IDEALAB

Integrating Design, Engineering, and Analysis

Dan Aukes
Assistant Professor
The Polytechnic School
Arizona State University

General Observation about Robot Design

IDEAB

Incendiary Idea:

ANYONE
can design robots

(they just need the right tools)

IDEAB

What do I mean?

IDEAB

I believe in the designer

- Even if they don't know the details of what they're doing
- Novices are generally bad at saying what they want
- Human guidance, intuition helps cut through NP-complexity

Jason was wrong earlier this morning

IDEAB

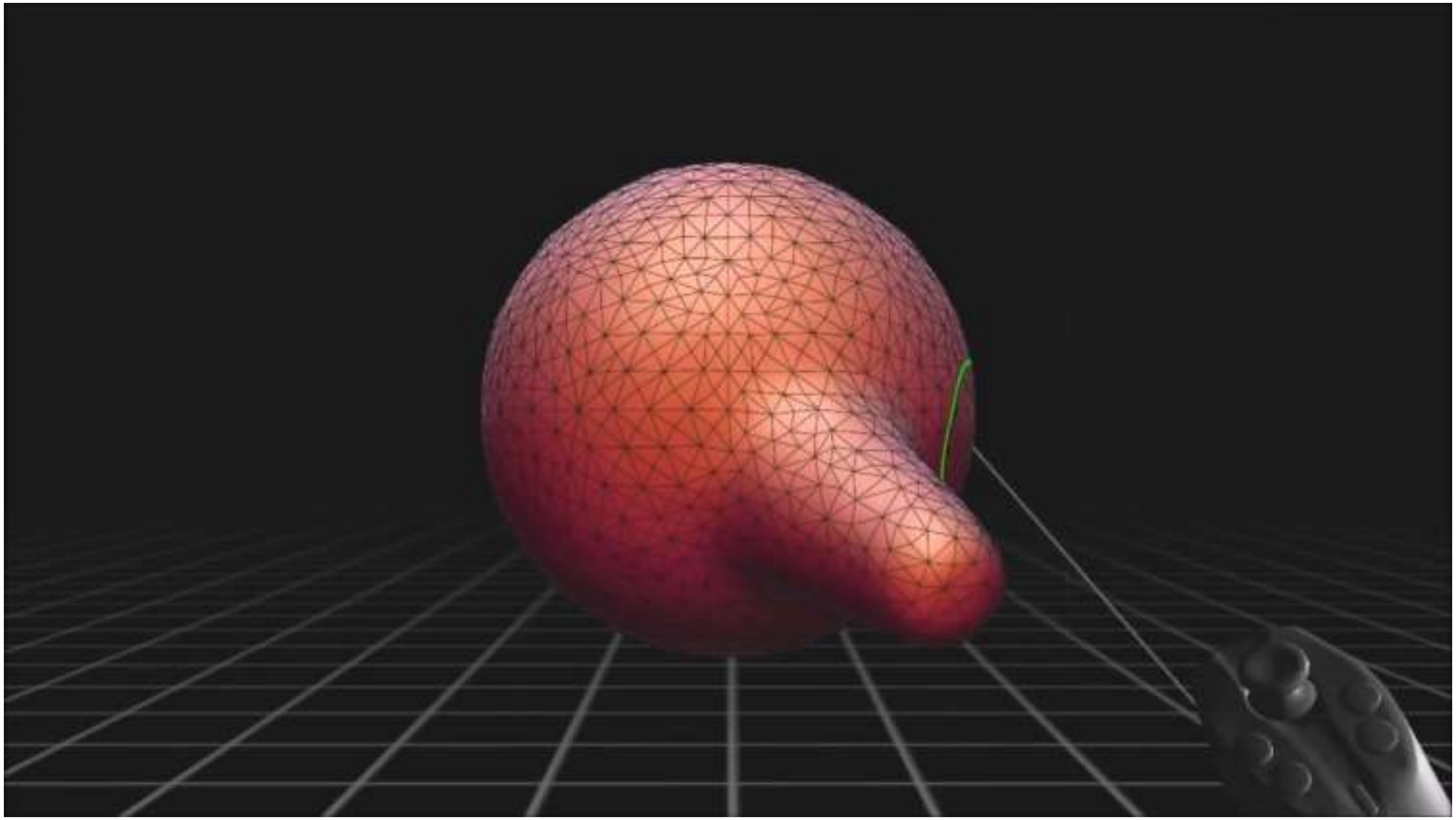
Design for the Novice

- What if you could "feel" how stiff your new robotic leg design is before you make it?
- What if your robot's geometry optimized itself to walk better?

- Simulation
  - Kinematics & Dynamics
  - FEA
- Optimization
- Interaction
  - Haptics
  - Virtual Reality
  - Wearable Devices

IDEAB

VRClay



What’s Needed

- Make it intuitive to design robots
- Bring expert designs to novices
- Give early meaning to designs
  - Manufacturing
  - Analysis
  - Interaction
  - Prototyping
- Applications:
  - STEM Education
  - New Manufacturing
  - Rapid Prototyping

IDEAB

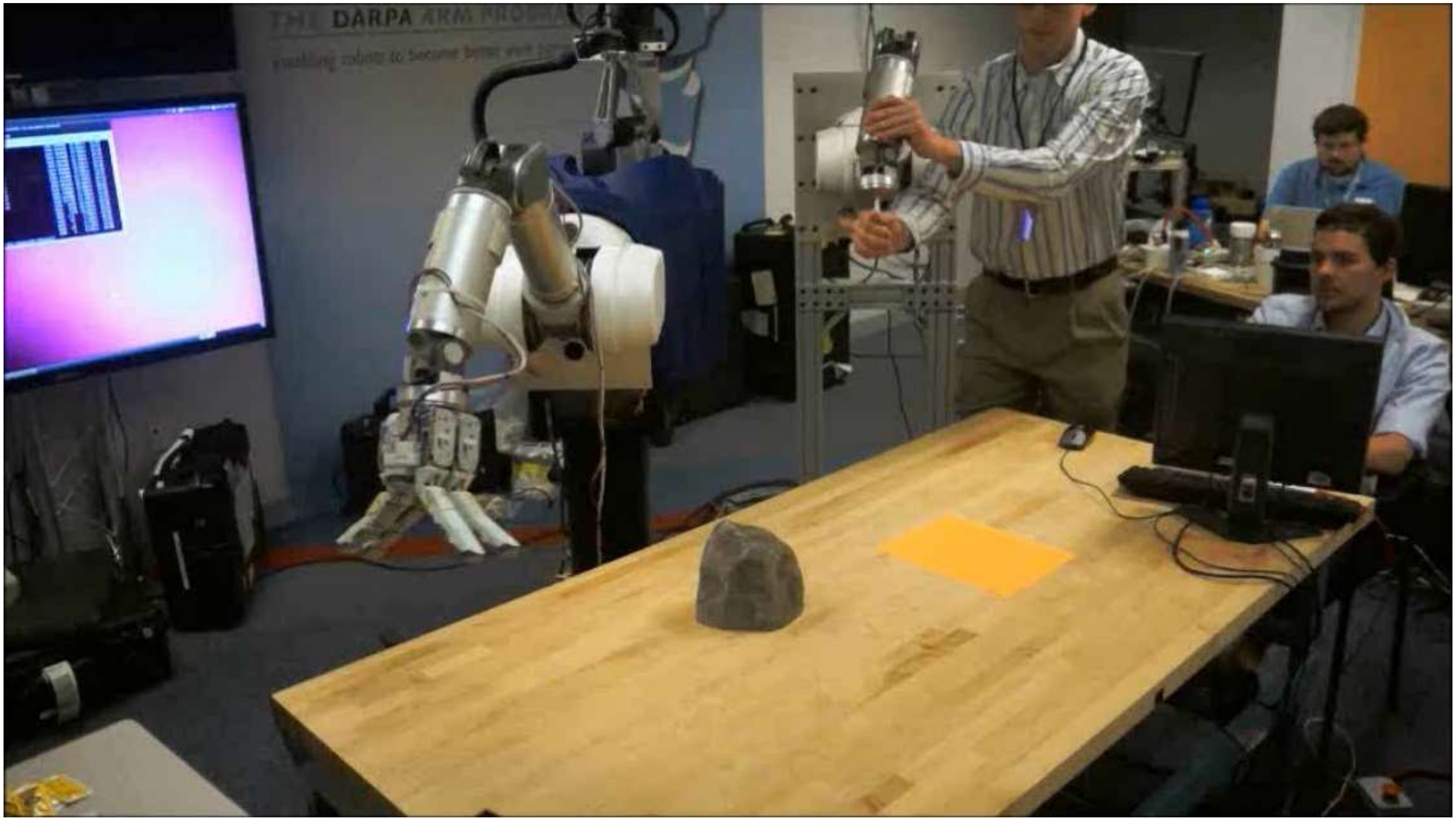
Minimalism by modularity

- Modular building blocks
  - Joints
  - Springs, Dampers, Masses
  - Actuators
- Connections & Interfaces
  - Electrical
  - Mechanical
  - To off-the-shelf components

Minimilasim via Underactuation

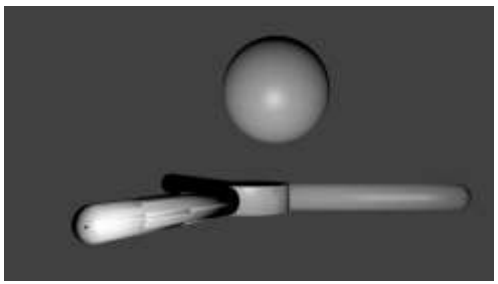
IDEAB

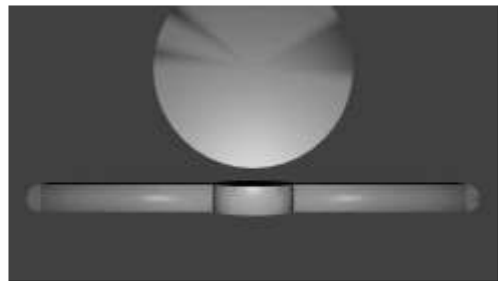
ARM-H Application



IDEAB

Dynamic Simulation in Robot Design





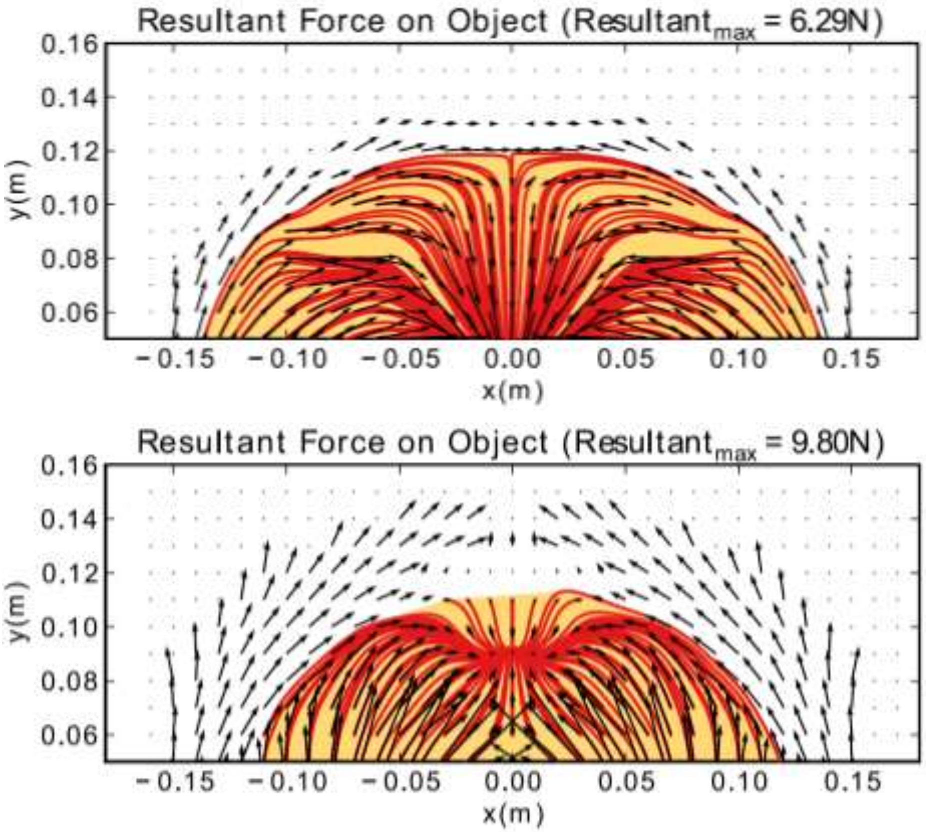
- Dynamic Test
  - Kinematics
  - Transmission
  - Finger Shape
  - Object Shape
  - Friction
  - Mass & Time
  - Full Workspace

- Prescribed-Position
  - Kinematics
  - Transmission
  - Finger Shape
  - ~~Object Shape~~
  - ~~Friction~~
  - ~~Mass & Time~~
  - Full Workspace

- Prescribed-Force
  - Kinematics
  - Transmission
  - Finger Shape
  - Object Shape
  - Friction
  - ~~Mass & Time~~
  - ~~Full Workspace~~

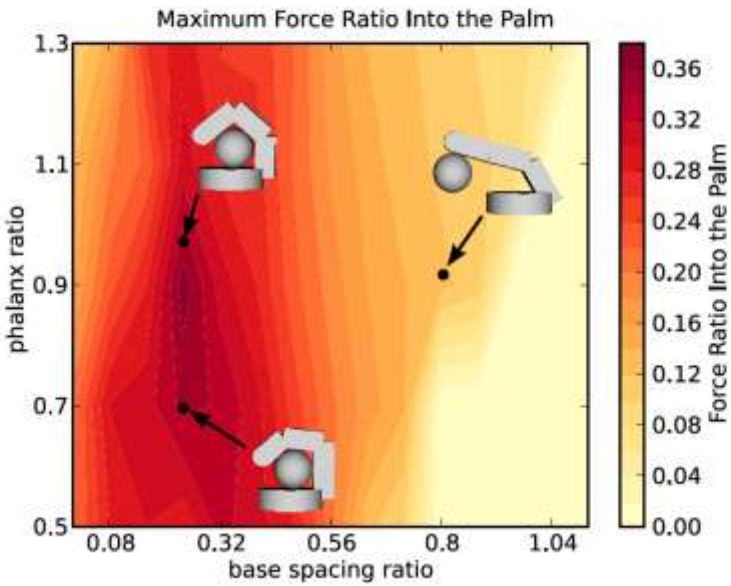
IDEAB

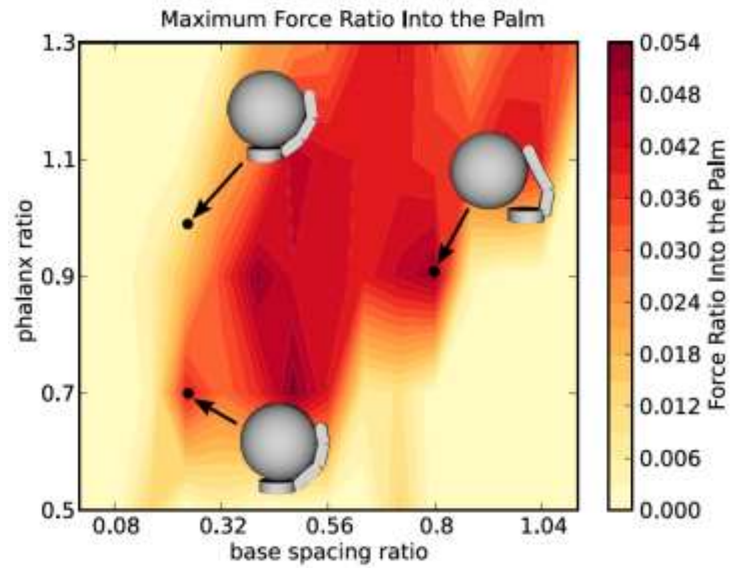
Locked vs. Unlocked



IDEAB

Design Variations & Performance

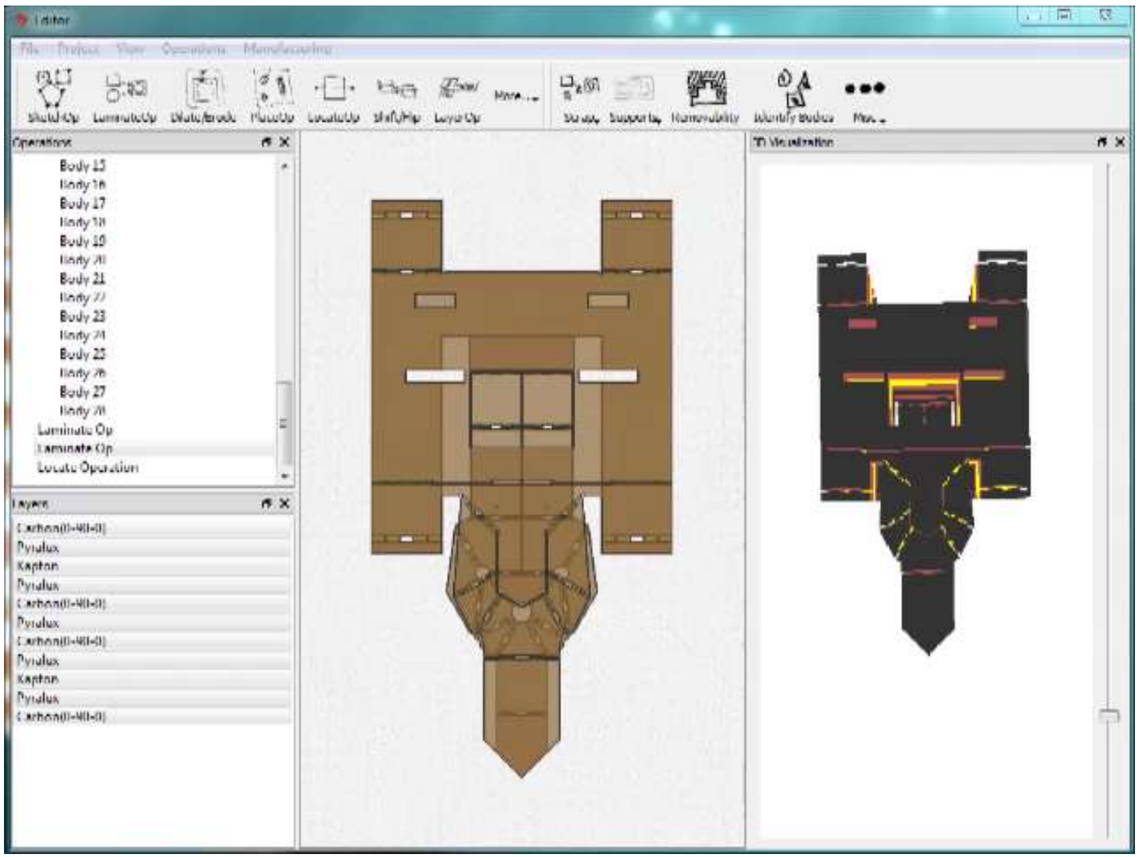




Minimality in Representation

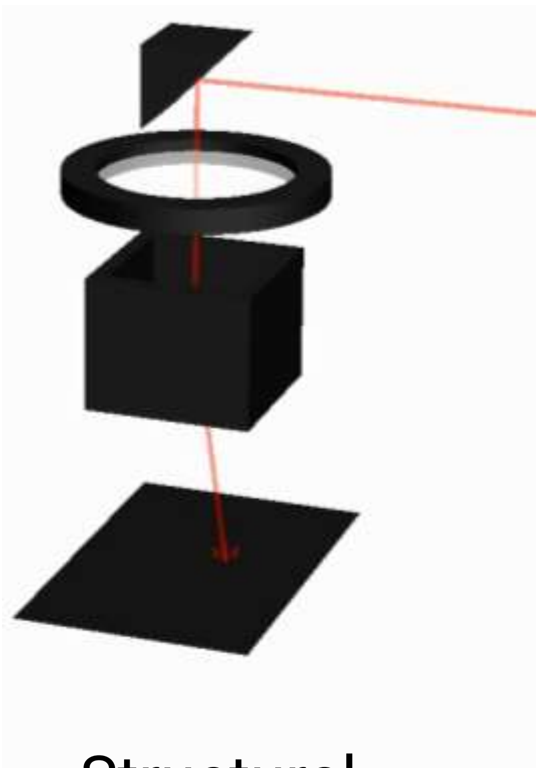
IDEAB

popupCAD

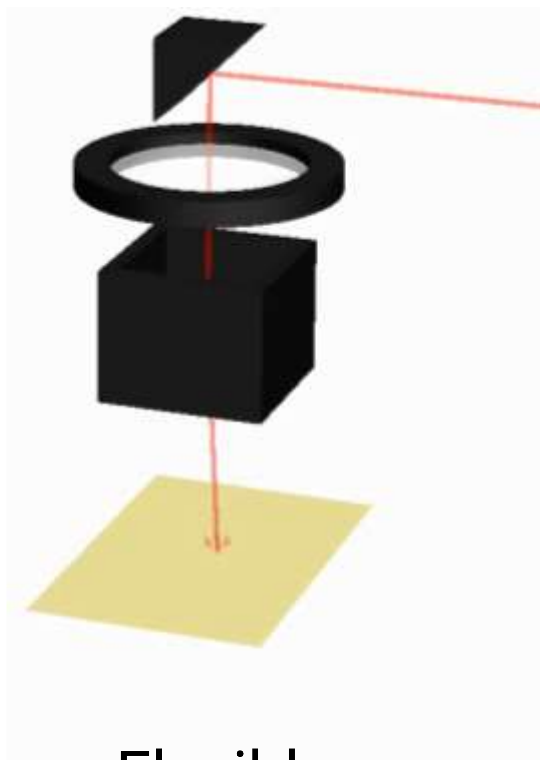


IDEAB

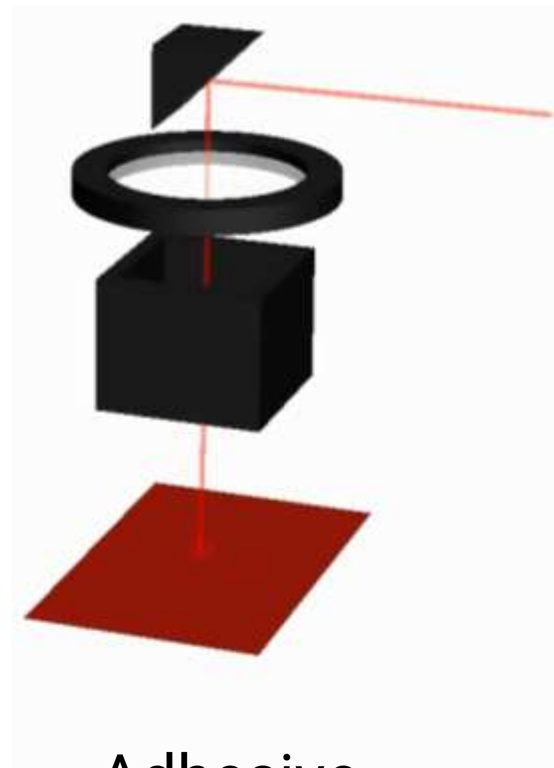
Cutting



Structural



Flexible

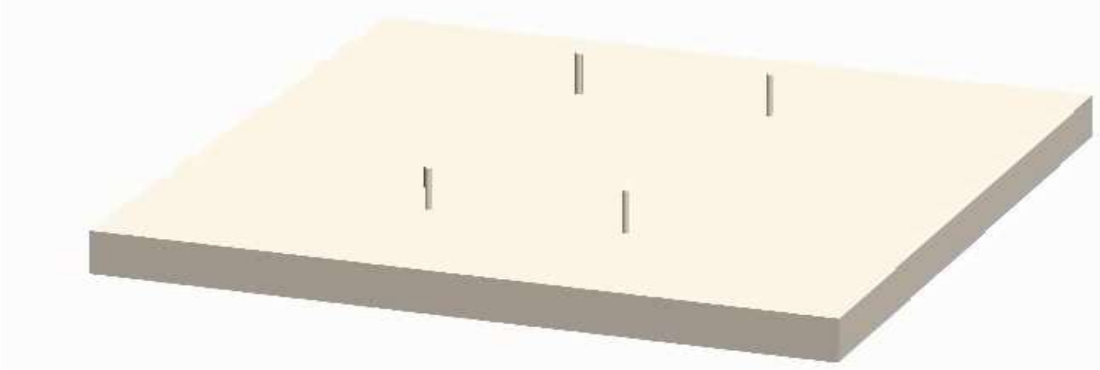


Adhesive

IDEAB

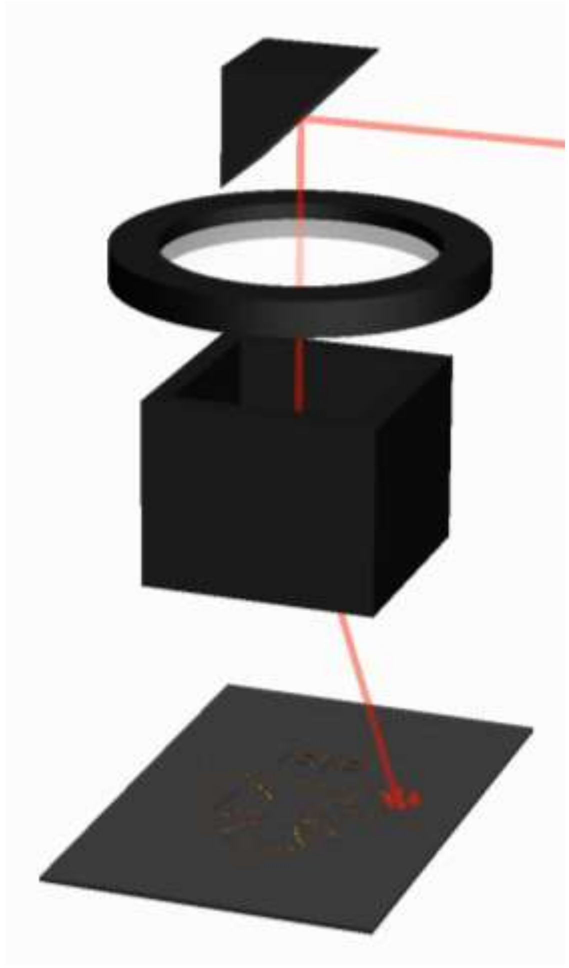
Stacking and Curing

Temperature
Pressure
Time

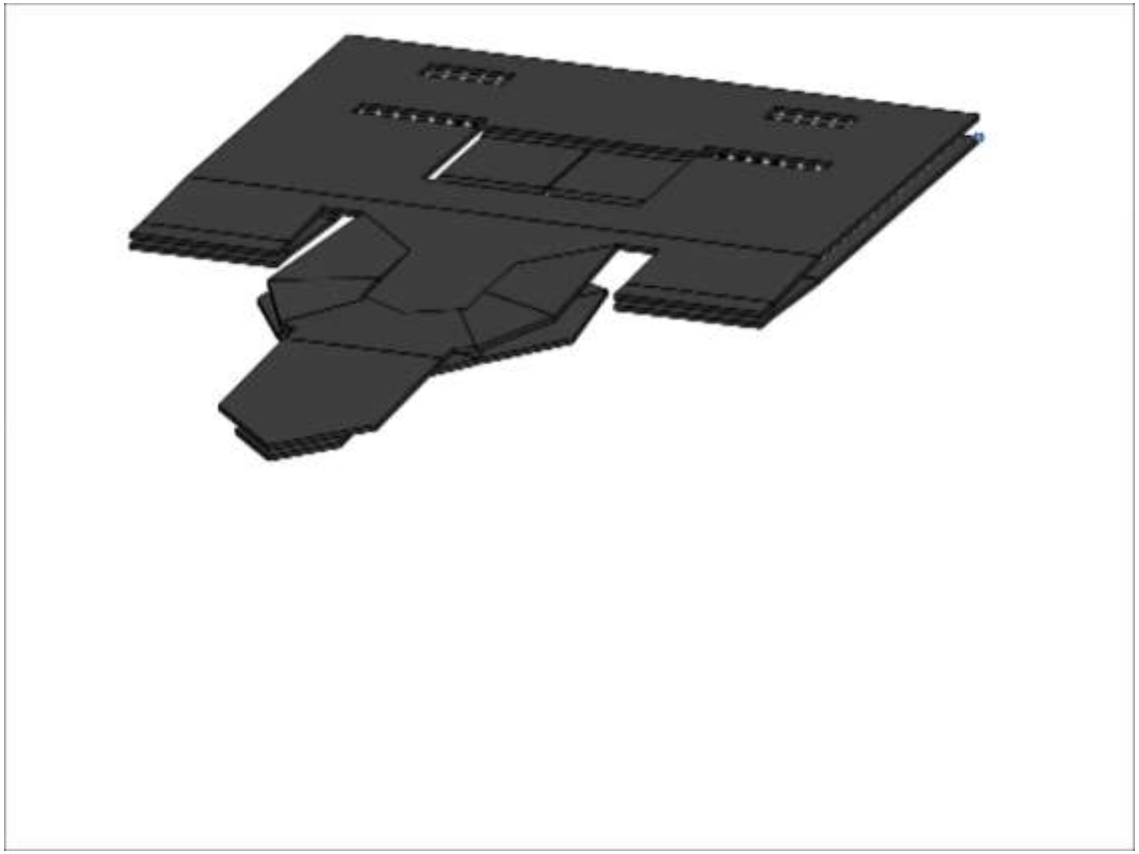


IDEAB

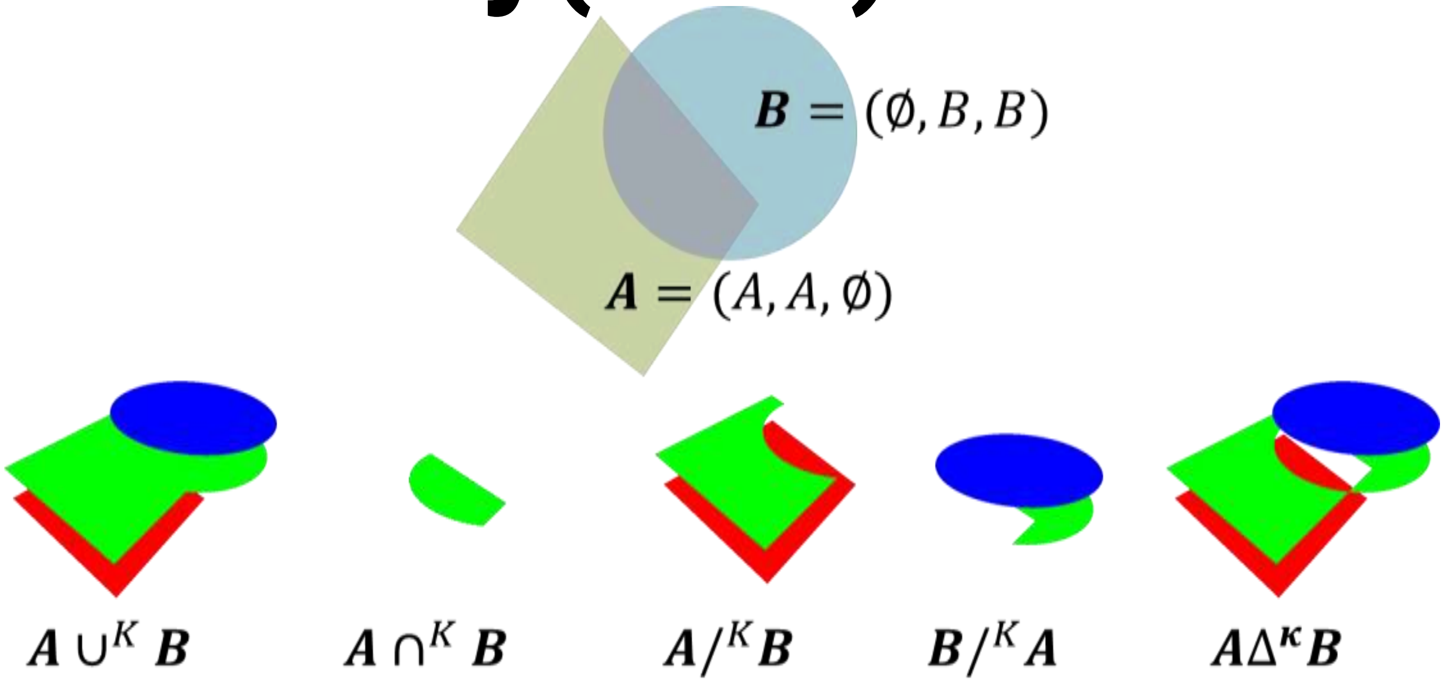
Release



Unfold



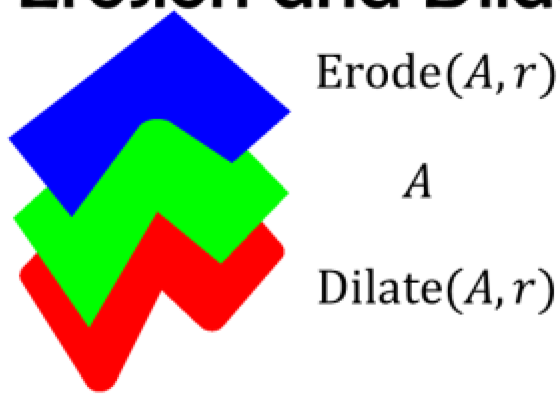
Layered Operations using Constructive Solid Geometry(CSG)



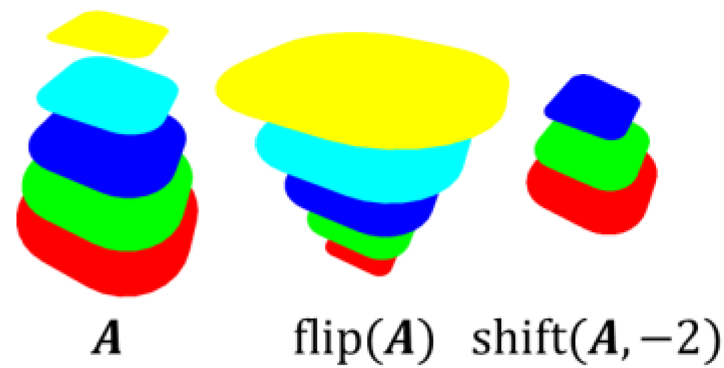
D. M. Aukes, B. Goldberg, M. R. Cutkosky, and R. J. Wood, *Smart Mater. Struct.*, 2014.

Other Operations

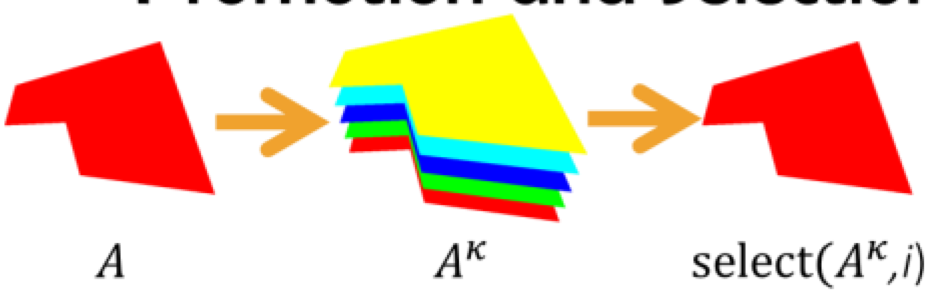
Erosion and Dilation



Shift and Flip



Promotion and Selection

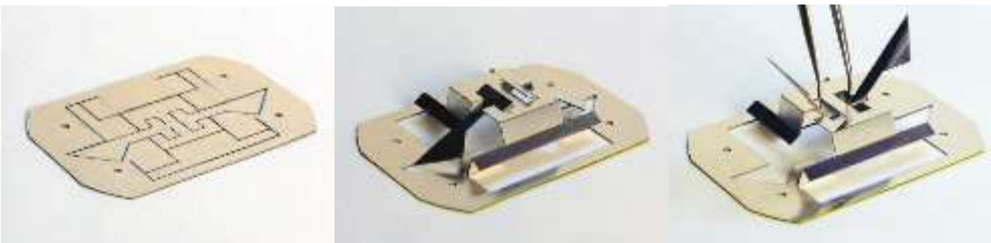


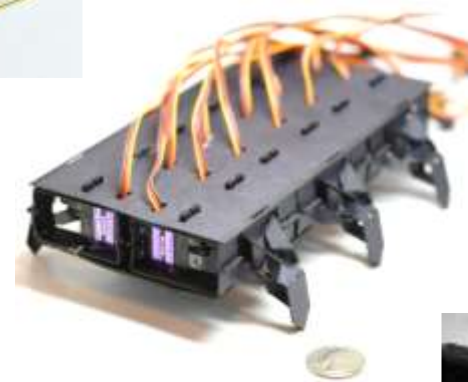
Null Laminate

$$\mathbf{0} = \emptyset^{\kappa}$$

D. M. Aukes, B. Goldberg, M. R. Cutkosky, and R. J. Wood, *Smart Mater. Struct.*, 2014.

popupCAD Designs





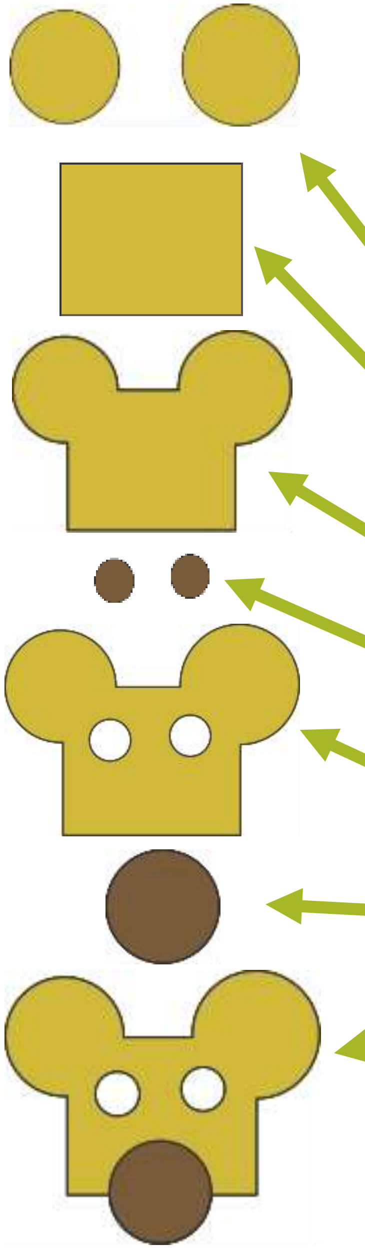
IDEAB

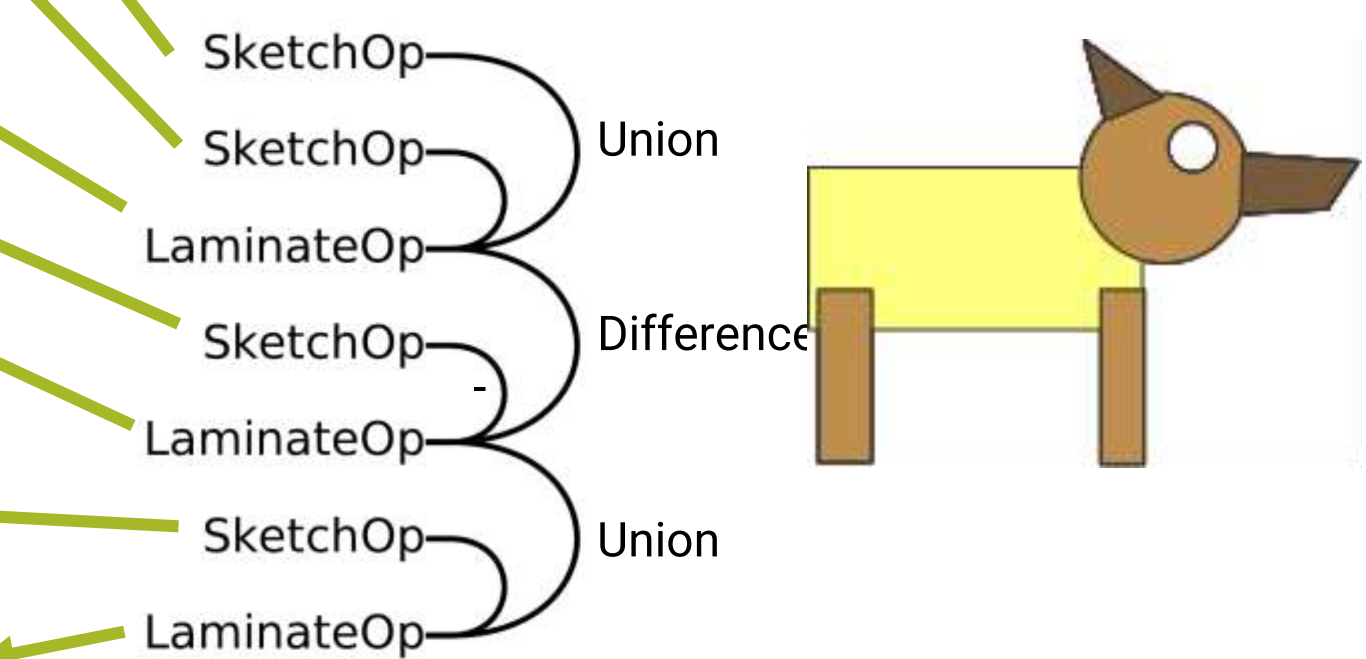
Several New Tools

- Abstracting the Design Process from the geometry
- Dynamic Simulation:
  - Understand ideal rigid body motion
- FEA-based stiffness analysis
  - Understand non-ideal bending of “rigid” links

IDEAB

Abstracting processes





New Dynamics System

- Written in Python
- Symbolic expressions using Sympy
- Kane’s Method for generating equtions of motion
  - Requires ability to perform vector operations(cross, dot, derivative, etc)
  - Reduced representation considers only named state variables

IDEAB

Rotations between Reference Frames

$$
{}^{A}\mathbf{R}^{B} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \mathbf{f} * \mathbf{f}^{T} \right) \cos(q) + \begin{bmatrix} 0 & -f_z & f_y \\ f_z & 0 & -f_x \\ -f_y & f_x & 1 \end{bmatrix} \sin(q) + \mathbf{f} * \mathbf{f}^{T}
$$

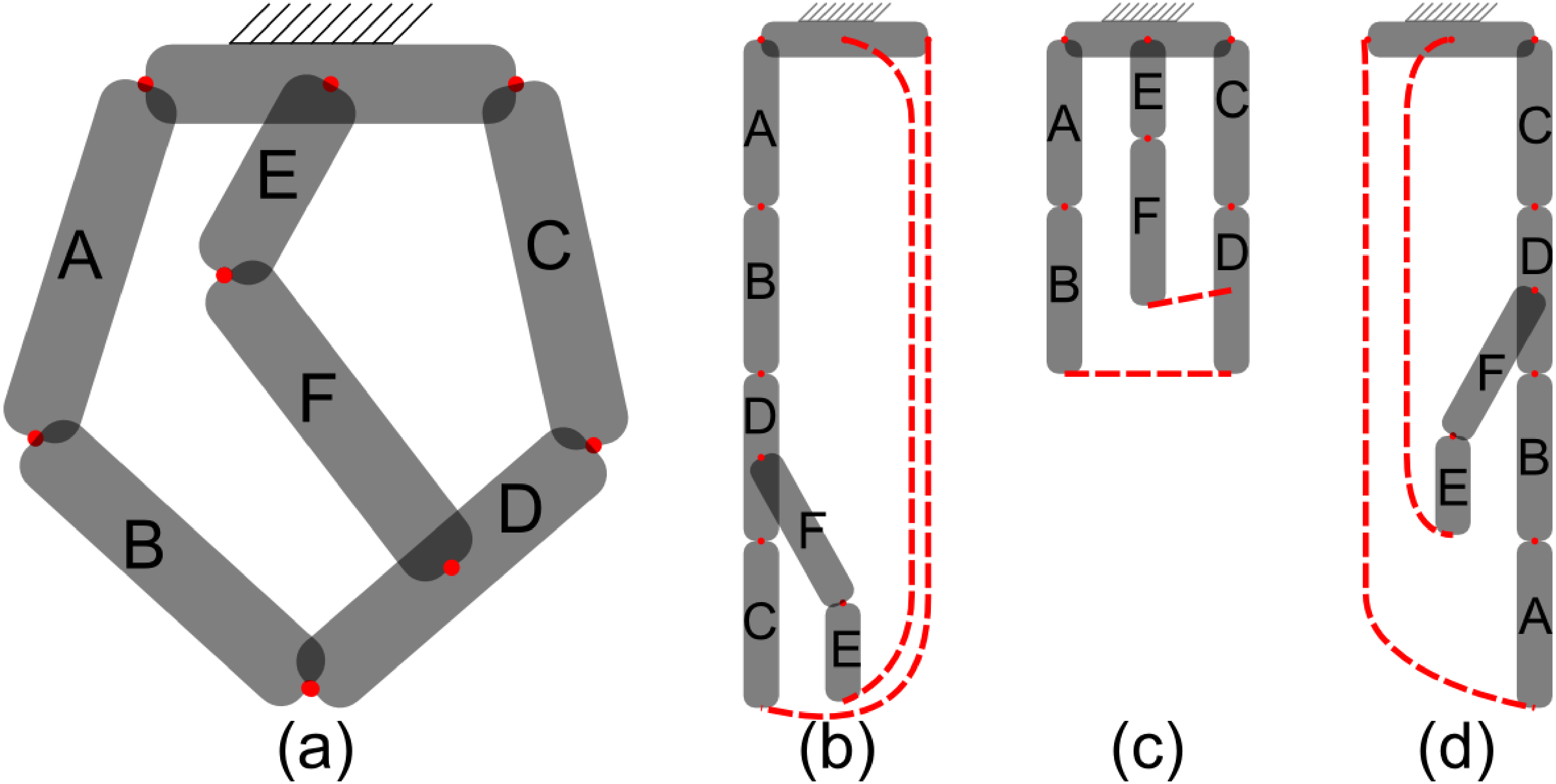
$$
{}^{A}\vec{\omega}^{B} = \dot{q}\hat{f}, \text{ where}
$$

$$
\mathbf{f} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^{T} \text{ and}
$$

$$
\dot{q} = \frac{\mathrm{d}(q)}{\mathrm{d}t}.
$$

IDEAB

Branching Topologies

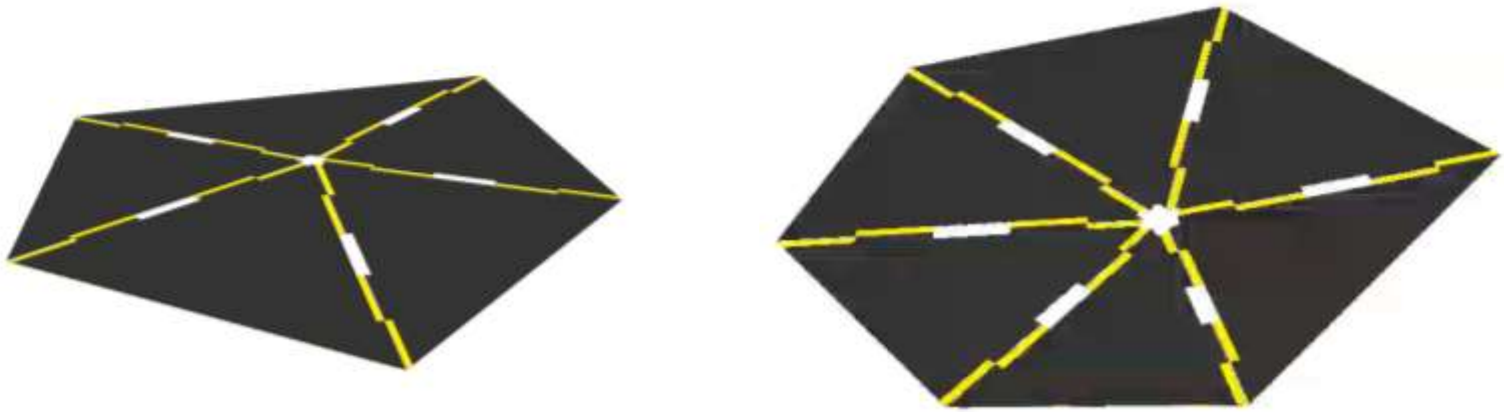


IDEAB

Vectors

```
>>> e = A.x+B.y
>>> e
A.x + B.y
>>> e.express(A)
A.x*(-sin(qB) + 1) + A.y*cos(qB)
>>> e.express(B)
B.x*cos(qB) + B.y*(-sin(qB) + 1)
>>> e.time_derivative(N,system)
-qA_d*N.x*sin(qA) + qA_d*N.y*cos(qA) +
A.x*(-qA_d*cos(qB) - qB_d*cos(qB)) + A
.y*(-qA_d*sin(qB) - qB_d*sin(qB))
>>>
```

Dynamic Simulation in Laminates



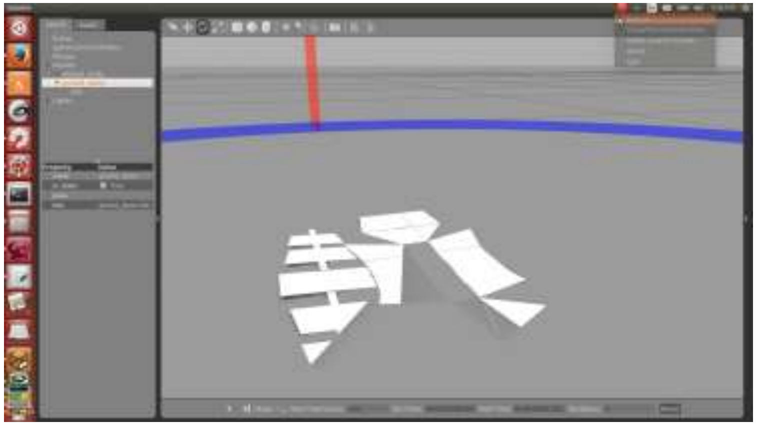
IDEAB

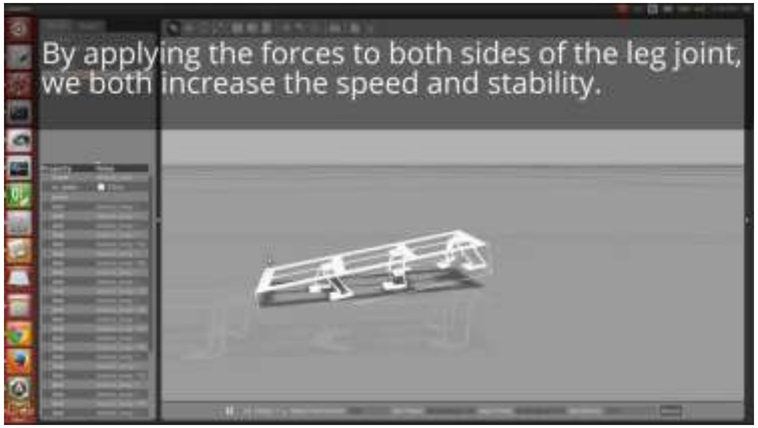
On Github

- https://github.com/idealabasu/pynamics

IDEAB

In Gazebo





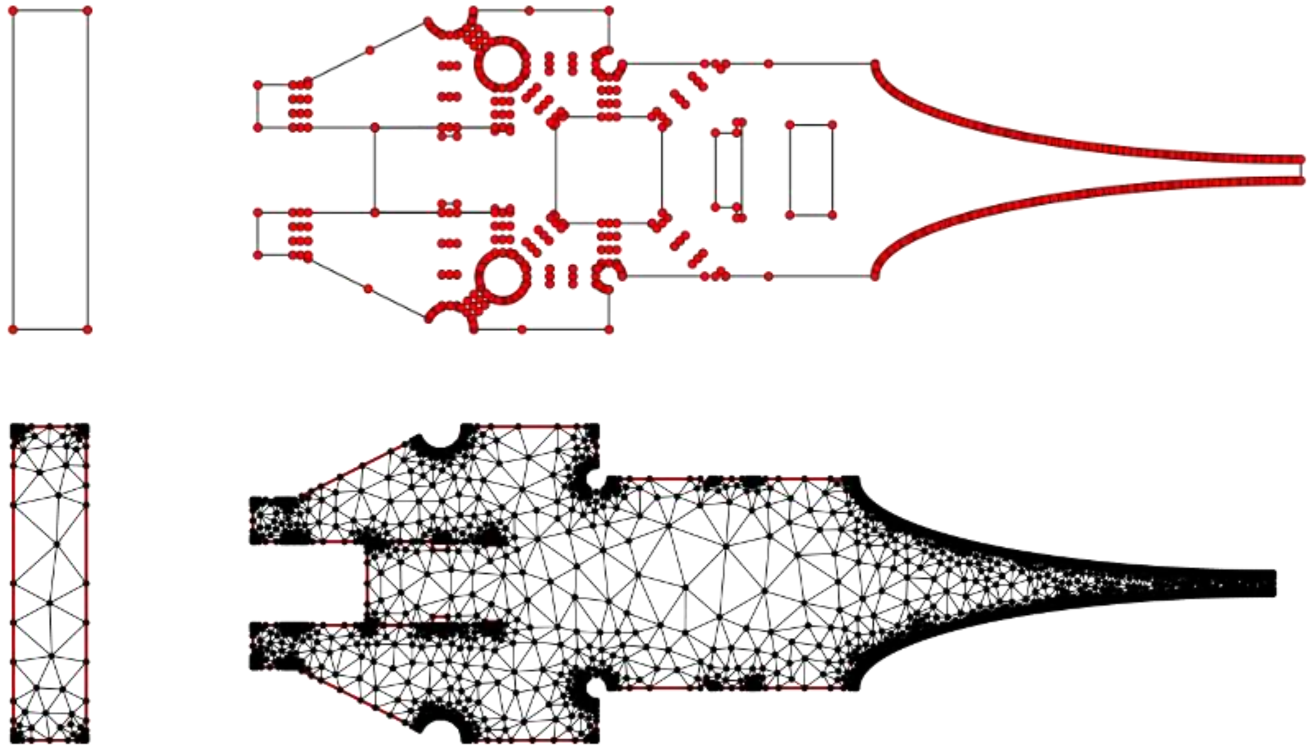
IDEAB

Finite Element Analysis

- Triangular elements: easy to generate from laminate shapes
- FEA package for Python
- New elements can be derived dynamically given new shape functions
  - Incorporates several interpolation methods: linear, quadratic, cubic, BCIZ
  - Works with classical laminate theory.
- On Github soon…

IDEALAB

Laminate Geometry



IDEAB

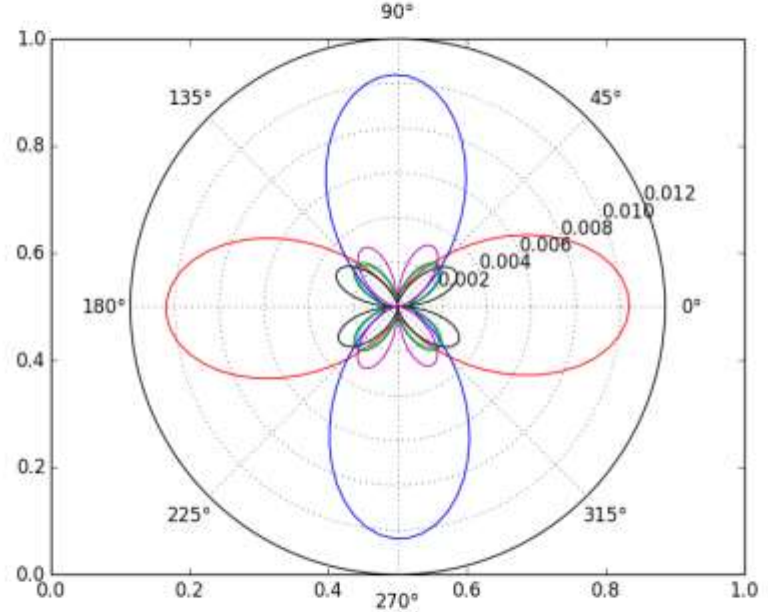
Laminate Stiffness

$$\left\{\frac{N}{M}\right\} = \left[\begin{array}{c|c} A & B \\ \hline B & D \end{array}\right]\left\{\frac{\varepsilon^{o}}{\kappa}\right\}$$

$$A_{ij} = \sum_{k=1}^{n} (\bar{Q}_{ij})_k (h_k - h_{k-1})$$

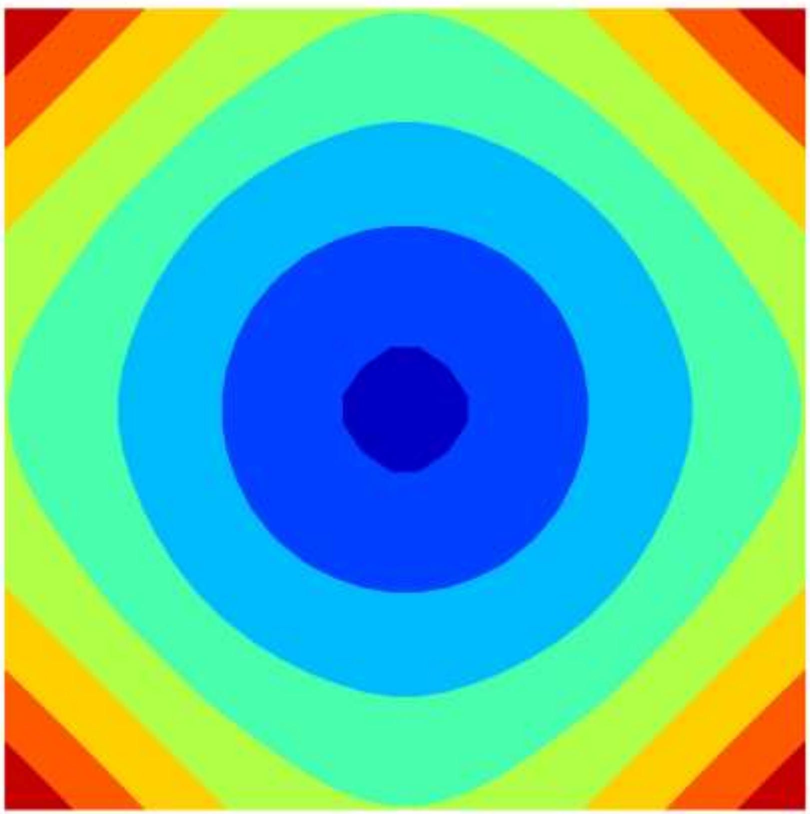
$$B_{ij} = \frac{1}{2} \sum_{k=1}^{n} (\bar{Q}_{ij})_k (h_k^2 - h_{k-1}^2)$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{n} (\bar{Q}_{ij})_k (h_k^3 - h_{k-1}^3)$$

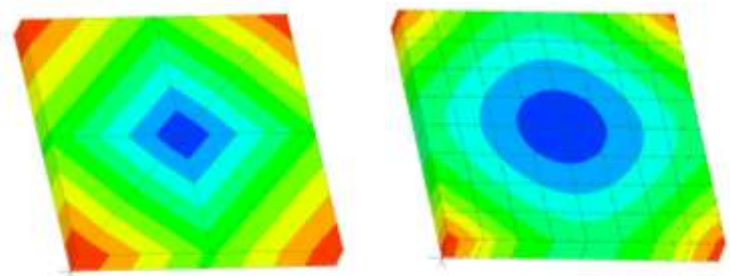


IDEAB

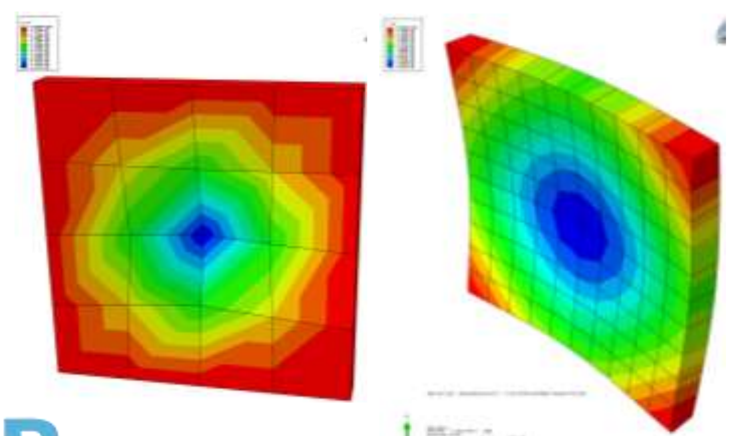
Laminate FEA



LISA



Abaqus



IDEAB