Design Automation

MAE 2250

Team not functioning?

1. Ask the TA to call a counselling meeting
   – Include head TAs Jeff/Katie
2. Assign clear goals and responsibilities
   – Deliverables and dates for each member
3. Send written summary
   – Email meeting minutes

- *Repeat*

Don’t internalize. Act swiftly.

Phases

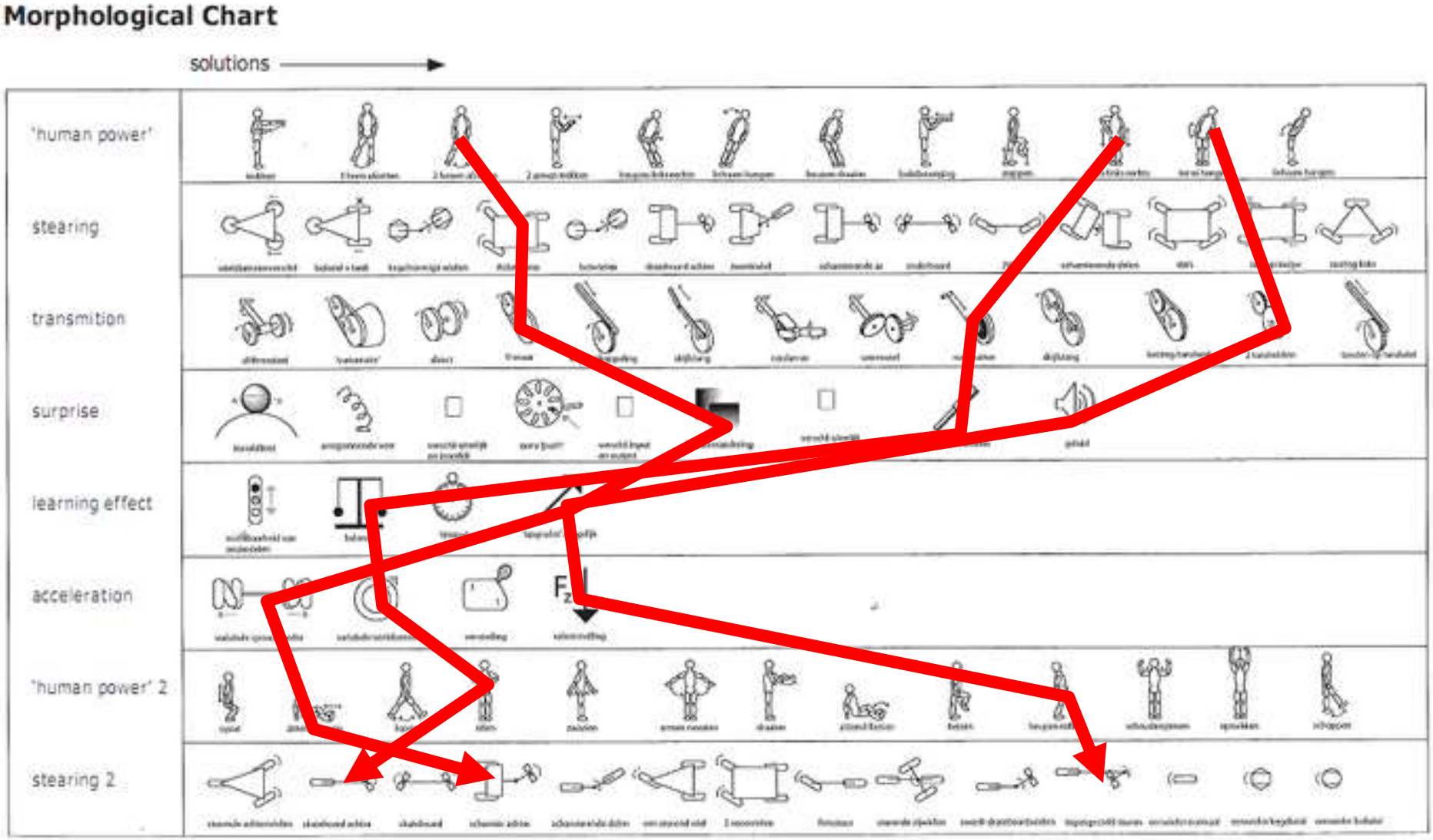
- Phase 0: Planning
- Phase 1: Conceptual design
- Phase 2: System design
- Phase 3: Detail design
- Phase 4: Testing and refinement
- Phase 5: Production ramp-up

Iterate

What is design?

- Design is a search process
  - In the space of all possible designs
  - Humans search this space using ‘creativity’
  - Design automation uses search algorithms (AI)

Design Space



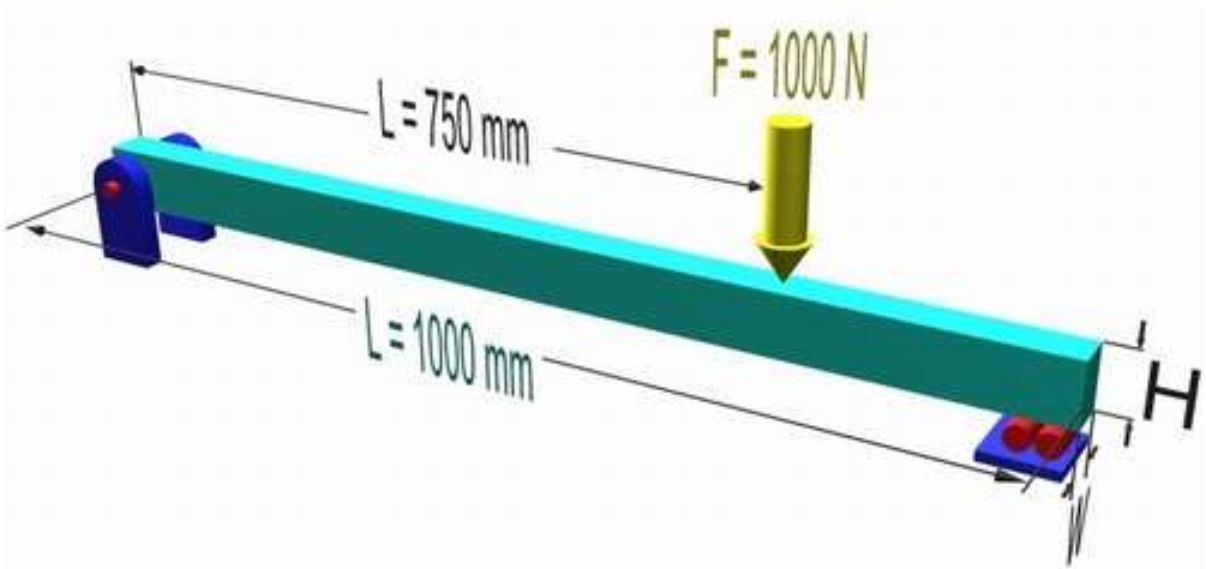
12^8 ~ 430M

Computational Design Space

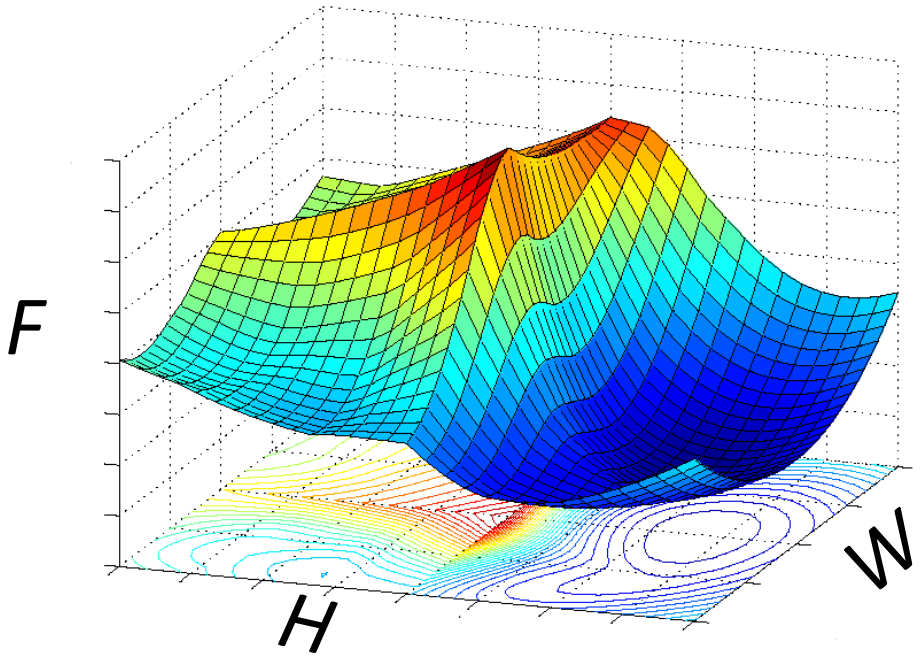
- What does this space look like?
  - Closed, parametric [detailed design]
    - Continuous, discrete
  - Open-ended [conceptual design]

Parametric design space

- You do the conceptual design and describe it using some free parameters
- Define a function which you which to optimize (maximize, minimize)
- Define constraints
- Let the computer find the setting of parameters that reach the optimum



Find H and W that minimize Deflection and Mass



Brute-force search

1. Scan all possible values at some interval
2. Evaluate function at each point
3. Keep the best value found

Why would this not work?

Random search

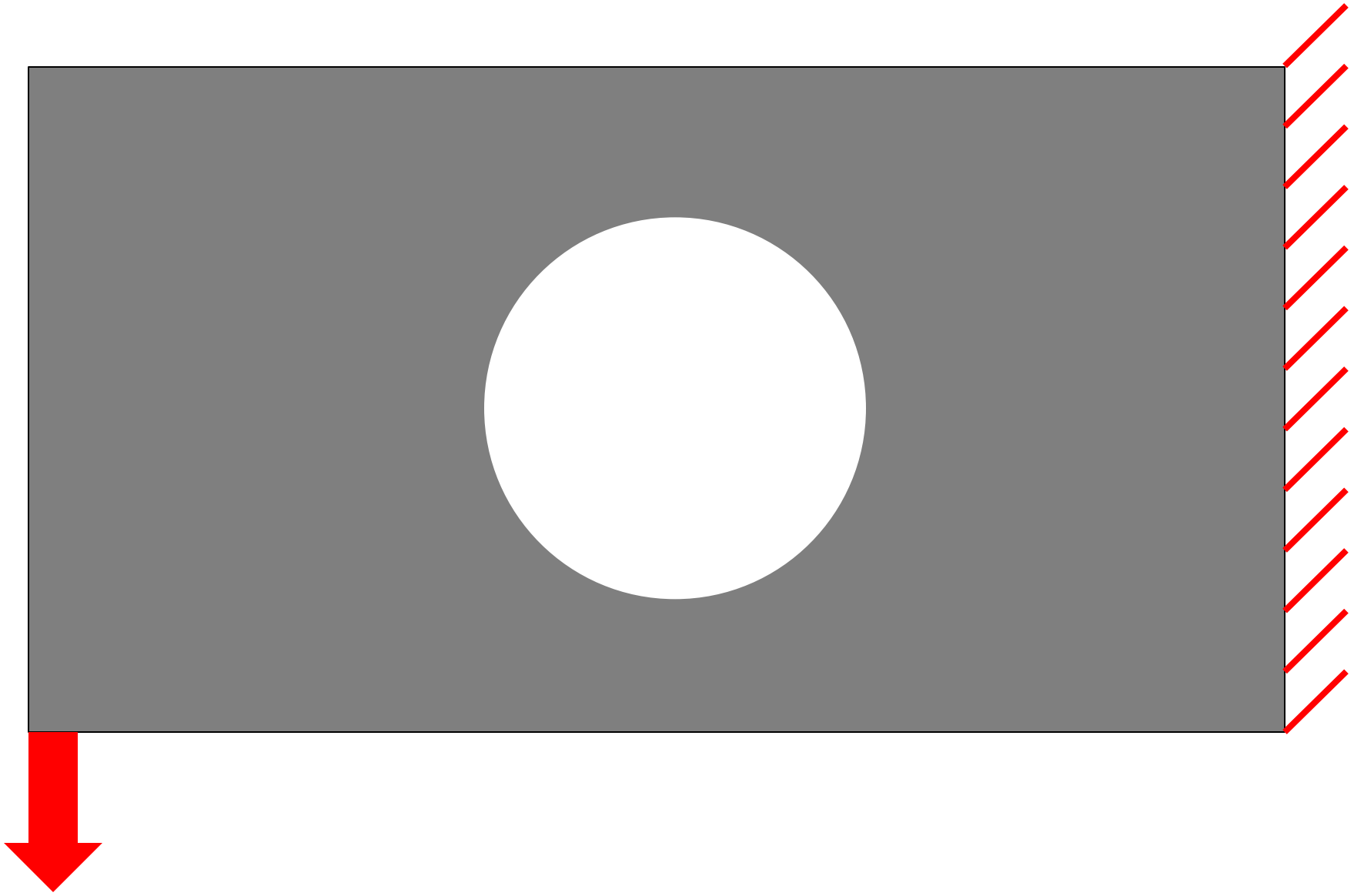
1. Start at some initial guess
2. Randomly chose another set of values for parameters
3. Check if they improve on your current values, if yes, keep, else discard
4. Repeat until no improvement made for a while

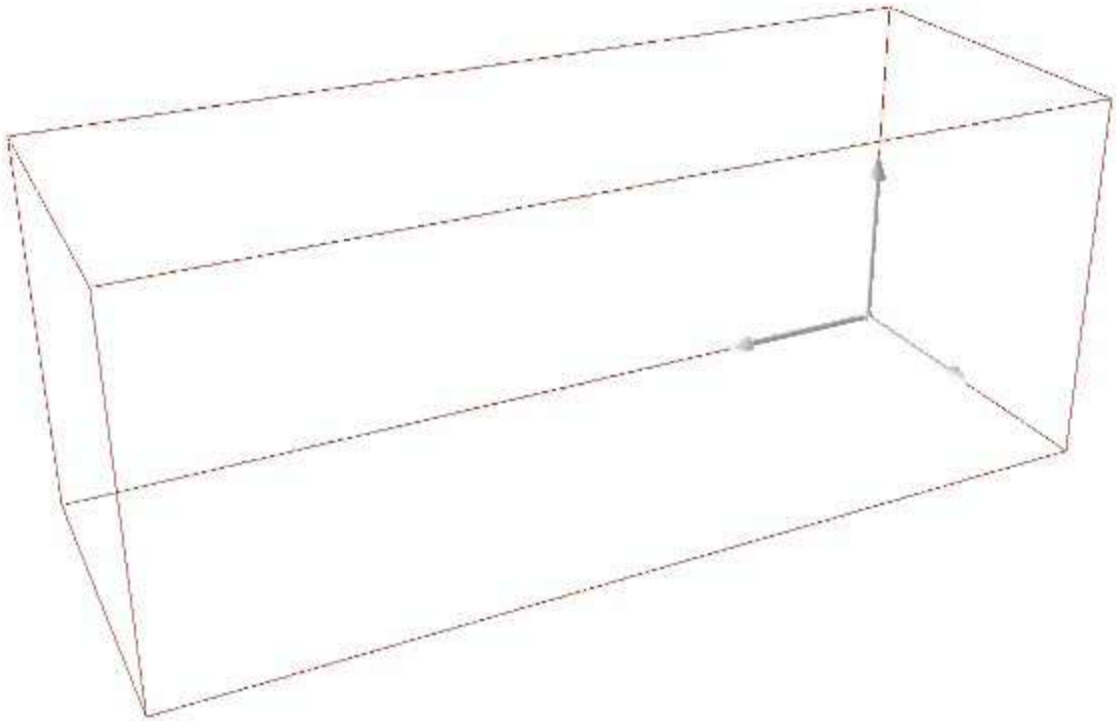
Why would this not work?

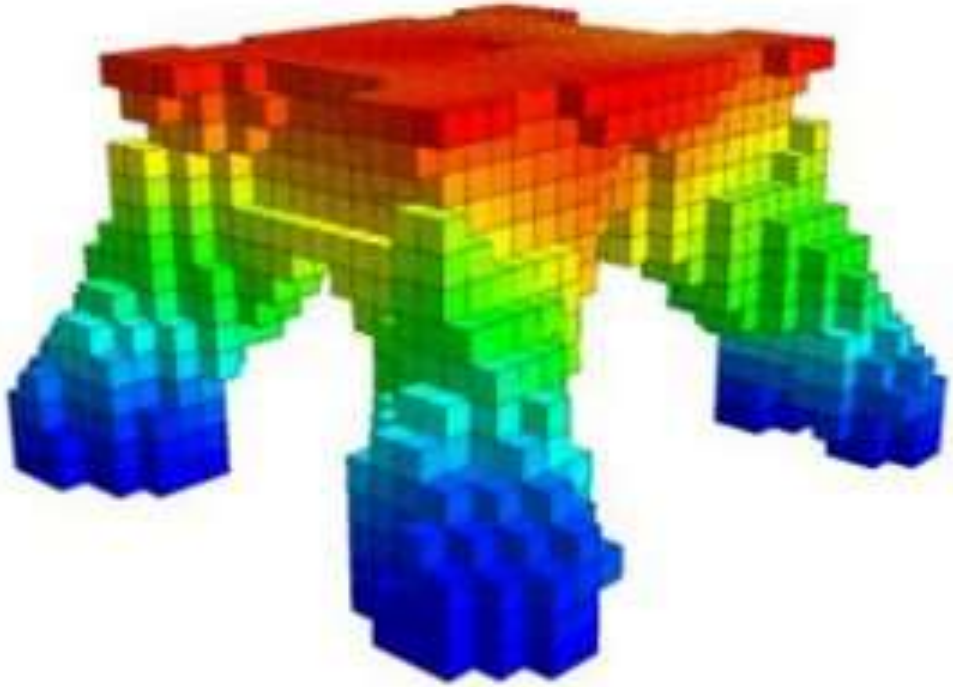
Gradient optimization Algorithm

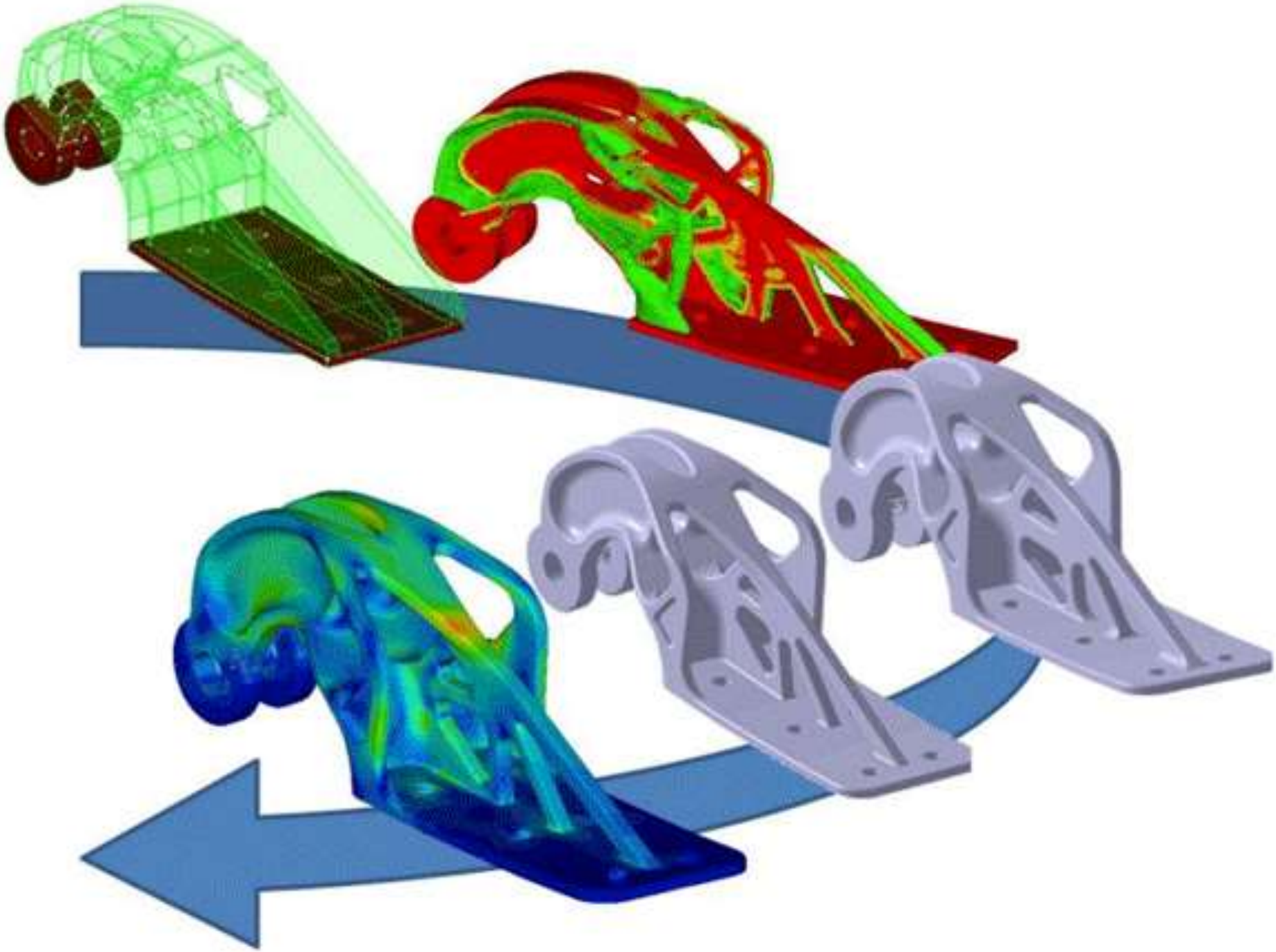
1. Start at some initial guess
2. Check derivatives in all dimensions
3. Move a little in the steepest direction
4. Repeat until all derivatives zero

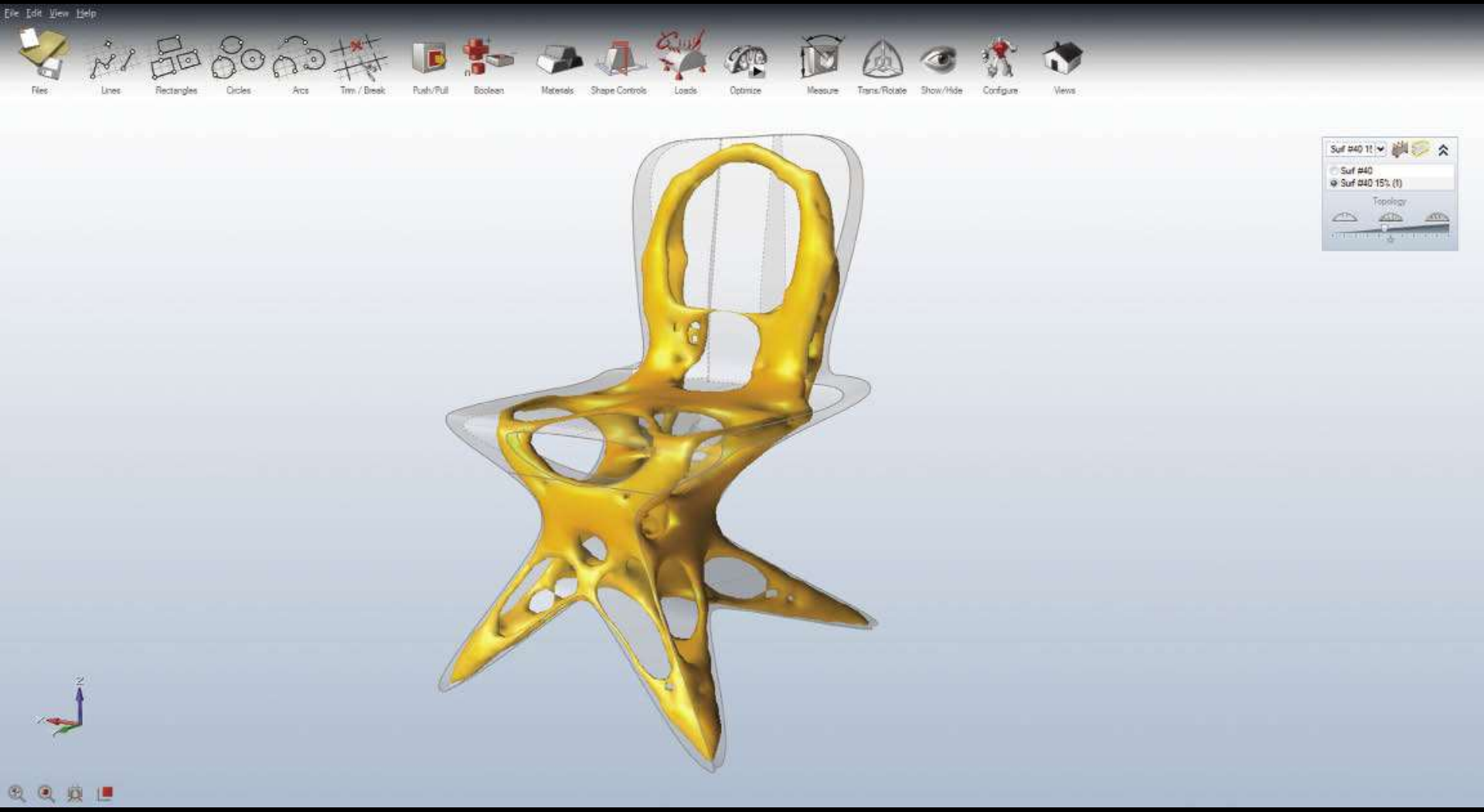
How can this algorithm fail?

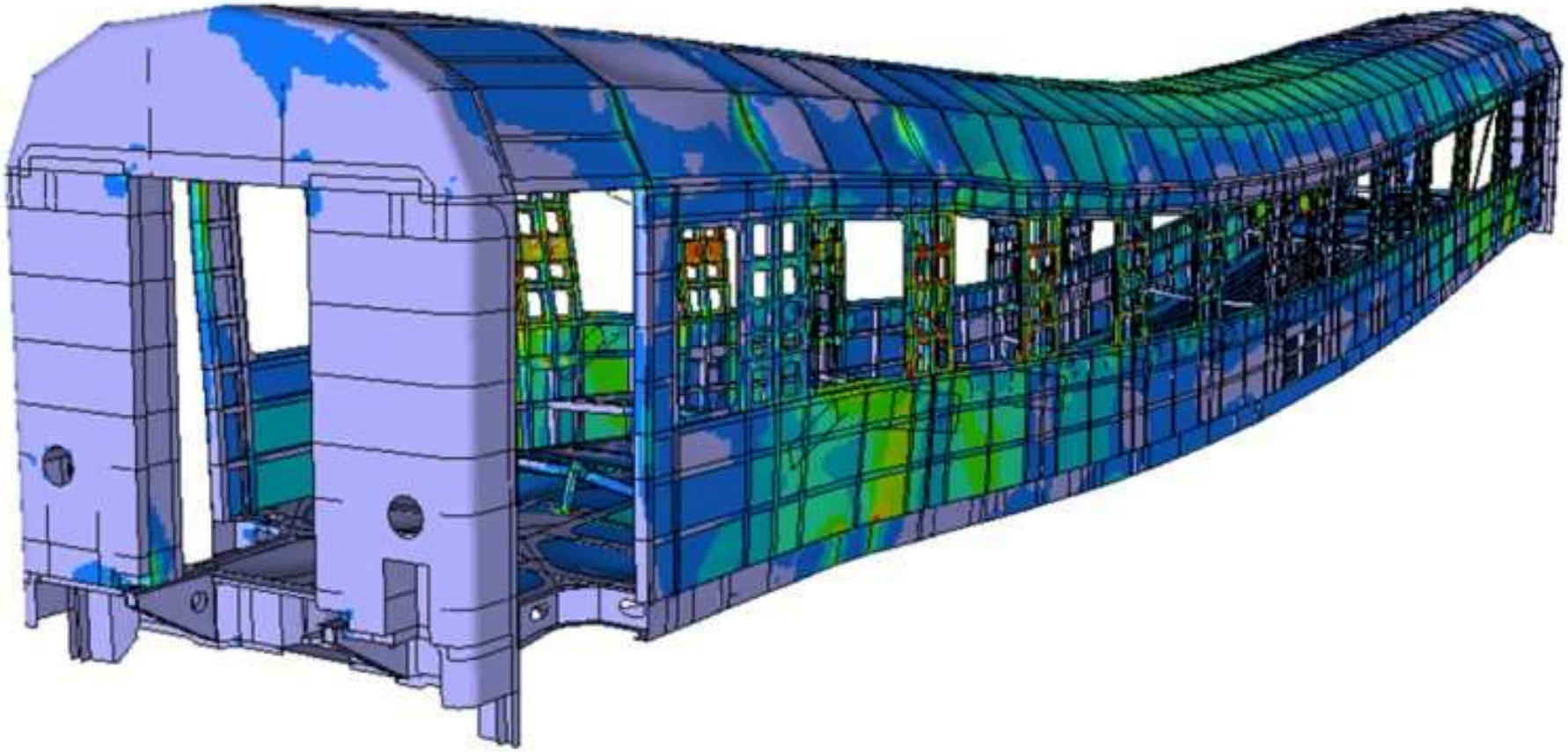




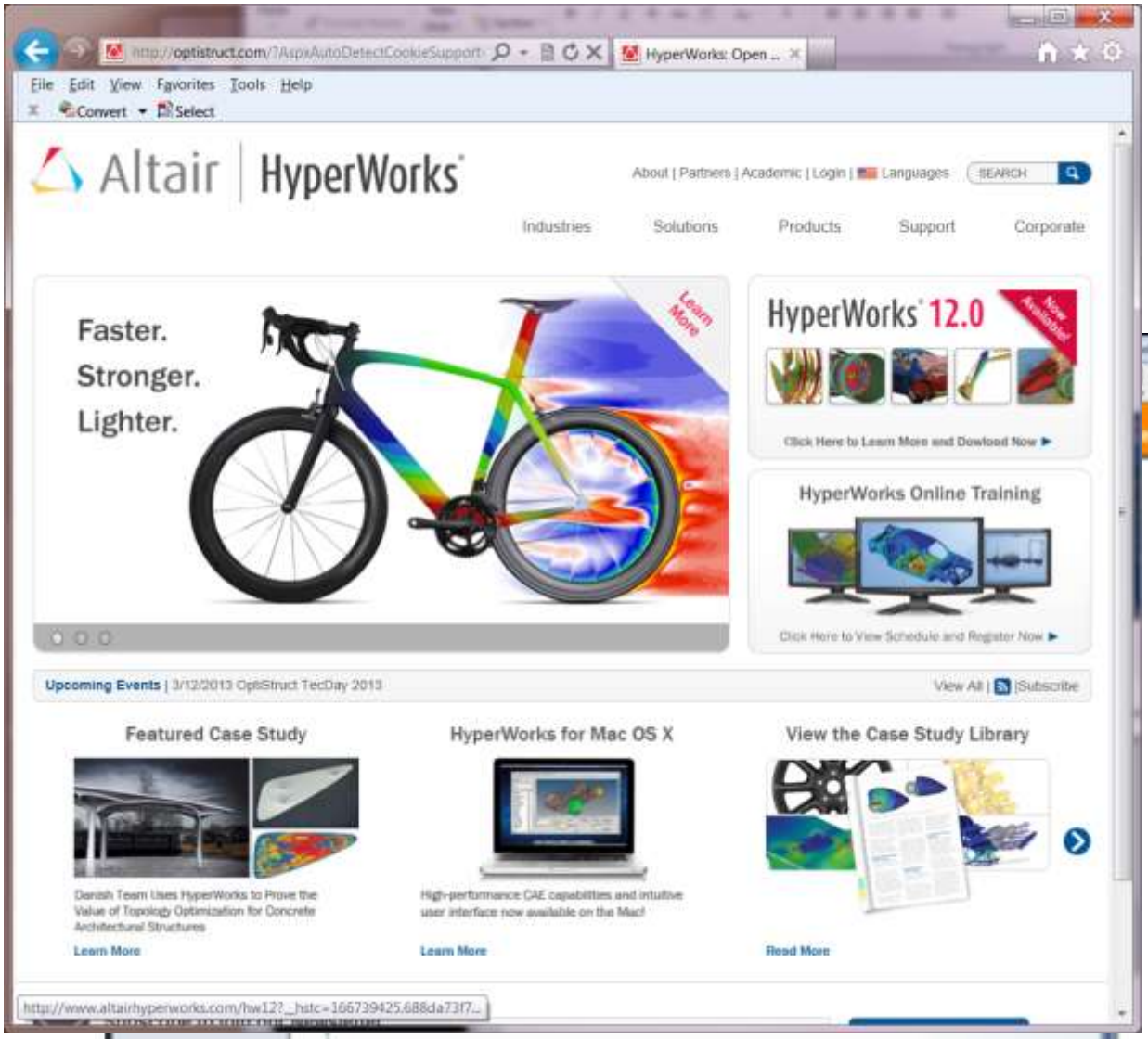


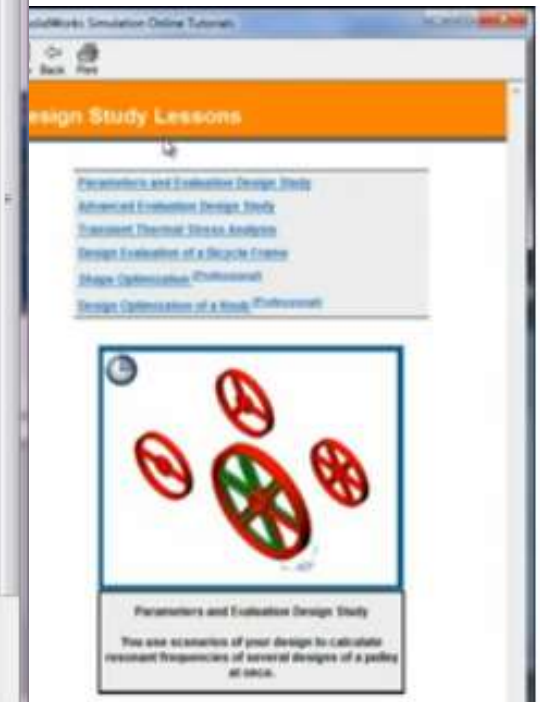


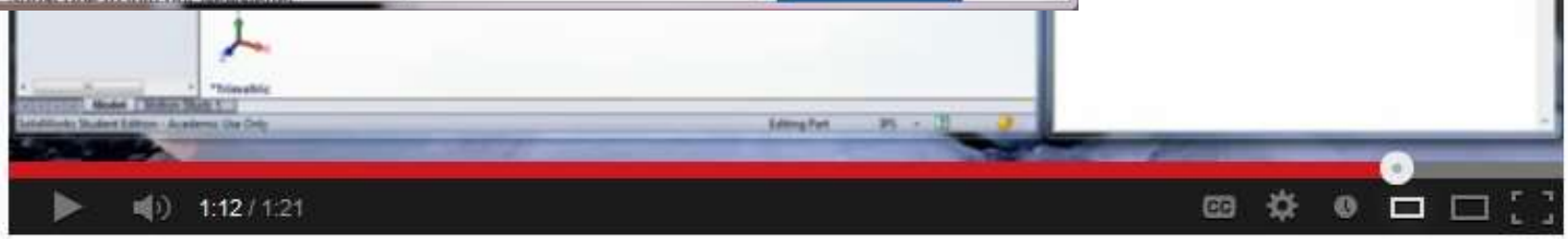

File Edit View Help
Files
Lines
Rectangles
Circles
Arcs
Trim / Break
Push/Pull
Boolean
Materials
Shape Controls
Loads
Optimize
Measure
Trans/Rotate
Show/Hide
Configure
Views
Suf #40
Suf #40 15% (1)
Topology

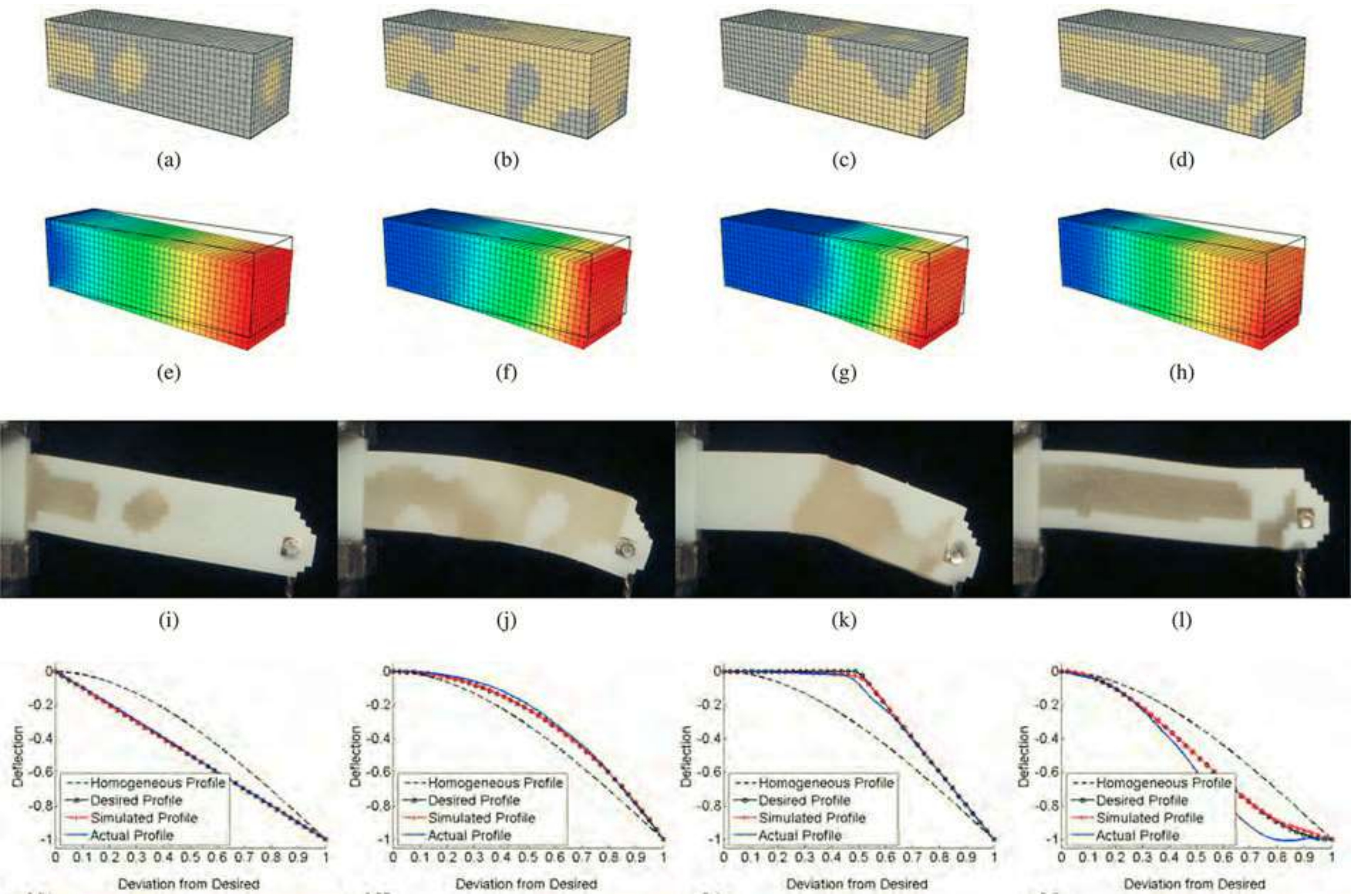


Vibration optimization


http://optistruct.com/?AspxAutoDetectCookieSupport=
HyperWorks: Open ...
File Edit View Favorites Tools Help
Convert Select
Altair | HyperWorks
About | Partners | Academic | Login | Languages
SEARCH
Industries
Solutions
Products
Support
Corporate
Faster.
Stronger.
Lighter.
Learn More
HyperWorks 12.0
Now Available!
Click Here to Learn More and Dowload Now
HyperWorks Online Training
Click Here to View Schedule and Register Now
Upcoming Events | 3/12/2013 OptiStruct TecDay 2013
View All | Subscribe
Featured Case Study
Danish Team Uses HyperWorks to Prove the Value of Topology Optimization for Concrete Architectural Structures
Learn More
HyperWorks for Mac OS X
High-performance CAE capabilities and intuitive user interface now available on the Mac!
Learn More
View the Case Study Library
Read More
http://www.altairhyperworks.com/hw12?_hstc=166739425.688da73f7...


Design Study Lessons
Parameters and Evaluation Design Study


1:12 / 1:21



(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)
Homogeneous Profile
Desired Profile
Simulated Profile
Actual Profile
Deflection
Deviation from Desired
0
-0.2
-0.4
-0.6
-0.8
-1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

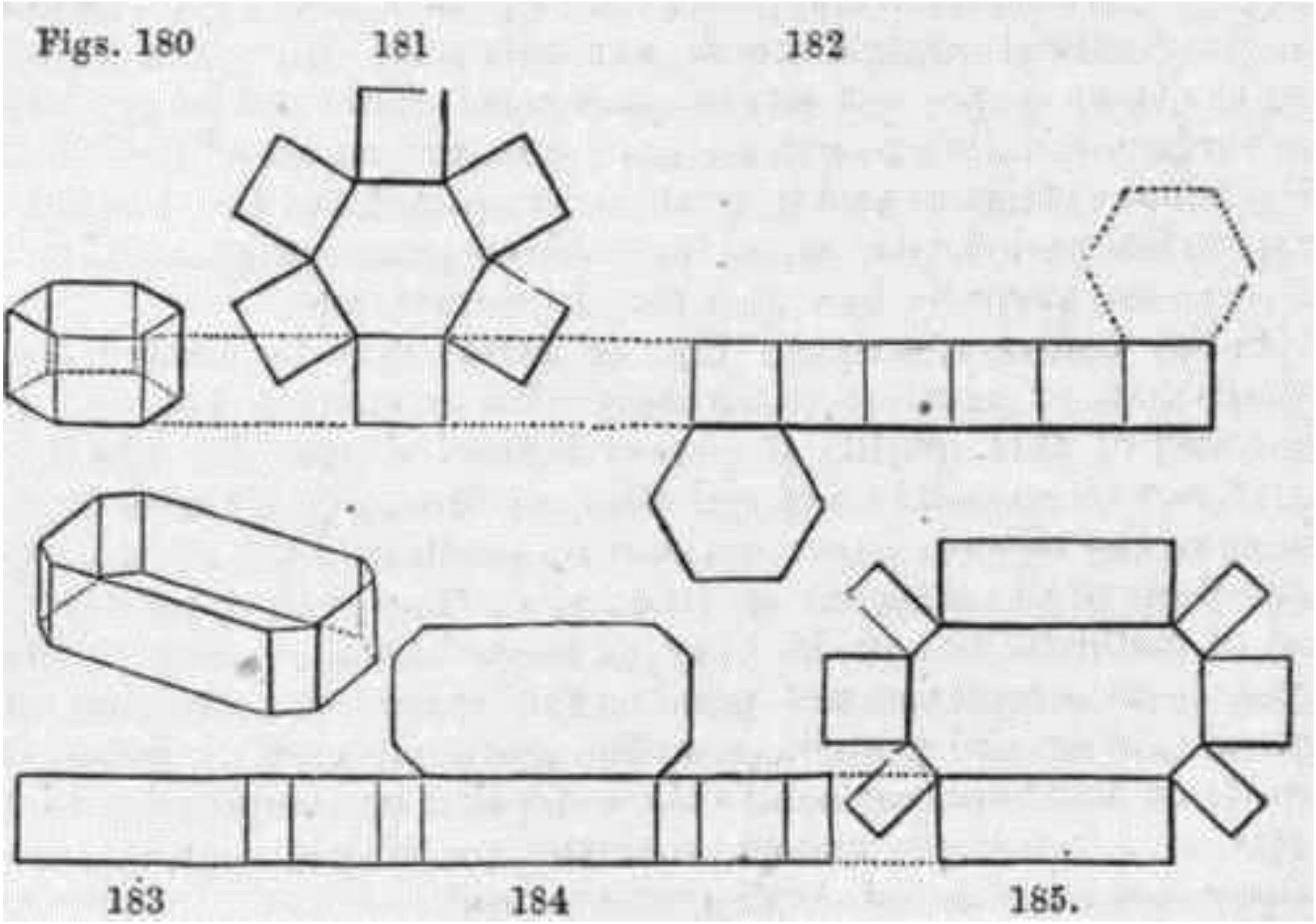
How can you use design automation in your project?

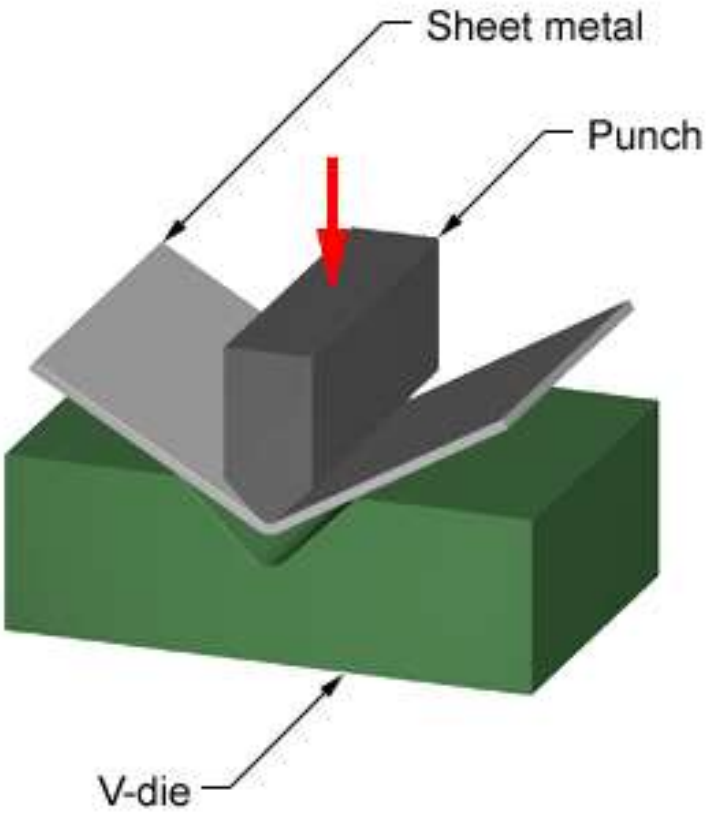
- Outline a problem that will be suitable
  - Quantitative objective(s)
  - Available simulation/analysis

Permutation design space

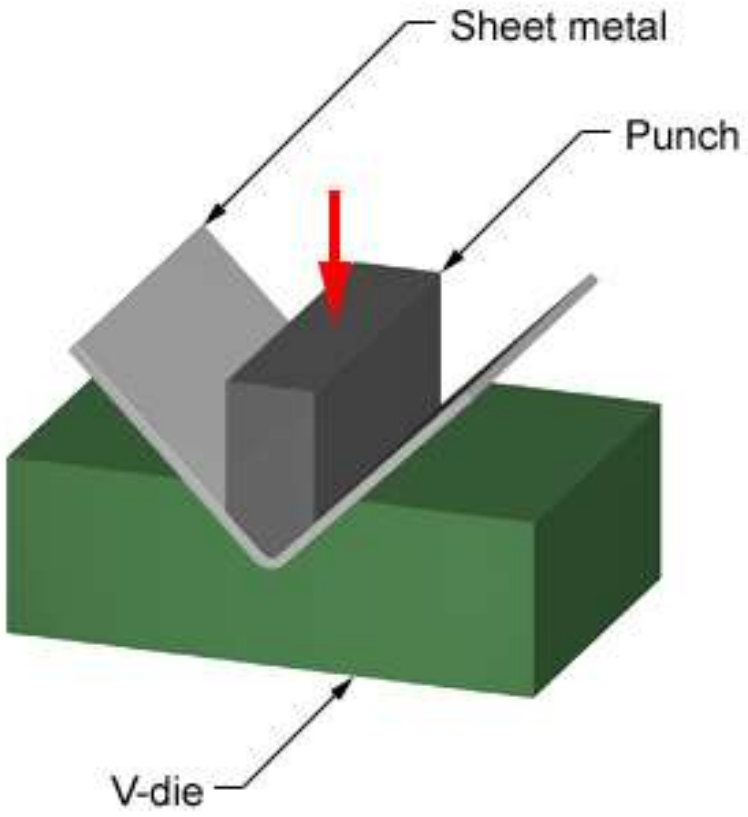
- The design space is discrete
- There is a finite number of possibilities
- Often characterized by a set of decisions

http://yorkshirelaser.co.uk/interface/fabrication

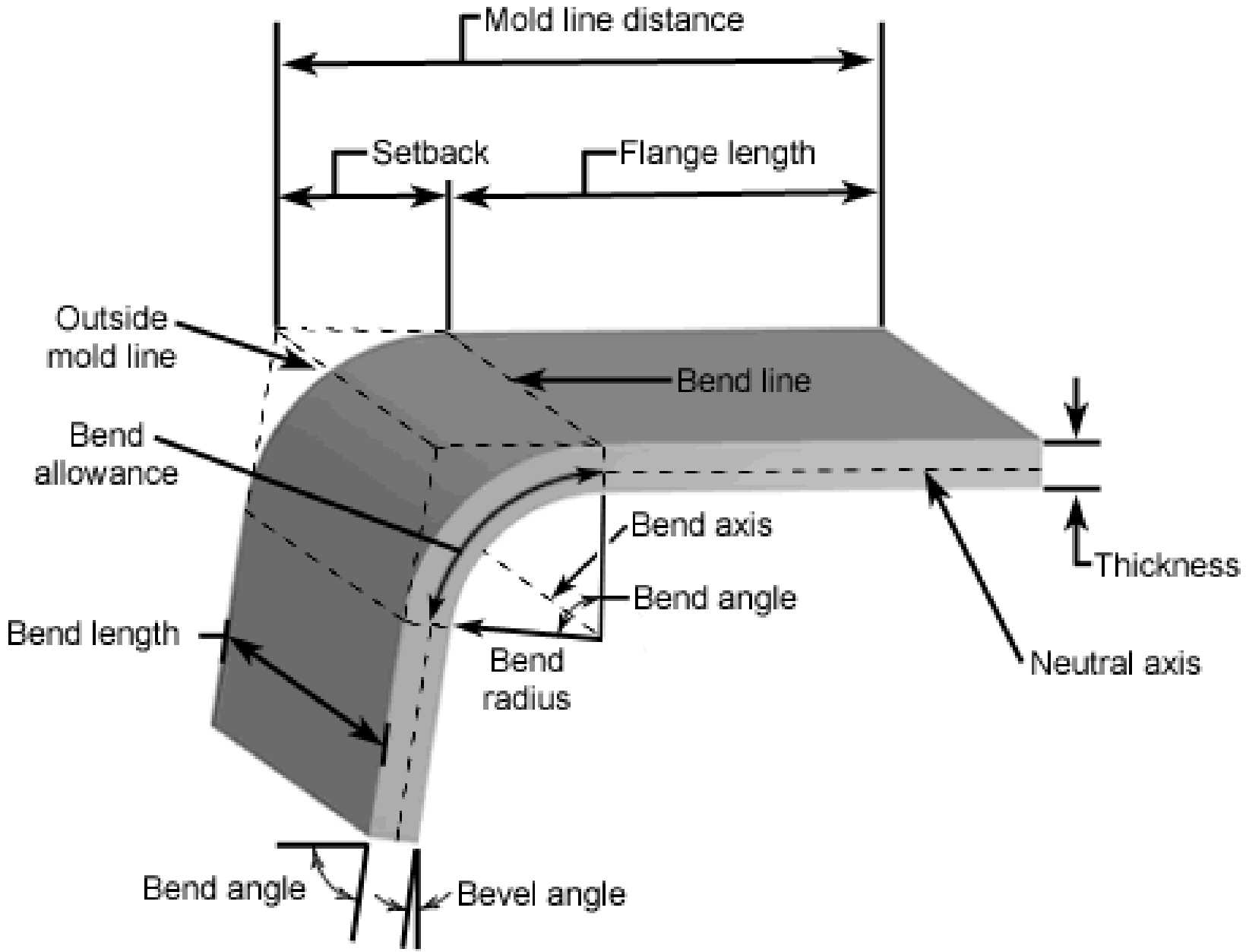


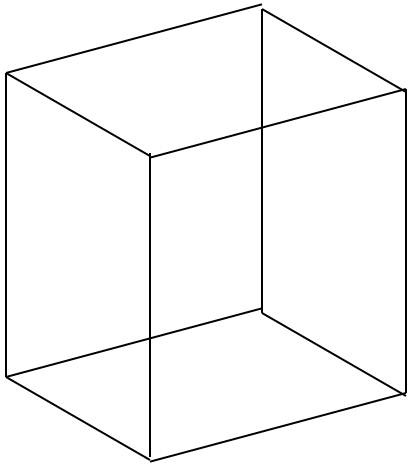


Air Bending

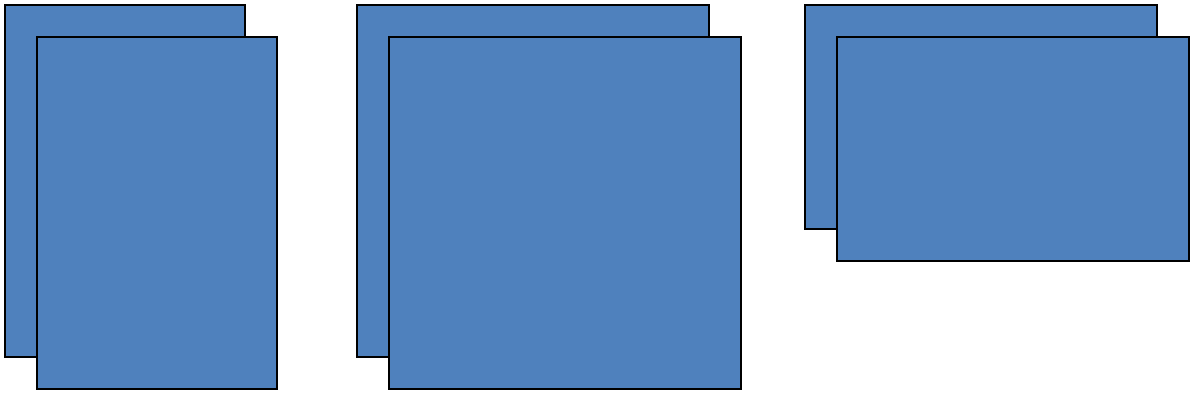


Bottoming

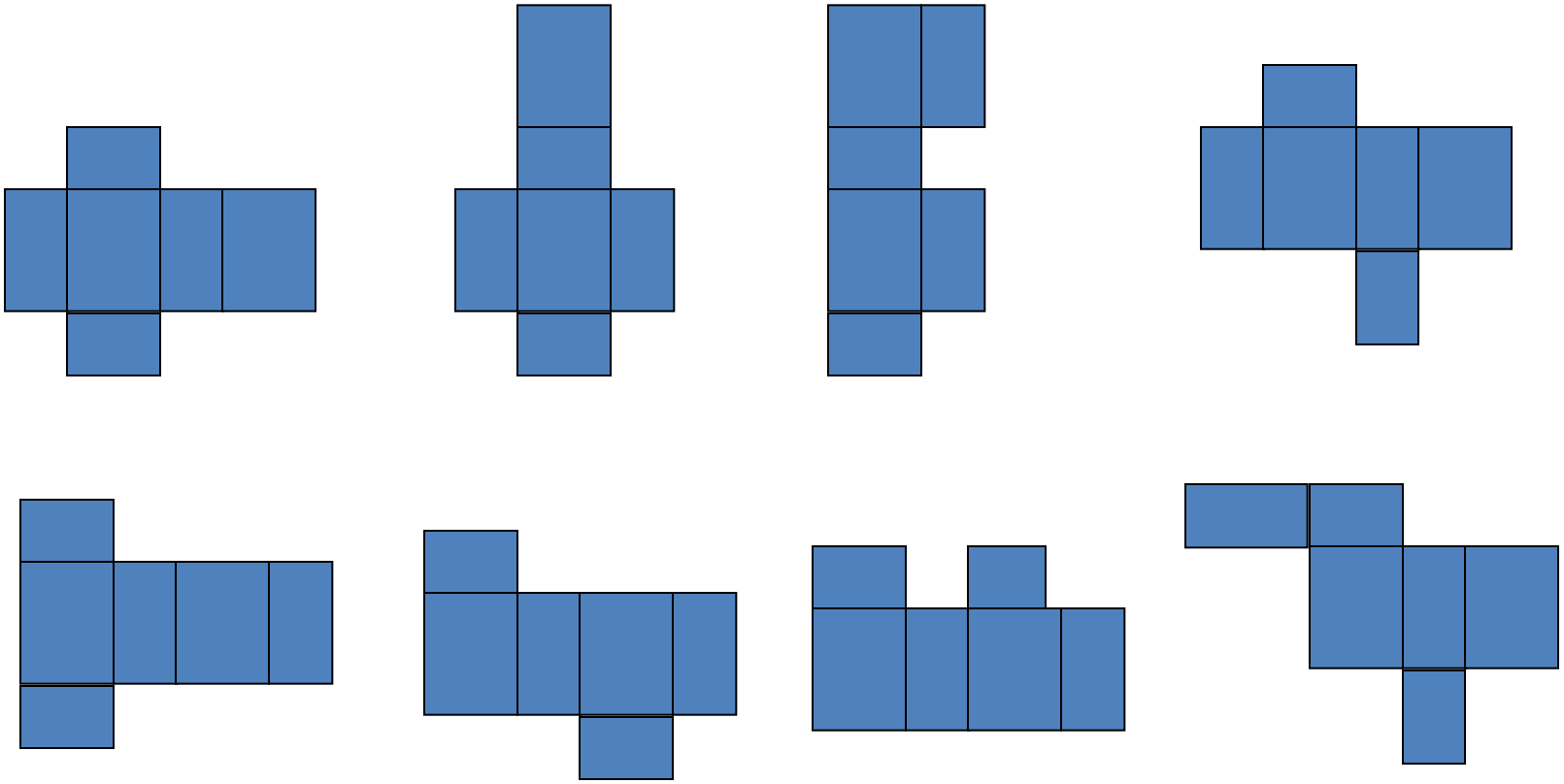

Mold line distance
Setback
Flange length
Outside mold line
Bend line
Bend allowance
Bend axis
Bend angle
Thickness
Bend length
Bend radius
Neutral axis
Bend angle
Bevel angle



Find unfolding pattern that minimizes total welding

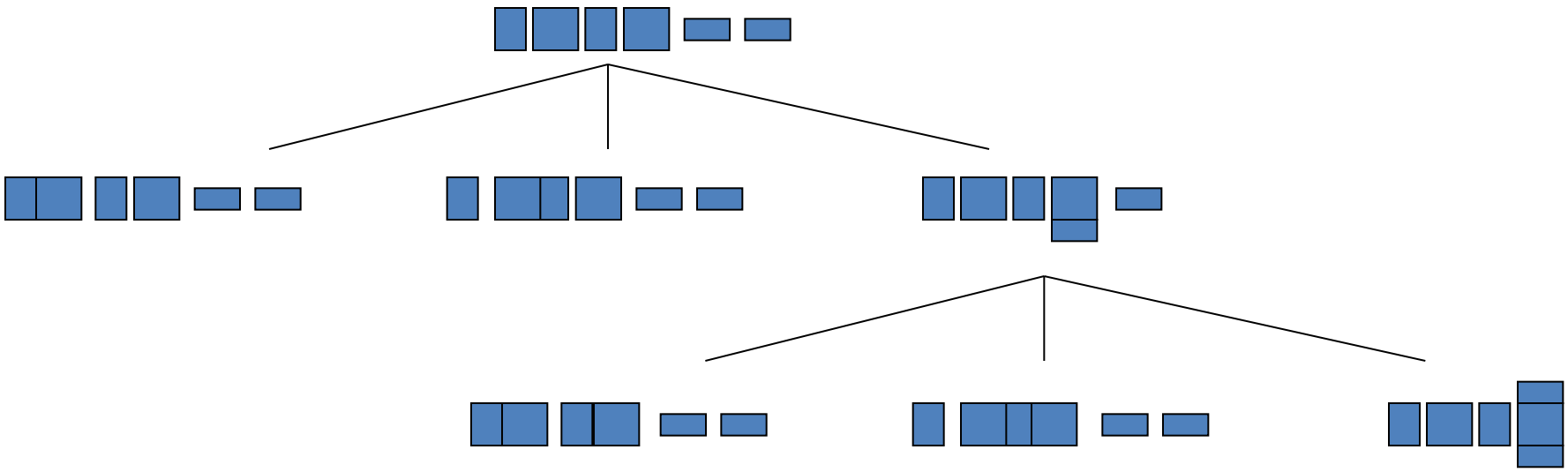


Permutation space



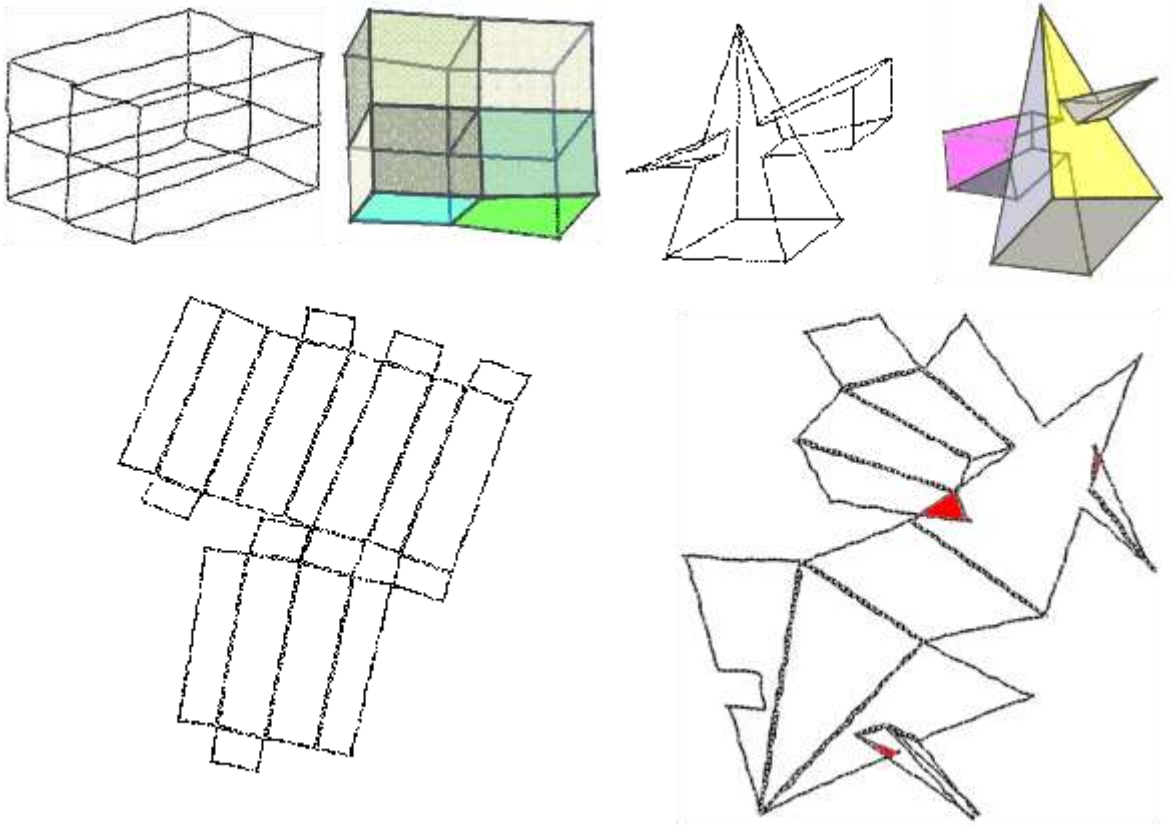
Lots more…

Searching the permutation space



- At each point Evaluate: Cost so far + optimistic remaining cost
- Explore most promising ‘leaf’ next
- This algorithm is called: A* (A-Star)

Finding optimal unfolding




©greeneyedmomma.com

Open-ended design

- Compose a set of given building blocks to meet a given design goal
- Example: Lego set
  - Known repertoire of blocks
  - Known interfaces
  - Known physics
  - How to put them together to achieve high level functionality?
  - Open-ended, unlimited complexity, exponentially intractable

Evolutionary Computation

- Study of self-organizing adaptive mechanisms based on Natural Selection, and applying them algorithmically to synthetic problems
  - Nonlinear optimization
  - Engineering Design
  - Models of natural systems (“Artificial Life”)
  - A Weak but General Method

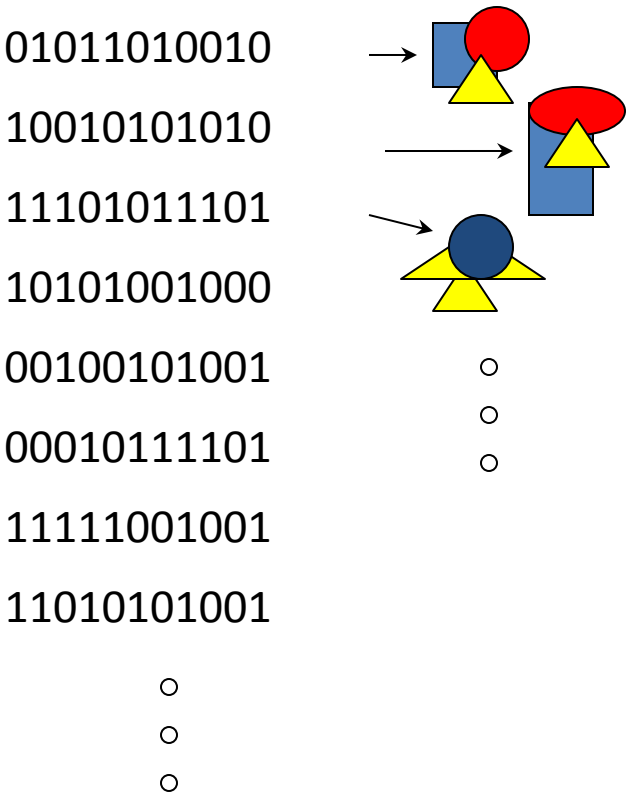
Alternative Evolutionary Systems

The study reveals that artifacts – like plant and animal life forms – can be arranged in continuous, chronological sequences. The existence of continuity implies that novel artifacts can only arise from antecedents artifacts – that new kinds of made things are never the pure creations of theory, ingenuity, or fancy…

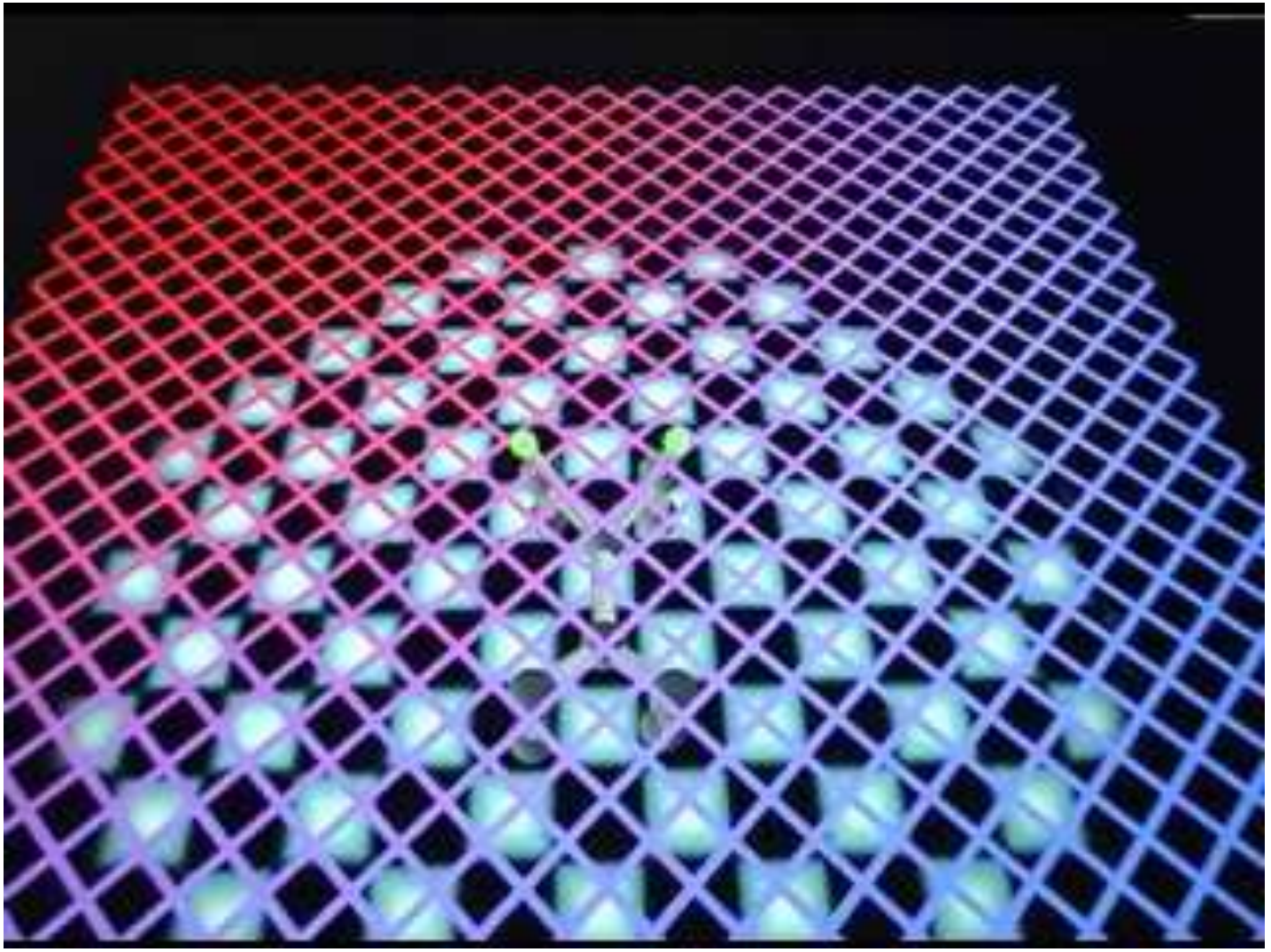
G. Basala (*The evolution of technology, 88*):

A basic evolutionary process

- Initialize a population of solutions
- Repeat
  - Selection (e.g. fitness proportionate)
  - Replication (e.g. duplication)
  - Variation (e.g. mutation)
- Until satisfactory solution found

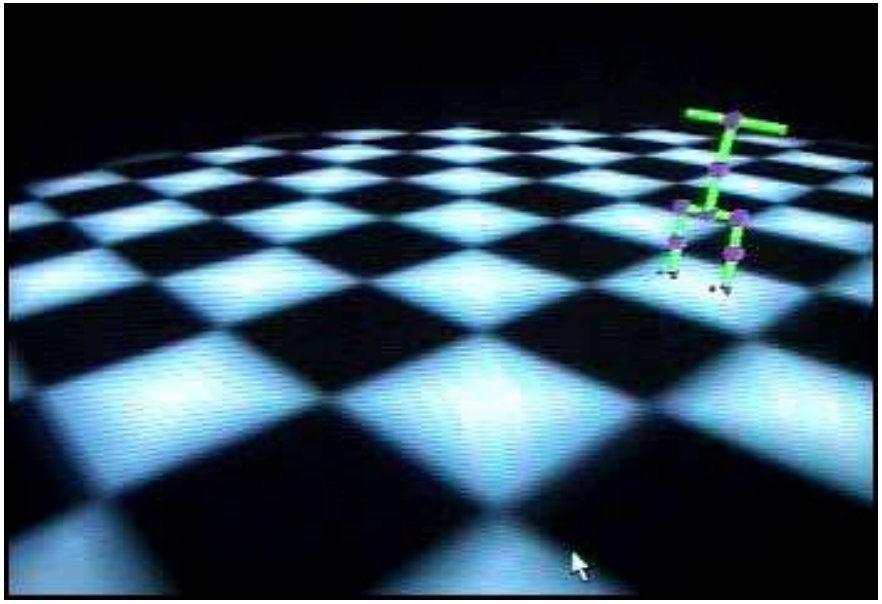


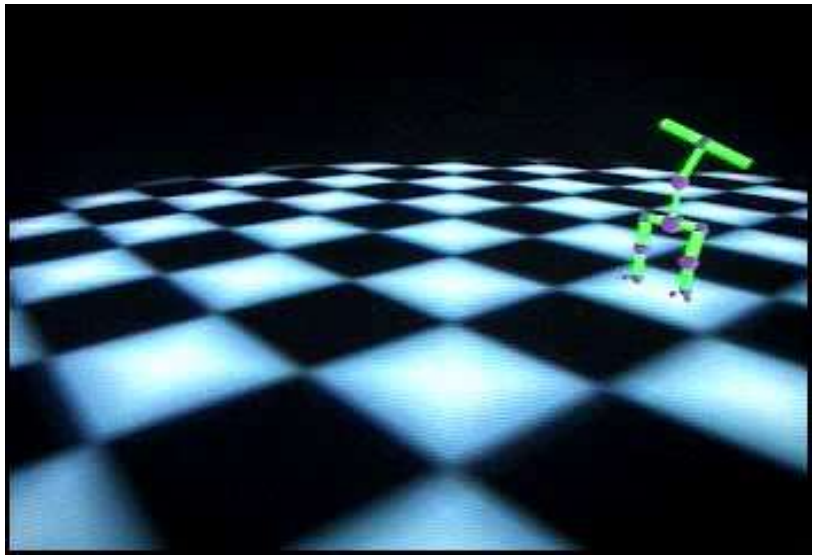
Evolving a gradient-Following brain



Joshua Bongard et al (2002)

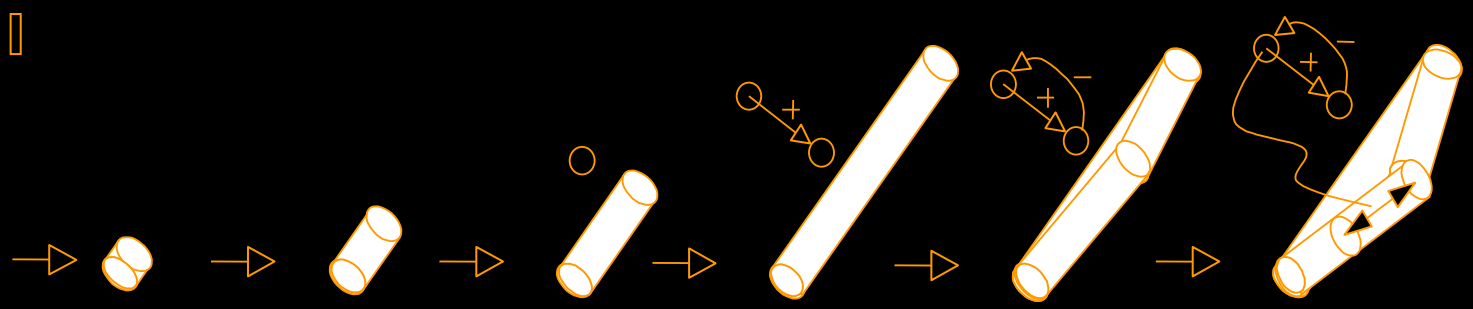
Evolving controllers

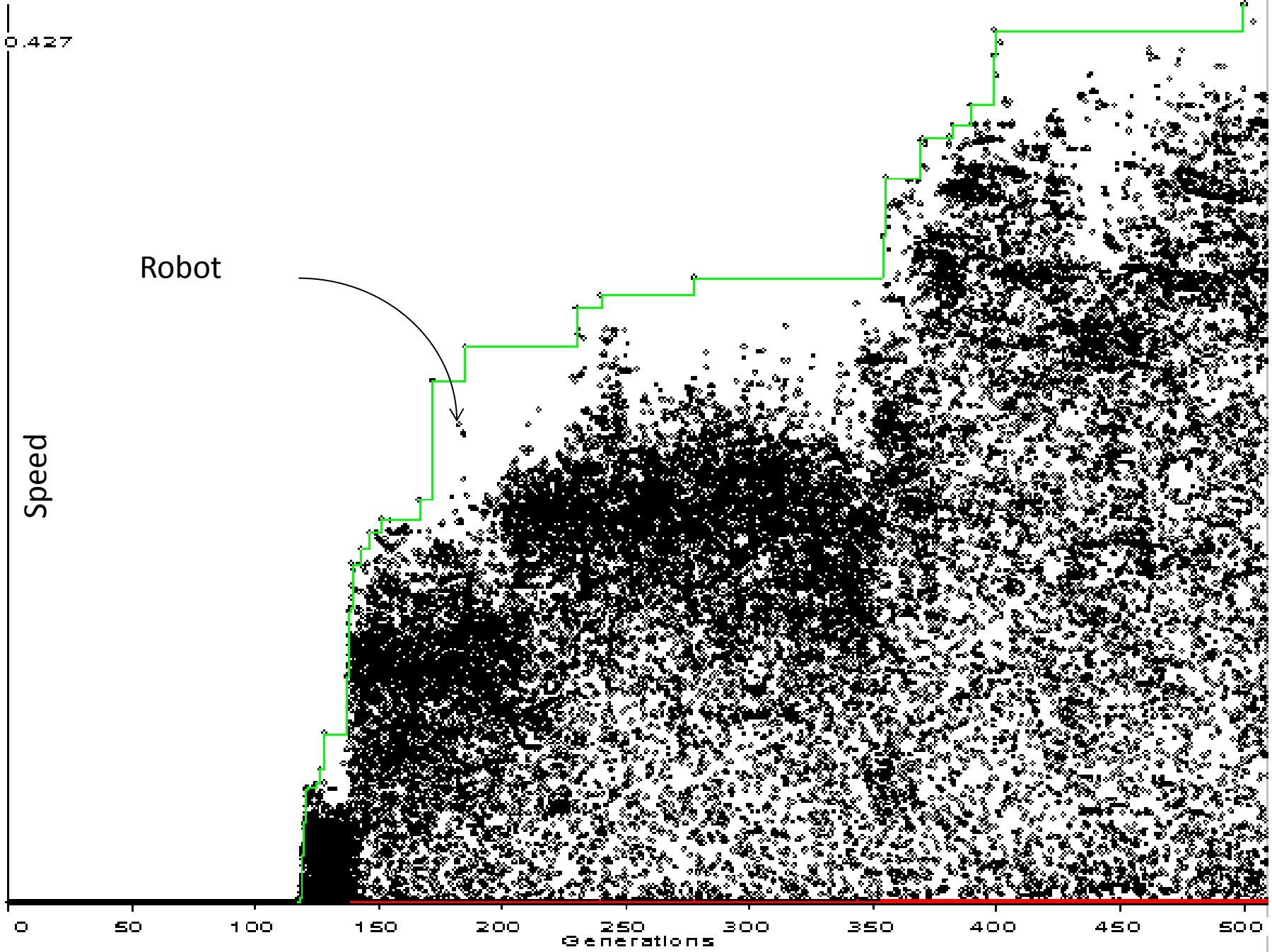


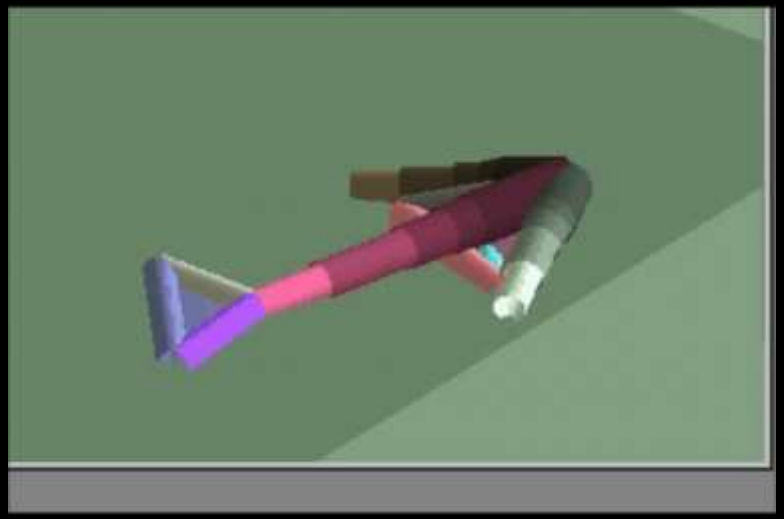


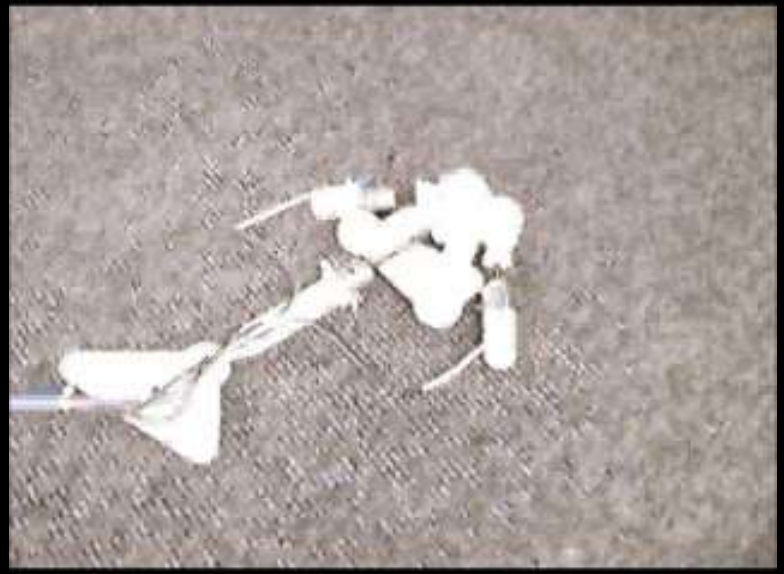
Joshua Bongard et al (2002)

Evolution

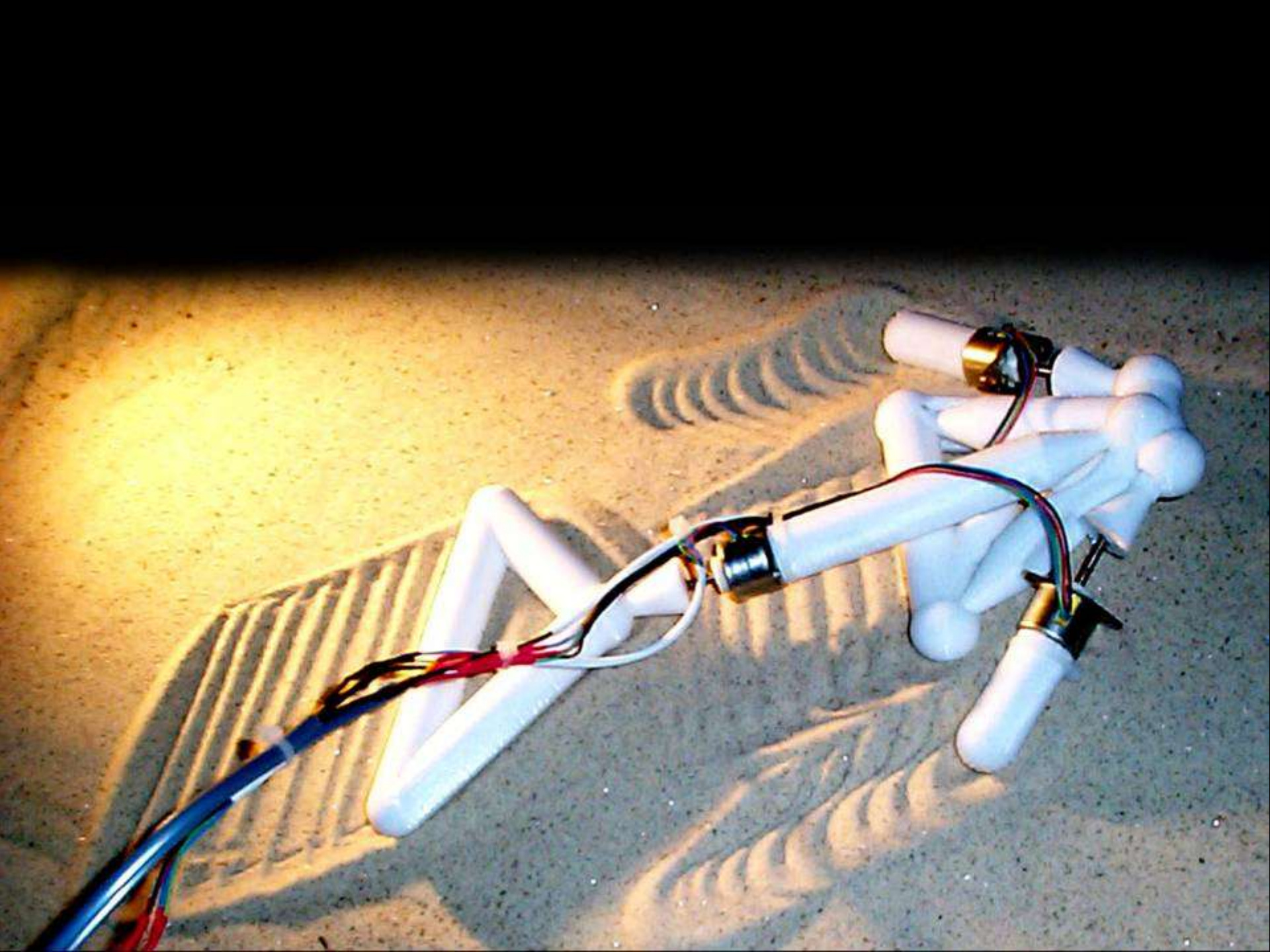


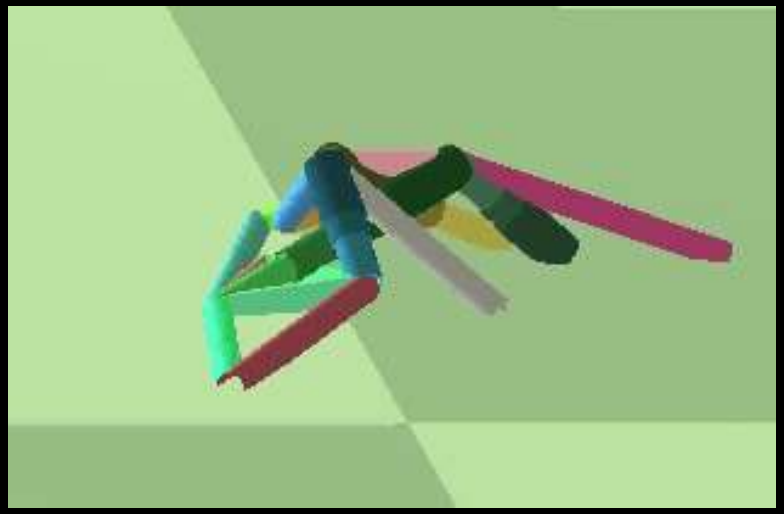

0.427
Robot
Speed
0
50
100
150
200
250
300
350
400
450
500
Generations

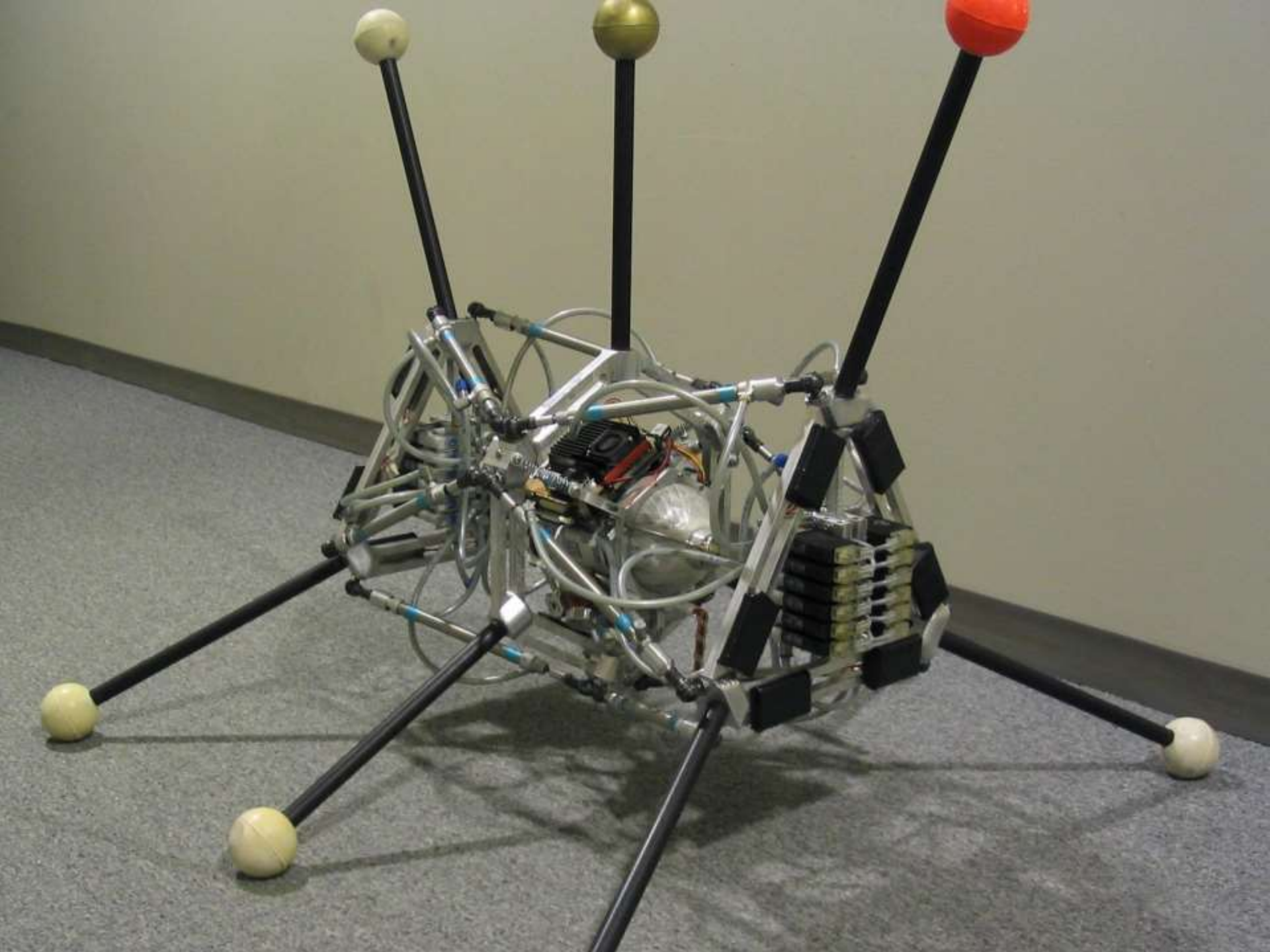


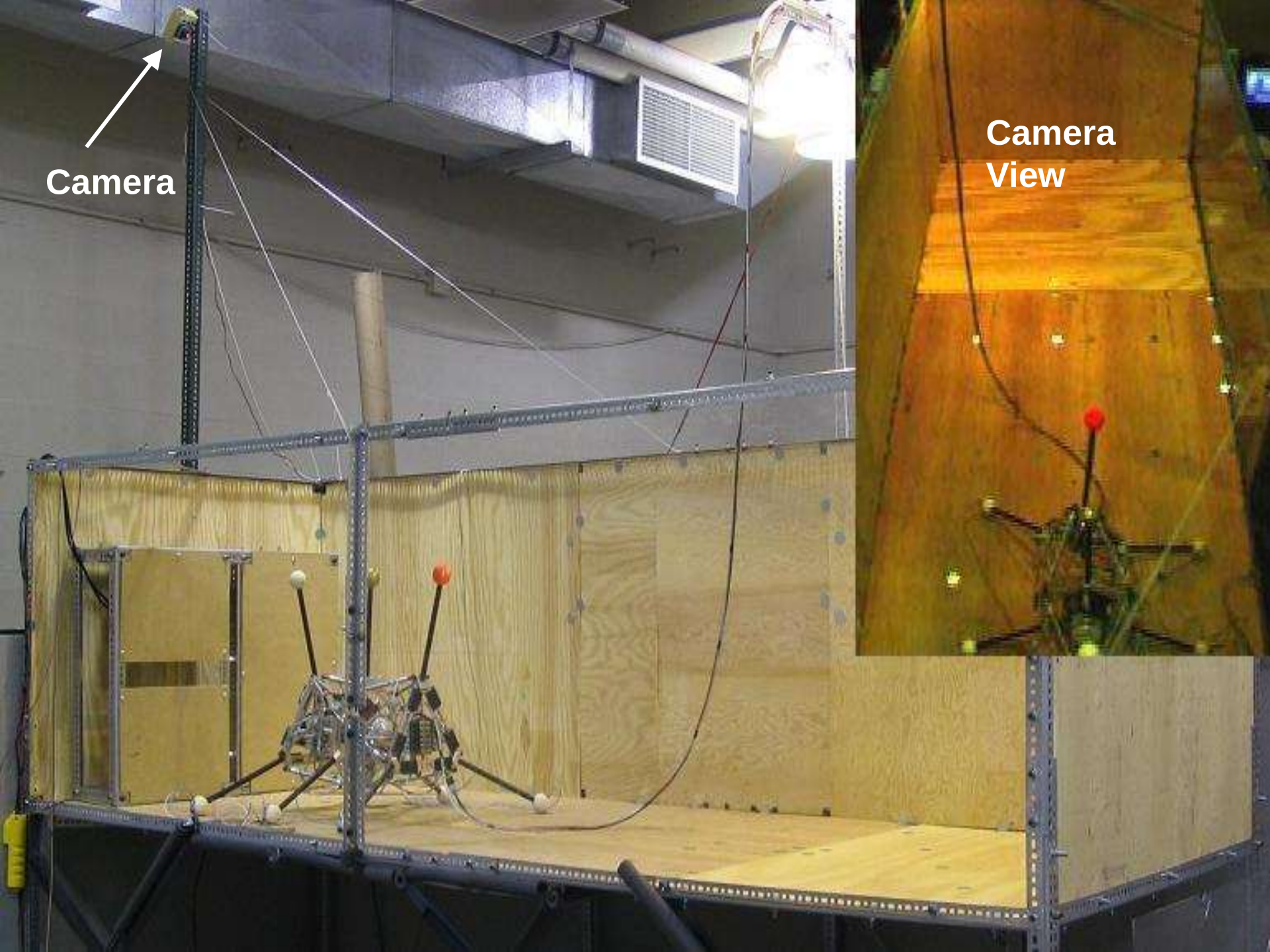


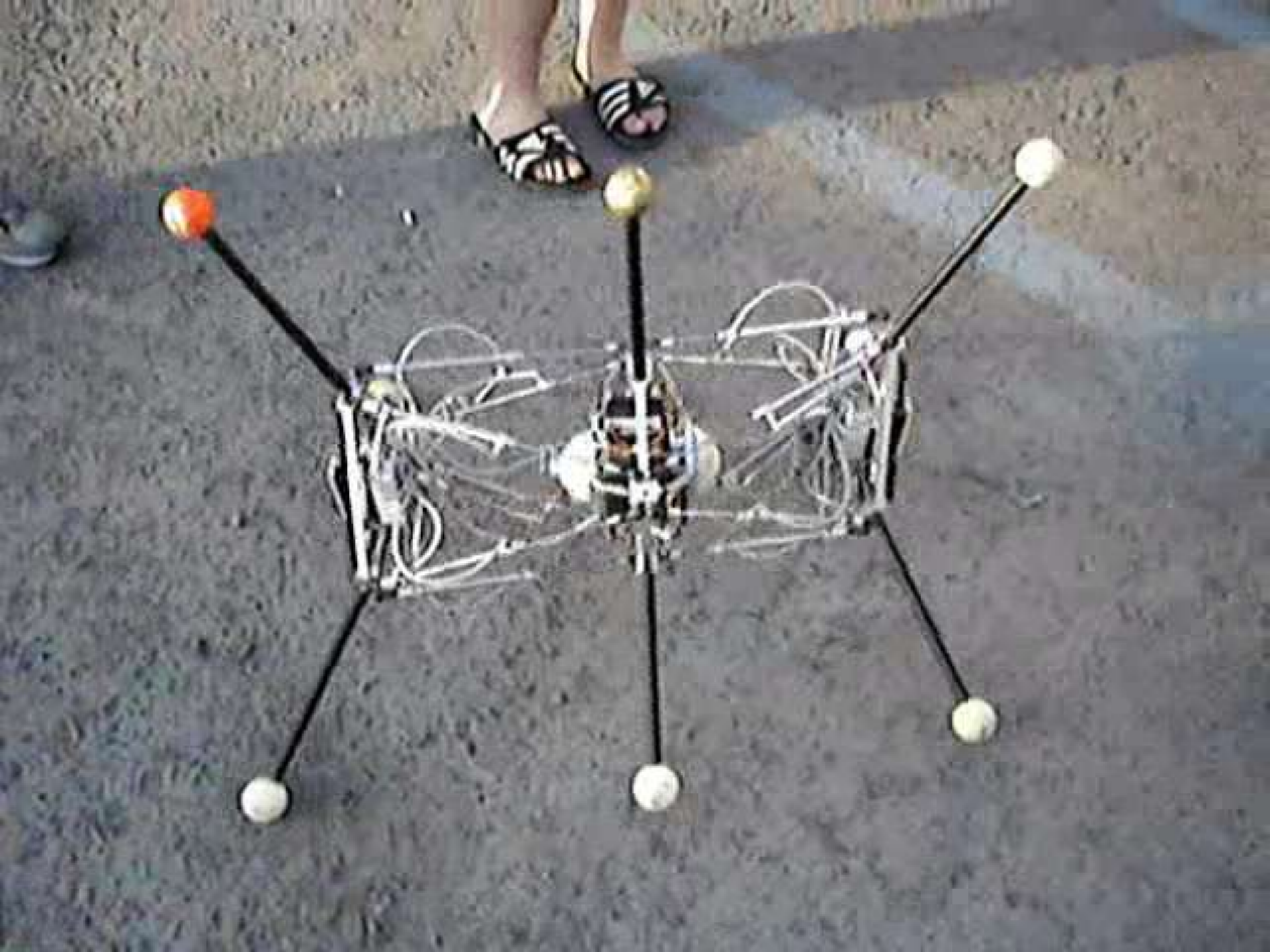
Lipson & Pollack, Nature 406, 2000

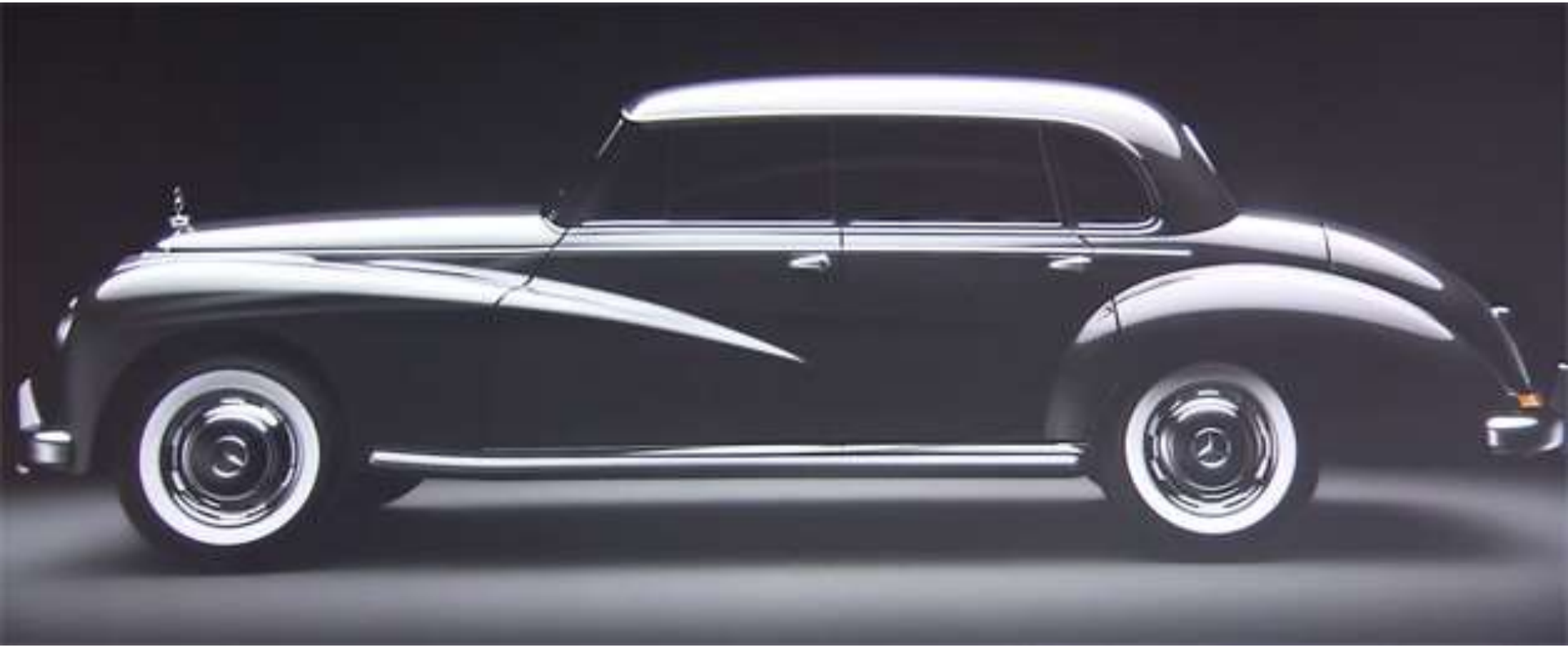


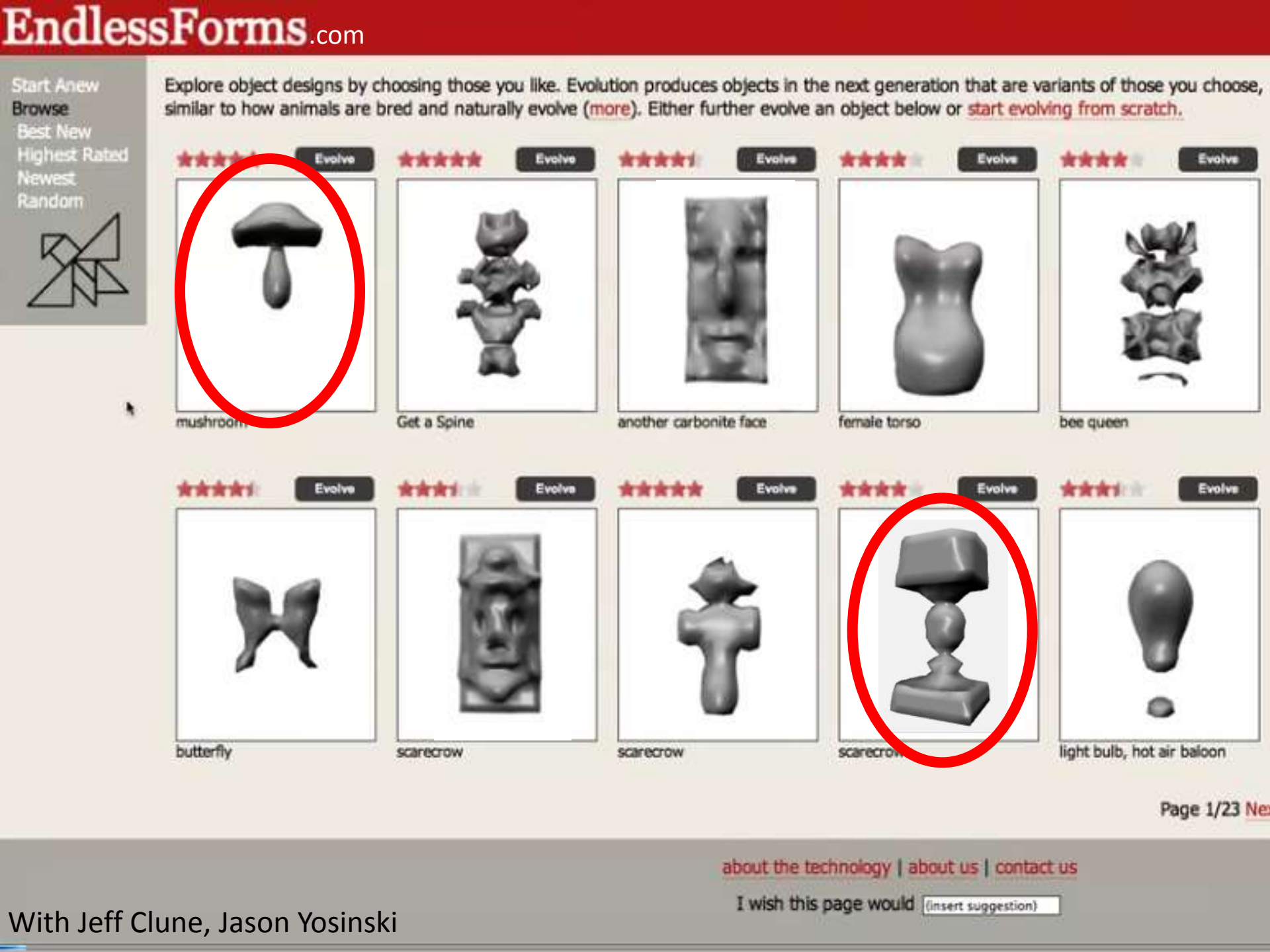





Camera
Camera View







With Jeff Clune, Jason Yosinski

Evolve

lamp

Evolve

lamp

Evolve

lamp square base

Evolve

lamp

Evolve

lamp

Evolve

square lamp

Evolve

Square lamp with square base

Evolve

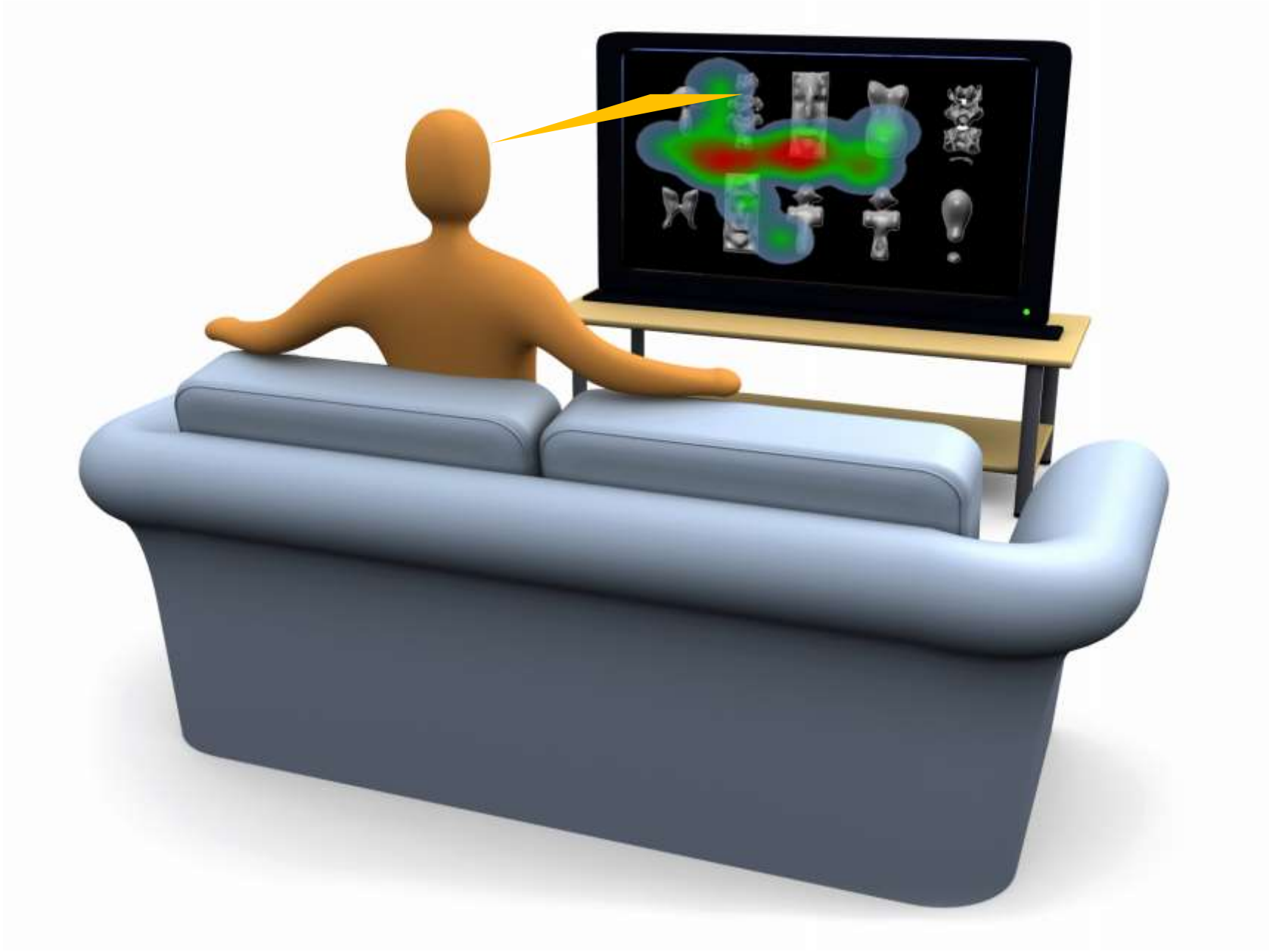
lamp

Evolve

meltlamp

Evolve

shaggy lamp



With Jeff Clune, Jason Yosinski

Developmental processes

Rules + Seed

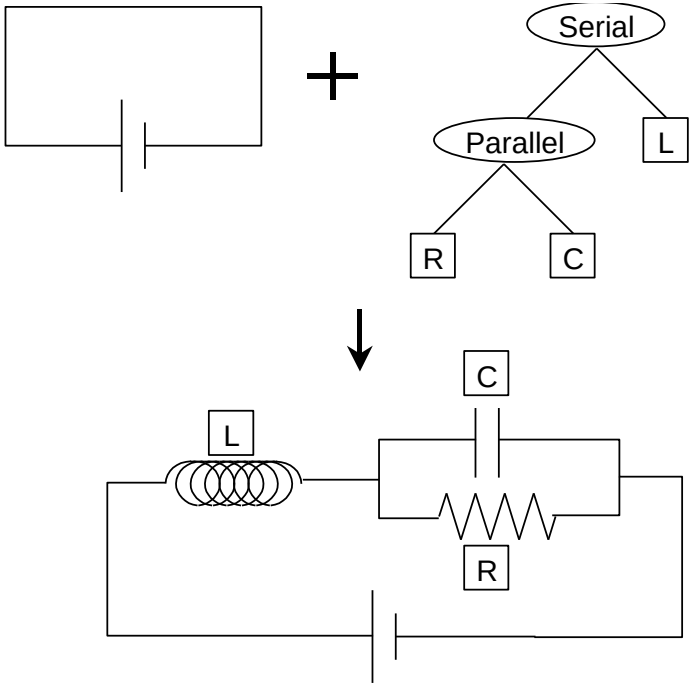
A → B
B → AB

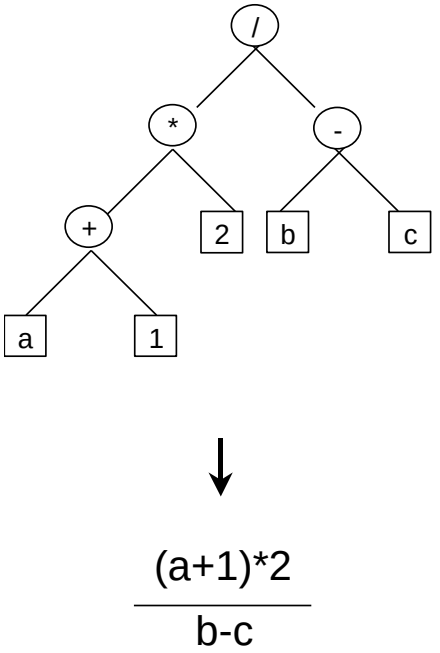
A

A
B
AB
BAB
ABBAB
BABABBAB
ABBABBABABBAB

Fibonacci

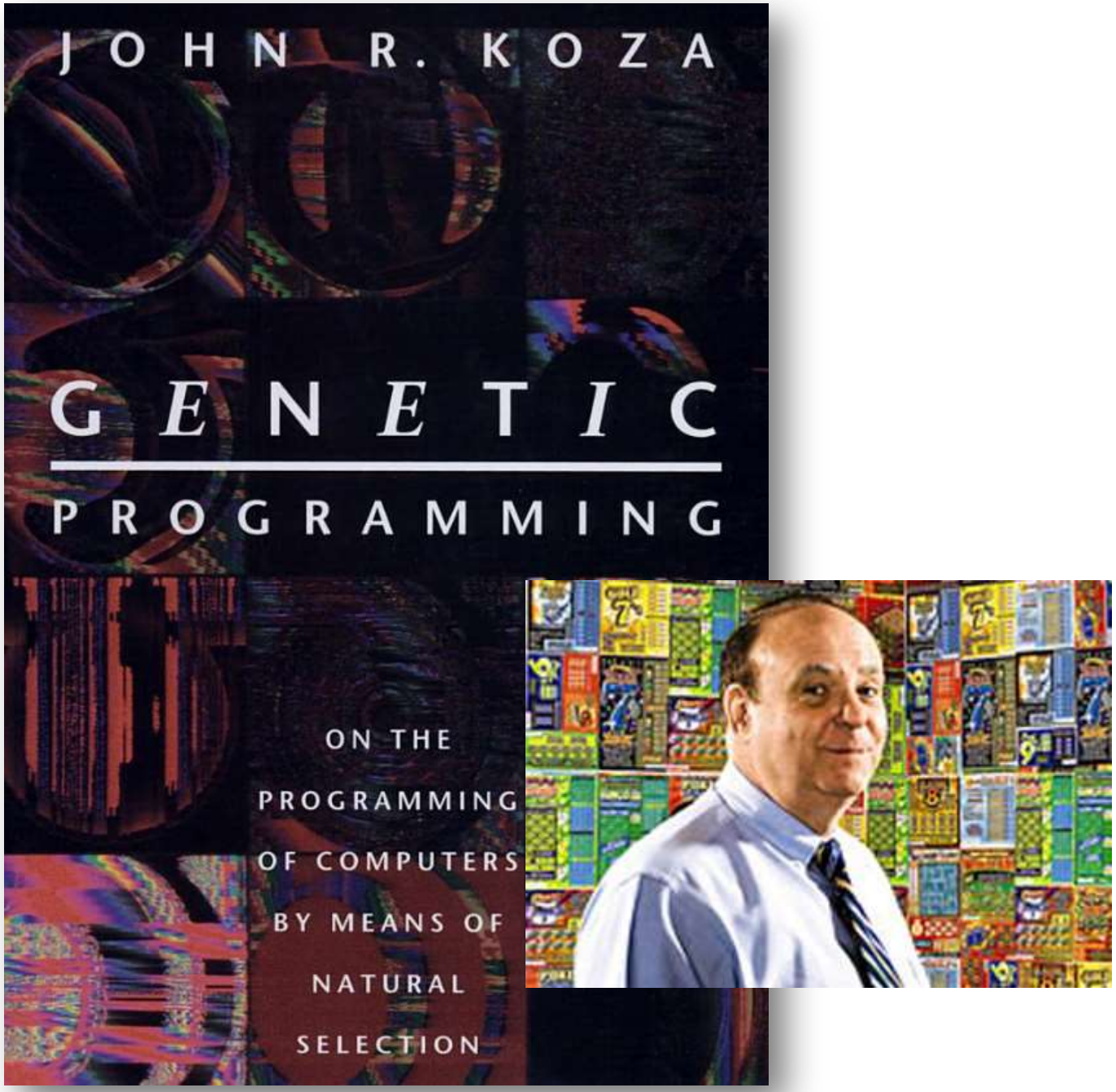
Background: Top-down and Bottom-up Tree Encodings

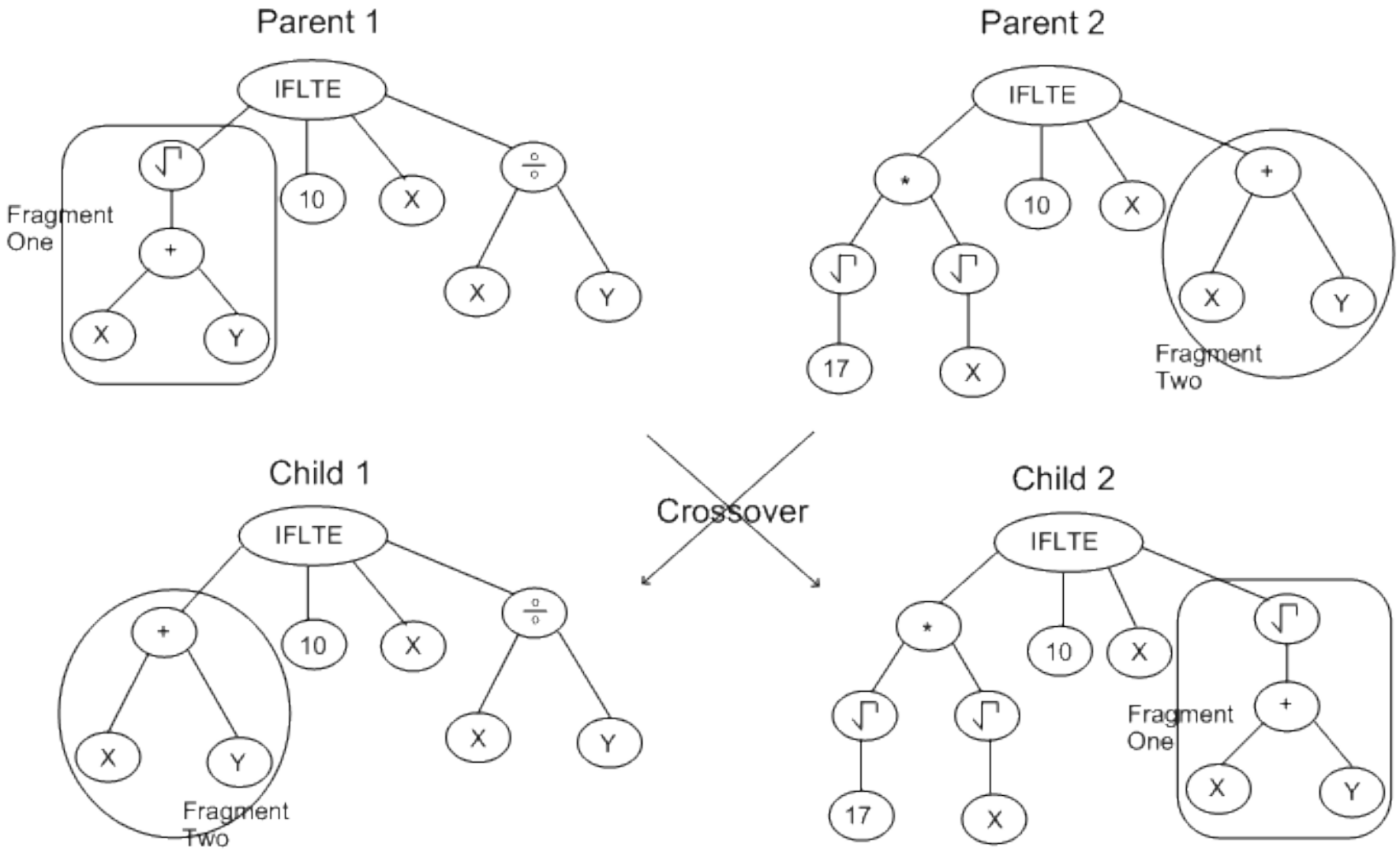




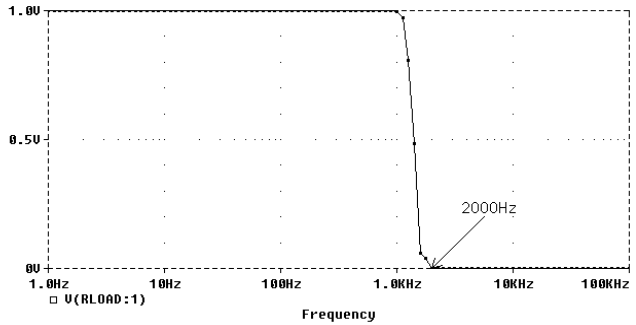
Top down: Embryo + variation operators. E.g. Circuit

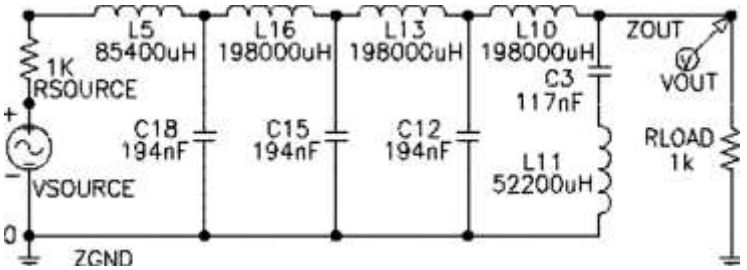
Bottom up: Composition of terminals, e.g. Symbolic expression


JOHN R. KOZA
GENETIC
PROGRAMMING
ON THE
PROGRAMMING
OF COMPUTERS
BY MEANS OF
NATURAL
SELECTION

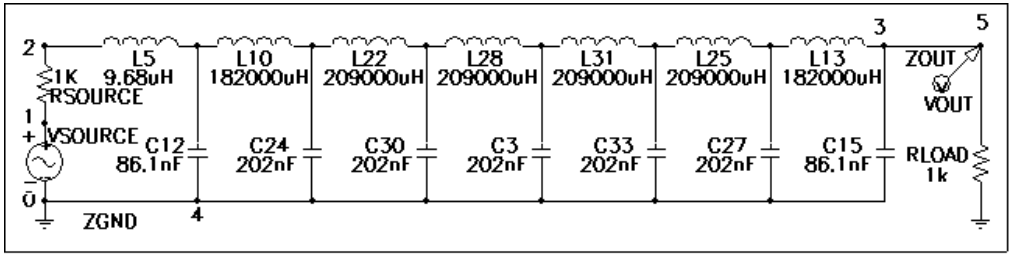

Parent 1
IFLTE
Fragment One
10
X
X
Y
+
X
Y
Parent 2
IFLTE
*
10
X
+
17
X
X
Y
Fragment Two
Crossover
Child 1
IFLTE
+
10
X
X
Y
X
Y
Fragment Two
Child 2
IFLTE
*
10
X
17
X
Fragment One
+
X
Y

Example: Analog Circuits





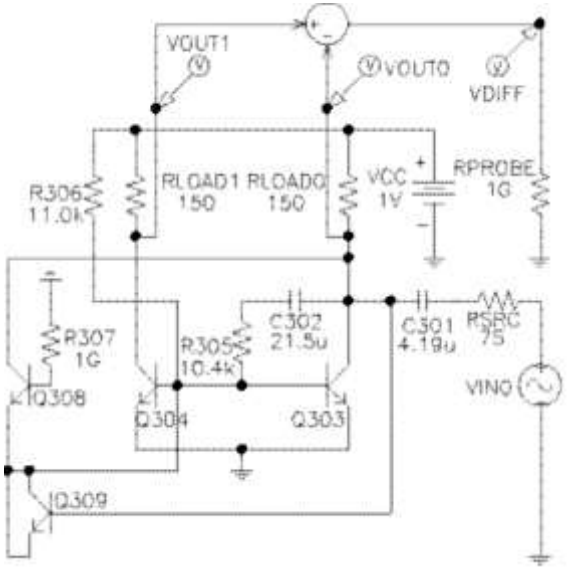
U. S. patent 1,227,113—George Campbell—AT&T—1917

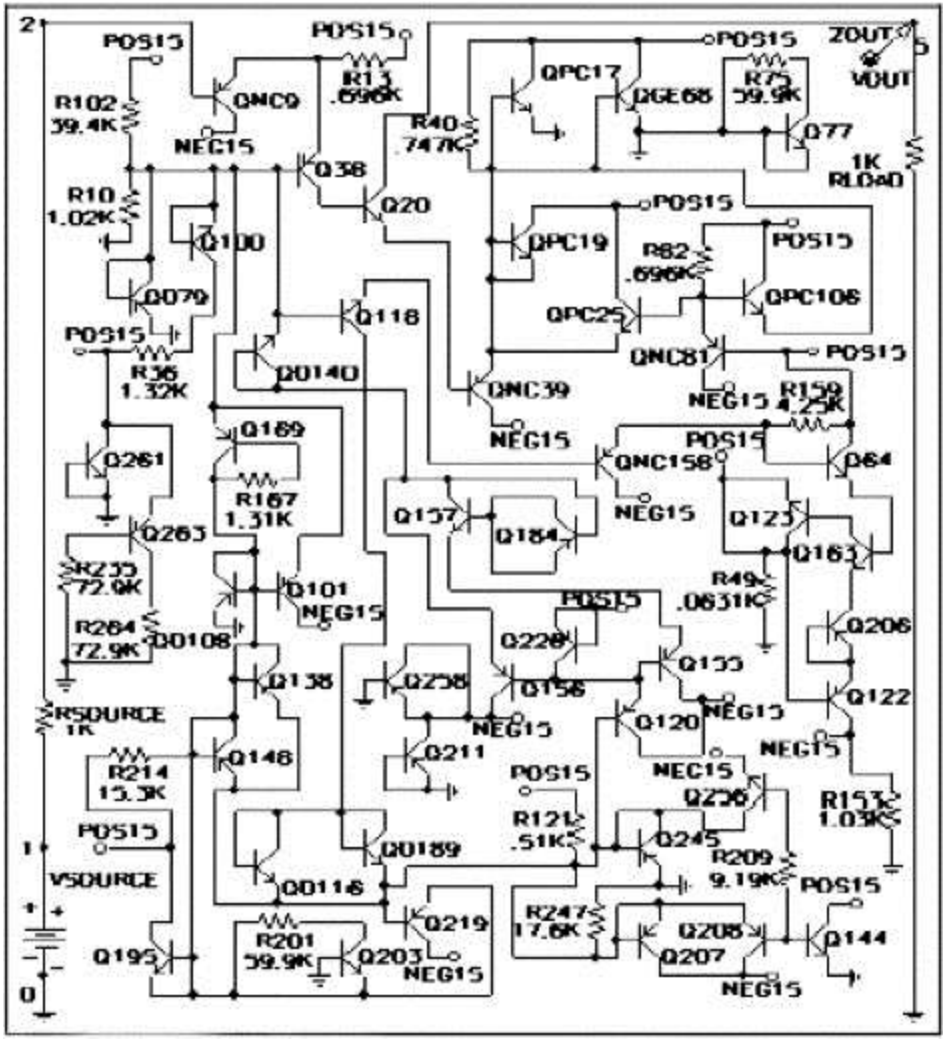


U. S. patent 1,538,964—Otto Zobel—AT&T—1925

One criterion for innovation:
Patentability

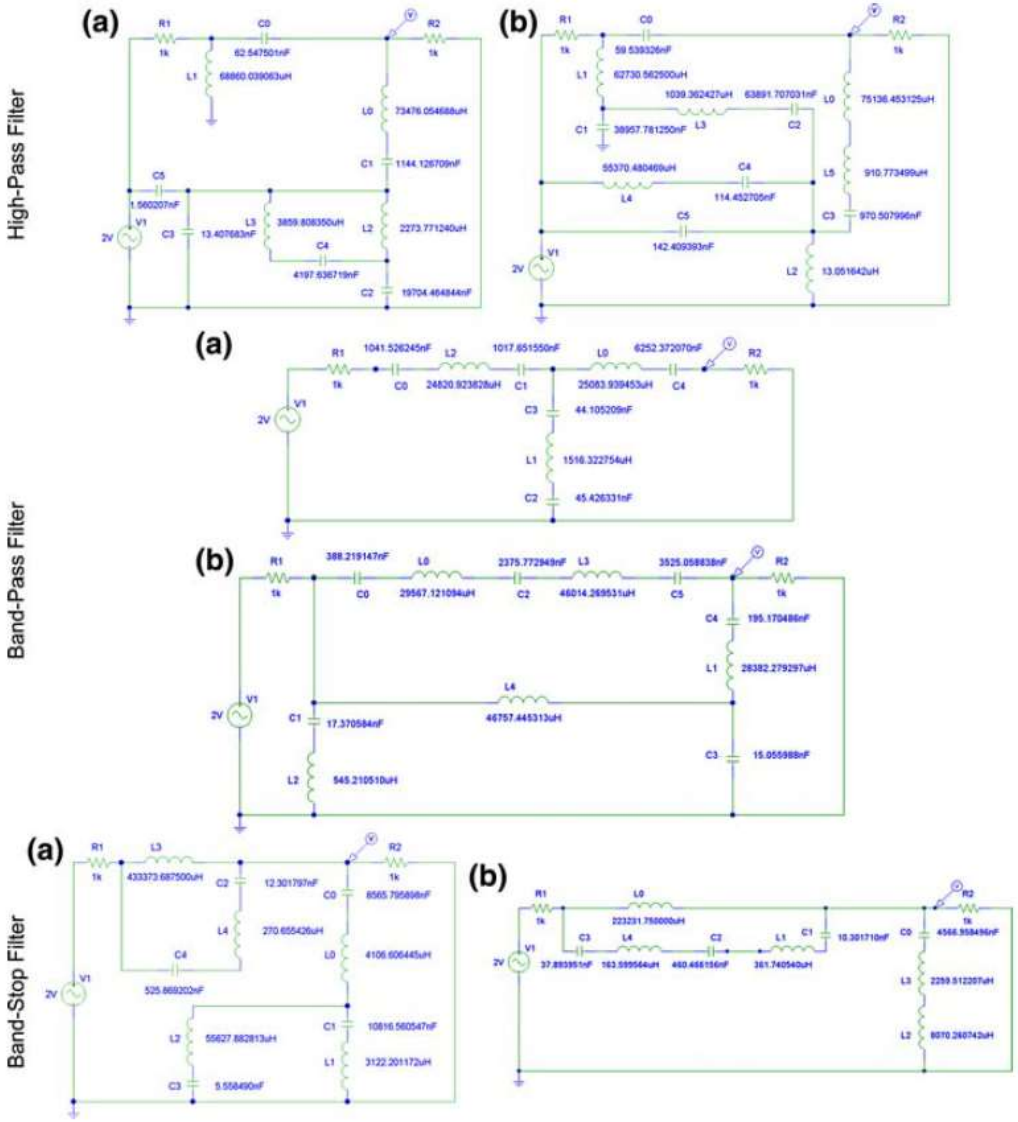
Post 2000 US patent
LOW-VOLTAGE balun circuit




2
POS15
POS15
R13
.696K
R102
39.4K
QNC9
NEG15
Q38
Q20
R10
1.02K
Q100
Q79
POS15
R36
1.32K
Q118
Q0140
Q169
Q261
R167
1.31K
Q263
R233
72.9K
Q101
NEG15
R264
72.9K
Q0108
Q138
RSOURCE
1K
R214
13.3K
Q148
POS15
1
VSOURCE
0
Q195
Q0189
Q0116
R201
59.9K
Q203
NEG15
Q219
Q211
Q258
NEG15
QPC17
POS15
ZOUT
VOUT
5
R40
.747K
QGE68
R75
59.9K
Q77
1K
RLOAD
POS15
QPC19
R82
.696K
POS15
QPC108
QPC25
QNC81
POS15
QNC39
NEG15
R159
4.25K
POS15
QNC158
Q64
NEG15
Q157
Q184
Q123
Q163
R49
.0631K
Q206
POS15
Q228
Q156
Q133
Q120
NEG15
Q122
NEG15
NEC15
Q256
R153
1.03K
POS15
R121
.51K
Q245
R209
9.19K
POS15
R247
17.6K
Q208
Q207
Q144
NEG15

Koza, 1992

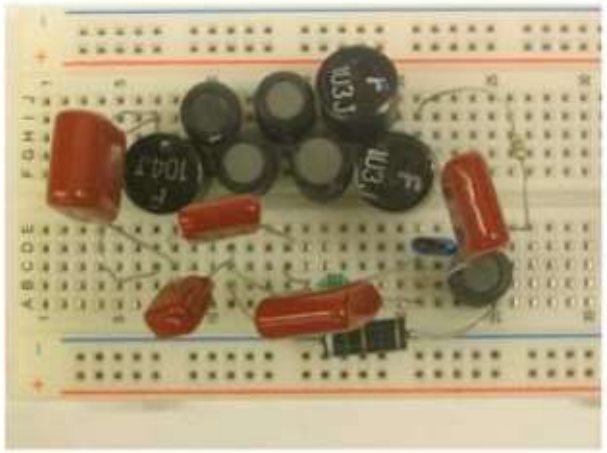
Resilient Circuits

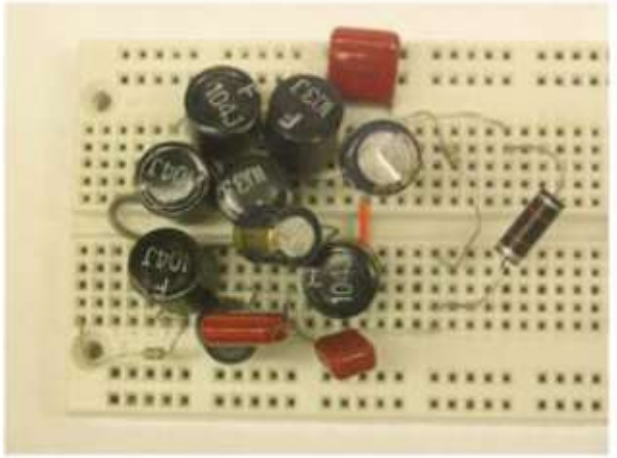


Kyung-Joong Kim

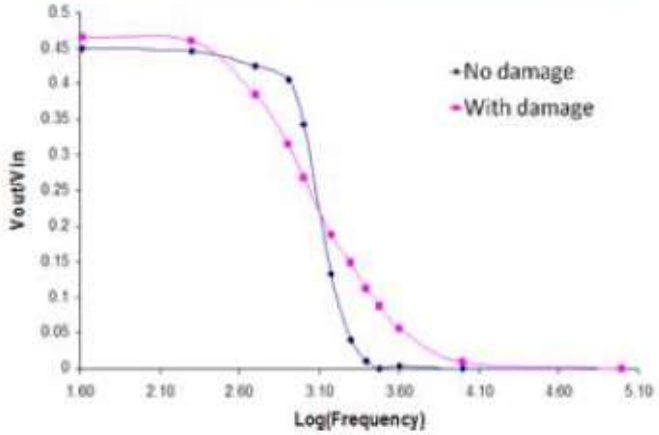
Resilient circuits

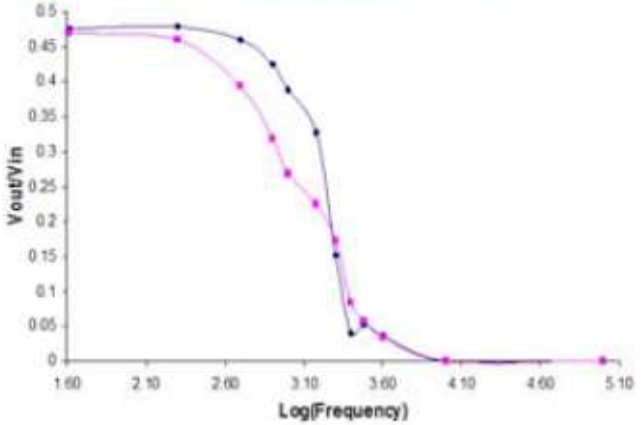
Real circuits



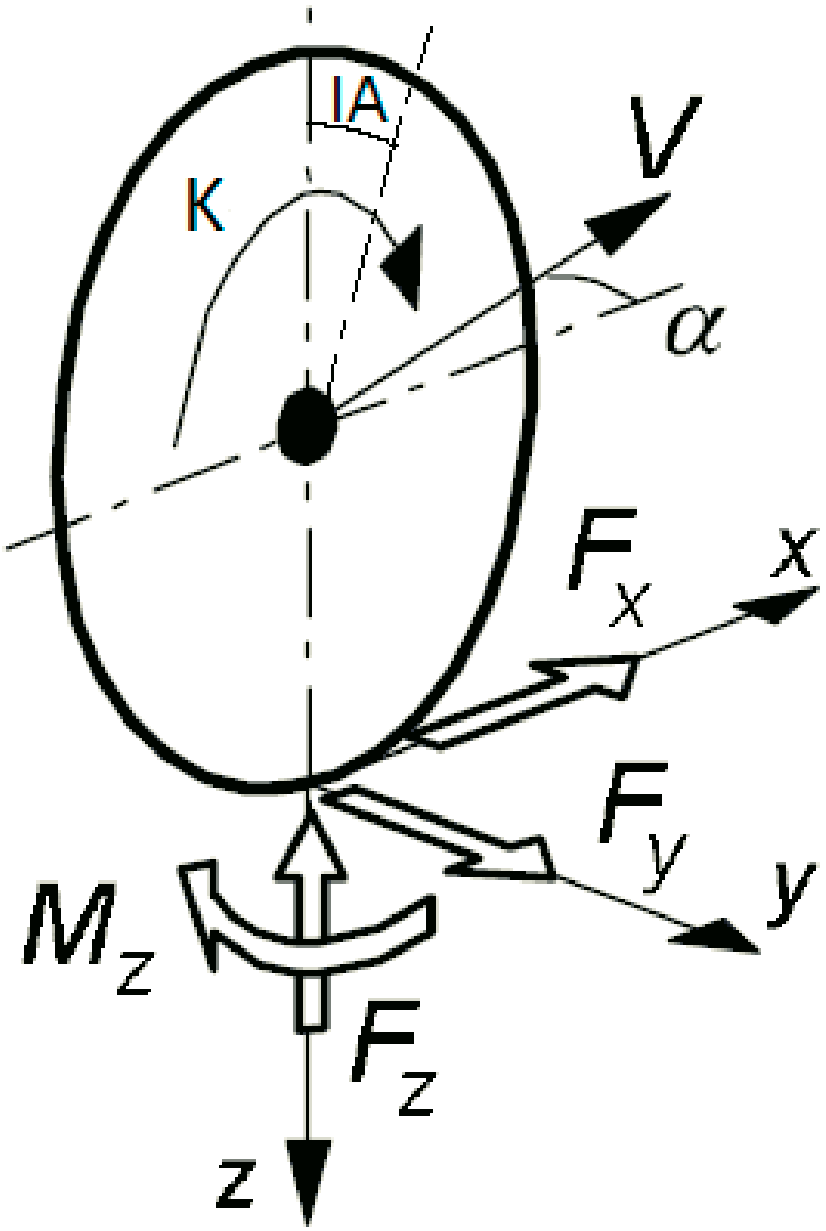


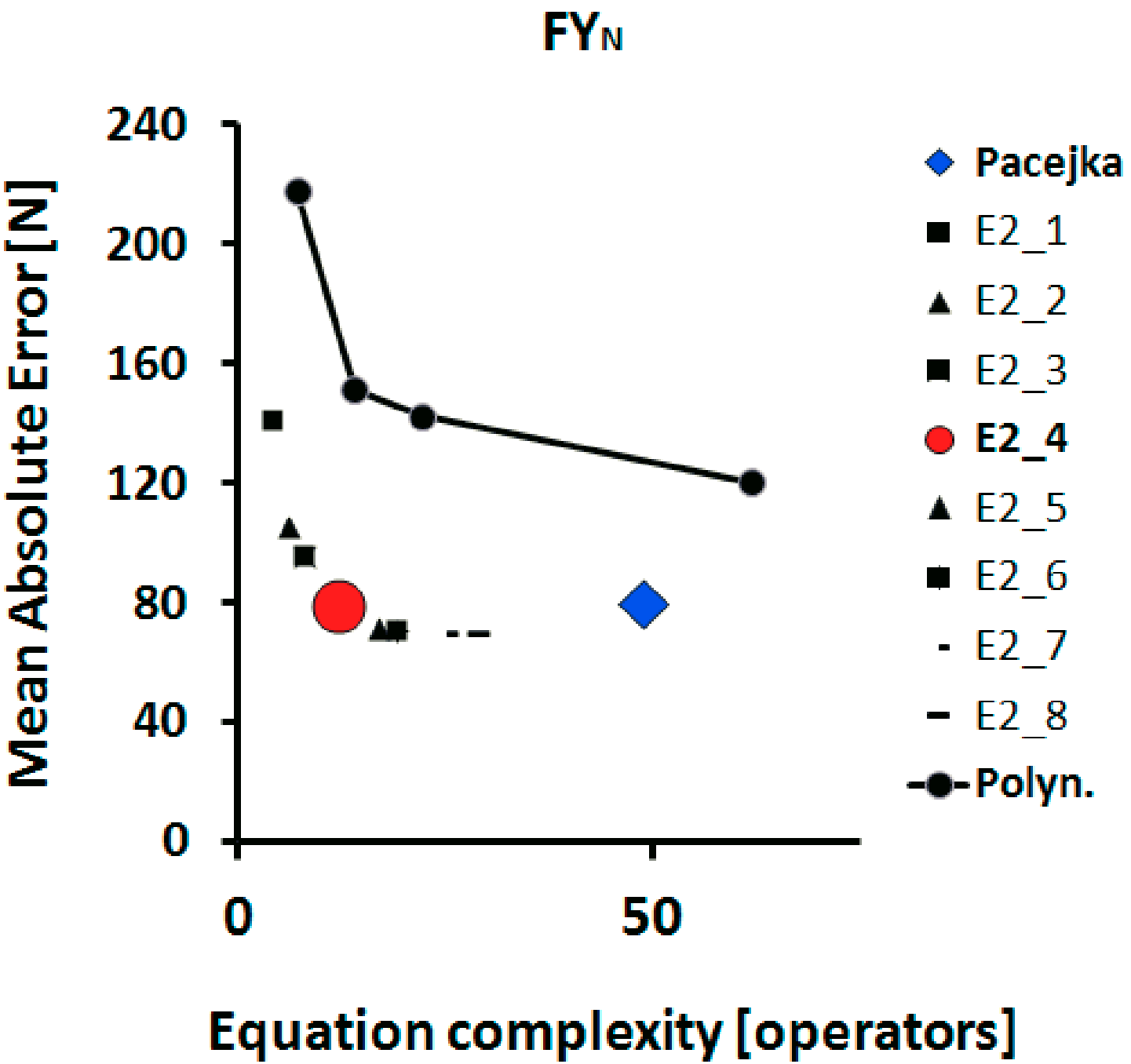
Output response of real circuits



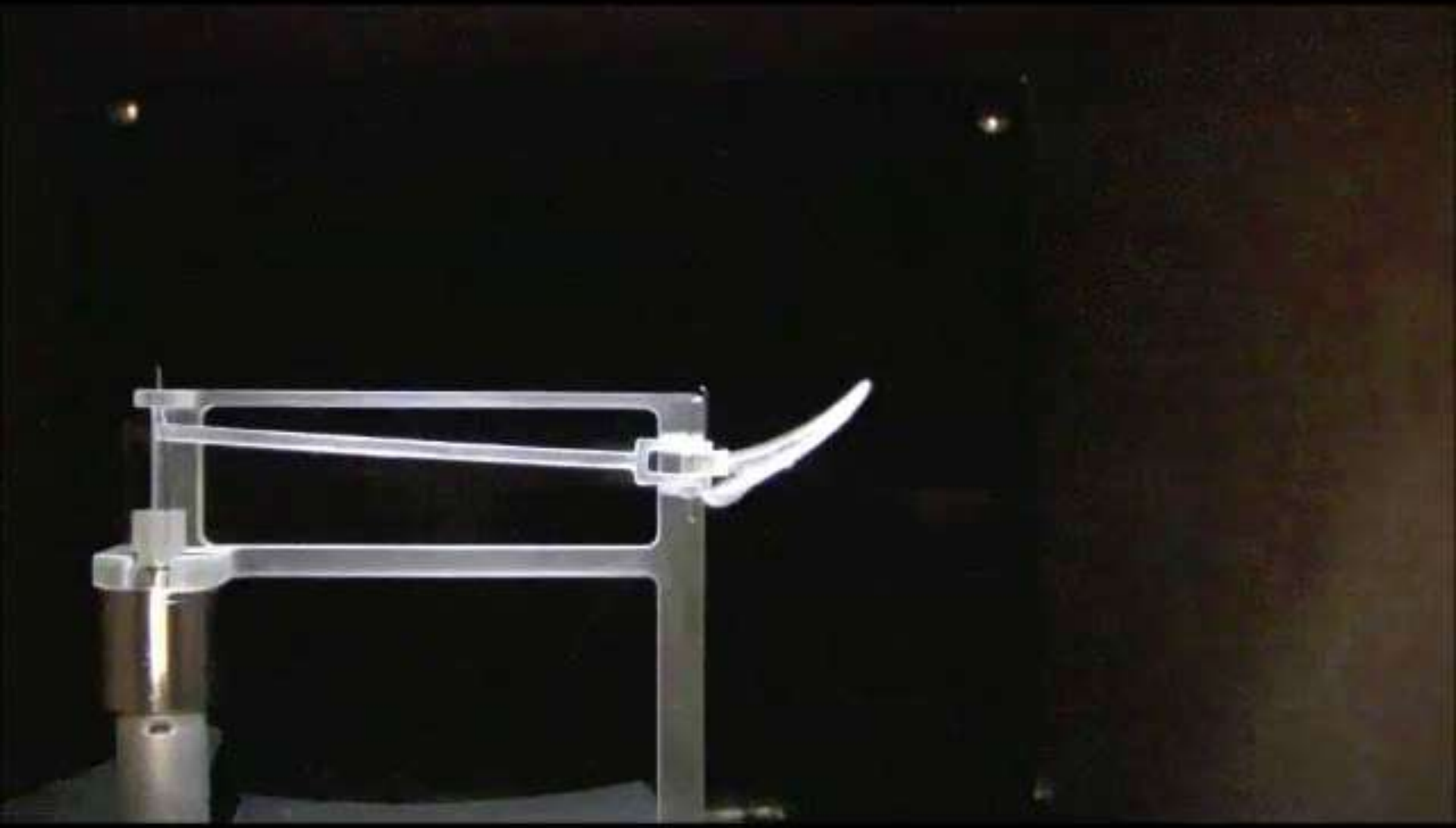


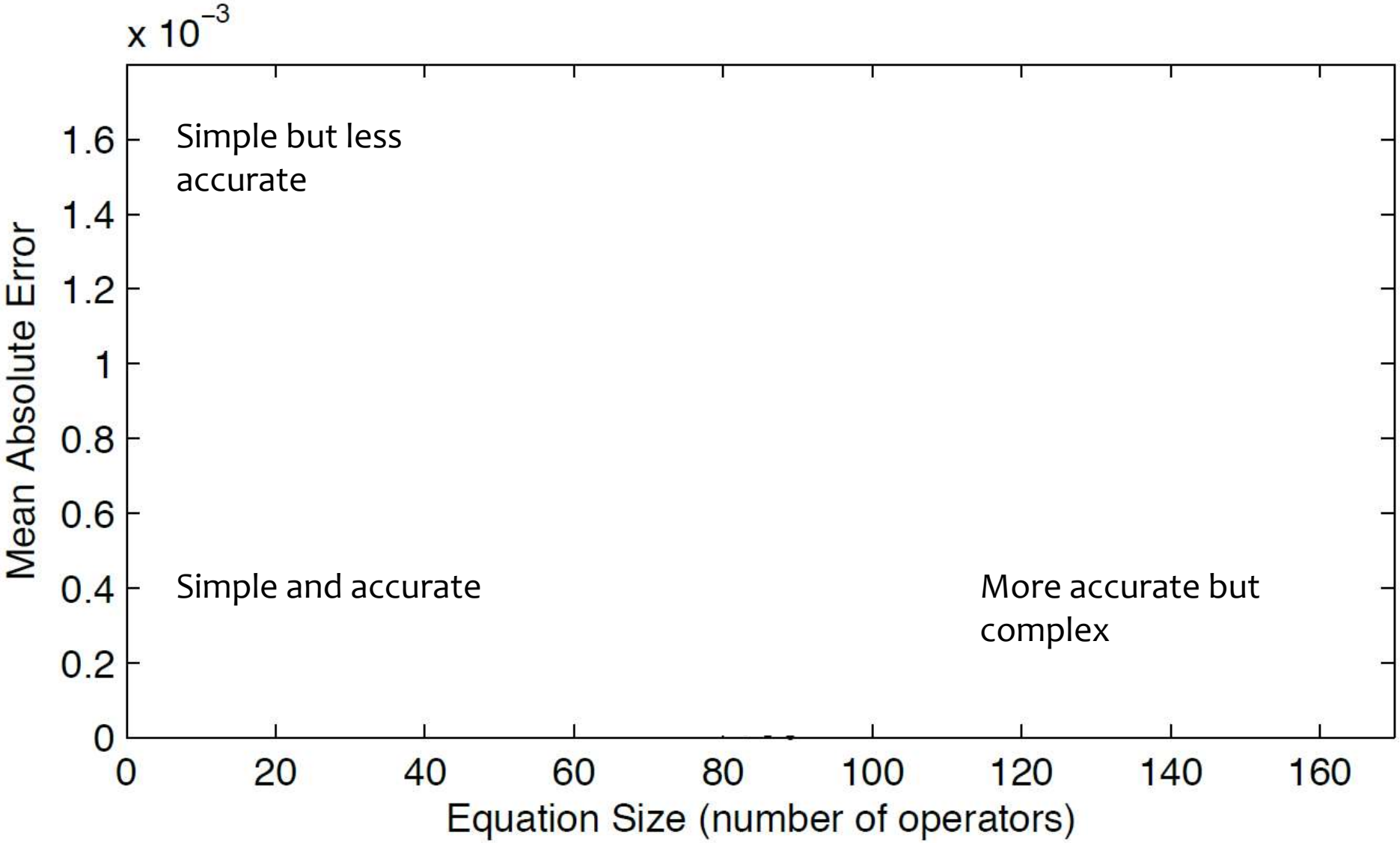
Kyung-Joong Kim


IA
K
V
α
F_x
x
F_y
y
M_z
F_z
z

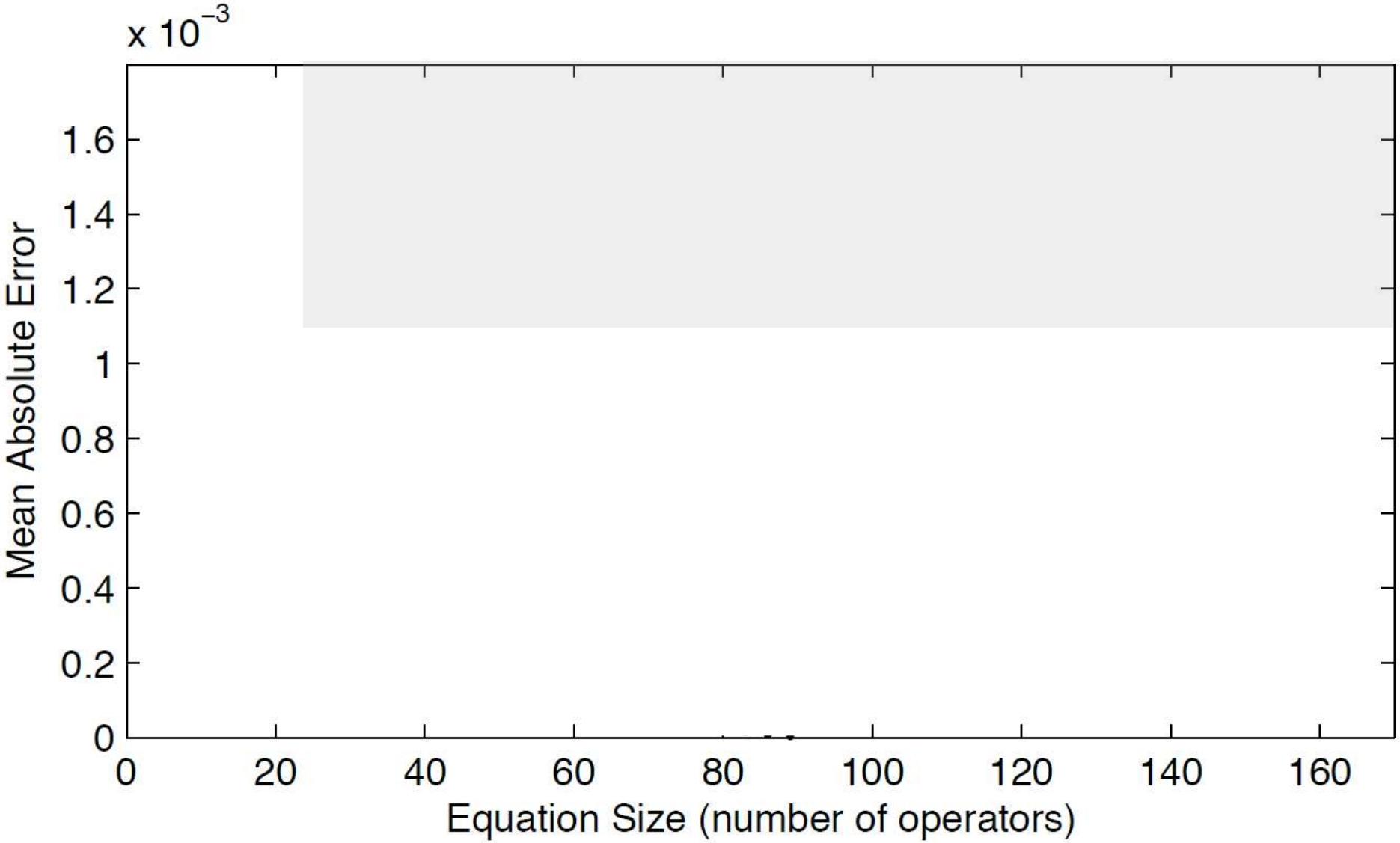


Ingmar Zanger, John Amend



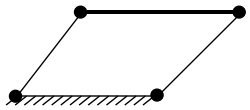


Charles Richter

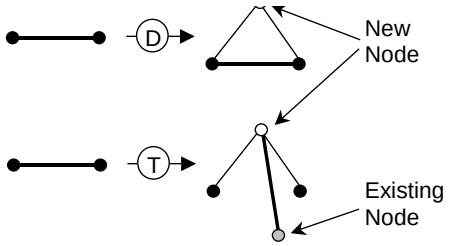


Charles Richter

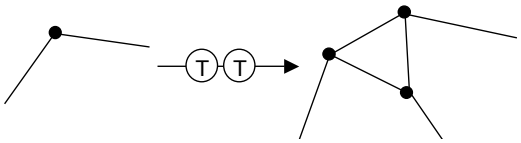
Top down encoding of a mechanism



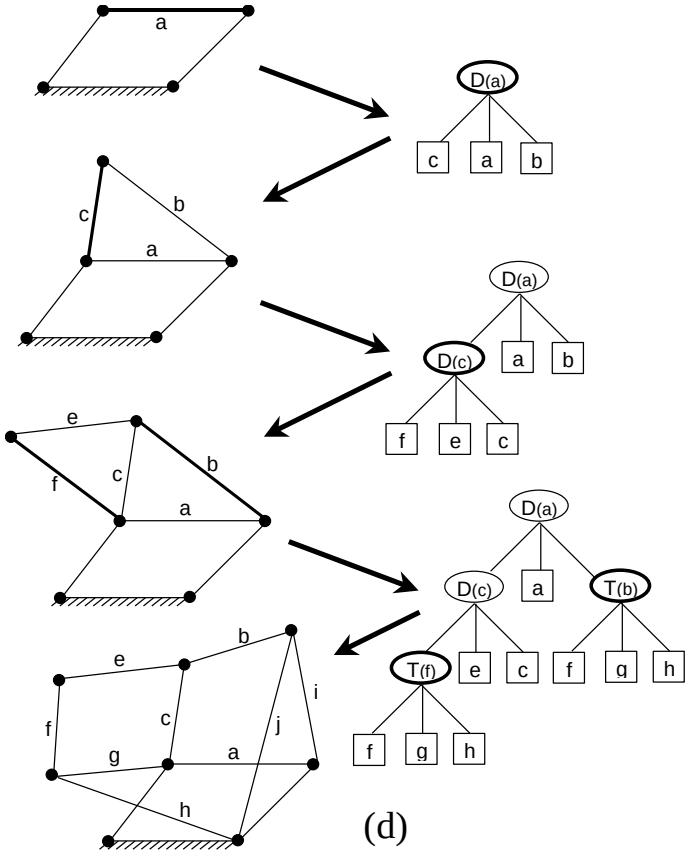
Start with Embryo with desired # of DoF, e.g. a four-bar mechanism (1 DoF)



Two variation operators maintain DoF



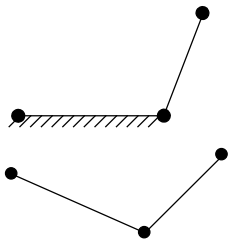
E.g. Transform dyad into tryad



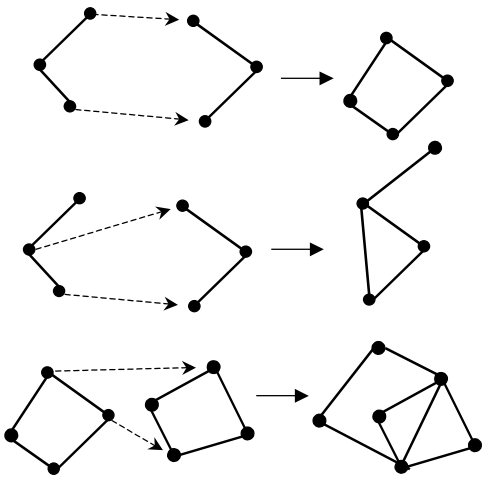
Example: A tree that constructs this 1-DoF compound mechanism

Operators provably DoF invariant

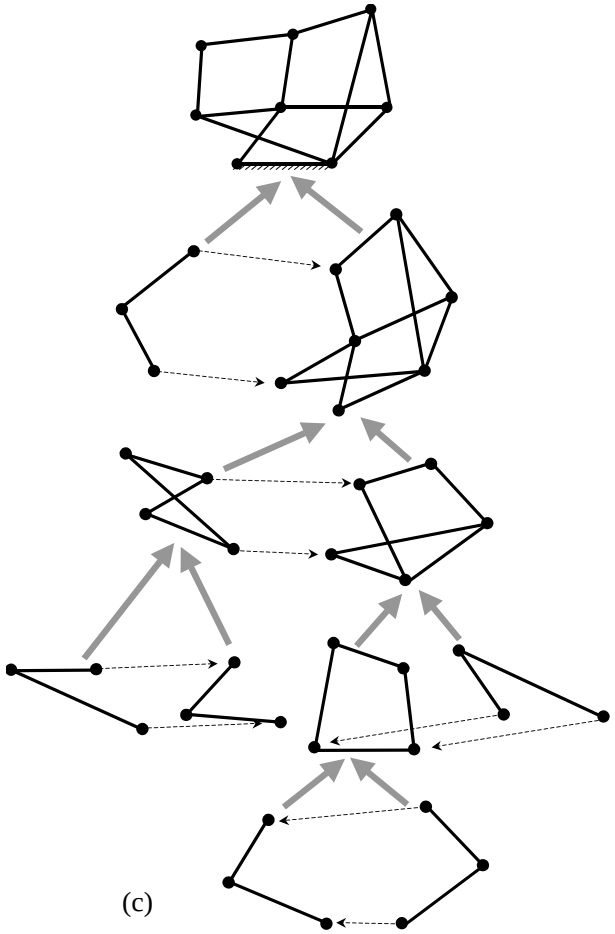
Bottom-up encoding of a 1-DoF mechanism



1-DoF terminals



Join substructures hierarchically at exactly two nodes (maintains DoF)



Example: A tree that constructs this 1-DoF compound mechanism

Operators provably DoF invariant

Intermediate Conclusions

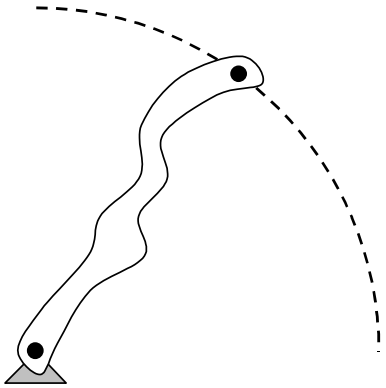
- Proposed two new DoF-invariant representations for kinematic mechanisms
  - top-down and bottom-up tree encodings

But wait, there's more…

A hard test problem

- The straight line problem
  - Devise a mechanism that traces a straight line without a reference to an existing straight line
- Human-competitive problem
  - Of great practical importance in the 18^{th} and 19^{th} century.
  - has baffled the world's greatest kinematic inventors for a century, many solutions put forward.
- Now forgotten
  - with advent of precision manufacturing

The Straight Line Problem

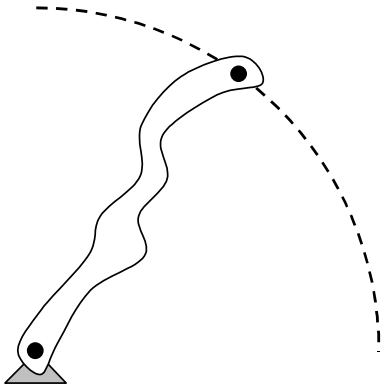


It is easy to think of a mechanism that traces an exact circle without having a circle built in: A **compass**.

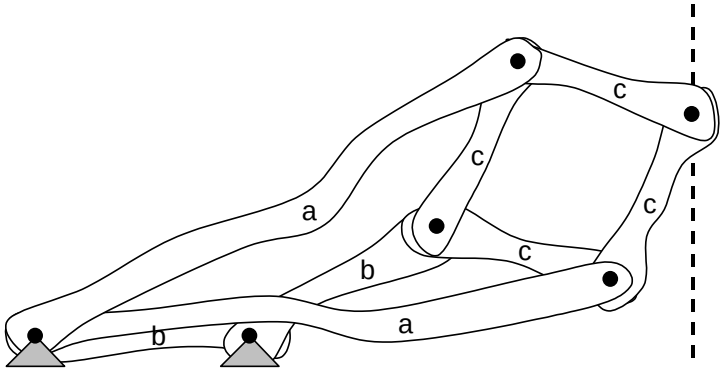
?

Can you think of a linkage mechanism that will trace a straight line without reference to an existing straight line?

The Straight Line Problem



It is easy to think of a mechanism that traces an exact circle without having a circle built in: A **compass**.



One solution: The Peaucellier (1873)

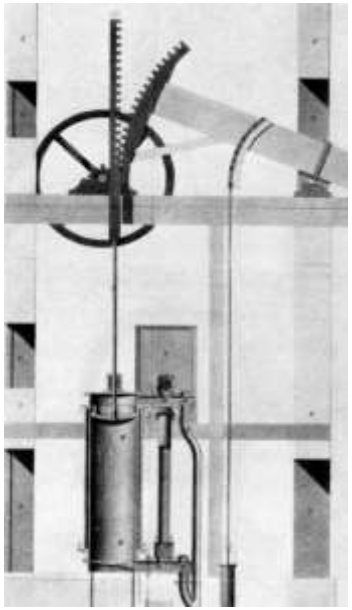
The straightness of the links themselves does not matter

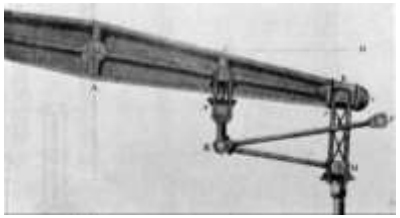
The Straight-Line problem

- Needed to guide the piston of the steam engine.
  - *The* breakthrough that made steam engines a success

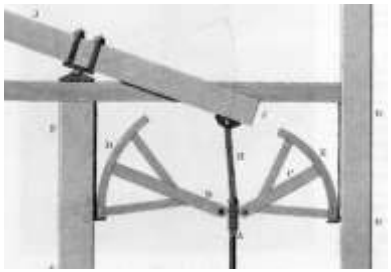
"Though I am not over anxious after fame, yet I am more proud of the parallel motion than of any other mechanical invention I have ever made"

James Watt, cf. 1810 [15]



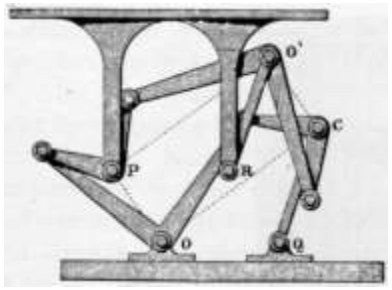


Watt's first straight line mechanism (1784)

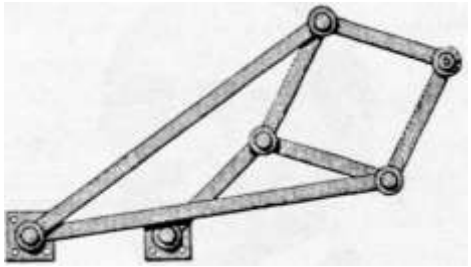


James Watt's original patents used racks and sectors, and many other cumbersome solutions

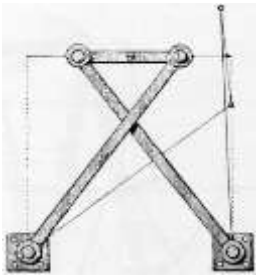
More established solutions



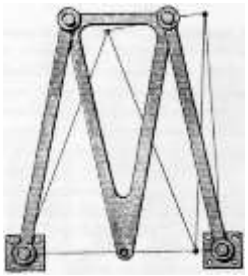
Silverster-Kempe's (1877)



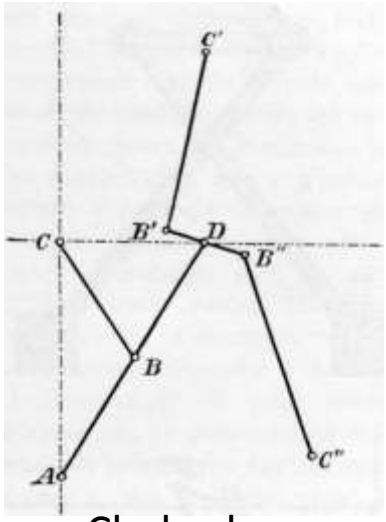
Peaucelier (1873)



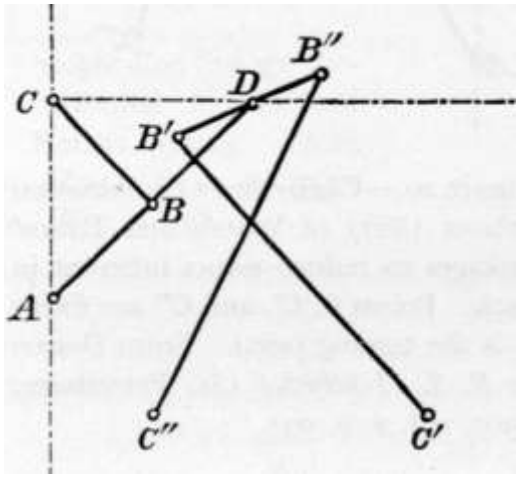
Chebyshev (1867)



Robert (1841)



Chebyshev (1867)

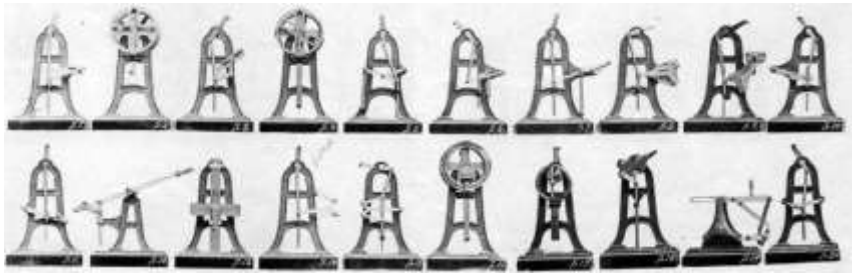


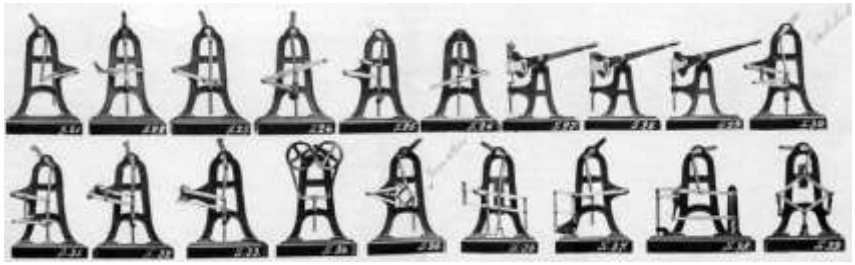
Chebyshev-Evans (1907)

Source: Kempe A. B., (1877), *How To Draw A Straight Line*, London

See **http://kmoddl.library.cornell.edu**

Considered fundamental technology

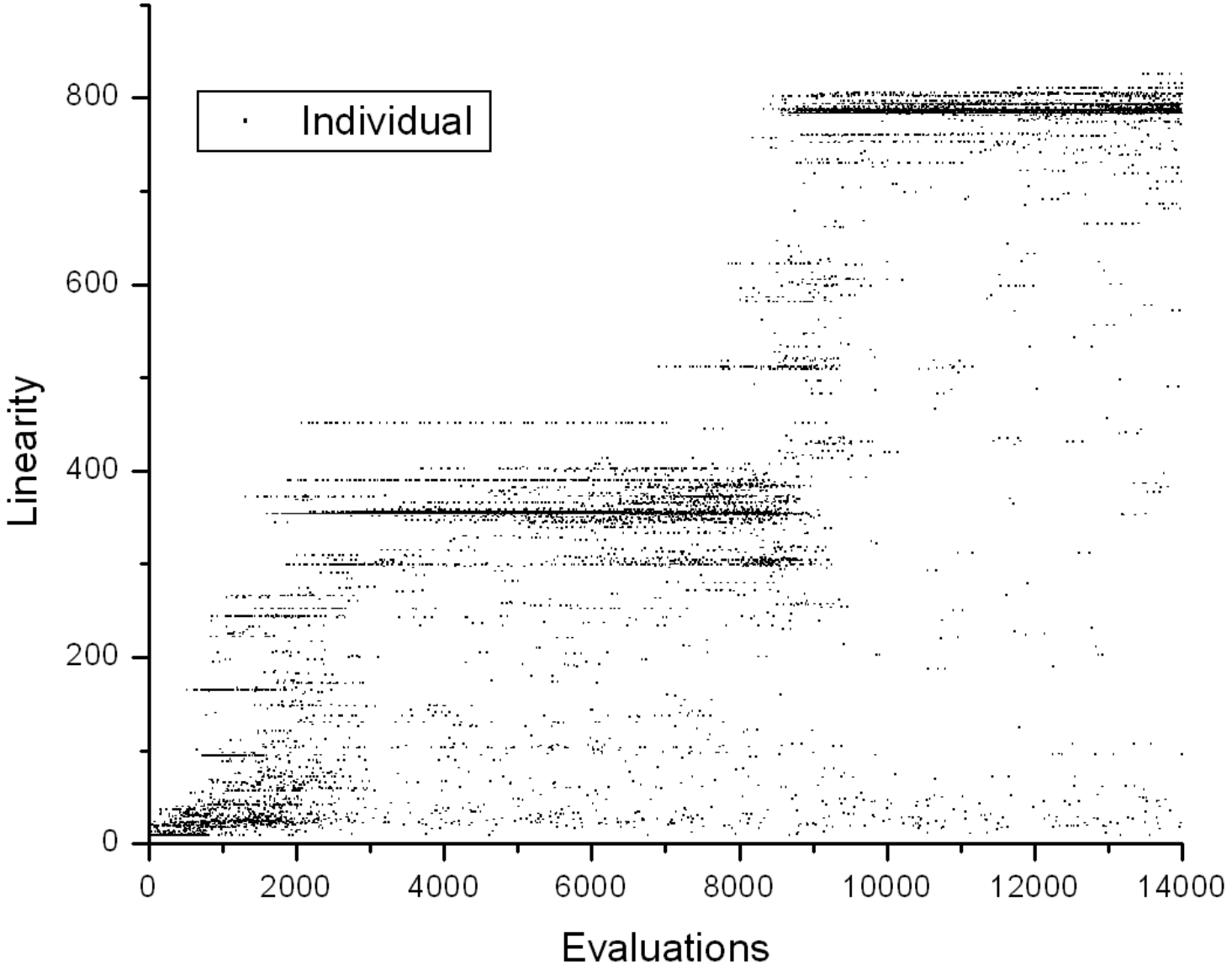




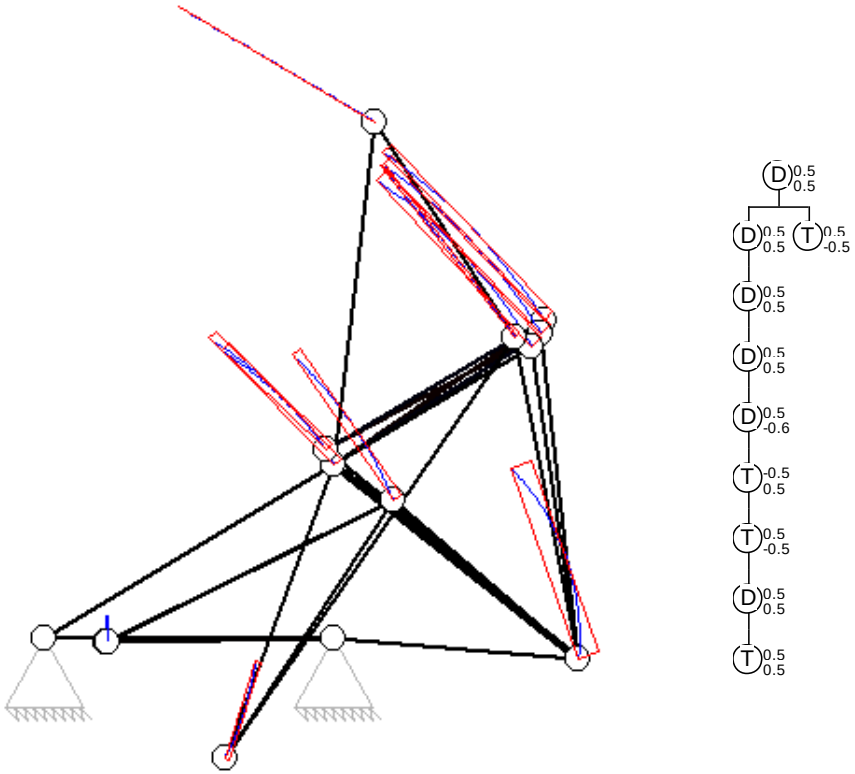
Cornell University acquired in 1882 about 40 straight-line mechanism models and used them in the early engineering curriculum.

See videos at
Cornell University Digital Library of Kinematic Models
http://kmoddl.library.cornell.edu

Evolving Straight line mechanisms

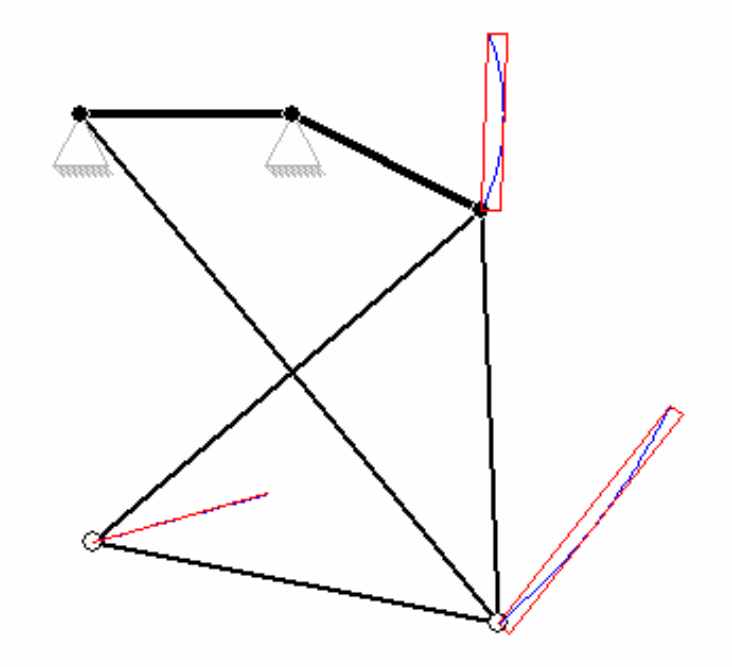


Some results



Linearity 1:4979

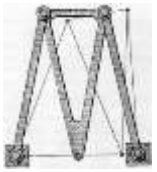
Some results



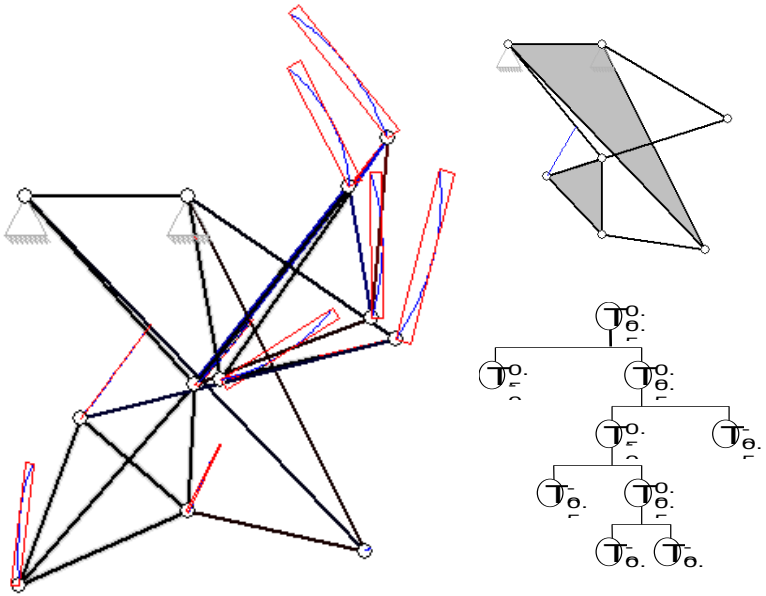
Linearity 1:5300

Infringes on Robert's Linkage (1841)

Published: Kempe A. B., (1877), *How To Draw A Straight Line*, London

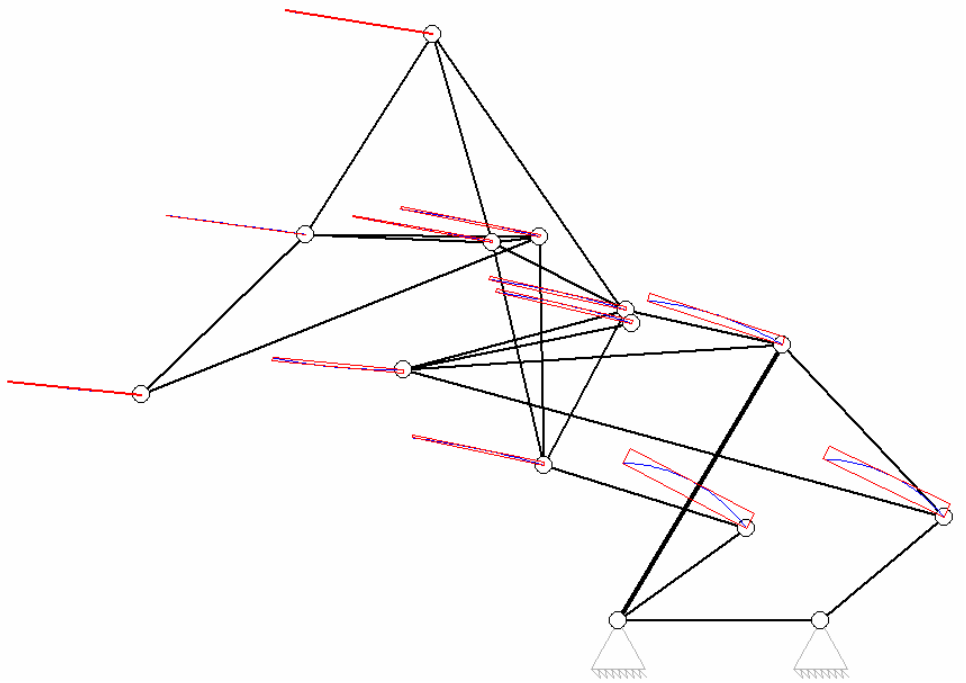


Some results



Linearity 1:12819

Some results



Linearity 1:28340

Many more solutions were produced