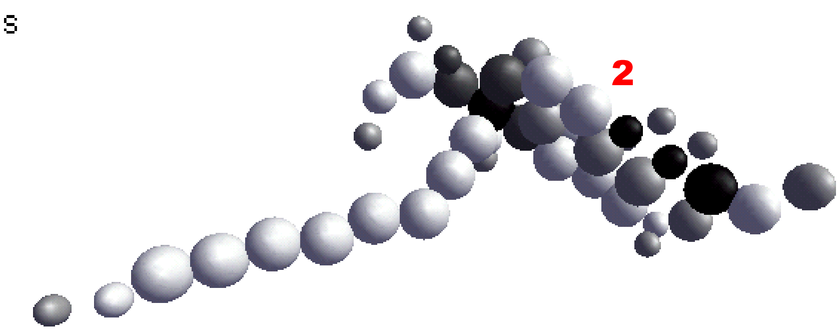
Bongard, J. C. and R. Pfeifer (2001)
Repeated Structure and Dissociation of Genotypic and Phenotypic Complexity in Artificial Ontogeny, in *Proceedings of The Genetic and Evolutionary Computation Conference*,
pp. 829-836.

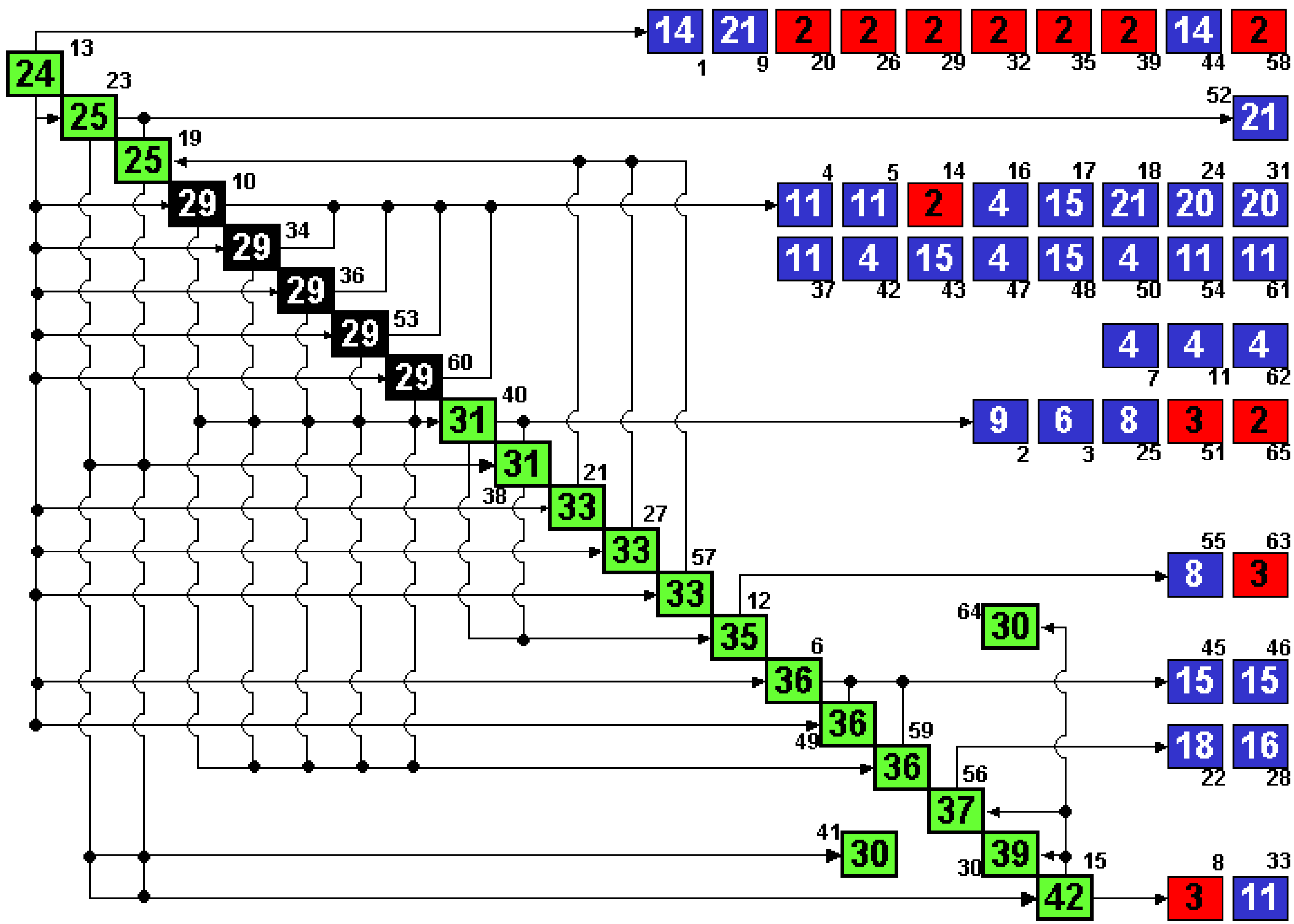
S: sensor , M: motor




Maximum fitness
Average fitness
Number of genes in fittest agent
Average number of genes / agent
Fitness
Number of genes
Generations
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0
20
40
60
80
100
120
140
160
180
200
10
20
30
40
50
60
70
80
1
2

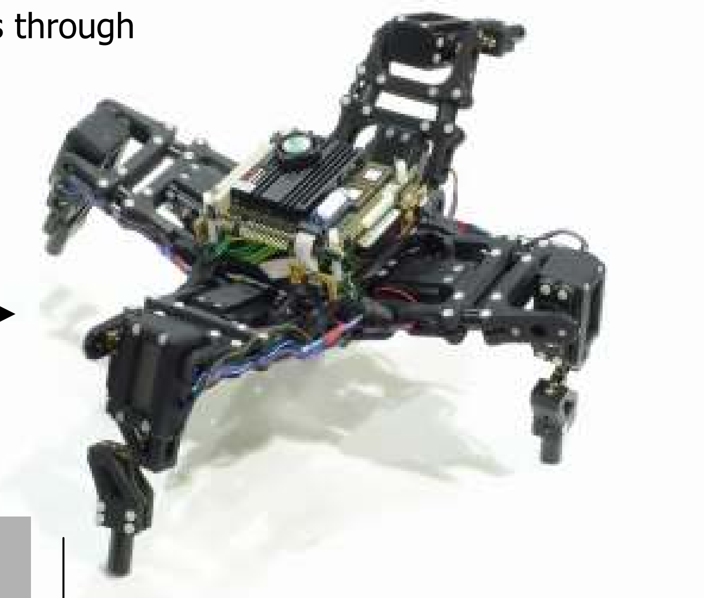

1

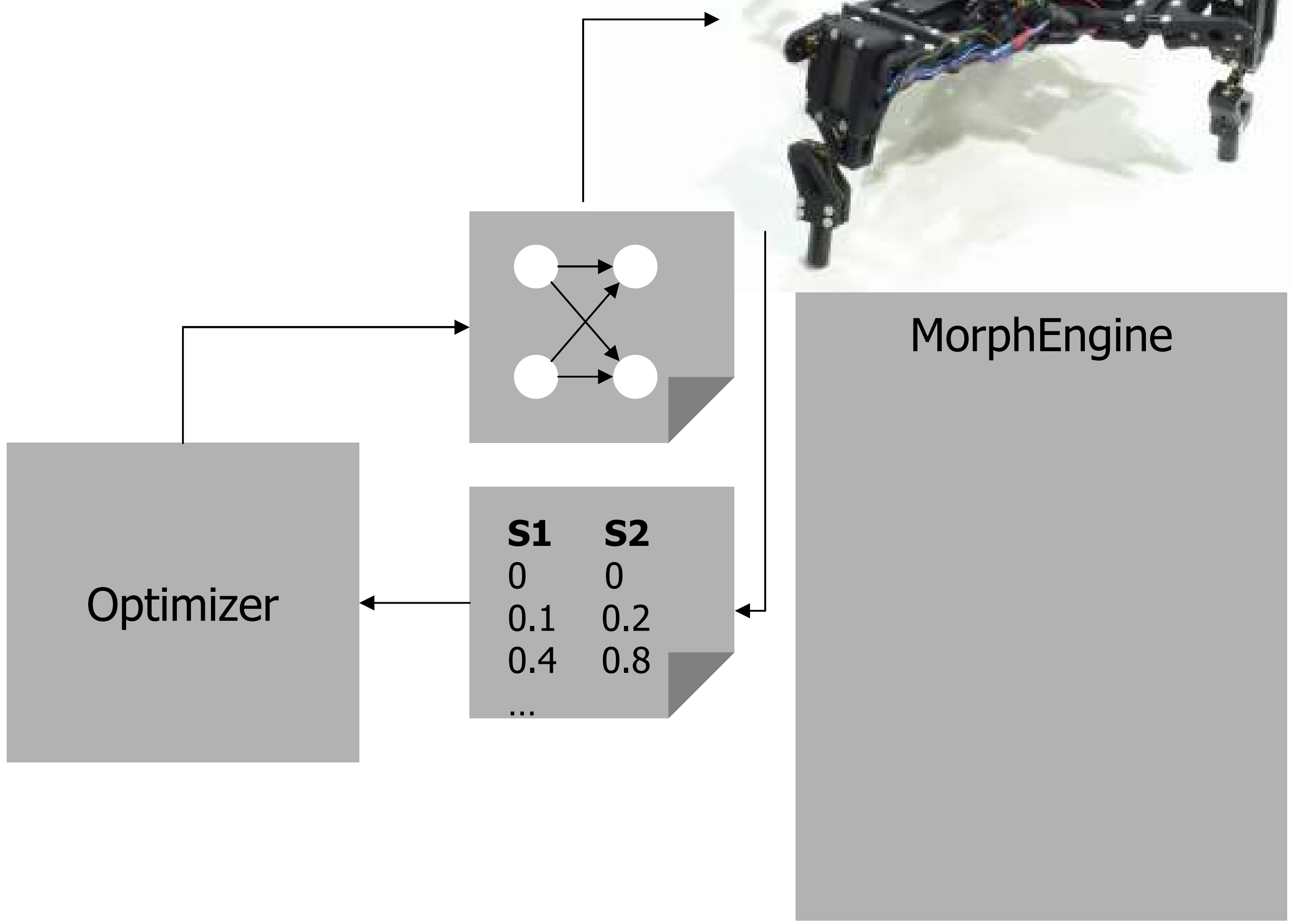

2



Regulatory gene
Targetted regulatory gene
Structural gene affecting morphogenesis
Structural gene affecting neurogenesis

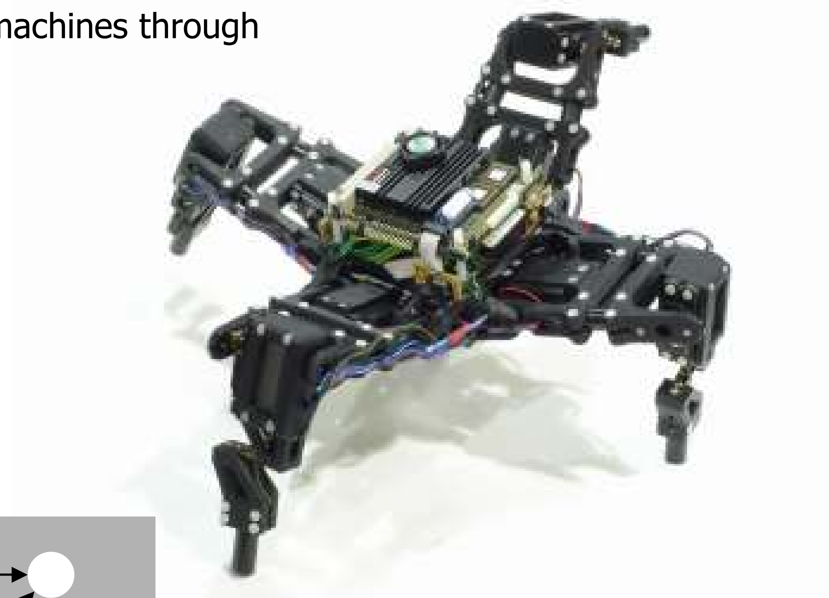
Bongard, J., Zykov, V., Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314: 1118-1121.

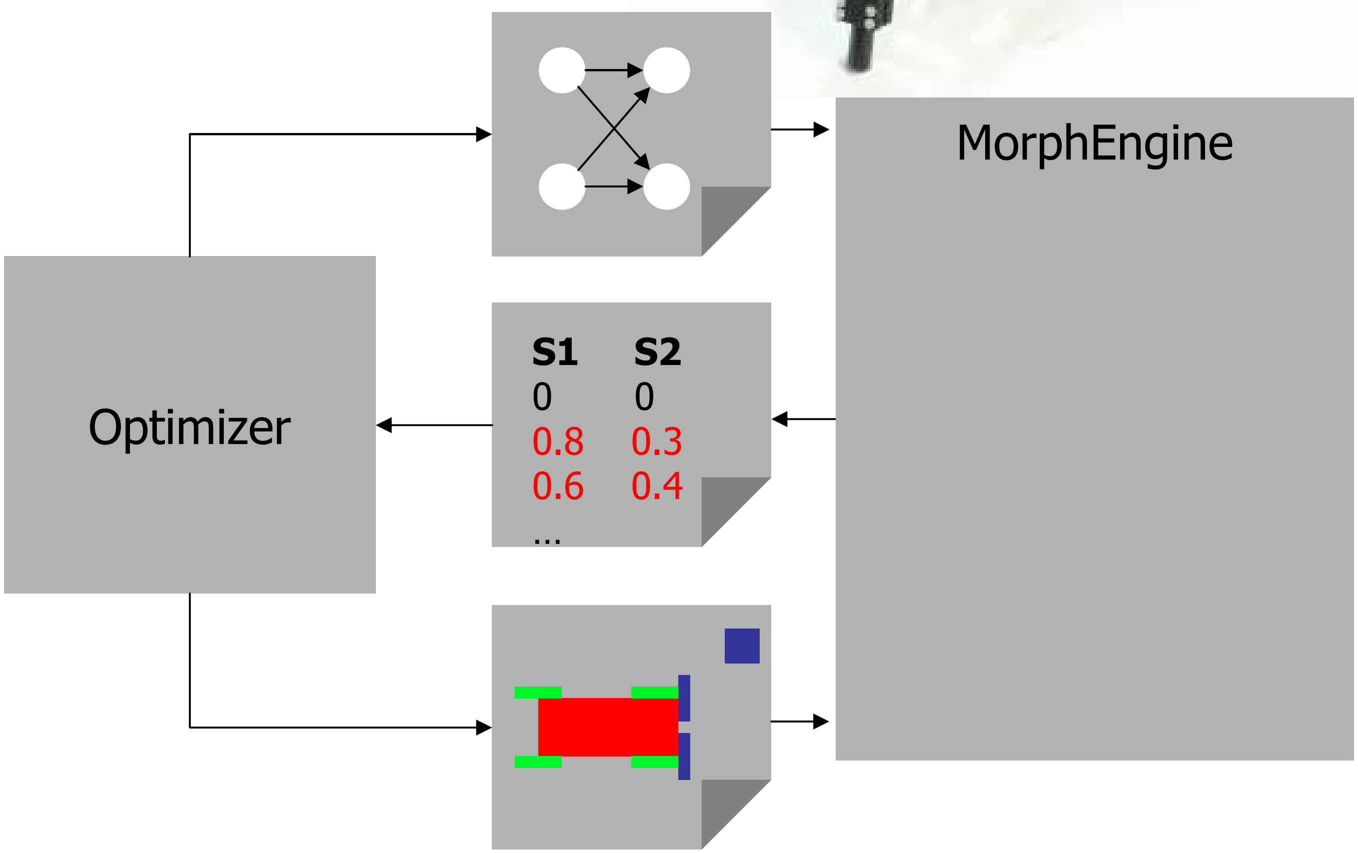




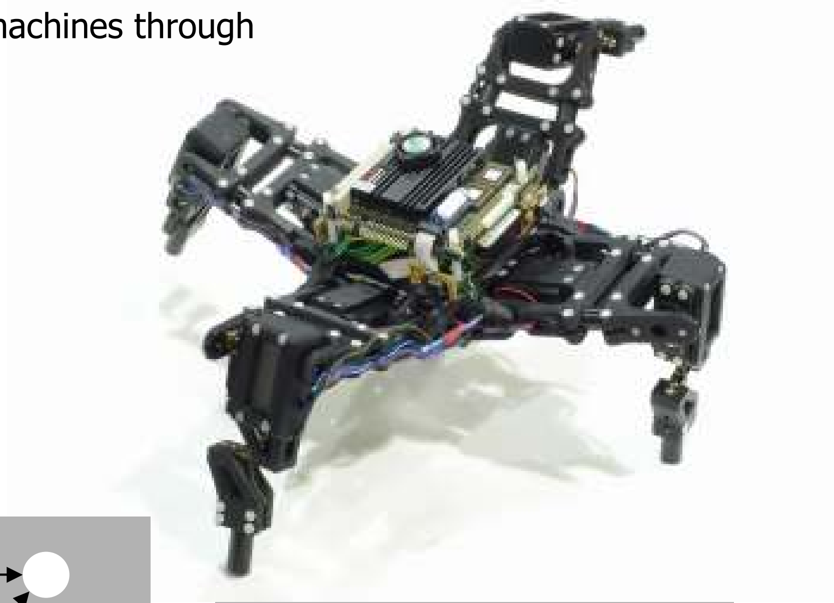
Evolving Self-Models

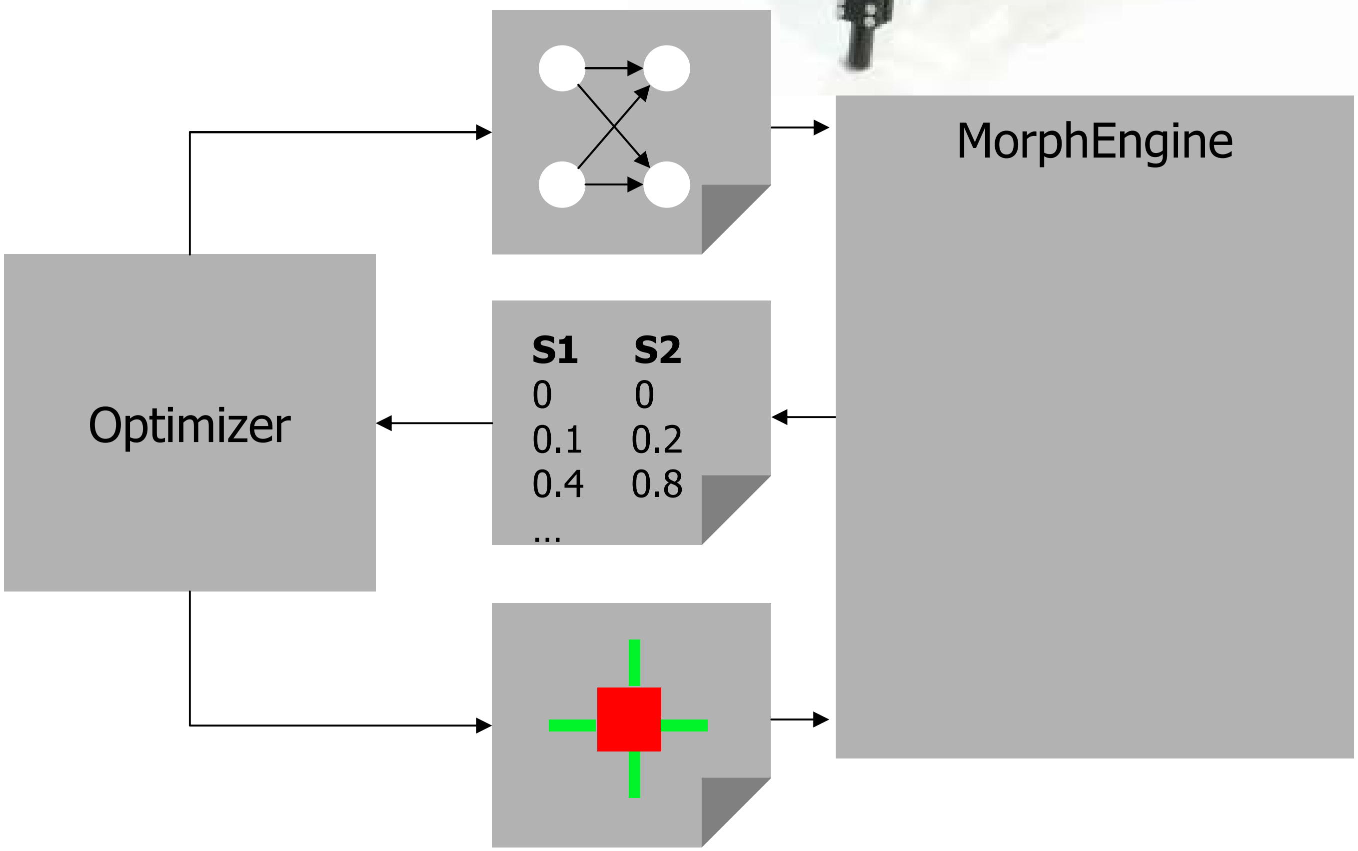
Bongard, J., Zykov, V., Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314: 1118-1121.

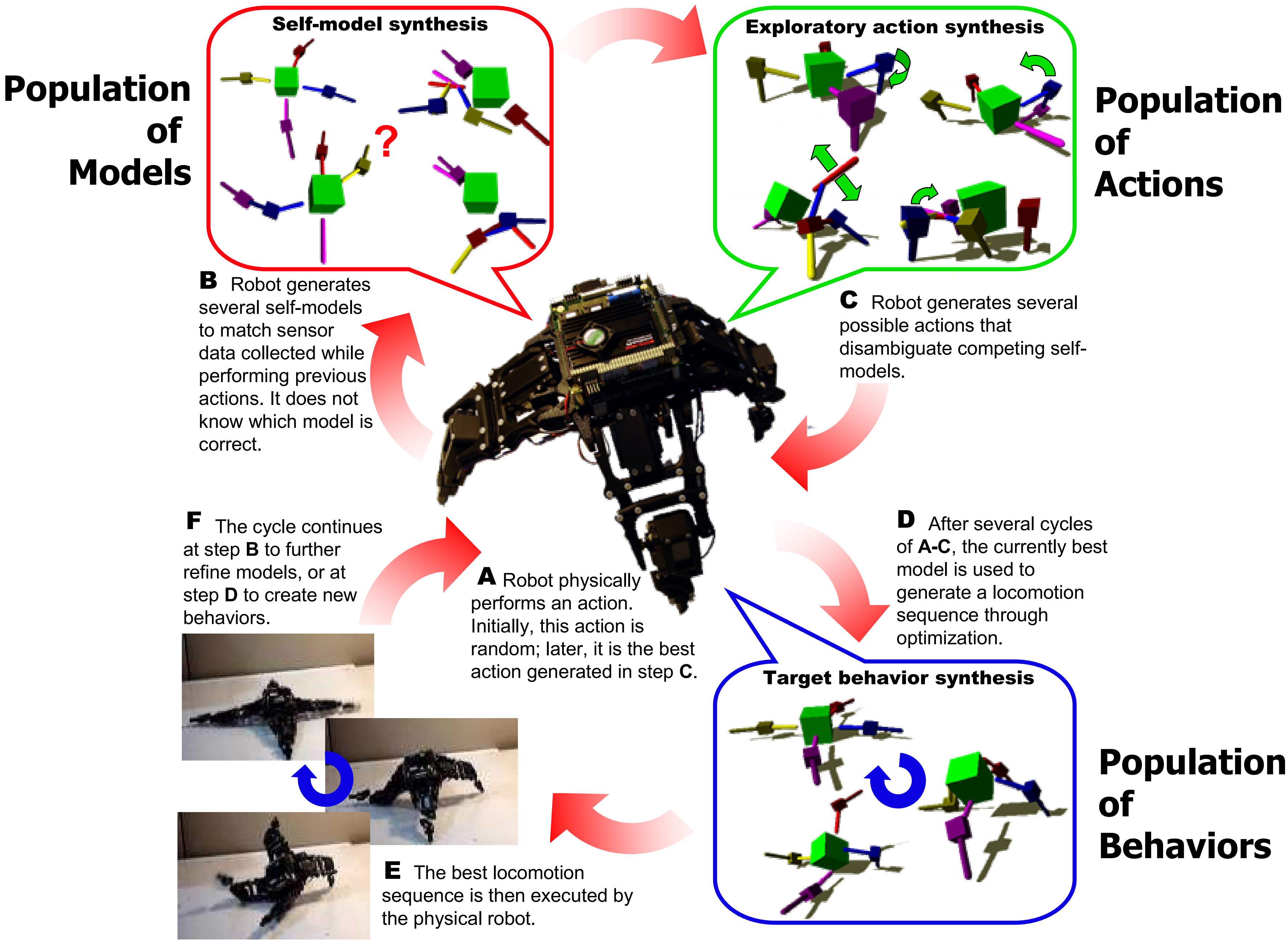




Bongard, J., Zykov, V., Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314: 1118-1121.

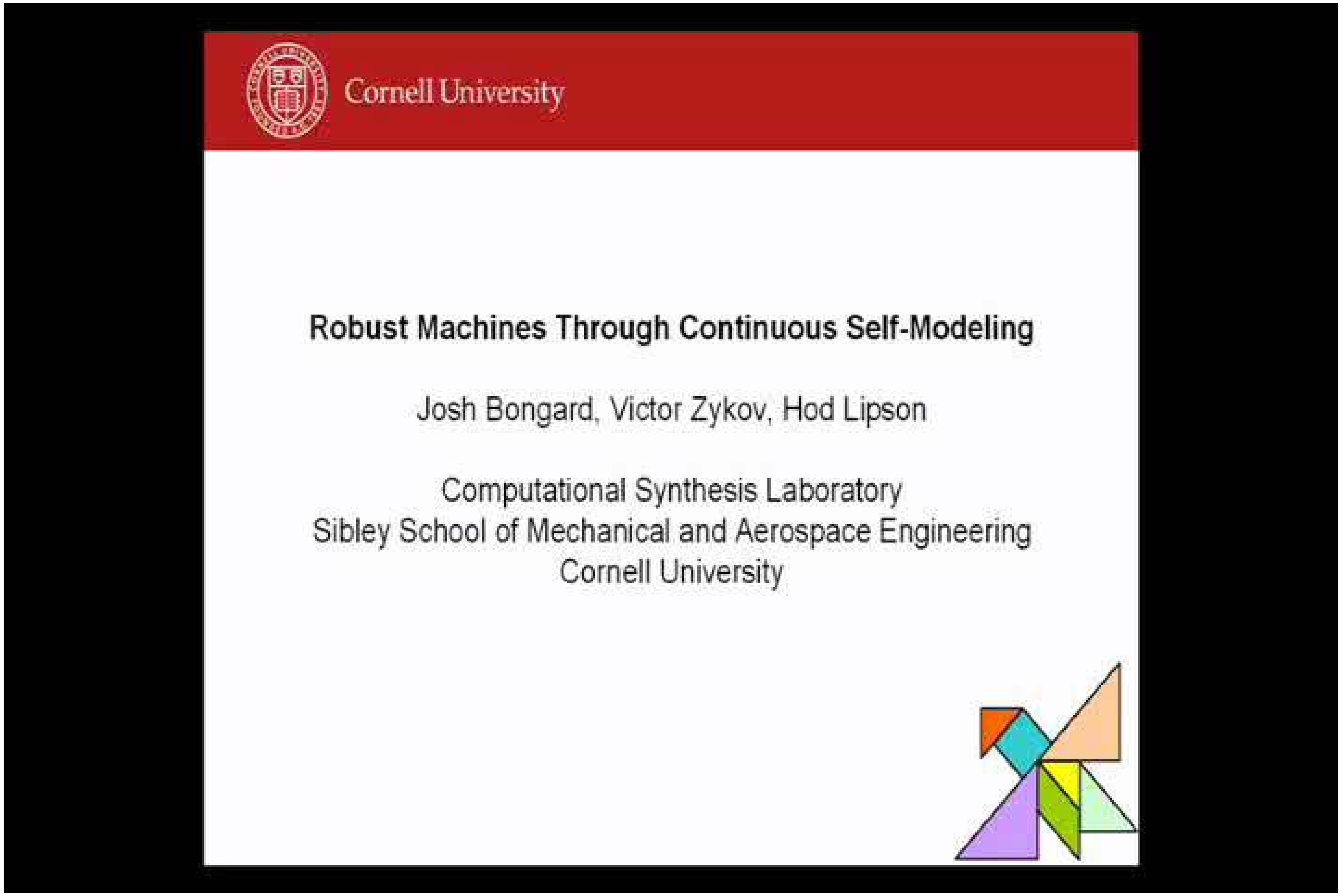





Population of Models
Self-model synthesis
?
B Robot generates several self-models to match sensor data collected while performing previous actions. It does not know which model is correct.
Exploratory action synthesis
Population of Actions
C Robot generates several possible actions that disambiguate competing self-models.
F The cycle continues at step B to further refine models, or at step D to create new behaviors.
A Robot physically performs an action. Initially, this action is random; later, it is the best action generated in step C.
D After several cycles of A-C, the currently best model is used to generate a locomotion sequence through optimization.
Target behavior synthesis
Population of Behaviors
E The best locomotion sequence is then executed by the physical robot.

Automating Robot Recovery

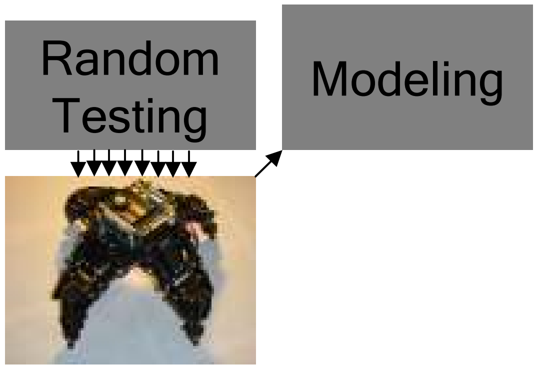
Bongard, J., Zykov, V., Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314: 1118-1121.



Comparative modeling performance

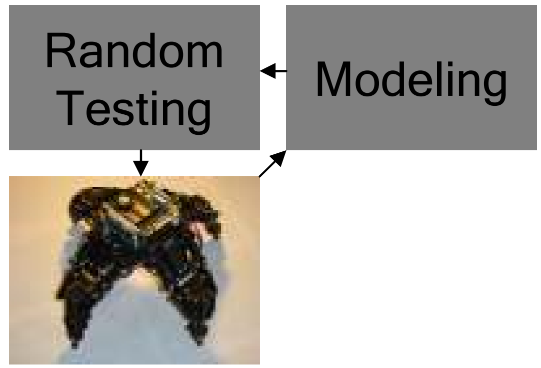
Intact robot

Batch testing



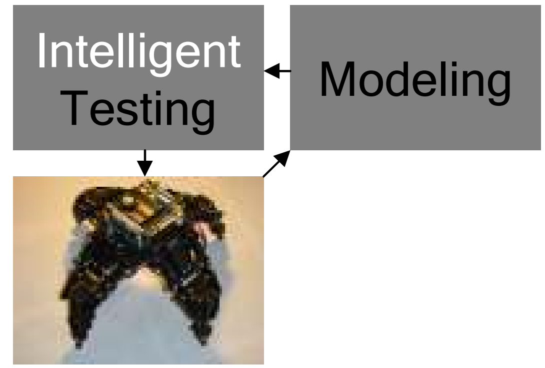
30 trials

Random testing



30 trials

Intelligent testing (EEA)



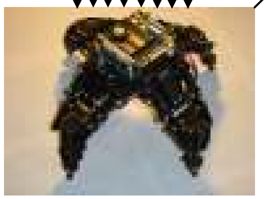
30 trials

Intact robot

Batch testing

Random Testing

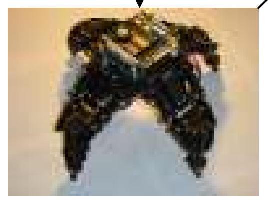
Modeling



Random testing

Random Testing

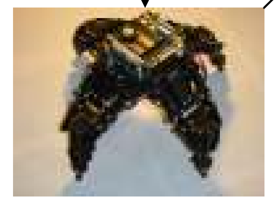
Modeling



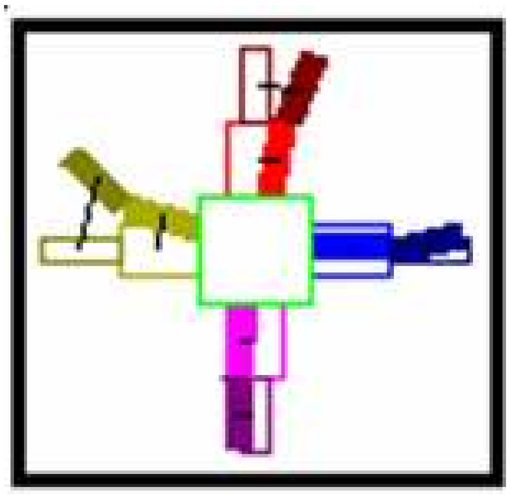
Intelligent testing (EEA)

Intelligent Testing

Modeling

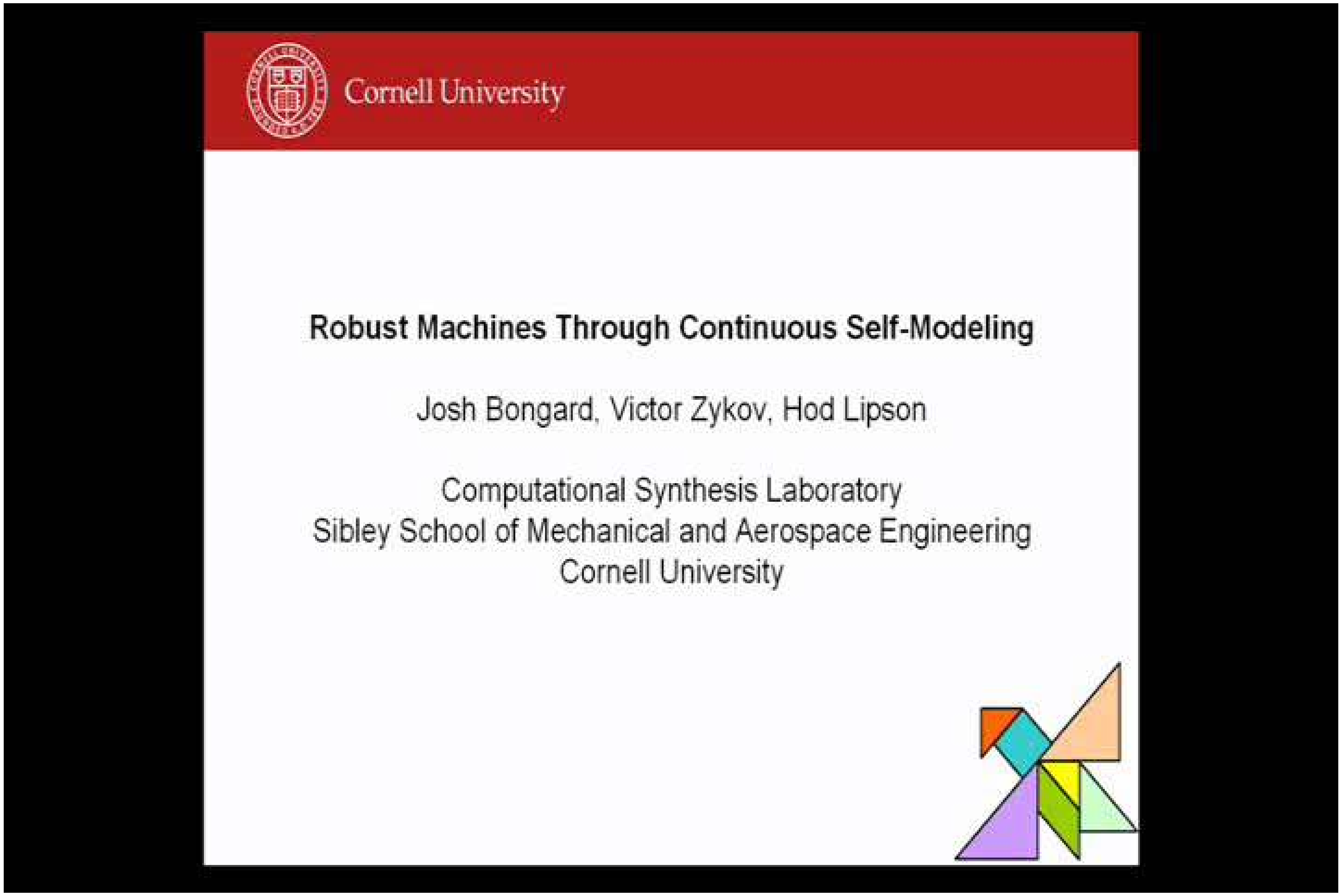


	Batch testing	Random testing	Intelligent testing (EEA)
Topology correct?	23.3% success	26.7% success	**43.3**% success
Model error:	9.62 +/- 1.47 cm	9.7 +/- 1.45 cm	7.31 +/- 1.22 cm



Bongard, J., Zykov, V., Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314: 1118-1121.

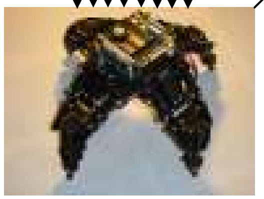
Automating Robot Recovery



Intact robot

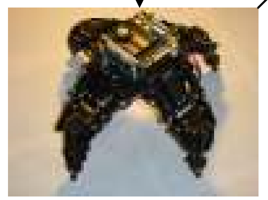
Batch testing

Random Testing → Modeling



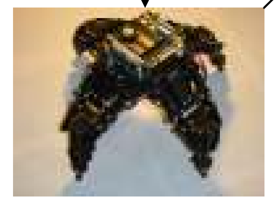
Random testing

Random Testing ← Modeling



Intelligent testing (EEA)

Intelligent Testing ← Modeling



Topology correct? 23.3% success | 26.7% success | **43.3**% success

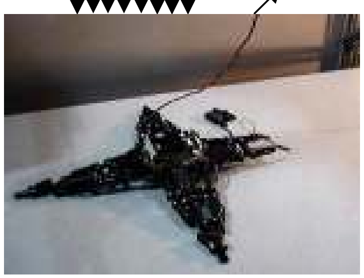
Model error: 9.62 +/- 1.47 cm | 9.7 +/- 1.45 cm | 7.31 +/- 1.22 cm

---

Damaged robot

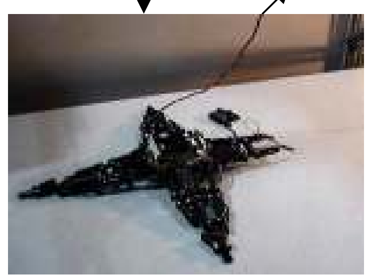
Batch testing

Random Testing → Modeling



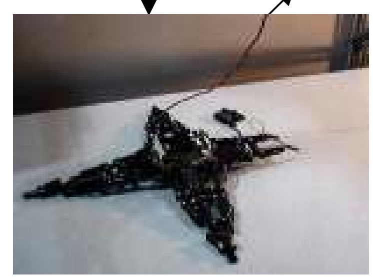
Random testing

Random Testing ← Modeling

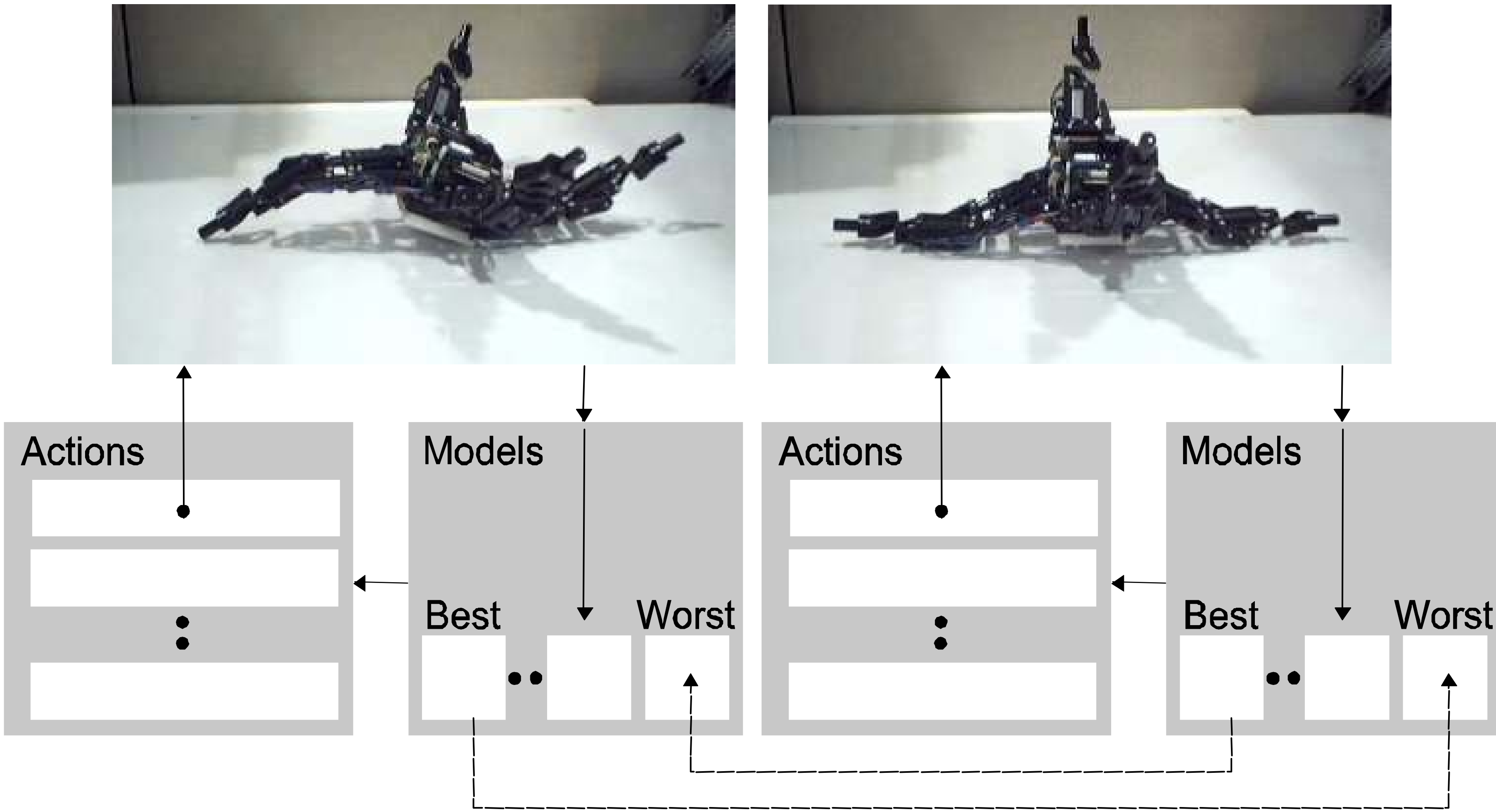


Intelligent testing (EEA)

Intelligent Testing ← Modeling



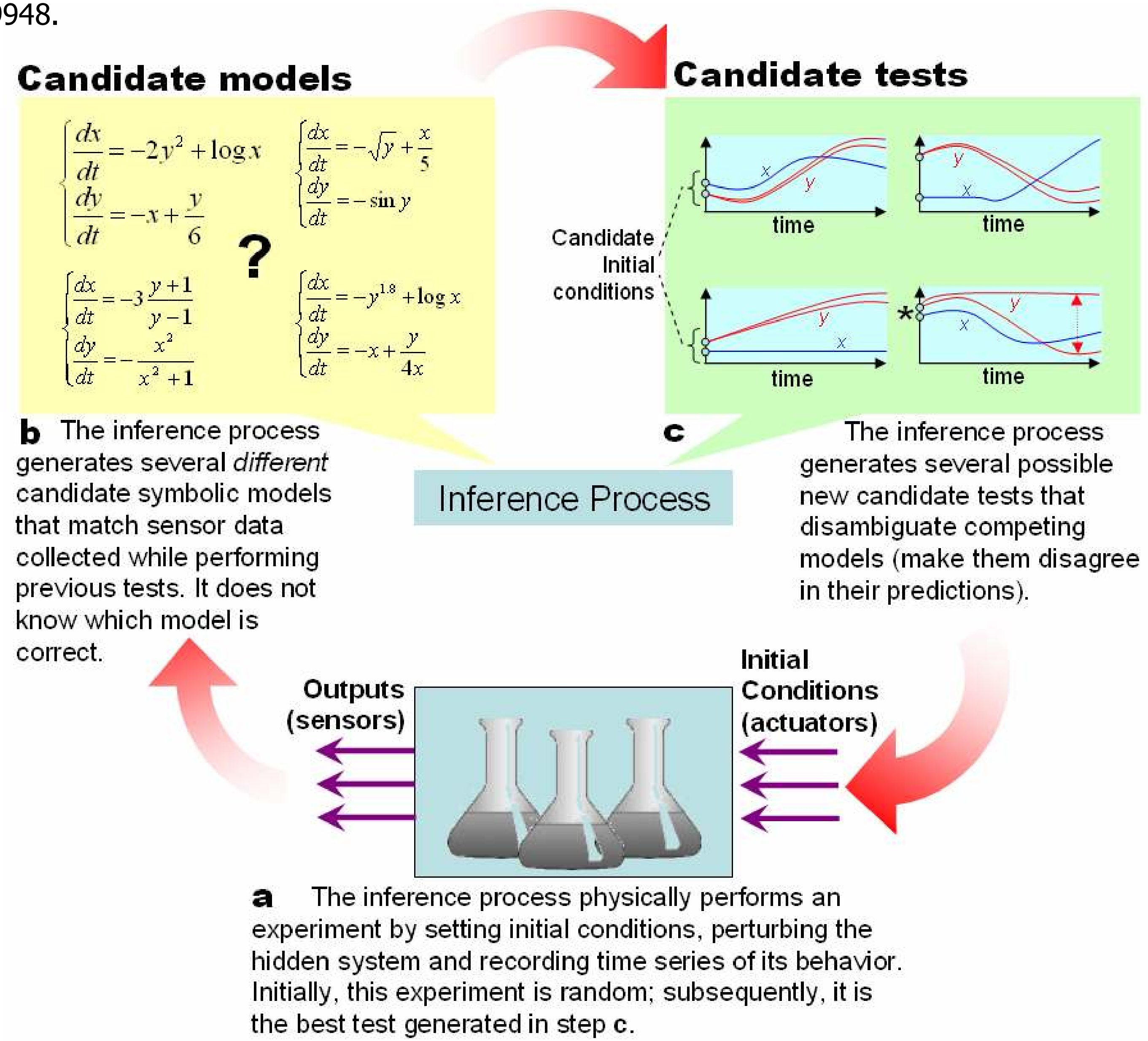
Model error: 5.60 +/- 2.98 cm | 4.55 +/- 3.22 cm | 2.17 +/- 0.55 cm



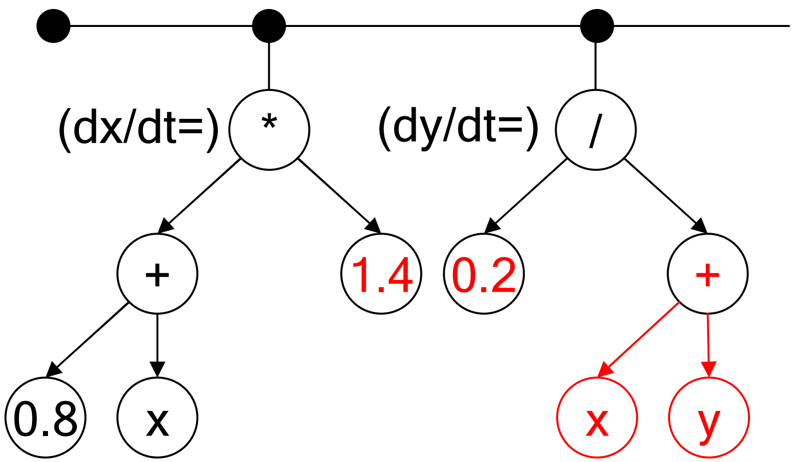
Bongard, J. (2007) Exploiting Multiple Robots to Accelerate Self-Modeling, *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, New York, NY, pp. 214-221.

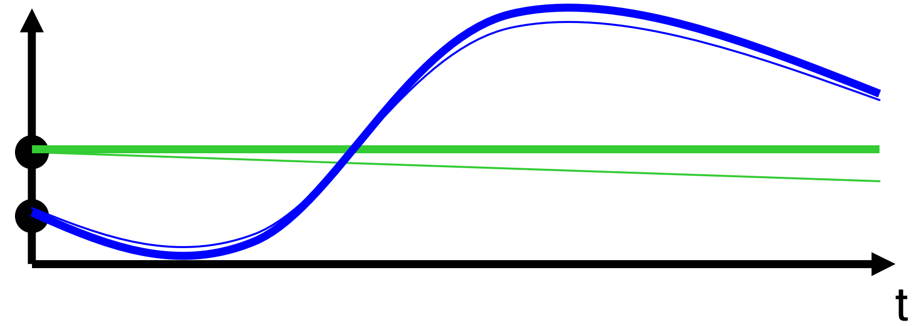
Bongard J. and Lipson H.(2007). Automated reverse engineering of nonlinear dynamical systems.
Proceedings of the National Academy of Sciences,
104(24): 9943-9948.

Automating System Identification

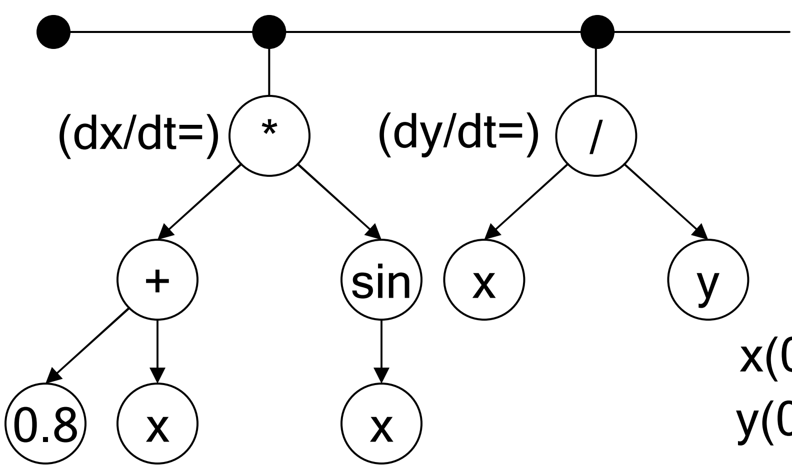


Encoding and optimizing of models

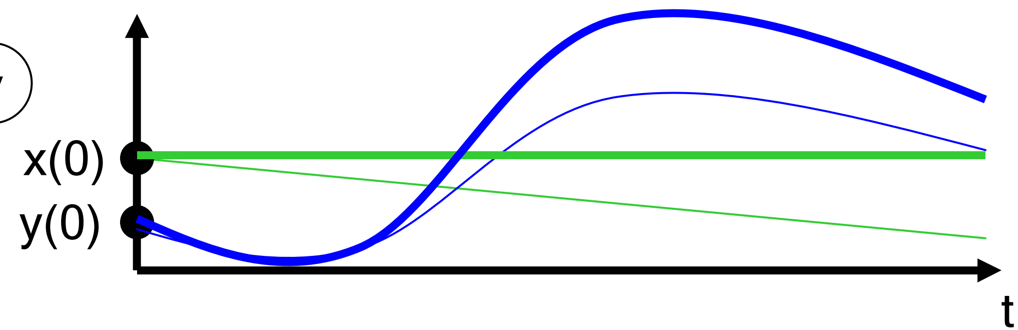


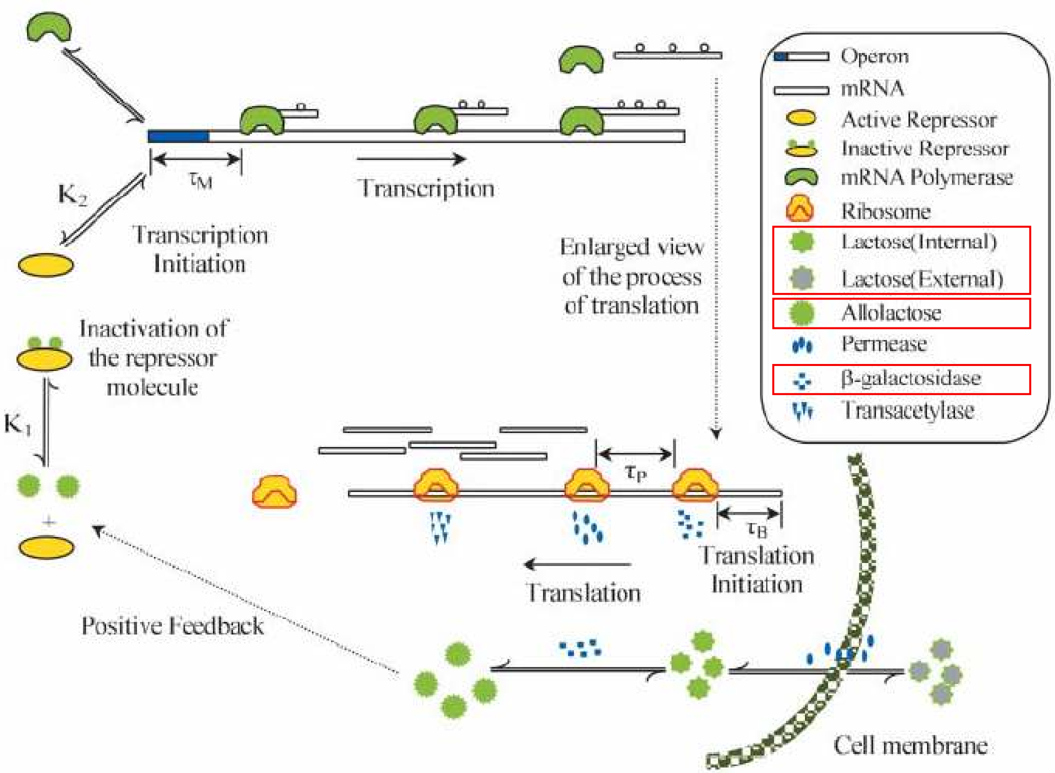


$$\frac{dx}{dt} = 1.4(0.8+x)$$
$$\frac{dy}{dt} = 0.2 / (x+y)$$



$$\frac{dx}{dt} = (0.8 + x) * \sin(x)$$
$$\frac{dy}{dt} = x / y$$

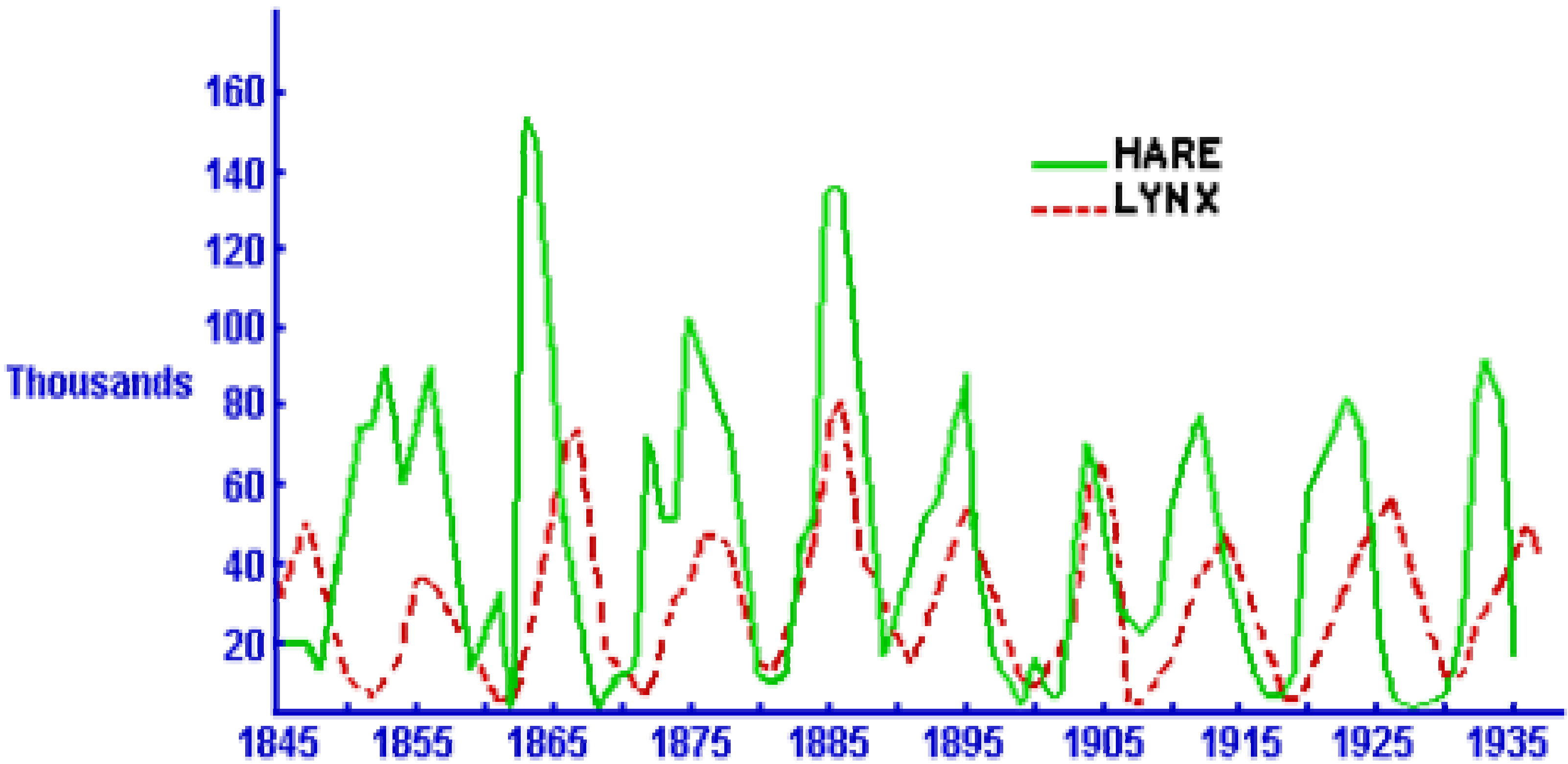




The *lac* operon from *E. coli* (G = concentration of beta-galactosidase; A = allolactose; L = lactose)	
Target system	$dG/dt = A^2/(A^2+1) - 0.01G + 0.001$ $dA/dt = G(L/(L+1) - A/(A+1))$ $dL/dt = -GL/(L+1)$
Best model	$dG/dt = 0.96A^2/(0.96A^2+1)$ $dA/dt = G(L/(L+1) - A/(A+1))$ $dL/dt = -GL/(L+1)$

Historical data reporting approximated populations of snowshoe hare (H) and Canadian lynx (L)

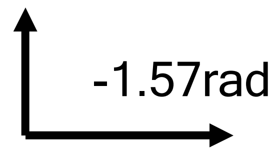
Target System

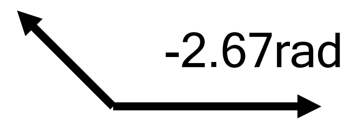


Best model

$$\frac{dH}{dt} = 3.42 \times 10^{6} - 67.82H - 10.97L$$
$$\frac{dL}{dt} = 3.10 \times 10^{5} + 32.66H - 63.16L$$

Evolving Coupled, Nonlinear Models — Application III: Mechanical pendula





$d\theta/dt = 1.004\omega + 0.0001$
$d\omega/dt = -19.43\sin(1.104\theta\mathbf{+0})$

$d\theta/dt = 1.0039\omega - 0.0003$
$d\omega/dt = -22.61\sin(1.101\theta\mathbf{-2.673})$

$d\theta/dt = 1.008\omega + 0.0028$
$d\omega/dt = -19.43\sin(1.0009\theta\mathbf{-1.575})$

$d\theta/dt = \omega$
$d\omega/dt = -9.8L\sin(\theta)$

Model for an idealized single pendulum with no friction


Boston Dynamics

Behavior chaining

Enables a robot to learn multiple, dynamic behaviors gradually.

Learns one behavior,
then gradually incorporates new behaviors into its existing repertoire.

All behaviors are incorporated into the same monolithic controller.

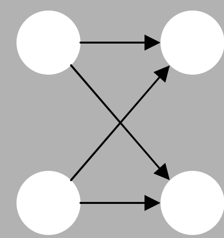
Is more scalable than other approaches that require building
a new controller component for each new behavior.

Builds on the idea of scaffolding, and robot shaping:
gradually changing the environment
to guide the learner toward a complex behavior
it might not have learned otherwise.

Behavior Chaining:

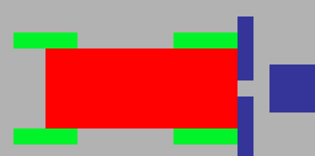
```
p = random controller
Eval(p)

For g = 1:generations

   tmp = CopyOf(p);
   c = Mutate(tmp);
   Eval(c)

   if (Fitness(c) > Fitness(p))
      p = c
   else
      delete c

      if Success(p)
         MoveObjFurther()

      if Failure(p)
         LengthenEvalTime()
```

CTRNN Controller



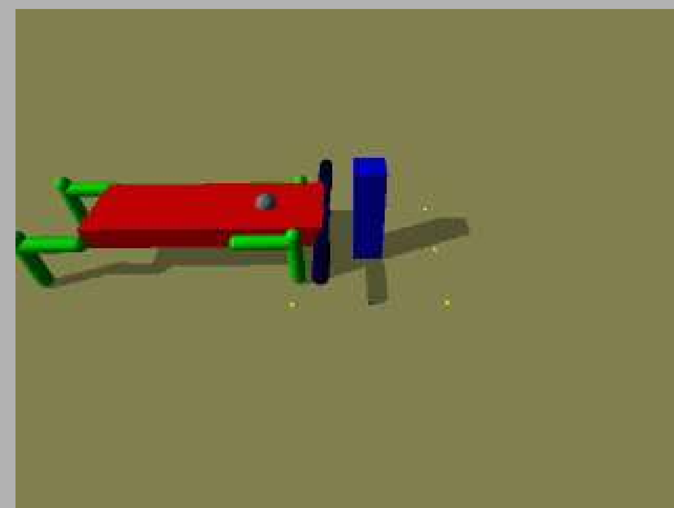
Sensor values

S1	S2
0	0
0.1	0.2
0.4	0.8
…	

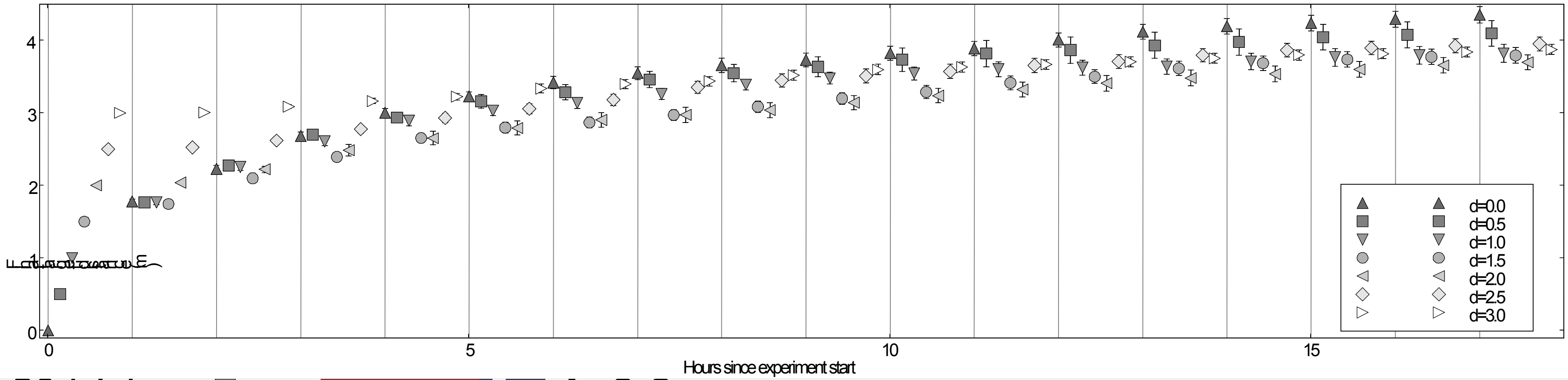


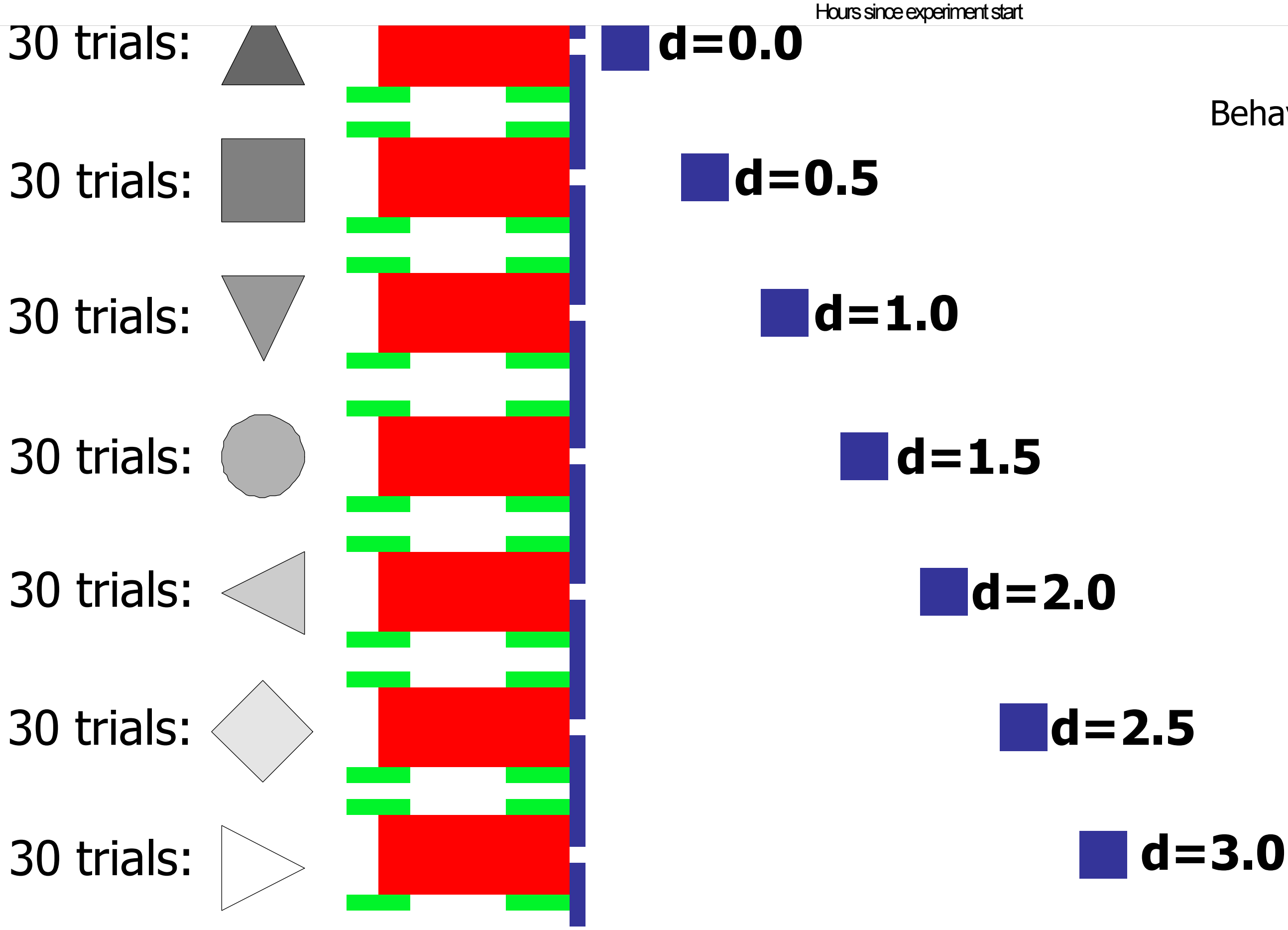
Morphology

MorphEngine

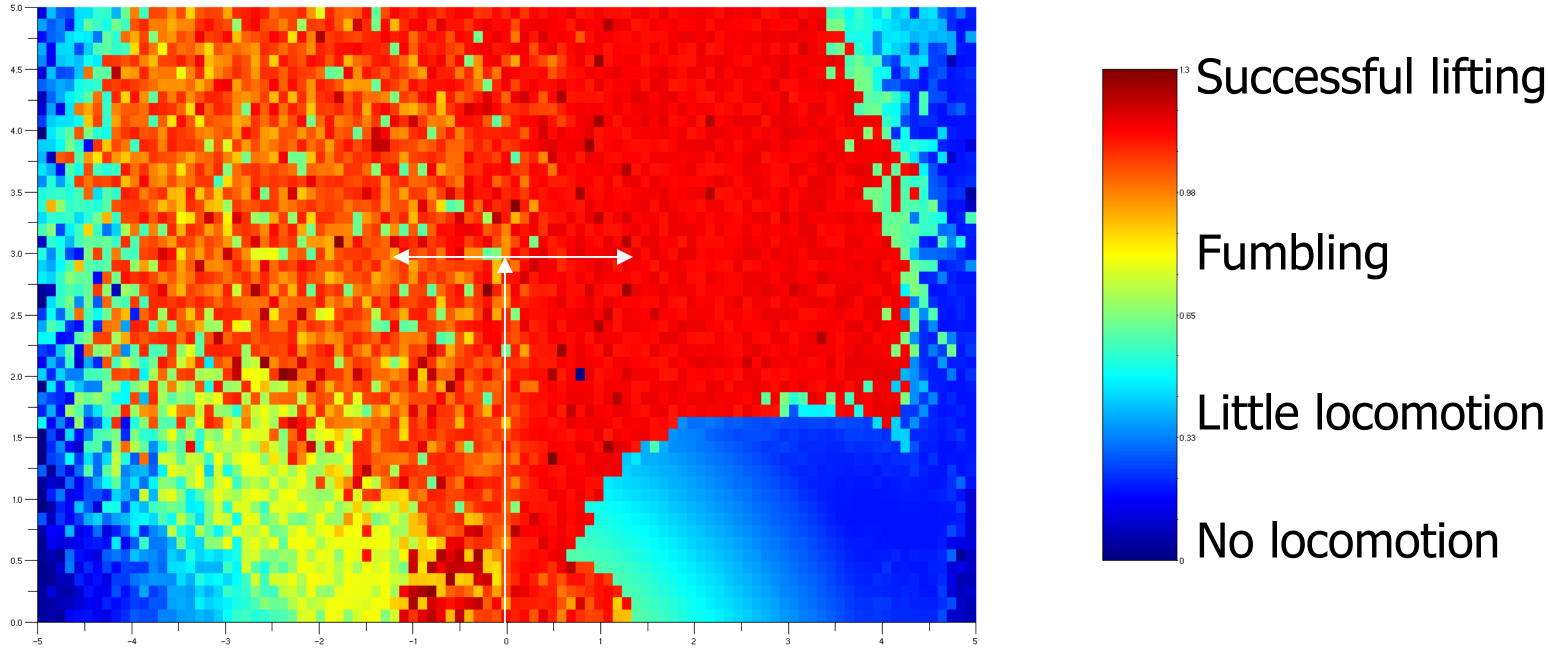


Bongard, J. (2008)
Behavior Chaining: Incremental Behavioral Integration for Evolutionary Robotics, *ALife XI*, to appear.





Bongard, J. (2008)
Behavior Chaining: Incremental Behavioral Integration for Evolutionary Robotics,
ALife XI, to appear.



Courtesy of Josh Auerbach

The Tool Set

MorphEngine
MorphEngine `Unplugged'
Physical Simulation

Example Investigations

Evolving Robot Morphologies and Controllers Together
Evolving Self-Models
Evolving Coupled, Nonlinear Models
Evolving Robots Capable of Multiple Behaviors

Conclusions

Proximate and Ultimate Mechanisms of Cognition
Summary

Nikolaas Tinbergen (1907–1988)

Four levels of description: not mutually exclusive

Proximate mechanisms:

1. **Causation (Mechanism):** what are the stimuli that elicit the response, and how has it been modified by recent learning?

2. **Development (Ontogeny):** how does the behaviour change with age?

Ultimate mechanisms:

3. **Evolution (Phylogeny):** how does the behaviour compare with similar behaviour in related species, and how might it have arisen through the process of phylogeny?

4. **Function (Adaptation):** how does the behavior impact on the animal's chances of survival and reproduction?

Proximate mechanisms:

1. Causation (Mechanism): how is stimuli combined in the **central nervous system**?

2. Development (Ontogeny): how does the **central nervous system** develop during growth?

Ultimate mechanisms:

3. Evolution (Phylogeny): how did **central neural structure** in the current population evolve from distributed neural structures in ancestral populations?

4. Function (Adaptation): how does **centralized neural structure** impact the **robot’s** chances of survival and reproduction? (integration of sensor information?)

Proximate mechanisms:

1. Causation (Mechanism): what are the stimuli that trigger **competition between processes in the brain**? What form do these processes take?

2. Development (Ontogeny): How do **competitive processes** multiply during growth? Do some processes eventually 'win'?

Ultimate mechanisms:

3. Evolution (Phylogeny): how did **competitive neural processes** arise from less, or lack of competitive processes in ancestral populations?

4. Function (Adaptation): how do **competitive neural processes** affect the animal's chances of survival and reproduction?

Proximate mechanisms:

1. Causation (Mechanism): what are the stimuli that trigger **the learning of a new behavior**? What form do these processes take?

2. Development (Ontogeny): How are **behaviors gradually integrated** into a robot's exhibit behavioral repertoire?

Ultimate mechanisms:

3. Evolution (Phylogeny): how do **the mechanisms that allow for gradual behavior integration** arise from ancestral robots capable of exhibiting only one behavior?

4. Function (Adaptation): Obvious

The Tool Set

MorphEngine → www.cs.uvm.edu/~jbongard/2008_Telluride

MorphEngine `Unplugged' → Software Robots with Hardware Brains

Physical Simulation

Example Investigations

Evolving Robot Morphologies and Controllers Together → CNS?

Evolving Self-Models → Competitive Processes in the Brain?

Evolving Coupled, Nonlinear Models

Evolving Robots Capable of Multiple Behaviors → Gradual Behavior Integration?

Conclusions

Proximate and Ultimate Mechanisms of Cognition

Summary

Take Home Message

To truly understand cognition, we must pursue two lines of attack:

understand biological systems by replicating them in hard/software

cause analogues of these systems to evolve in artificial systems.