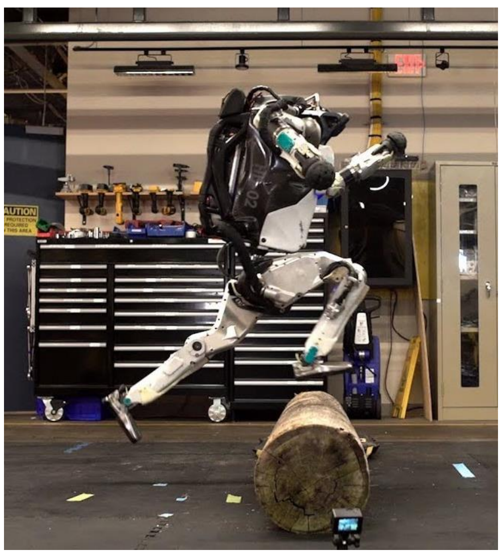
Computational Design Synthesis of Passive Dynamic Systems

Fritz Stöckli
Prof. Dr. Kristina Shea
Engineering Design + Computing Laboratory
Department of Mechanical and Process Engineering
ETH Zürich
May 8 2019

Robotic Systems

Active Robotic Systems

- Actuators and feedback control
- High task flexibility possible
- Responsive to environment
- High robustness



https://www.bostondynamics.com/atlas

Passive Robotic Systems

- No actuators and control
- No energy source necessary
- Potential to save energy and components

Passive dynamic walking, Mcgeer, T., 1990, International Journal of Robotics Research

Automated Topological Synthesis in Robotics

Active Dynamic Systems

- Evolving topology and control together

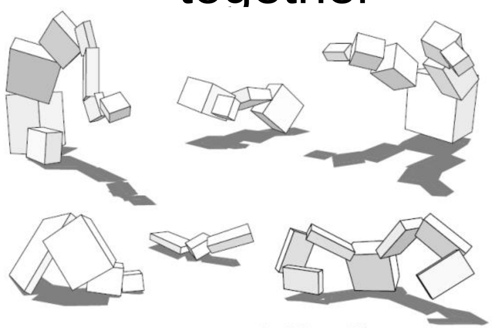
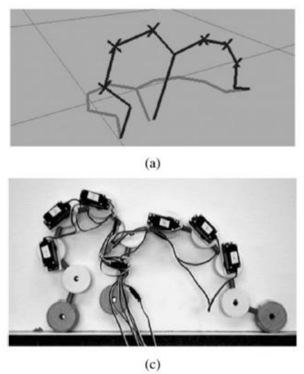


Figure 7: Creatures evolved for walking.

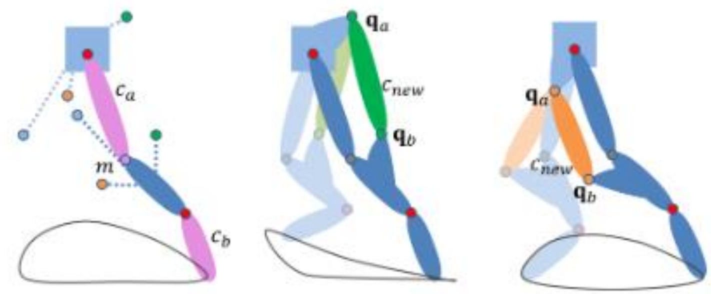
Evolving Virtual Creatures, Karl Sims, 1994



Generative Representations for the Automated Design of Modular Physical Robots, G.Hornby, H.Lipson, 2003

Kinematic Systems

- Not considering causes of motion (forces, masses, … do not matter)



Computational Design of Linkage-Based Characters, Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, Markus Gross

This Research: Passive Dynamic Systems

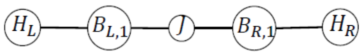
- Forces, masses, … are important
- Do not draw energy from a source
- No feedback control

CDS of Passive Dynamic Systems - Overview

Robotic Task

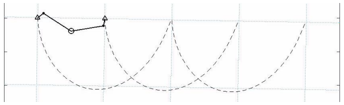
Graph Grammar

Robot Topology



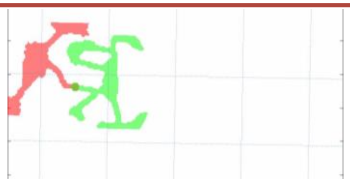
Simulation-Driven Parametric Opt

Multi-Body System



Rule-Based Topology Opt

Shape Embodiment



3D-Printing

Prototype

Example Problem: Brachiating

Robotic Task

Graph Grammar

Robot Topology

Simulation-Driven Parametric Opt

Multi-Body System

Rule-Based Topology Opt

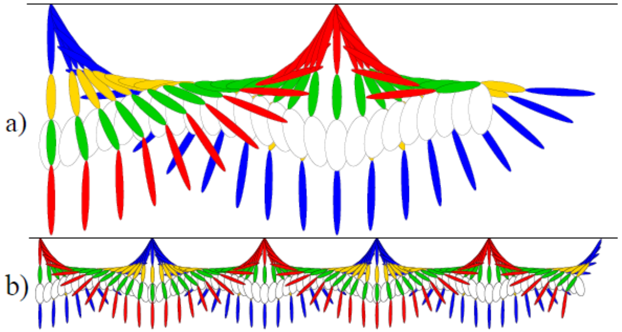
Shape Embodiment

3D-Printing

Prototype

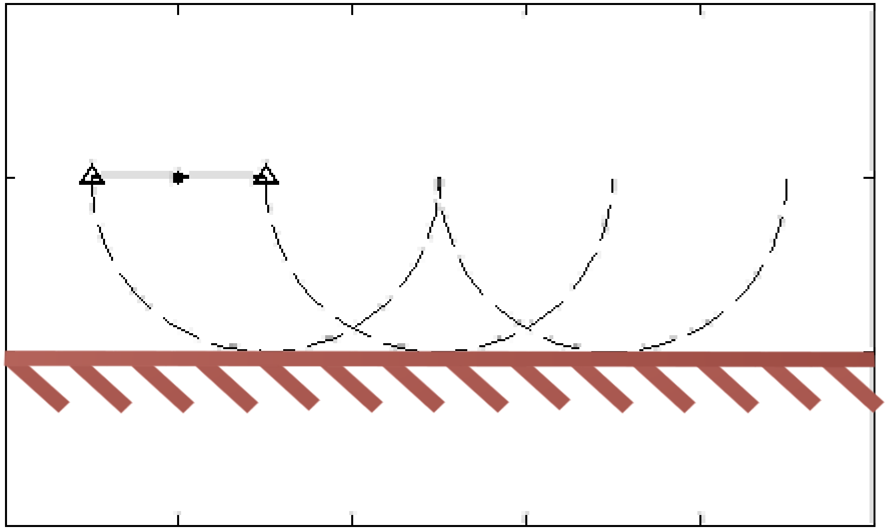
- Brachiating: The swinging locomotion of primates moving from one tree branch to the next.

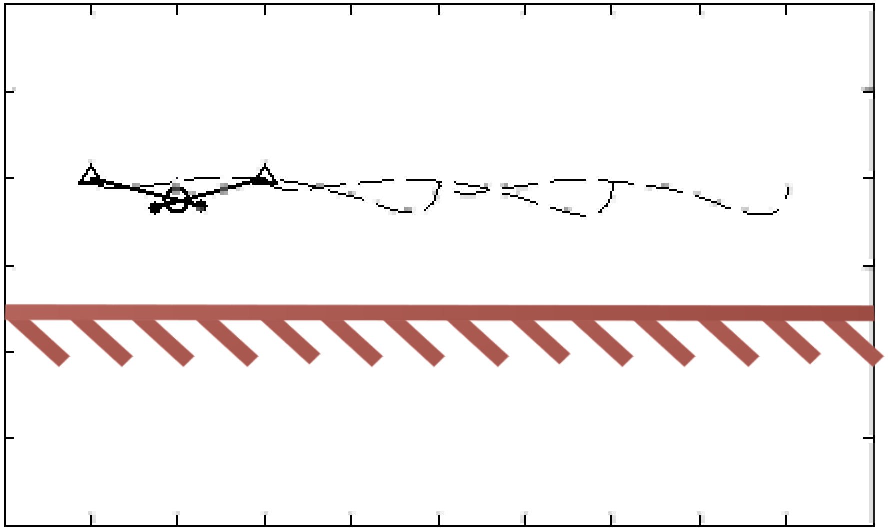
- Complex, bio-inspired models of passive dynamic brachiating exist:



A five-link 2D brachiating ape model with life-like zero energy-cost motions, Mario Gomes, Andi L. Ruina, 2005

Motivation for Complex Brachiating Topologies





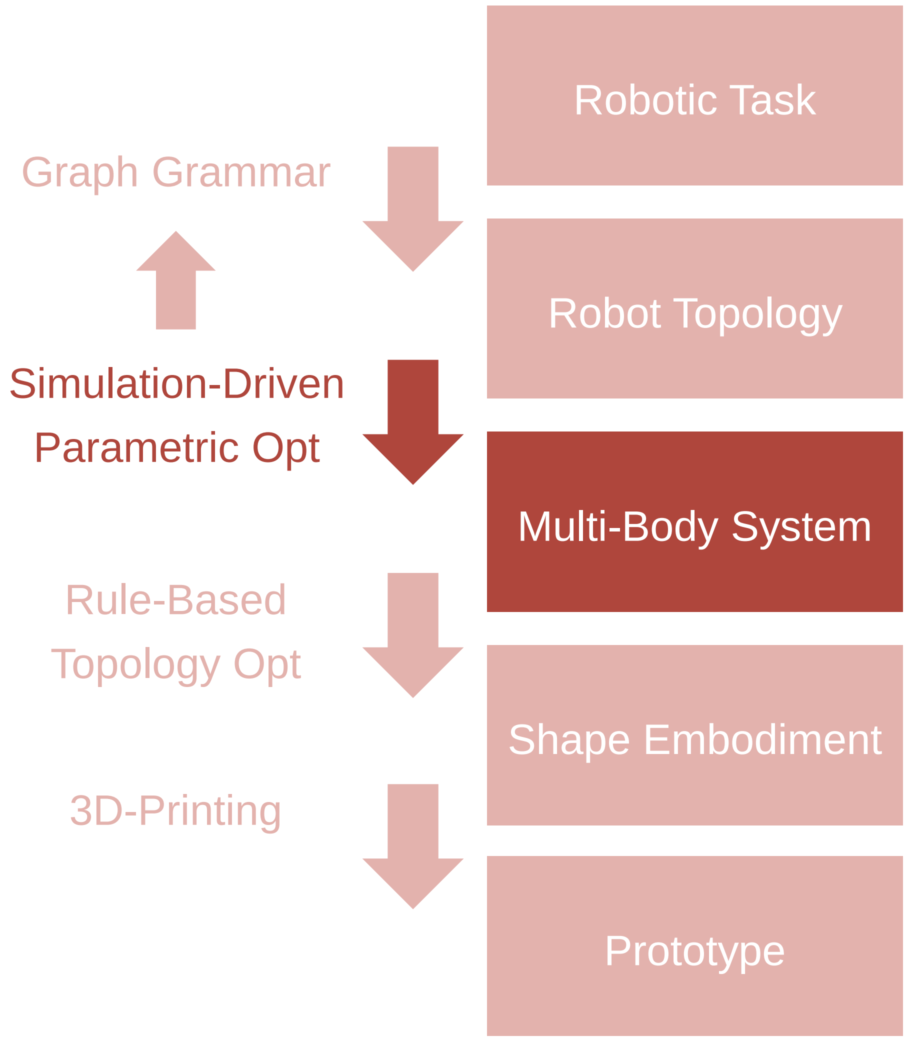
Single Pendulum

- Simplest possible solution

More Complex Solutions

- Might require less space
- Test for synthesis method

Simulation-Driven Parametric Optimization



- Multibody simulation
- Arbitrary 2D systems with revolute joints
- Closed kinematic chains possible
- Parametric Optimization
- Evaluation based on system trajectory

Multi-Body Dynamics

Equations of motion
(set of ODEs)

$$M(q,t)\ddot{q} - h(q,\dot{q},t) = 0$$

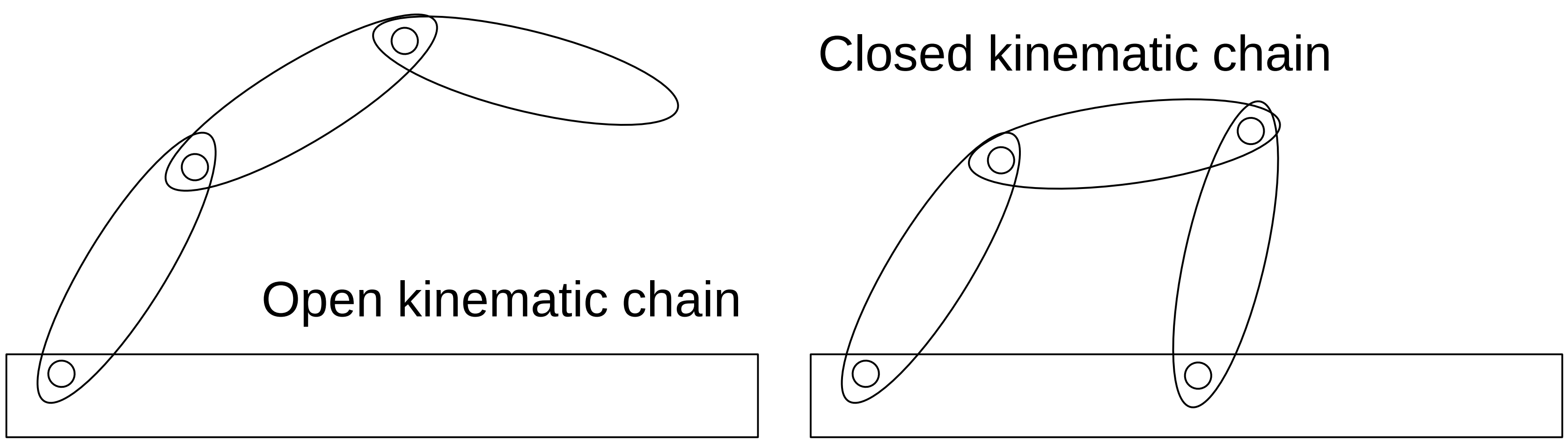
M Mass matrix

q System coordinates

h Forces (gravity, springs, …)

Motion trajectories can be calculated using numeric integreation.

Formulation works for open kinematic chains only.



Multi-Body Dynamics with Closed Kinematic Chains

Equations of motion for systems with closed kinematic chains (Differential-Algebraic System)

$$\boldsymbol{M}(\boldsymbol{q},t)\ddot{\boldsymbol{q}} - \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}},t) - \boldsymbol{W}(\boldsymbol{q},t)\boldsymbol{\lambda} = 0$$ Set of ODEs

$$\boldsymbol{g}(\boldsymbol{q},t) = 0$$ Set of algebraic Eqations

$\boldsymbol{g}$ Vector of Constraints (Same as $\mathbf{C}$ in Lecture 3 “Kinematics of Mechanisms”)

$\boldsymbol{\lambda}$ Vector of constraining forces

$$\boldsymbol{W}(\boldsymbol{q},t) = \frac{\partial \boldsymbol{g}(\boldsymbol{q},t)}{\partial \boldsymbol{q}}$$ Matrix of generalized force directions (How constraining forces act on system coordinates)

Multi-Body Dynamics with Closed Kinematic Chains

Transform into set of ODEs by taking second derivative of $\boldsymbol{g}$

$$\boldsymbol{M}(\boldsymbol{q},t)\ddot{\boldsymbol{q}} - \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}},t) - \boldsymbol{W}(\boldsymbol{q},t)\boldsymbol{\lambda} = 0$$

$$\ddot{\boldsymbol{g}}(\boldsymbol{q},t) = \boldsymbol{W}^T(\boldsymbol{q},t)\ddot{\boldsymbol{q}} + \boldsymbol{\zeta}(\boldsymbol{q},\dot{\boldsymbol{q}},t) = 0$$

Initial concitions:

$$\boldsymbol{g}(\boldsymbol{q}_0,t_0) = 0$$

$$\dot{\boldsymbol{g}}(\boldsymbol{q}_0,\dot{\boldsymbol{q}}_0,t_0) = 0$$

Solve for $\boldsymbol{\lambda}$ and $\ddot{\boldsymbol{q}}$

$$\boldsymbol{\lambda} = -(\boldsymbol{W}^T\boldsymbol{M}^{-1}\boldsymbol{W})^{-1}(\boldsymbol{W}^T\boldsymbol{M}^{-1}\boldsymbol{h} + \boldsymbol{\zeta})$$

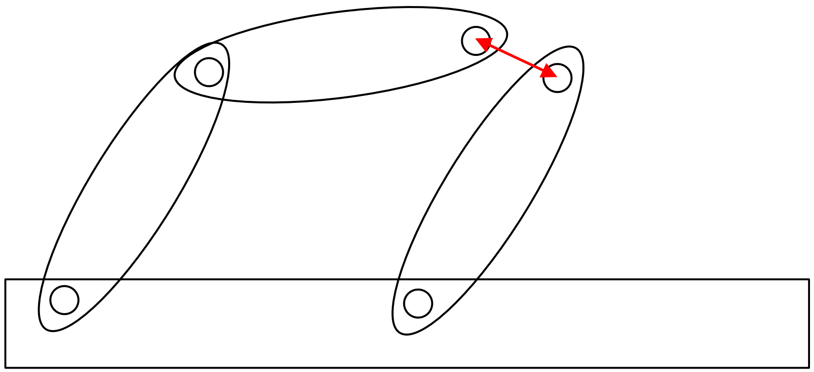
$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{h} + \boldsymbol{W}\boldsymbol{\lambda})$$

$$\boldsymbol{g}(\boldsymbol{q}_0,t_0) = 0$$

$$\dot{\boldsymbol{g}}(\boldsymbol{q}_0,\dot{\boldsymbol{q}}_0,t_0) = 0$$

Numerical problems and Stabilization

$$\ddot{q} = M^{-1}(h + W\lambda)$$
$$g(q_0, t_0) = 0$$
$$\dot{g}(q_0, \dot{q}_0, t_0) = 0$$

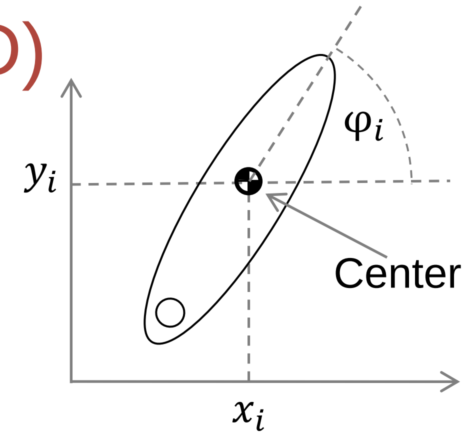


Numeric errors during integration can accumulate and break constraints

Baumgarte Stabilization:

Correct these errors during integration by replacing $\ddot{g} = 0$ by $\ddot{g} + 2\gamma\dot{g} + \gamma^2 g = 0$ (change ζ accordingly)

Body Coordinate Representation (2D)



For each body i glbal coordinates x_i, y_i, φ_i
mass m_i and moment of inertia I_i

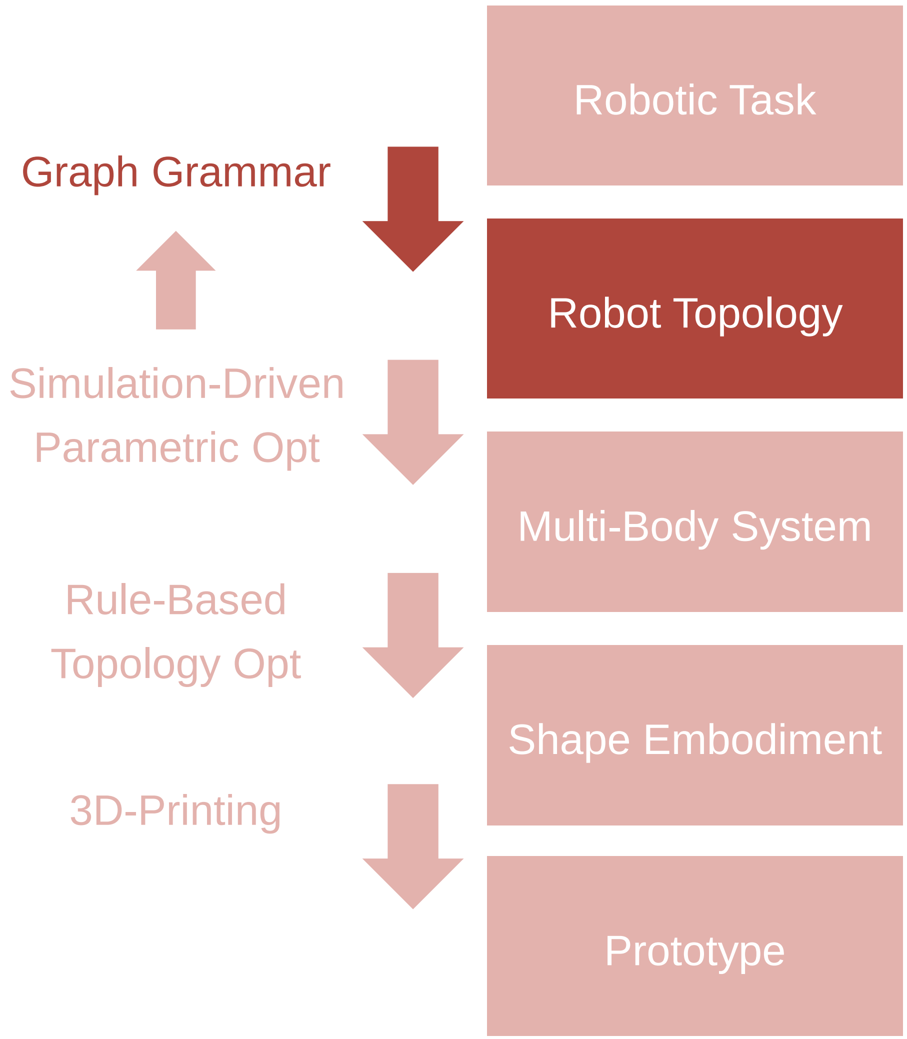
System coordinates: $\boldsymbol{q} = (x_1, y_1, \varphi_1, \dots, x_N, y_N, \varphi_N)^T$

Mass matrix: $\boldsymbol{M} = diag(m_1, m_1, I_1, \dots, m_N, m_N, I_N)$

Forces (here gravity only): $\boldsymbol{h} = (0, -m_1 g, 0, \dots, 0, -m_N g, 0,)^T$

Vector of Constraints to model joints: Same as in Lecture 3

Robot Topology Design Synthesis

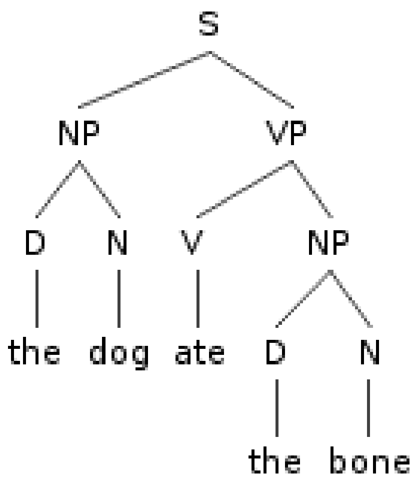


- Robot topology represented by graph
- Grammar rules used to automatically generate new systems

Engineering Design + Computing Laboratory

Origins of Transformational Grammar Rules in Linguistics

- A language is undefinable except for its grammar
  - proper ways to form valid statements
- Generative Grammars
  - Noam Chomsky - 1956
  - Rules that collectively define a language of feasible states
- A rule represents heuristic knowledge

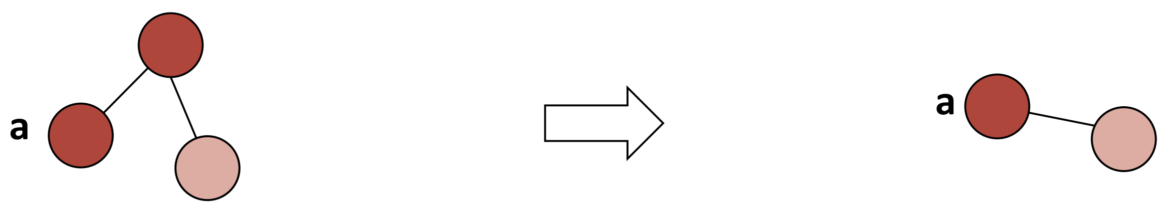


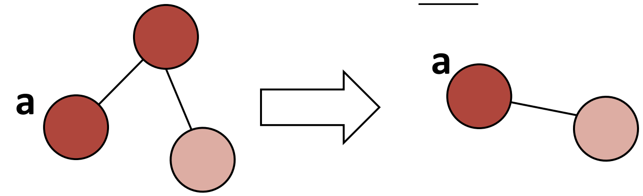
Graph Grammar(I)

- Graph rewriting system
- Rules used to change graph
  - Application conditions: Where the rule can be applied
  - Application procedure: What it does to the graph
- Rules represent heuristic knowledge

Left hand side of rule:
Pattern to find in graph

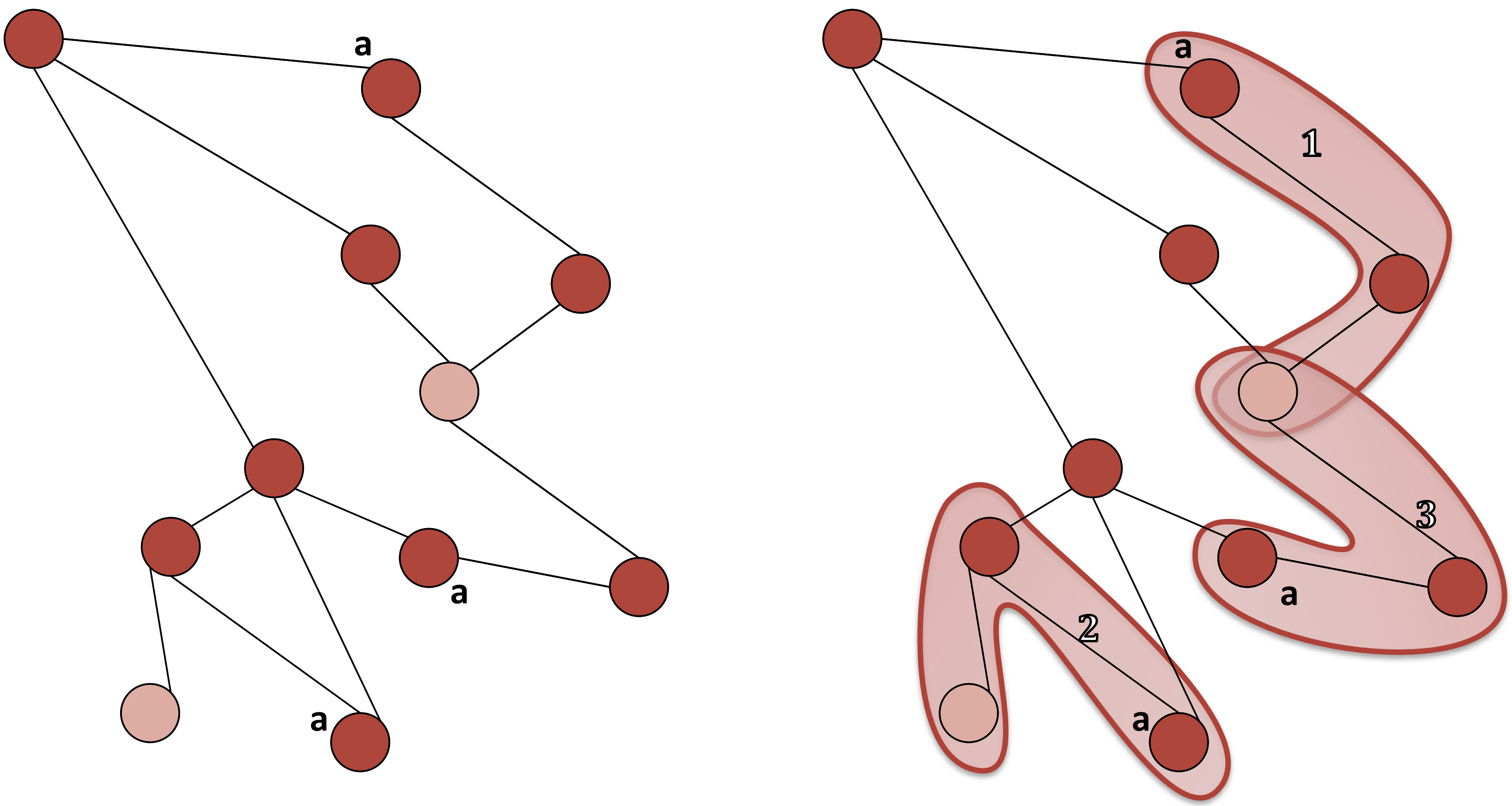
Right hand side of rule:
Replaces left hand side

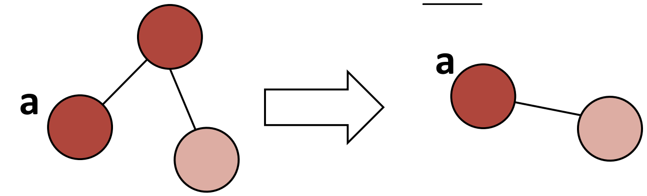




Graph Grammar(II)

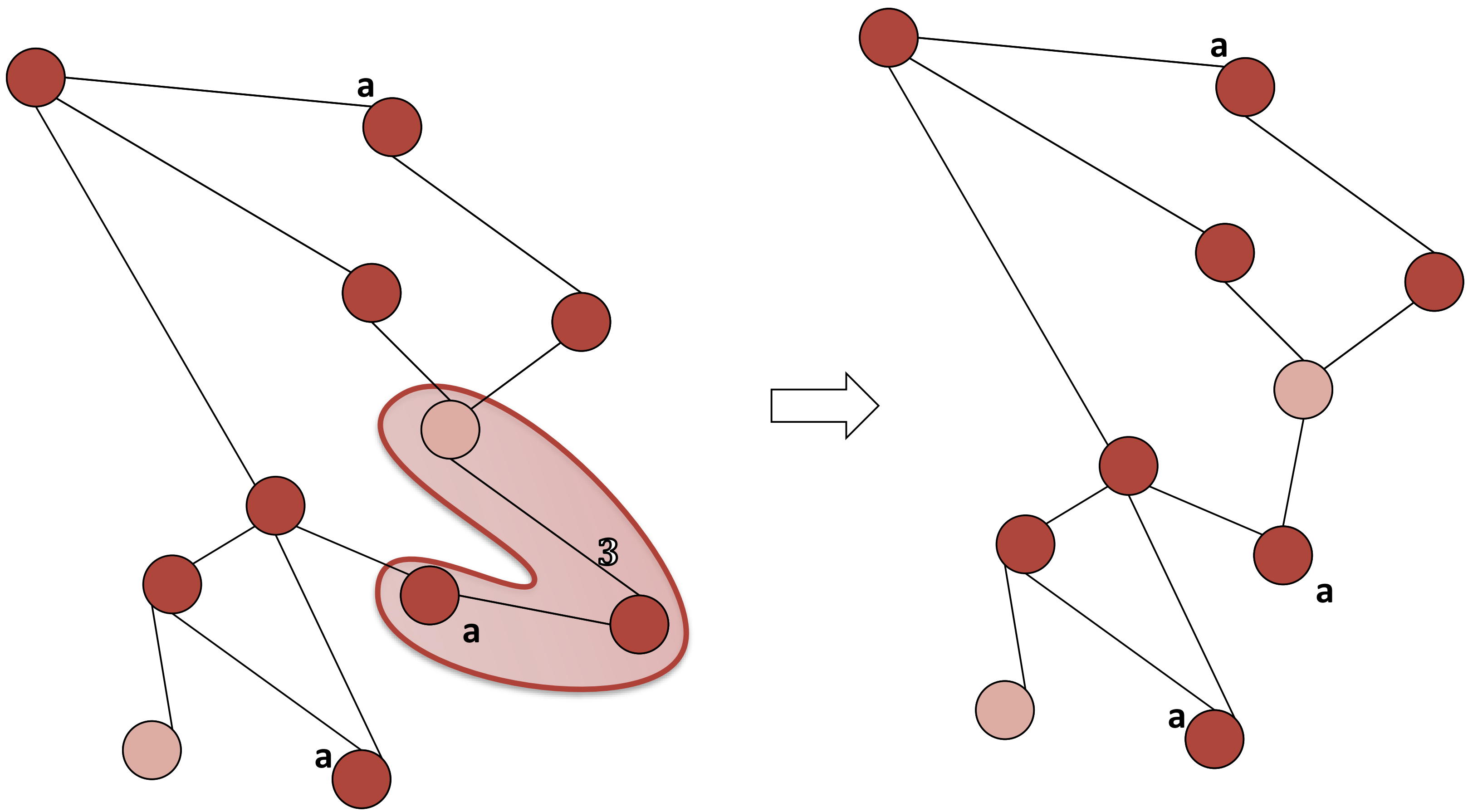
Recognize left hand side of rule in graph



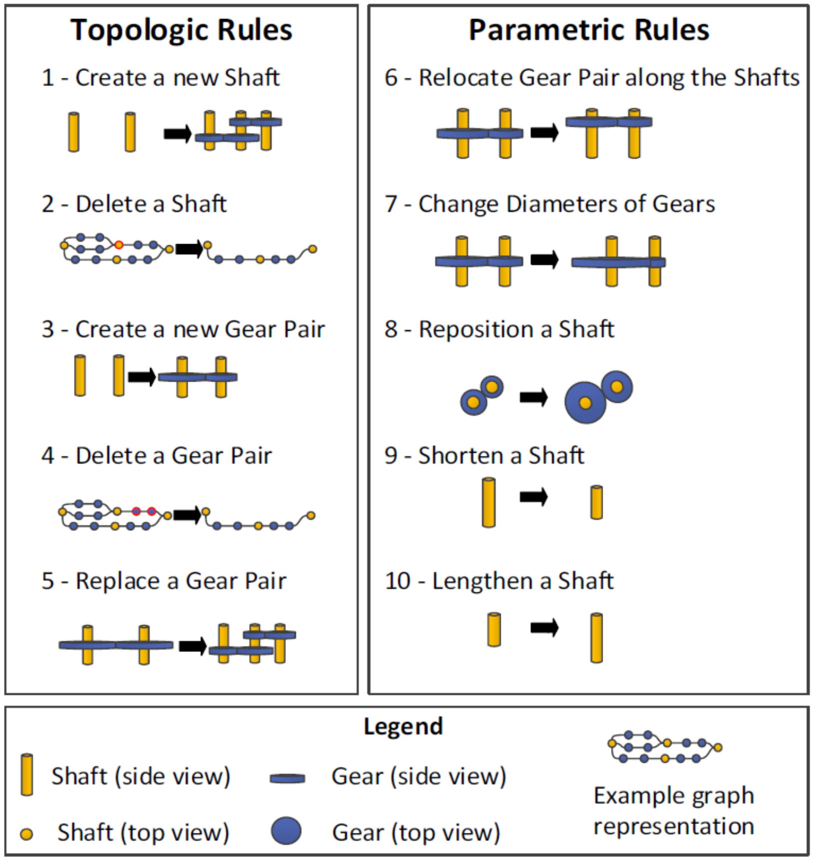


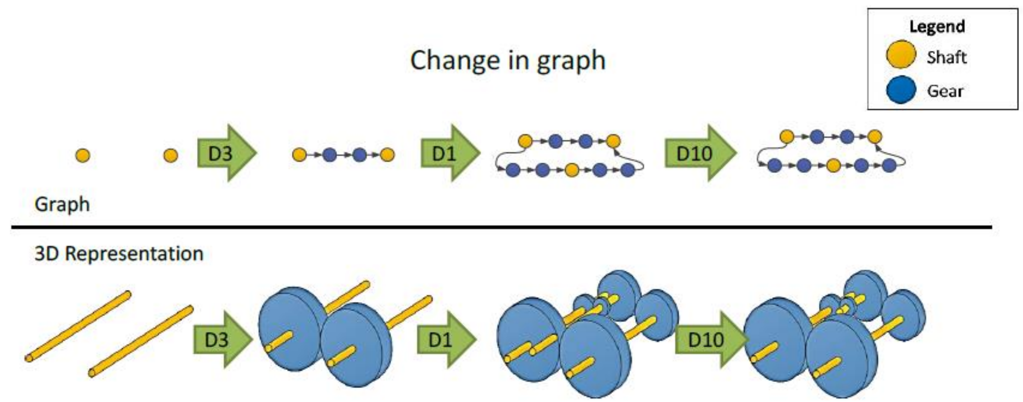
Graph Grammar(III)

Choose where to apply rule

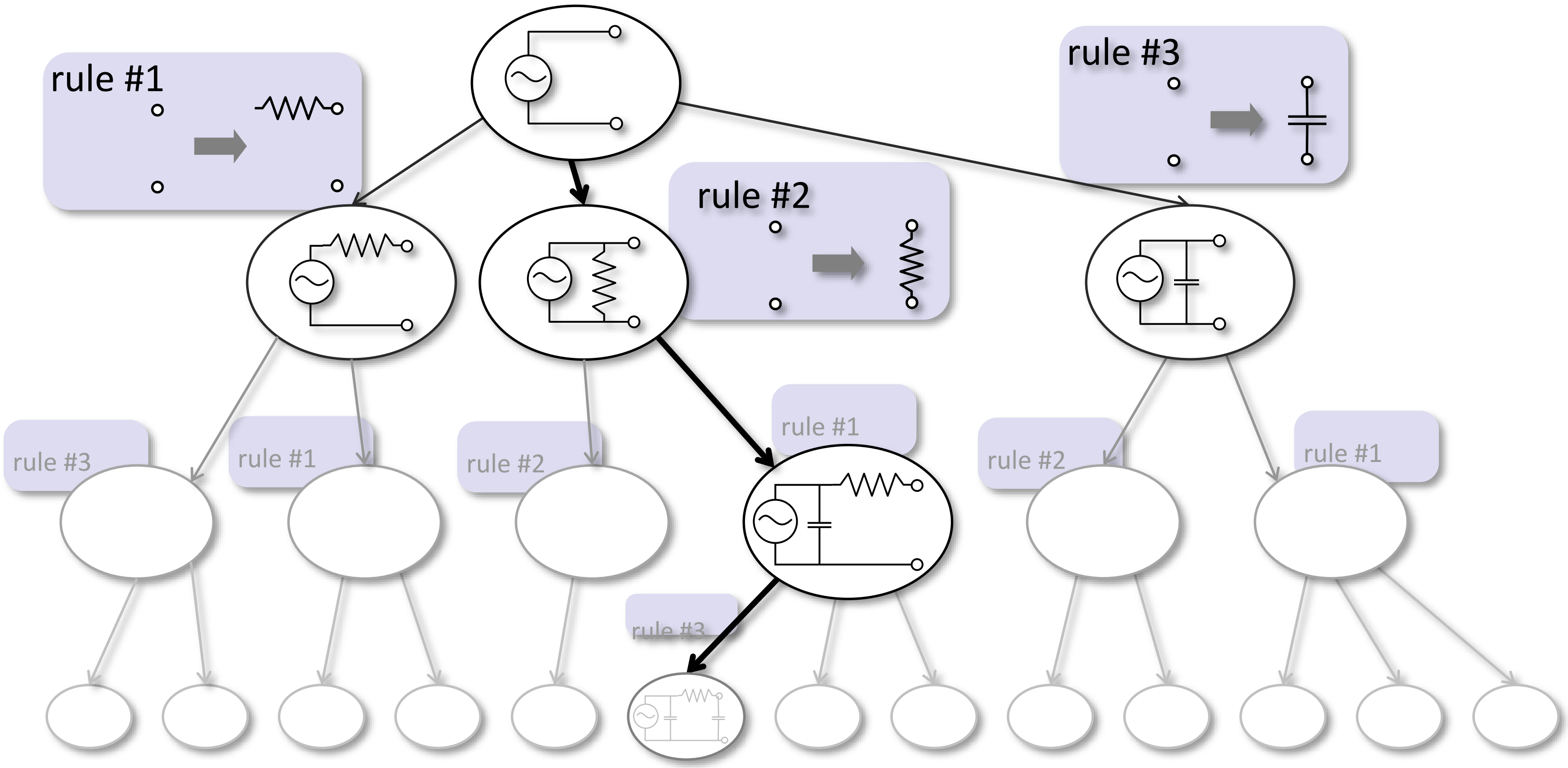


Example: Gear Box Design



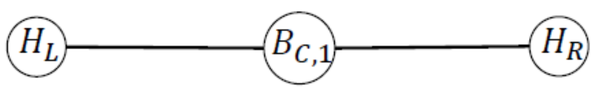


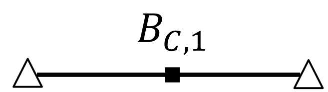
Example: Low-pass filters

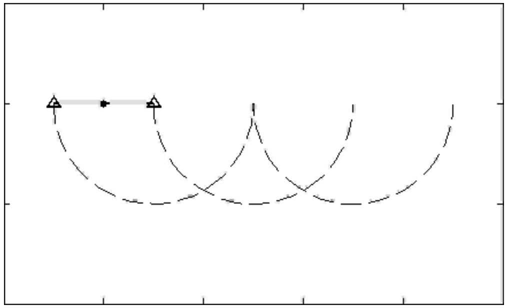


Design Rules for Passive Dynamic Systems (I)

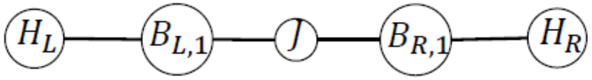
Graph Representation	Multibody System	Simulation

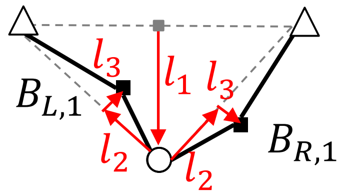


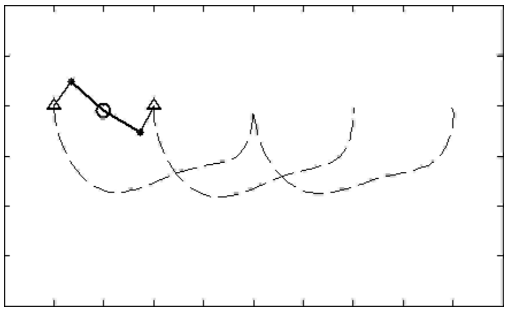




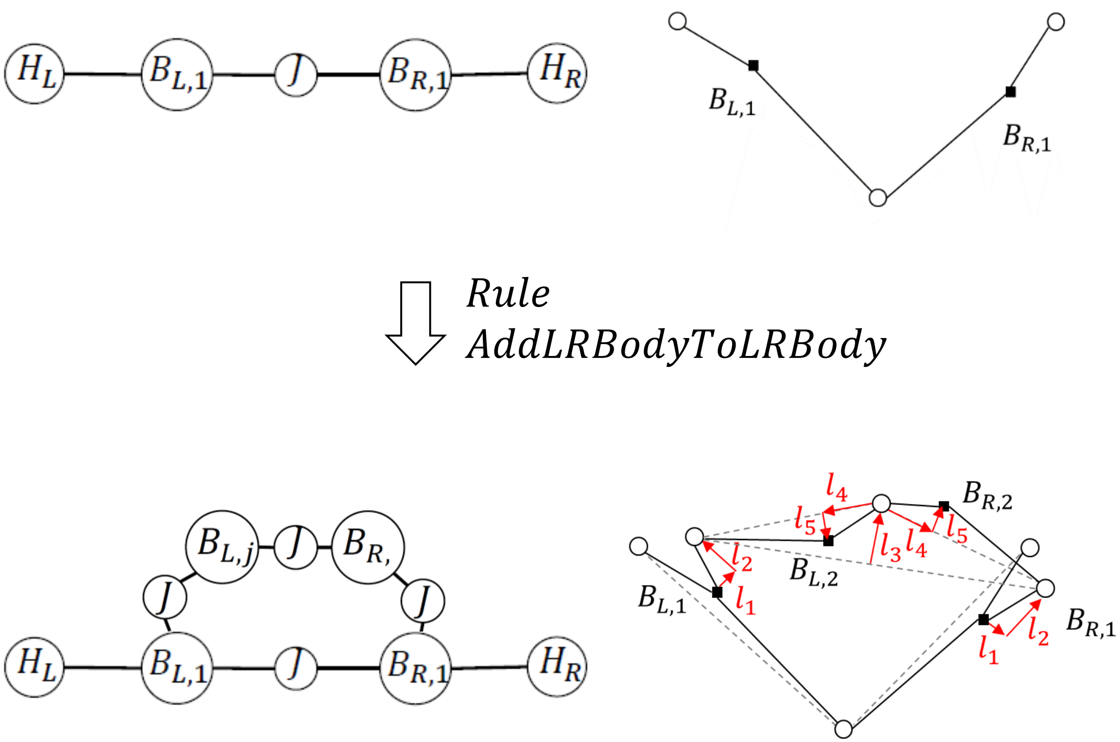
Rule
ReplaceCBodyByLRBody

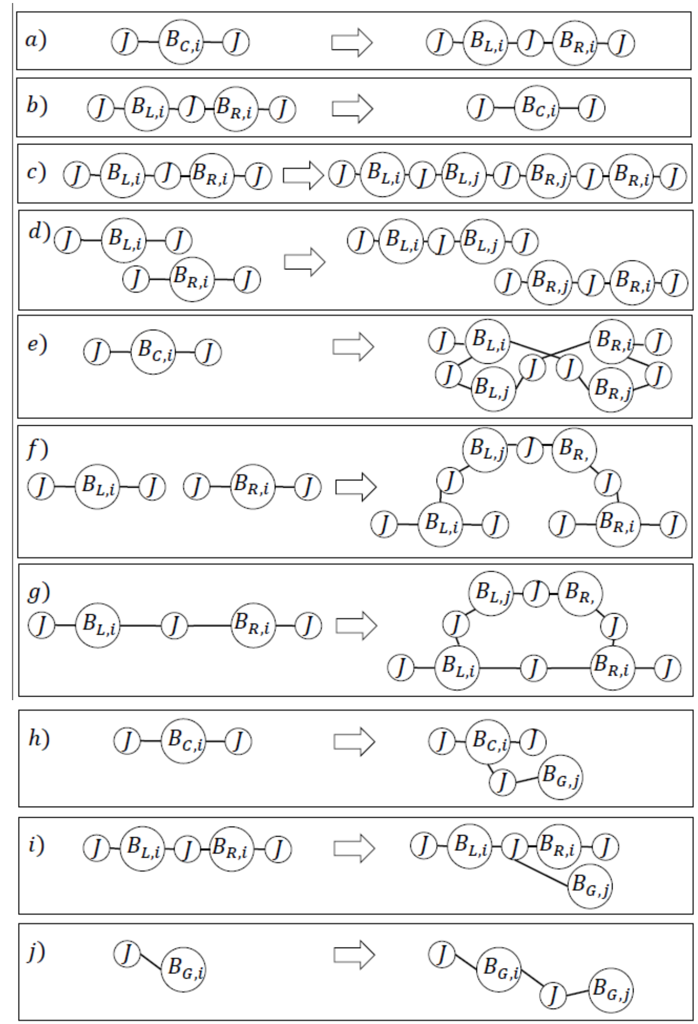






Design Rules for Passive Dynamic Systems (II)





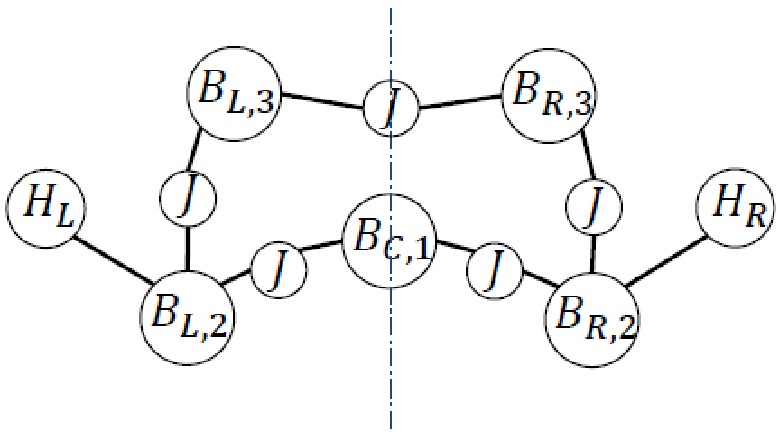
Symmetry for Brachiating (I)

Symmetry

- Is required for cyclic brachiating
- Similar as in walking between left and right leg

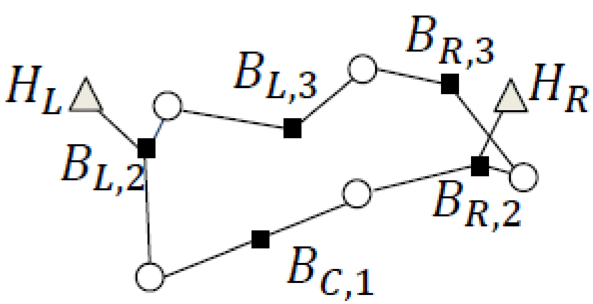
Symmetric Graph

- Rules generate symmetric configurations only
- Mirror symmetry



Symmetric Multibody System

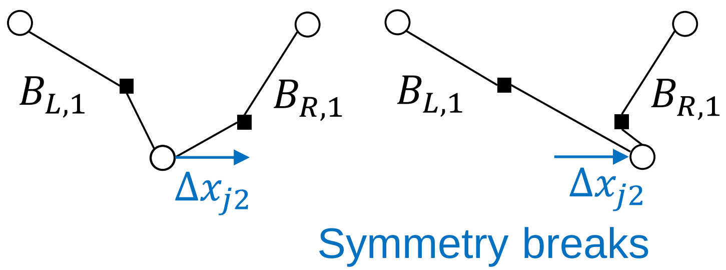
- Rules generate symmetric geometries only



Symmetry for Brachiating (II)

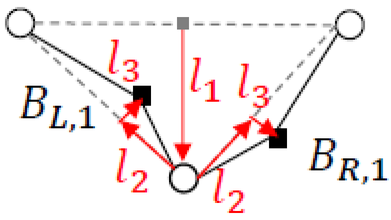
Arbitrary Parameterization

- Problem: Symmetry breaks

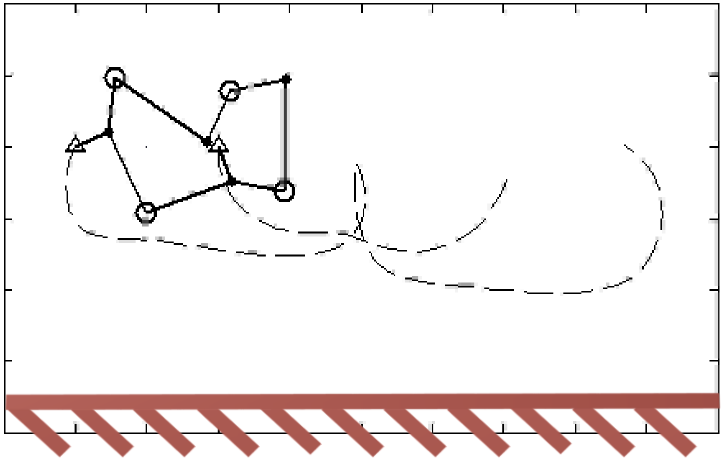


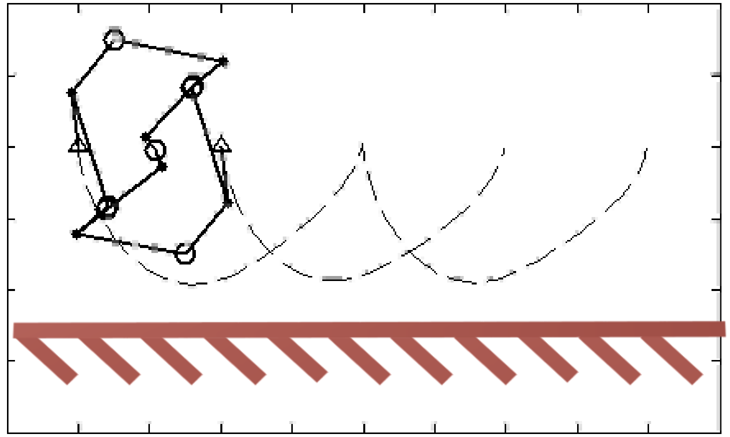
Symmetric Parameterization

- Symmetry is maintained when optimization variables are varied
- This is included in the design rules



Evaluation Criteria





Cyclic Locomotion

- Number of successful swings

$$f_{4a} = -n_{sw}$$

- Difference in states and hand position after first and last swing

$$f_{4b} = \Delta_{pos}(t_1)a_1 + \Delta_{vel}(t_1)a_2 + \Delta_{hand}(t_1)a_3 + \Delta_{hand}(t_{end})a_4$$

Space Requirement

- Lowest coordinate swept during the whole motion

$$f_5 = -y_{min}$$

Complexity

- Measured by the number of bodies

$$f_6 = -n$$

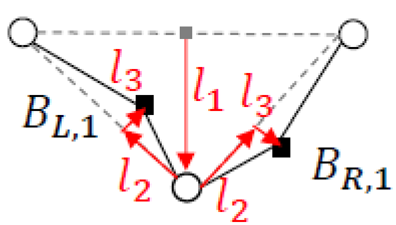
Synthesis and Optimization

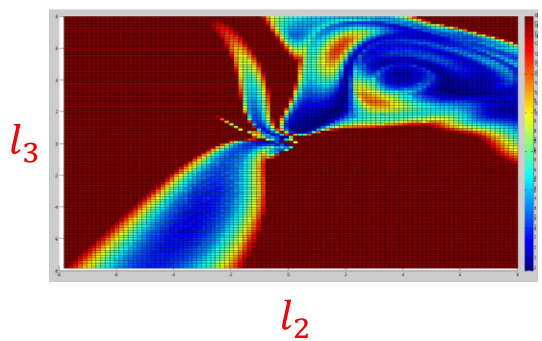
Parametric Optimization

- For each topology generated
- Multi-objective genetic algorithm (pop size: 200, generations: 80) From Matlab toolbox
- Highly non-linear, non-convex multi-modal optimization landscape

Topological Synthesis

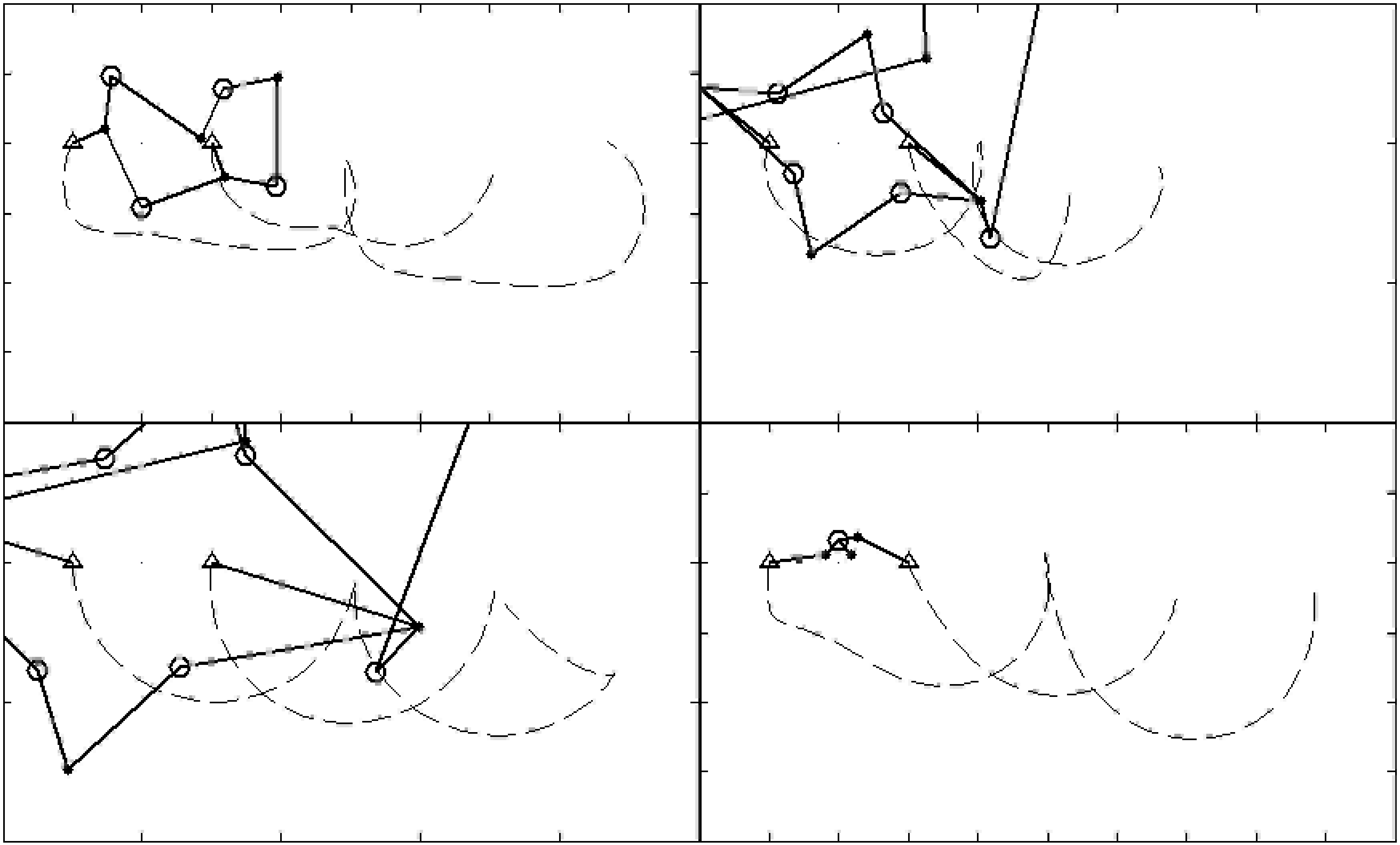
- Multi-objective burst algorithm (burst length: 3, max iterations: 500)





Cyclic locomotion:
blue: good performance
red: poor performance

Intermediate Solutions after some Generations



Results

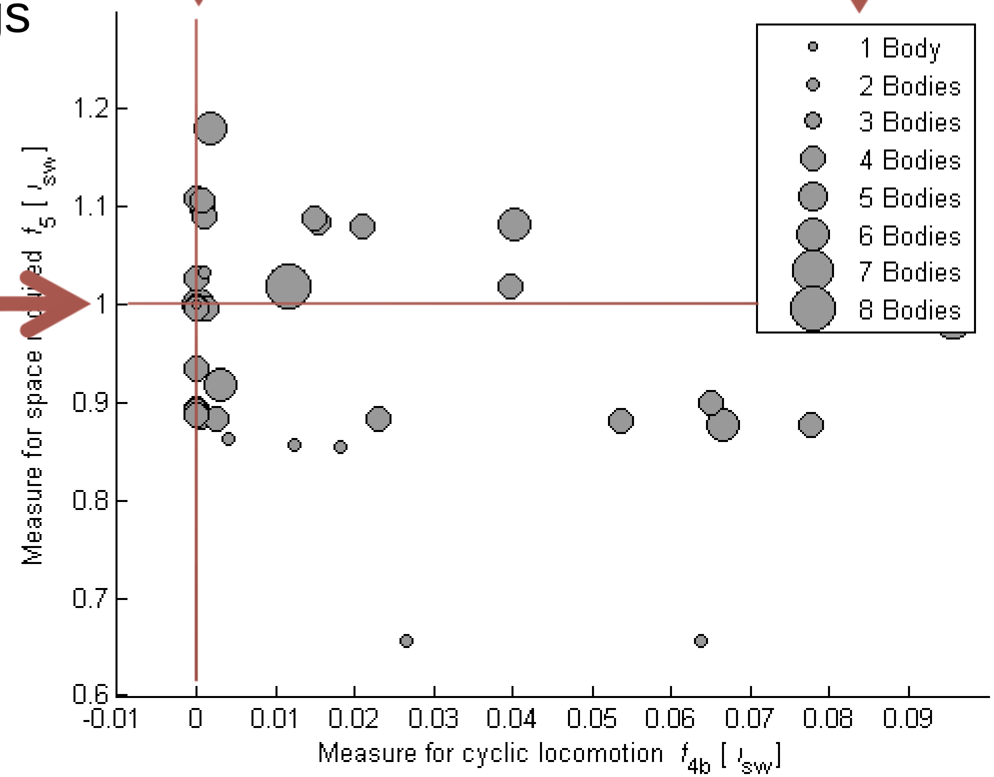
Evaluation Plot

- Final populations of eight different topologies
- All do three successful swings
- 3 Objectives

Ideal cyclic locomotion

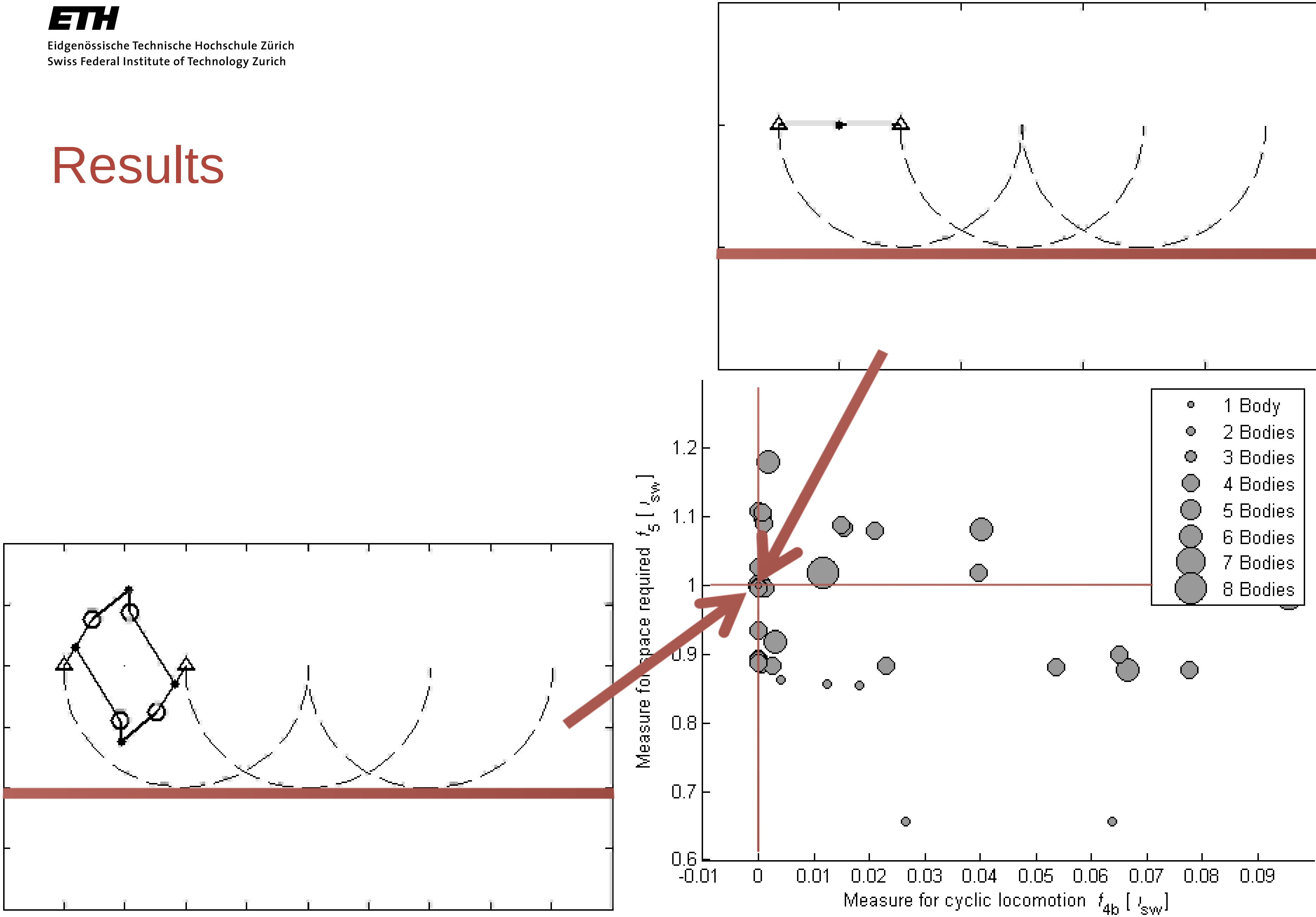
Number of bodies

Space requirement of single pendulum

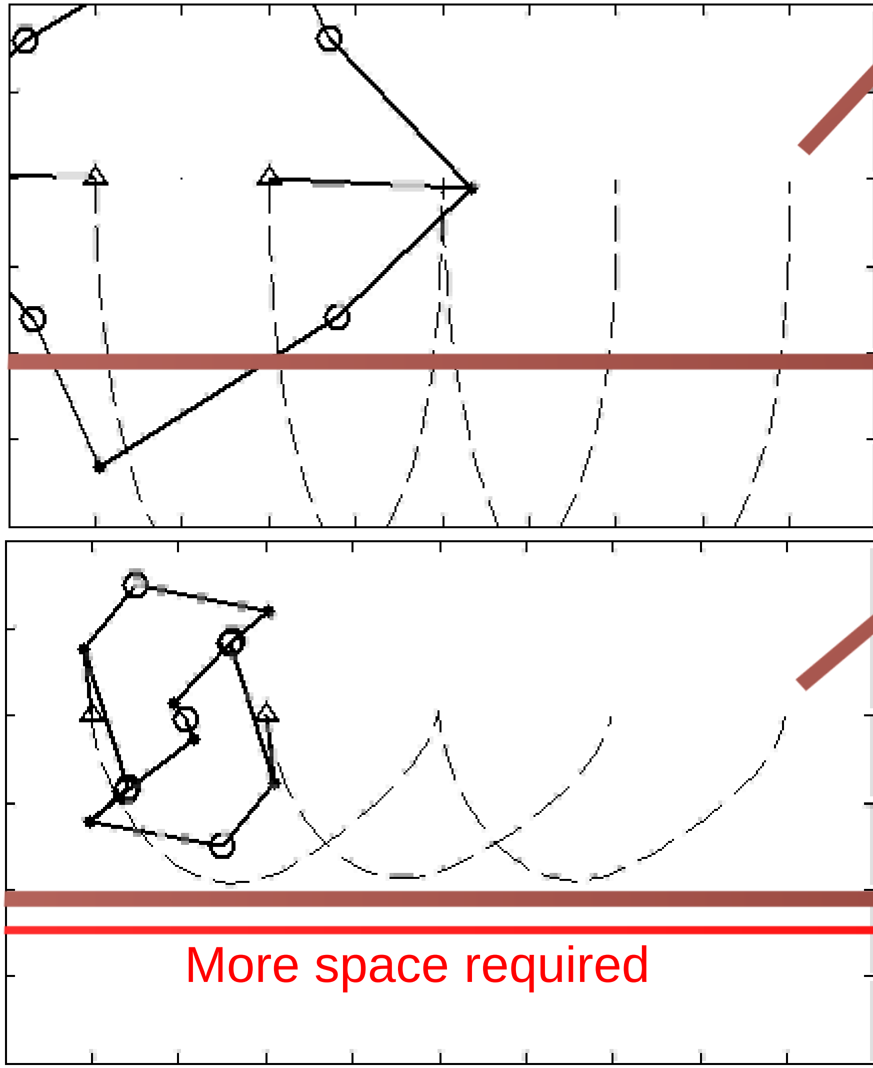


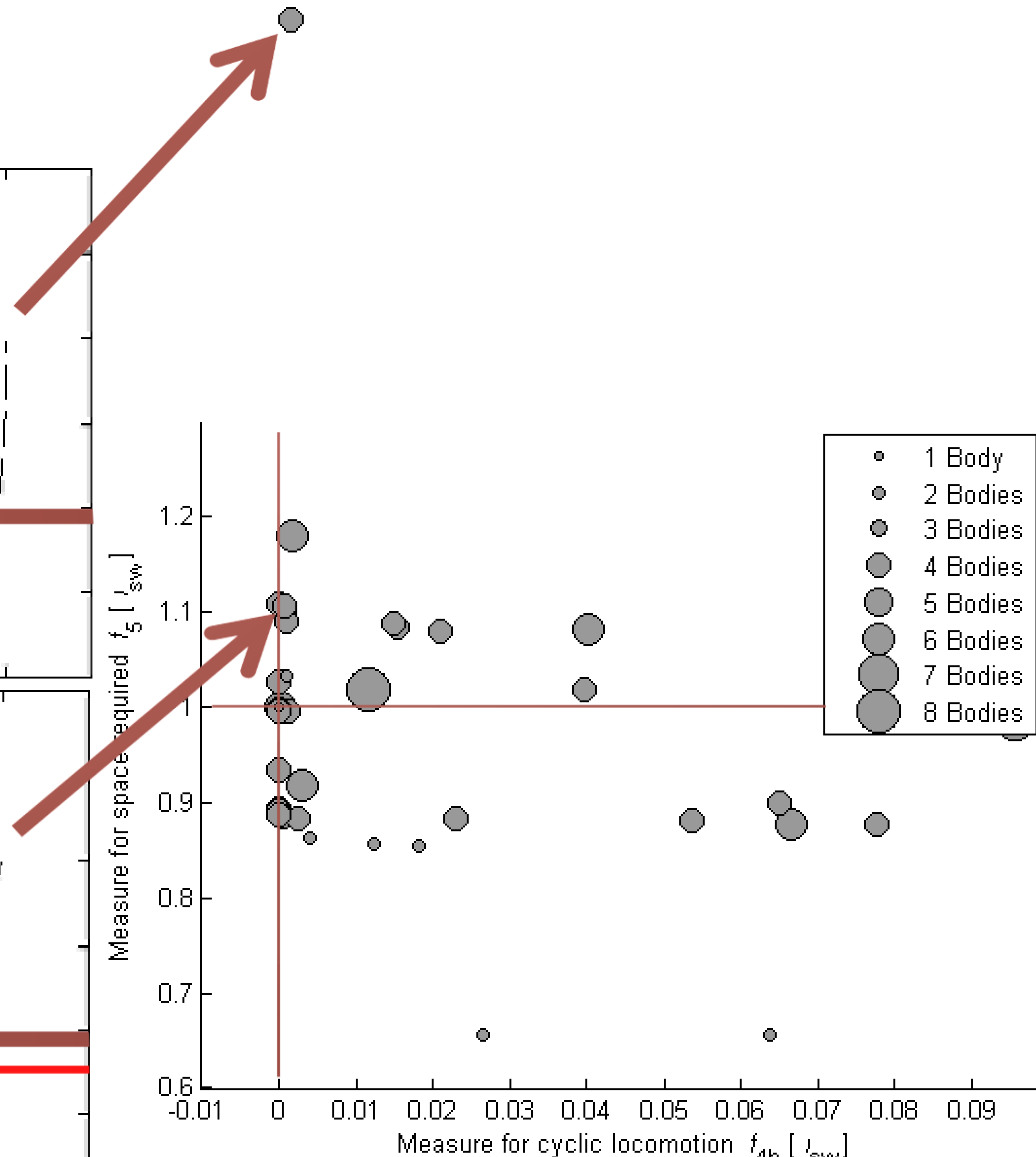
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Results



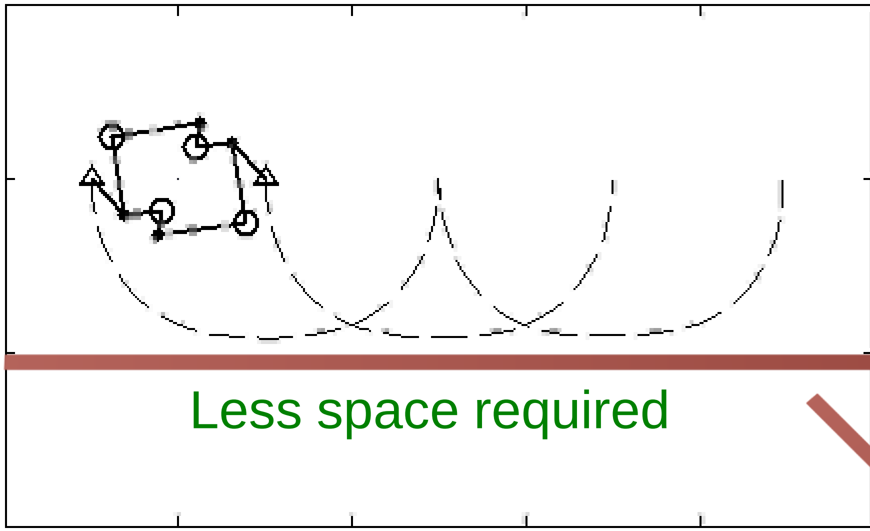
Results

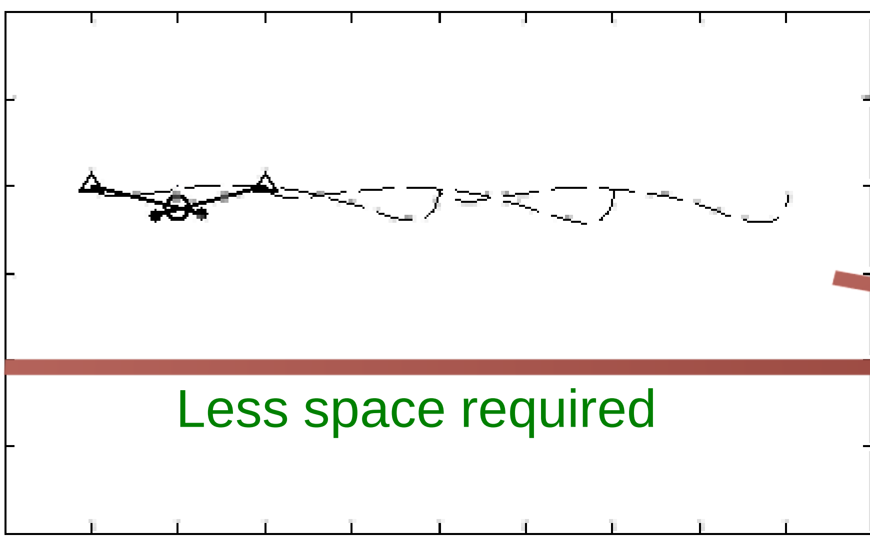

More space required

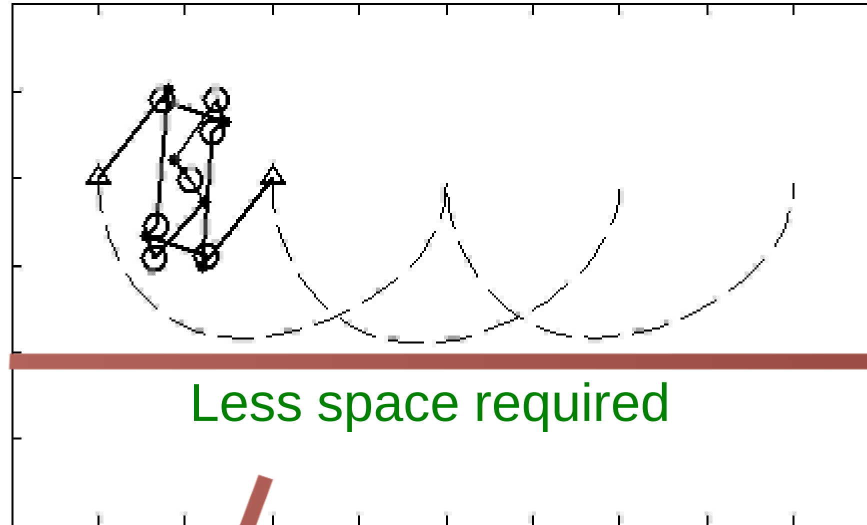

1 Body
2 Bodies
3 Bodies
4 Bodies
5 Bodies
6 Bodies
7 Bodies
8 Bodies
Measure for space required f_5 [l_{sw}]
Measure for cyclic locomotion f_{4b} [l_{sw}]
1.2
1.1
1
0.9
0.8
0.7
0.6
-0.01
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09

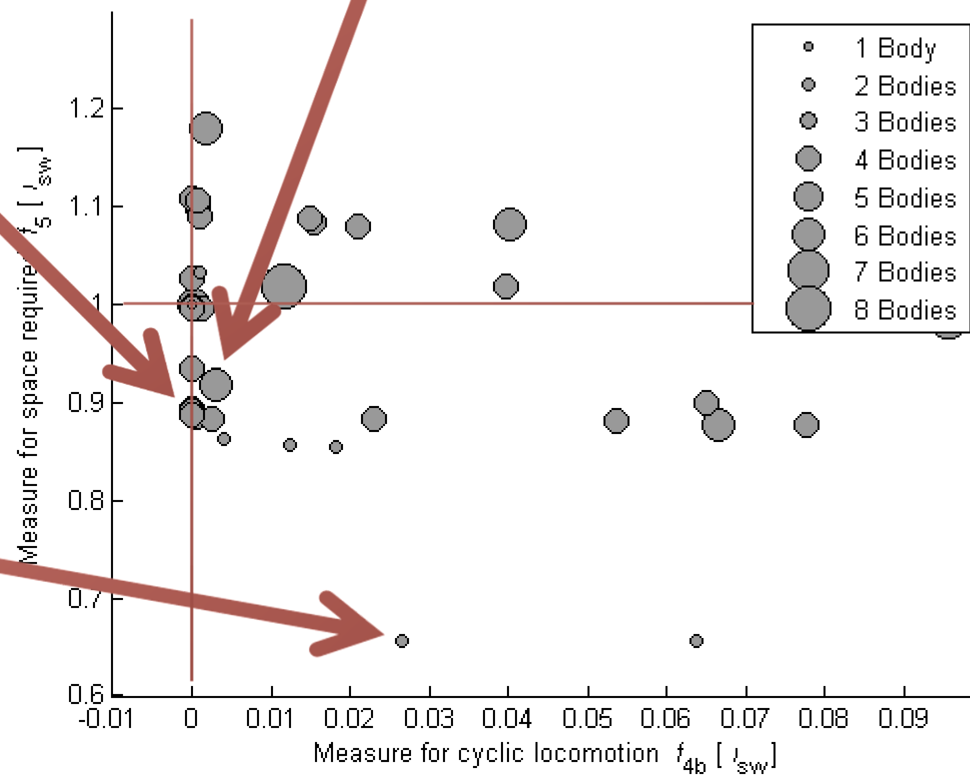
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Results

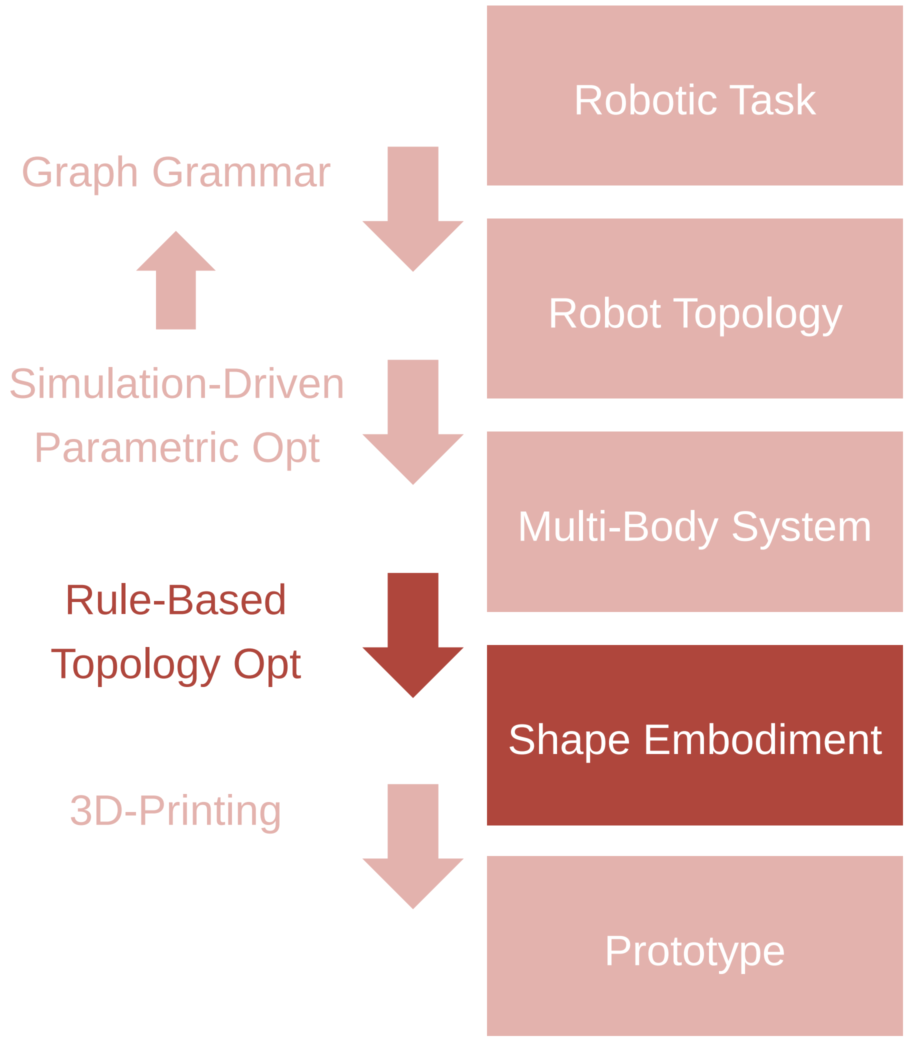








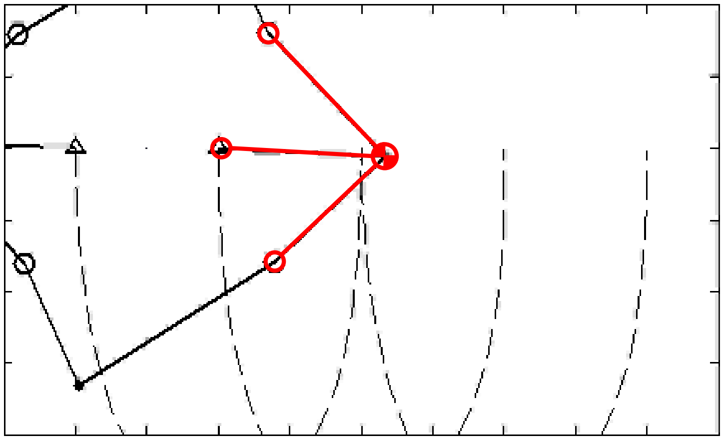
Shape Embodiment Design

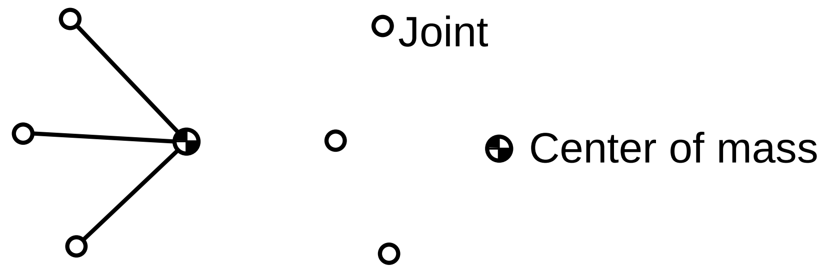


- Bodies defined by inertia properties only
- Topology optimization needed to find shapes of bodies

Automated Shape Design for Multi-Body Systems

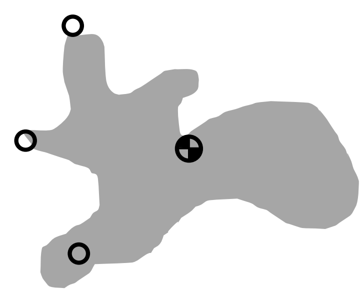
Multi-body system representation





Dynamic properties:
[Mass, moment of inertia, center of mass]

Find shape
- Matching dynamic properties
- Connecting all elements
- Avoiding collisions

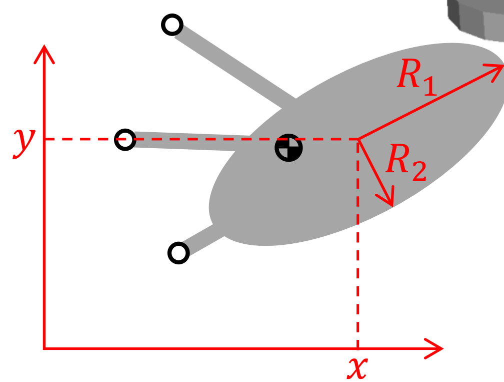


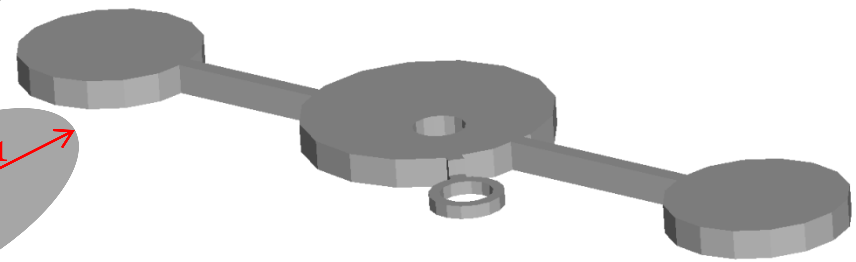
Assemble Geometric Primitives

Find shape with given dynamic properties

Ellipse

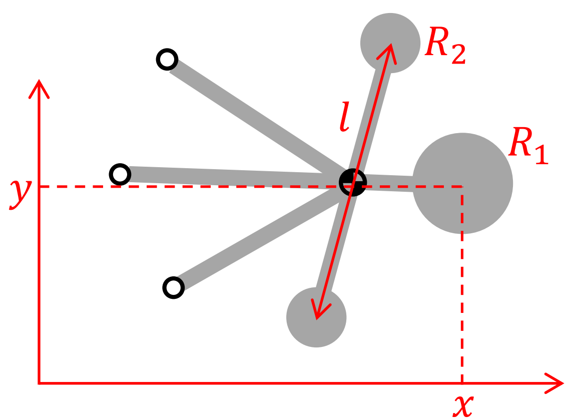
- 5 variables: x, y, R_1, R_2, α
- Sometimes no solution found
- Compact



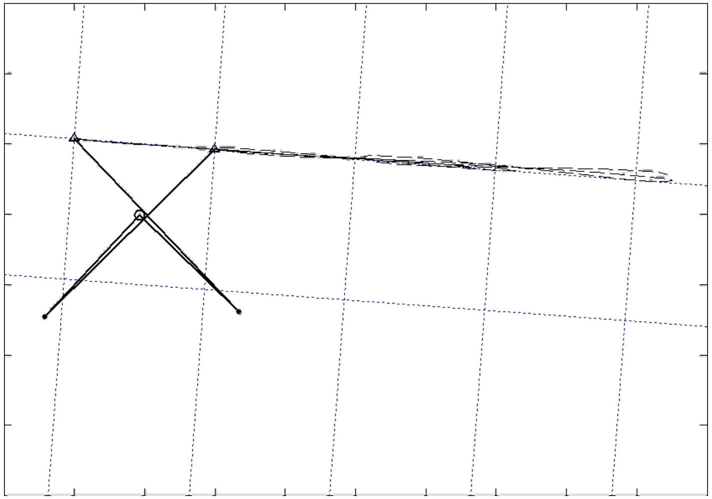


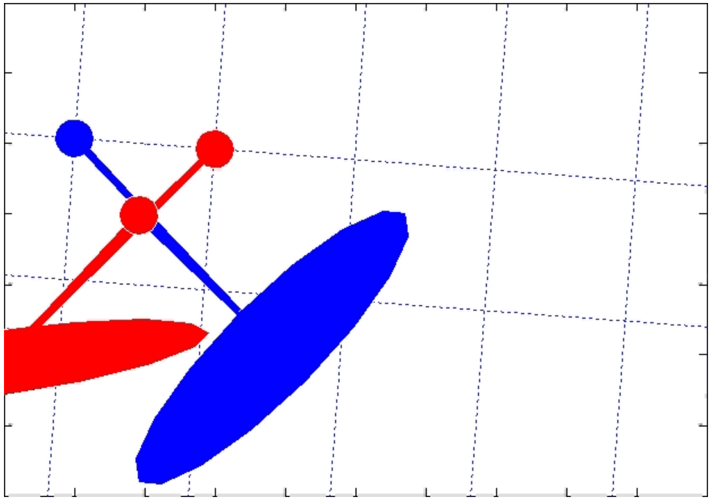
Three Circles

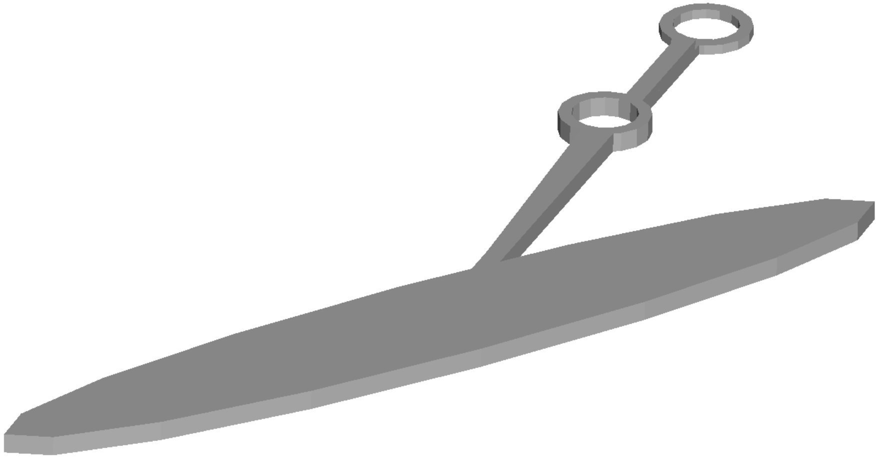
- 6 variables: $x, y, R_1, R_2, \alpha, l$
- Mass and Moment of inertia change more independently
- Uses more space



Assemble Geometric Primitives



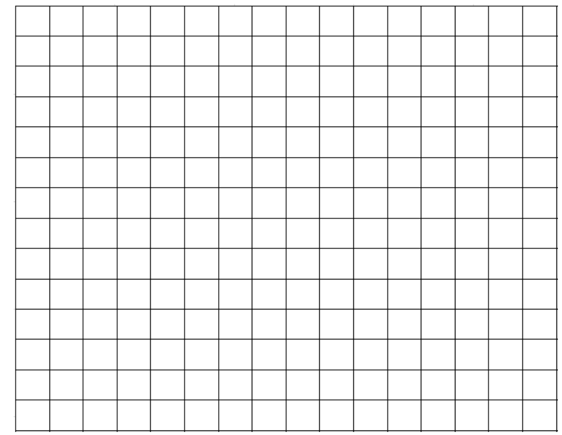




Finite Element Design Space

ρ_i Density of element *i* /
Material per 2D element *i*

m_0, I_0, c_{x0}, c_{y0} Desired dynamic properties



Feasibility/Satisfiability Problem:

$$\boldsymbol{m}^T\boldsymbol{\rho} = m_0$$

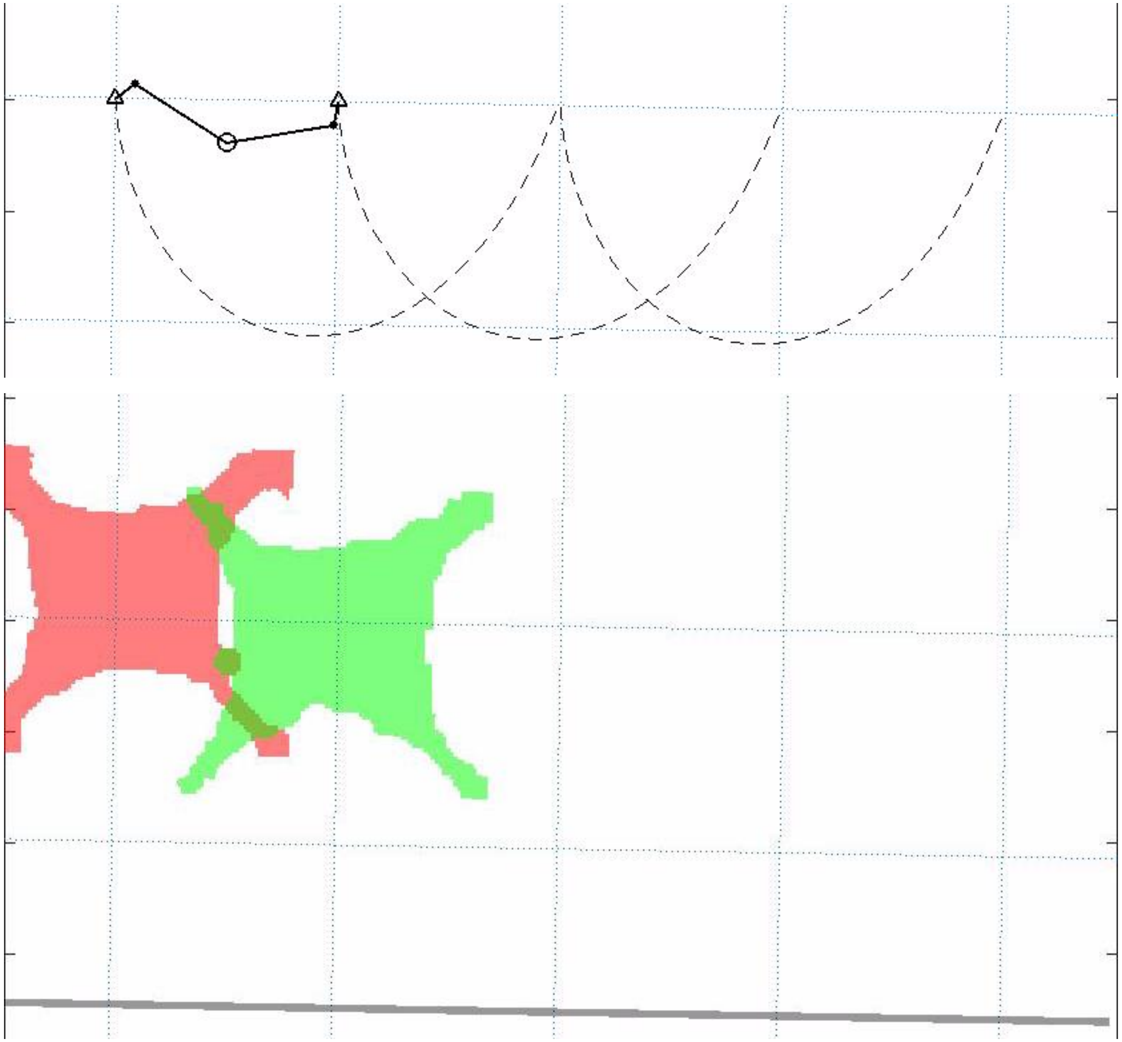
$$\boldsymbol{I}^T\boldsymbol{\rho} = I_0$$

$$\frac{\boldsymbol{c_x}^T\boldsymbol{\rho}}{\boldsymbol{m}^T\boldsymbol{\rho}} = c_{x0}$$

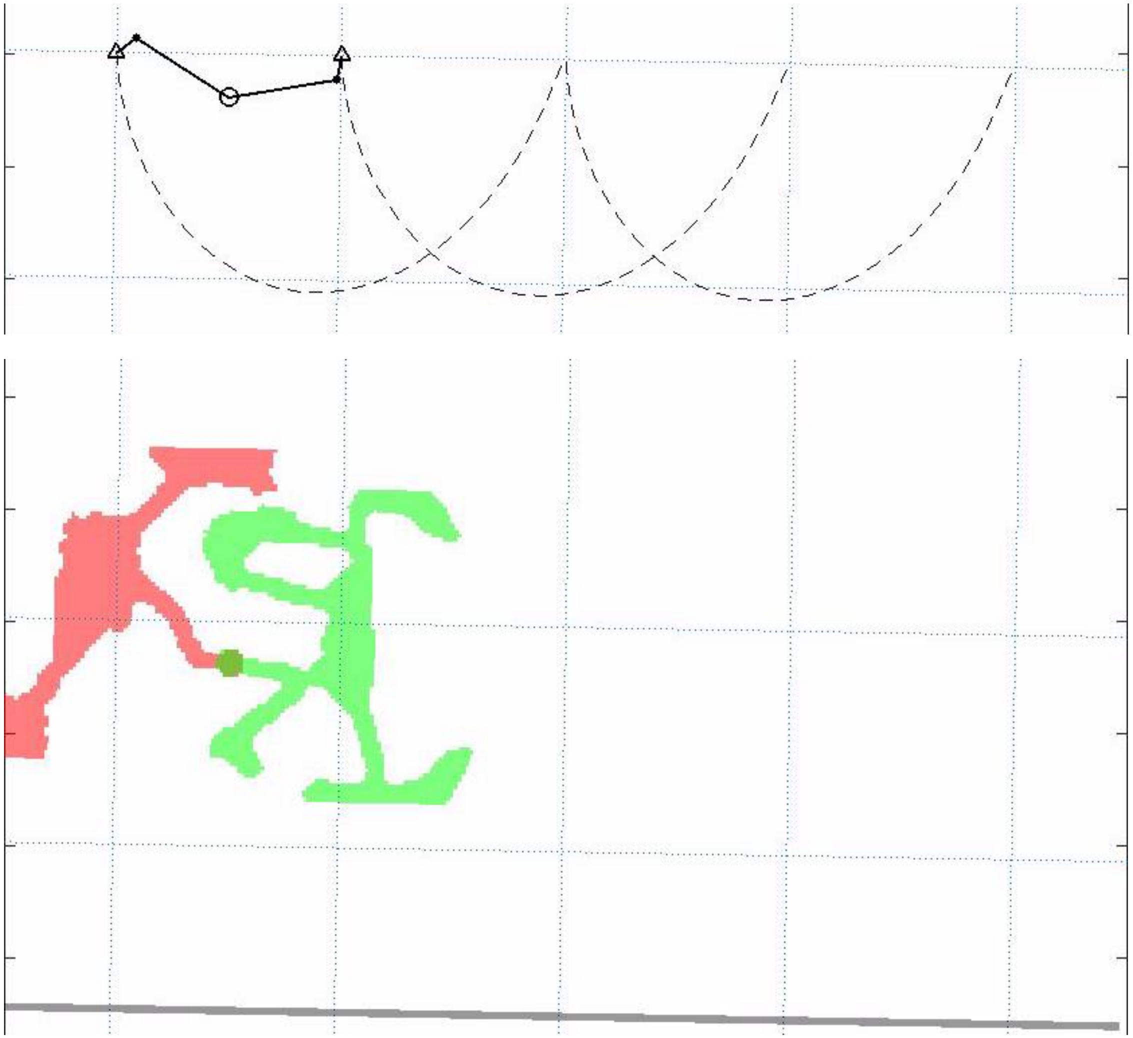
$$\frac{\boldsymbol{c_y}^T\boldsymbol{\rho}}{\boldsymbol{m}^T\boldsymbol{\rho}} = c_{y0}$$

$$0 \leq \rho_{min} \leq \rho_e \leq \rho_{max}$$

Finite Element Design Space



Finite Element Design Space



Fabrication

Robotic Task

Graph Grammar

Robot Topology

Simulation-Driven Parametric Opt

Multi-Body System

Rule-Based Topology Opt

Shape Embodiment

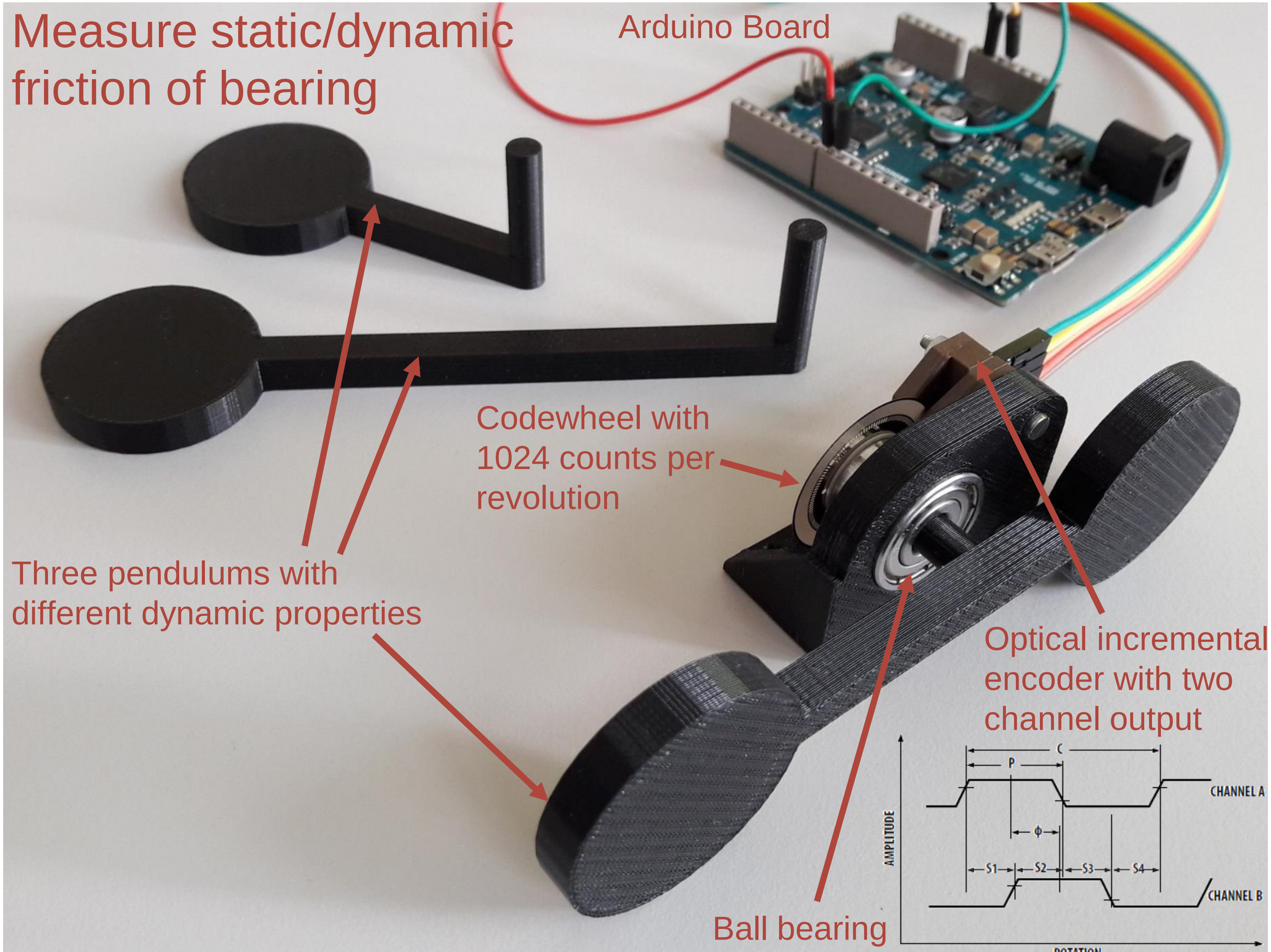
3D-Printing

Prototype

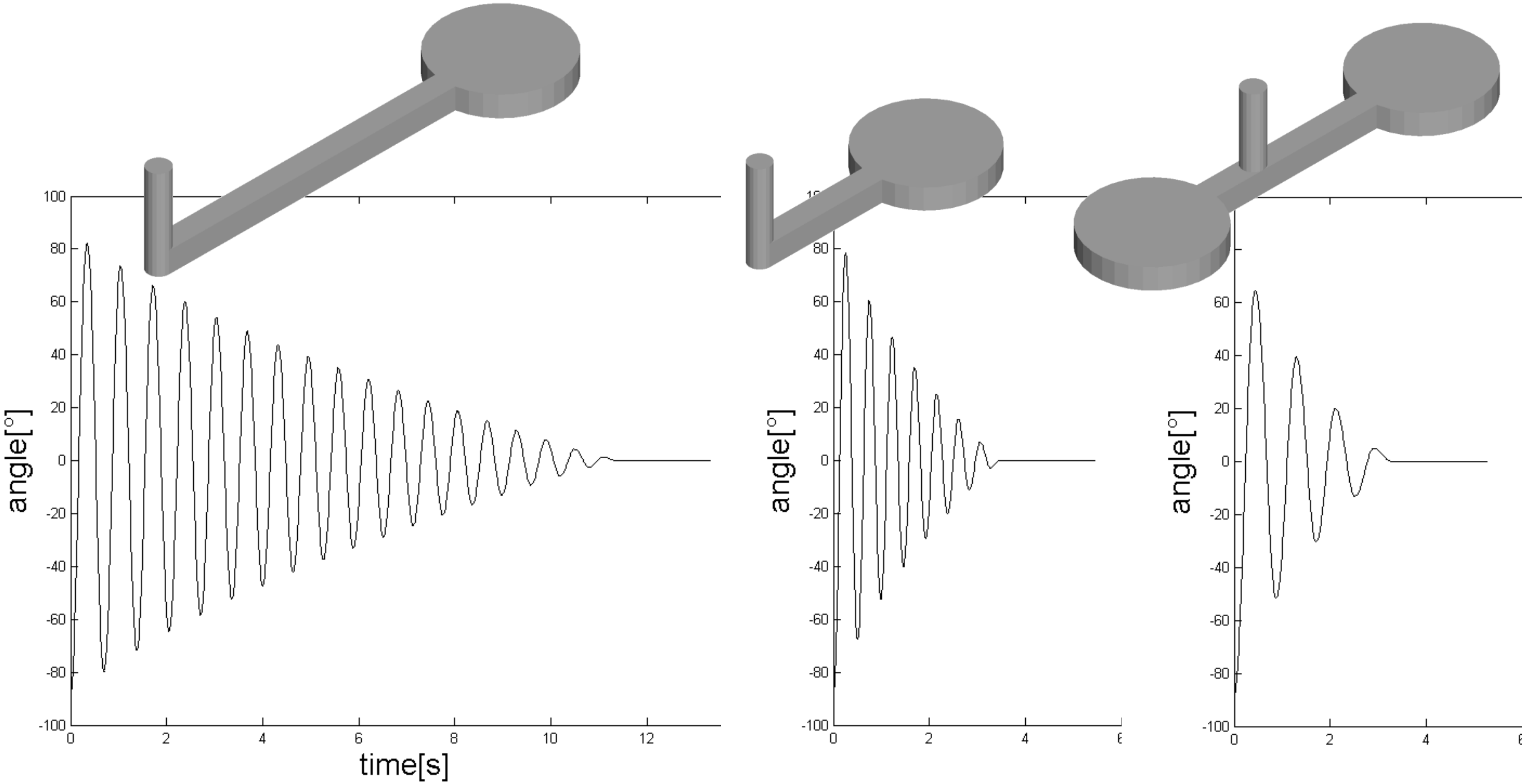
- Additive manufacturing works well for complex shapes
- Ball bearings for low friction

Experiment and Optimization based Design of a Passive Walking Robot, Fabio Modica, 2016, EDAC master thesis

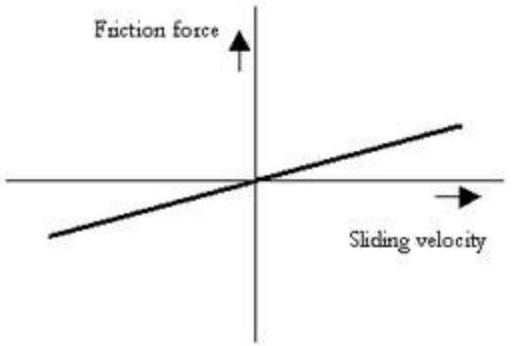
Measure static/dynamic friction of bearing



Measurement results



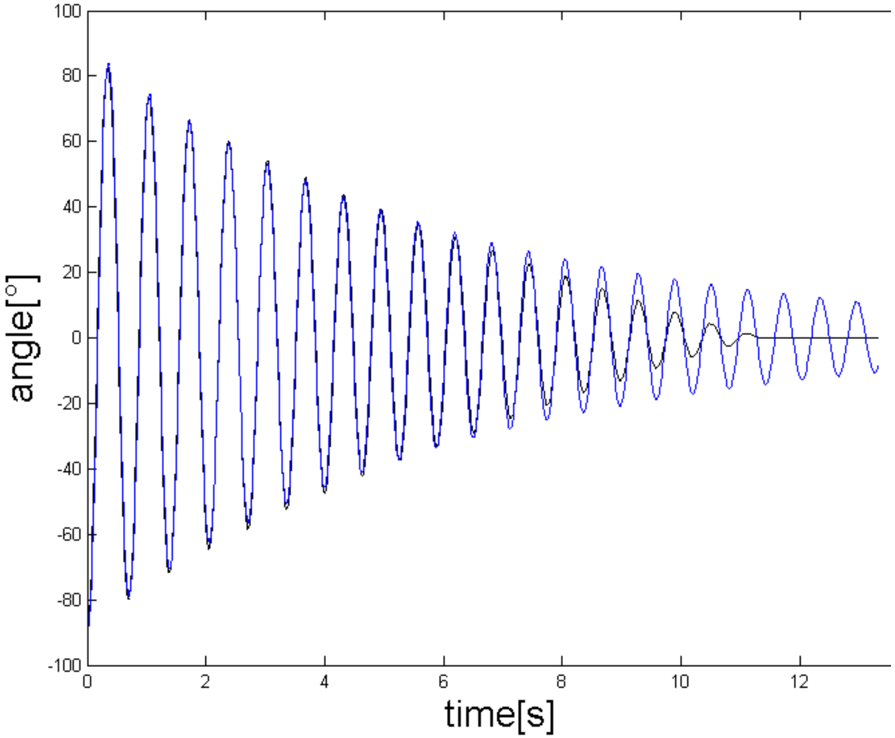
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

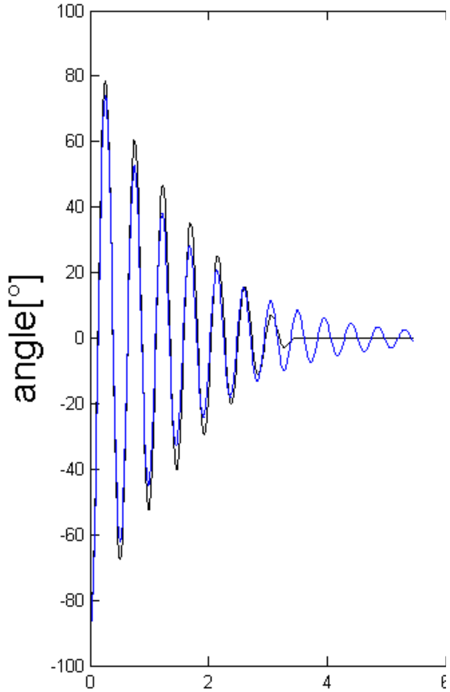


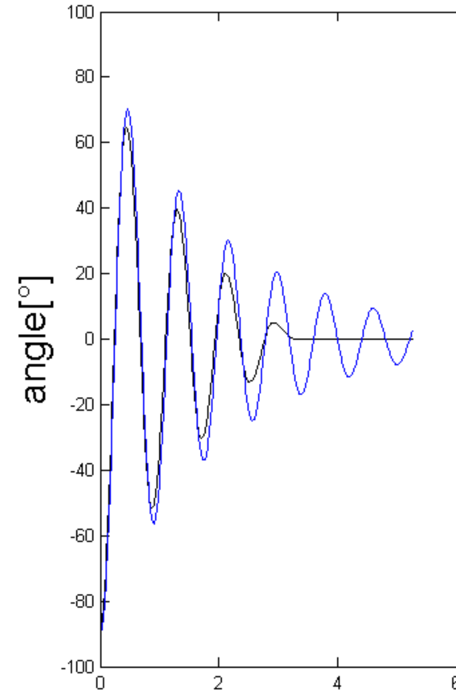
Fit Measurement to Simulation Model

$$\ddot{\varphi} I = -mglsin(\varphi) - d_F \dot{\varphi}$$

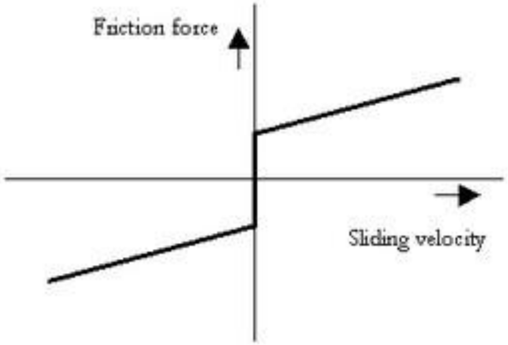
Friction Torque vellocity proportional (viscous friction):
Standard in robotics, good properties of ODE and control problem





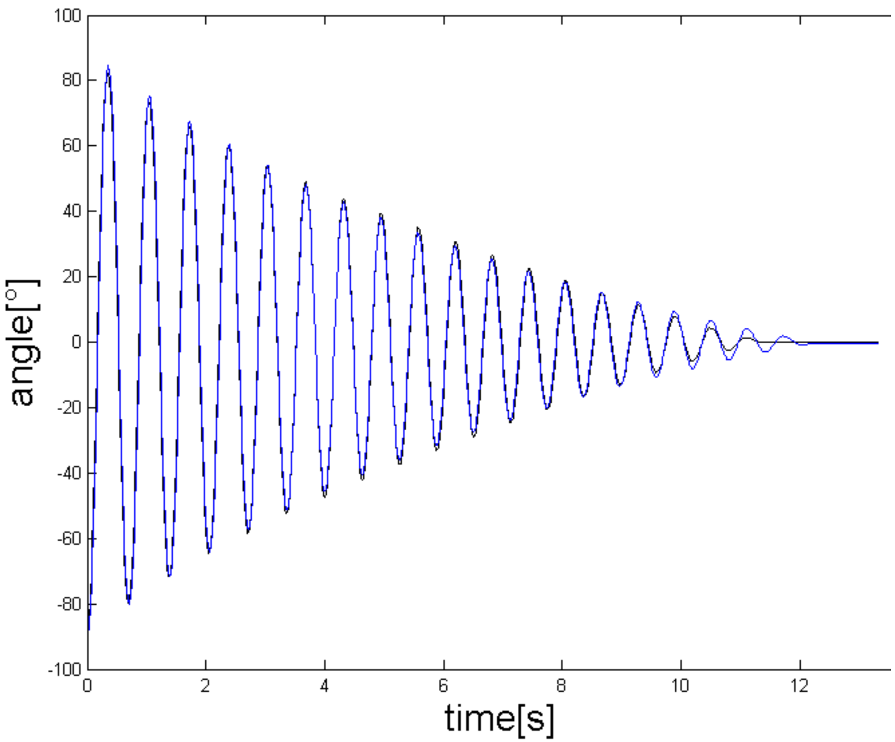


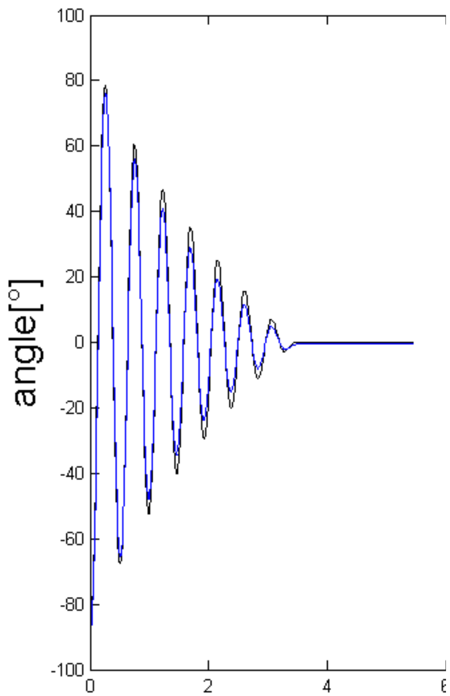
Fit Measurement to Simulation Model

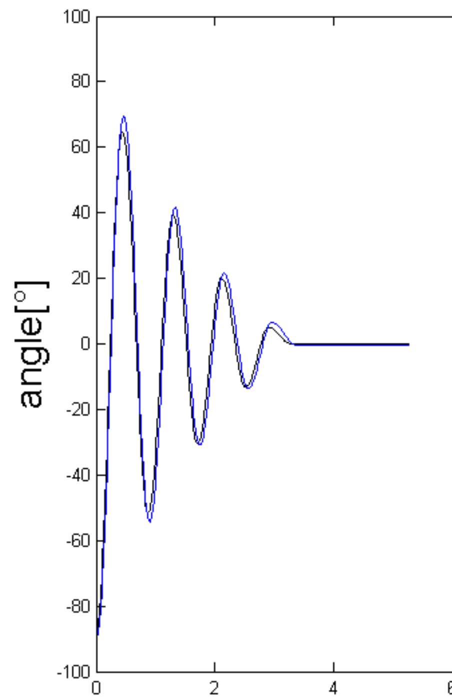


$$\ddot{\varphi} I = -mglsin(\varphi) - T_F$$

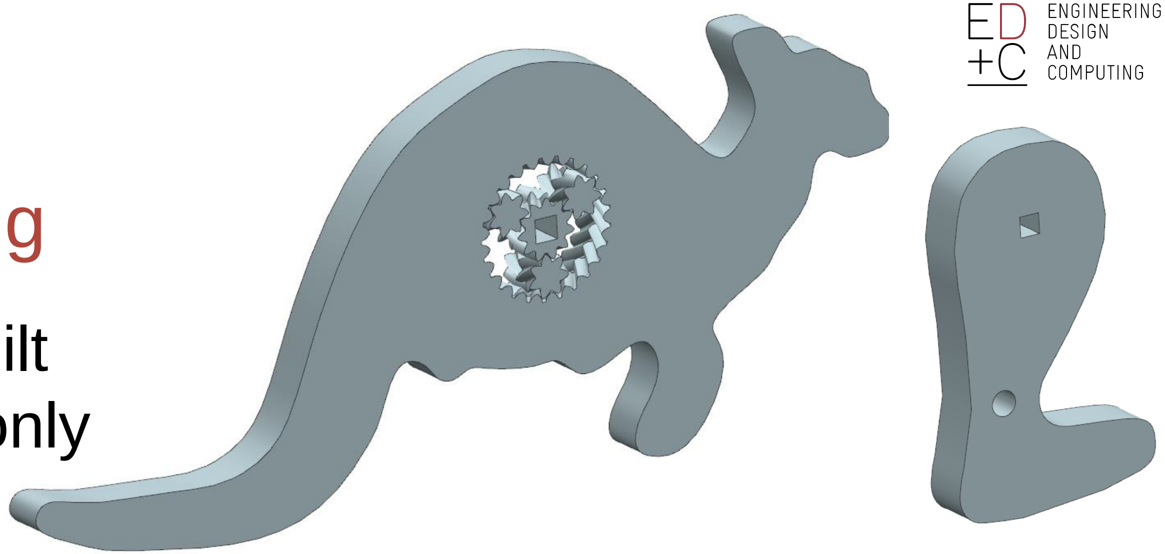
Combination of viscous friction and coulomb friction



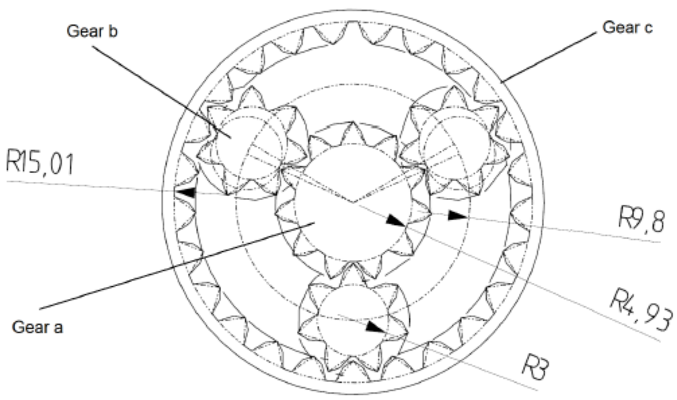


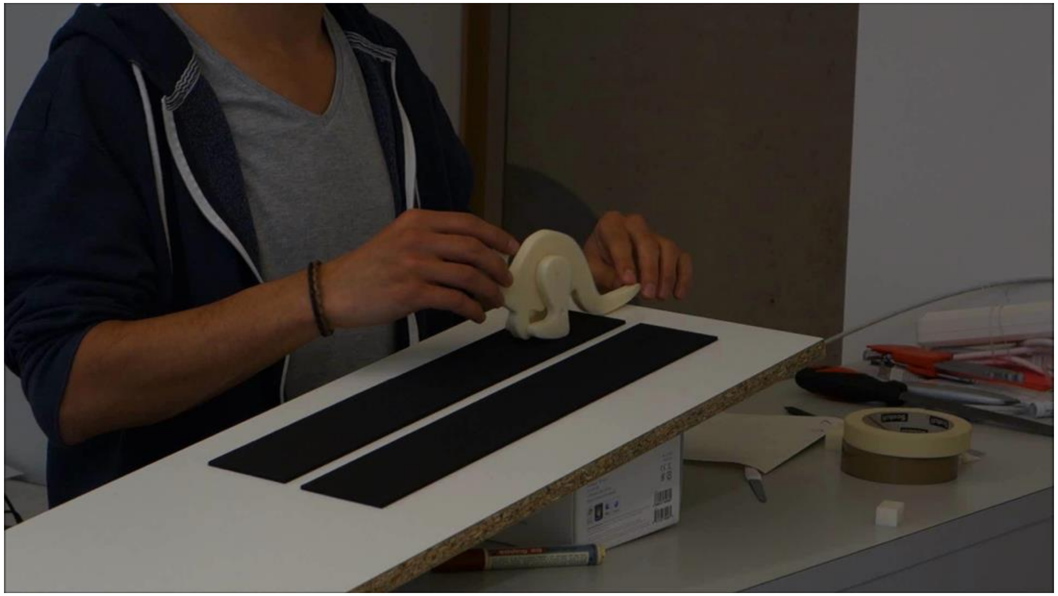


3D-Printed Bearing



- Passive walker built using FDM parts only
- Printed in one job





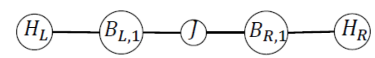
- Planetary gear bearing with clearance adapted to our FDM machine
- Very robust gait

CDS of Passive Dynamic Systems - Overview

Robotic Task

Graph Grammar ↓

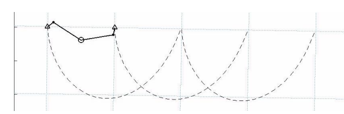
Robot Topology



Problem specific rules generate topologies preserving symmetry

Simulation-Driven Parametric Opt ↓

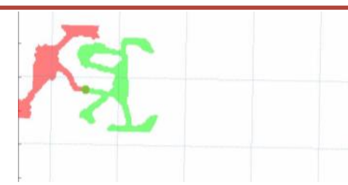
Multi-Body System



Evaluation based on system trajectories

Rule-Based Topology Opt ↓

Shape Embodiment



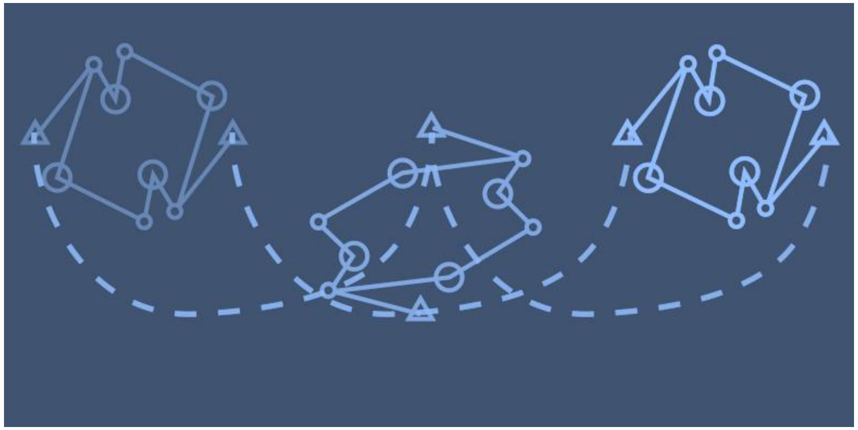
Topology optimization avoiding collision for given trajectories

3D-Printing ↓

Prototype

Graph Grammar ↑

Future Work



- Evaluation
  - Sensitivity analysis
- Additional joint types, friction, springs, …
- Other robotic tasks
- Prototyping
- Synthesis and optimization
  - Different strategies