MEAM 520
Lecture 26: Wrap-Up

Cynthia Sung, Ph.D.

Mechanical Engineering & Applied Mechanics

University of Pennsylvania

Previously: Types of Multi-Robot Problems/Solutions

Centralized: All robots are directed by a centralized planner/controller

If 1 robot is d-DOF, N robots are Nd-DOF

State space goes to O(Nd) dimensions

Distributed: Robots make their own decisions

Deadlock/Livelock

Communications complexity

Optimal

Computationally Tractable

Previously: Interesting Problems in Multi-Robots

- Planning
- Task Allocation

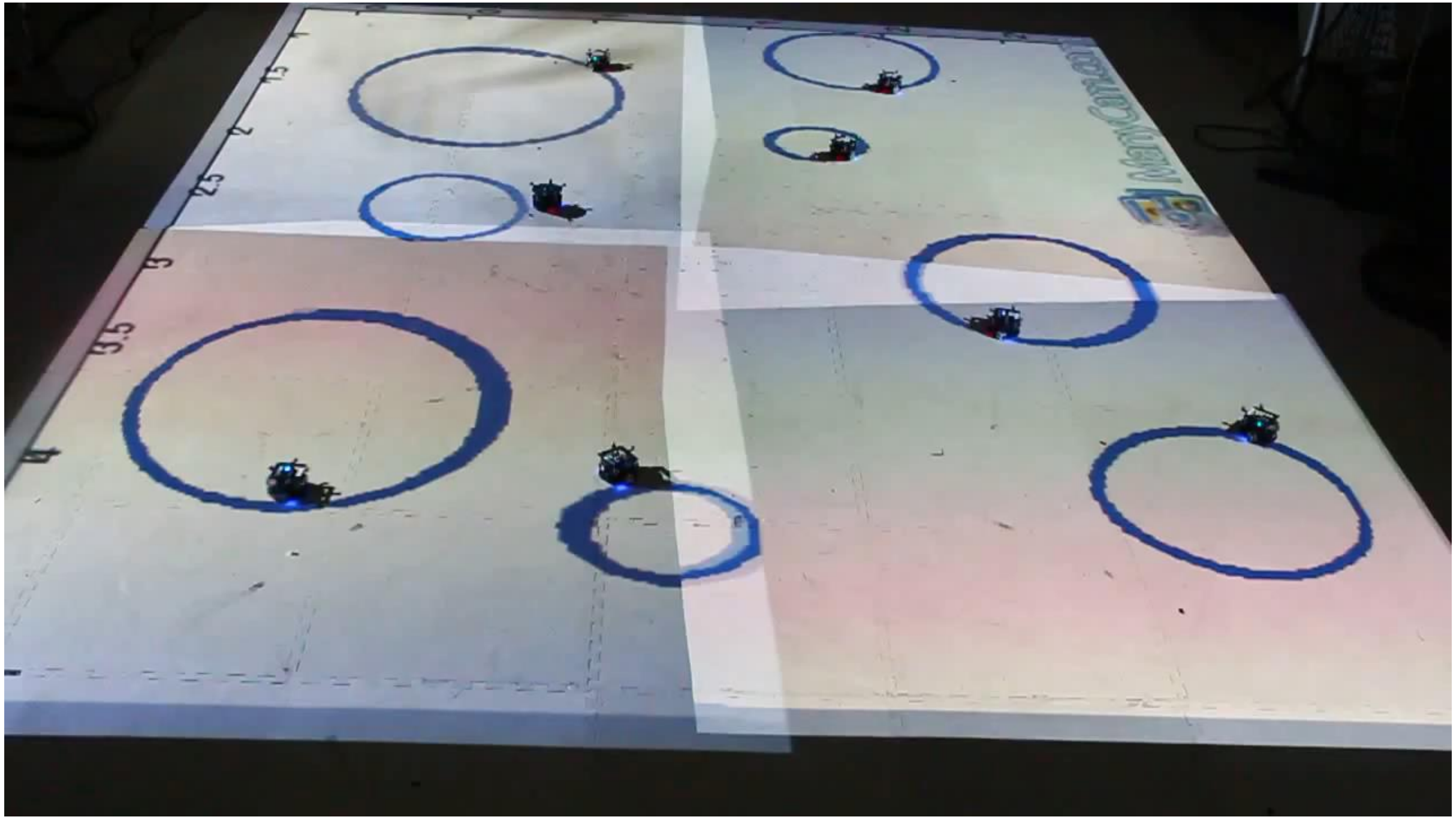
Strategy: Decompose into individual robots except at coordination

Dogar et al, ICRA 2015

https://www.youtube.com/watch?v=vBymMF6mrhI

Previously: Interesting Problems in Multi-Robots

- Planning
- Task Allocation
- Consensus



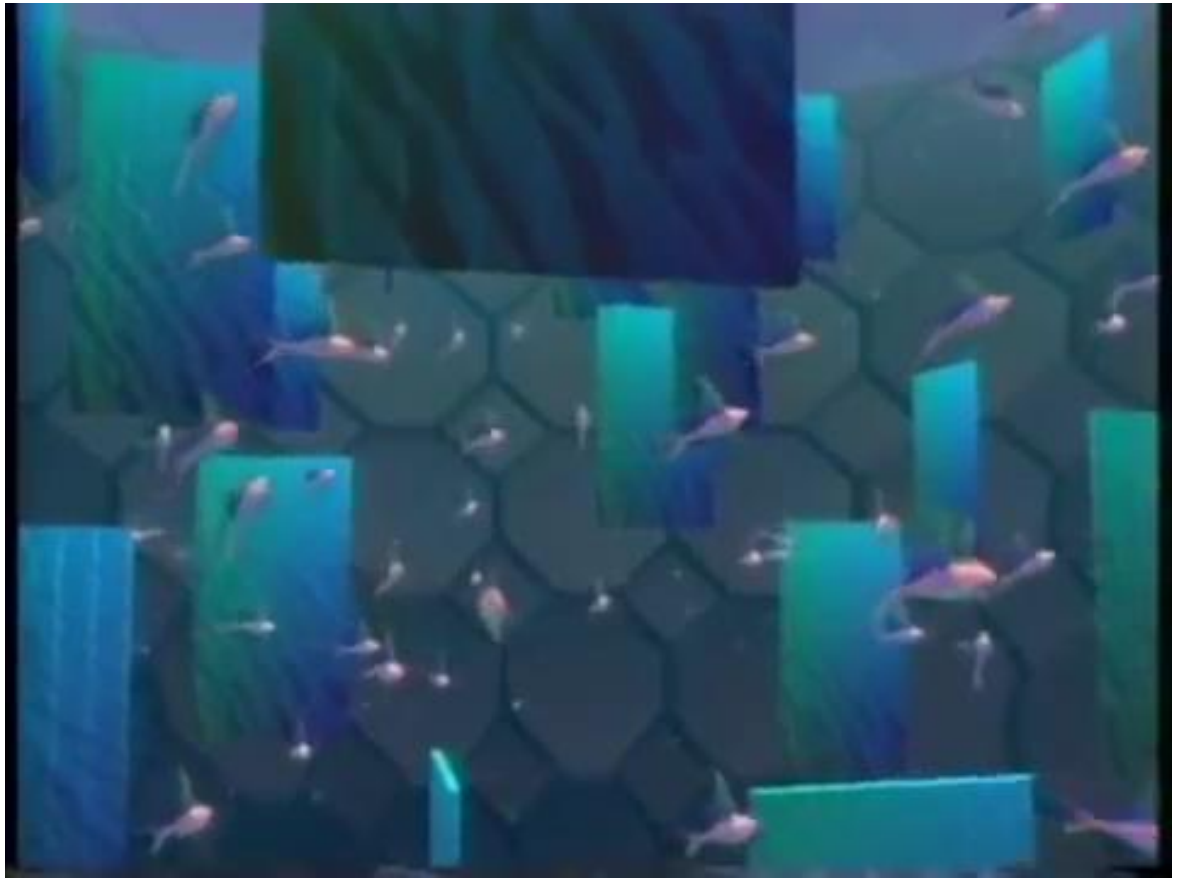
Strategy: Average values from neighbors

MRSL, Boston University, 2014
https://www.youtube.com/watch?v=YEFFXt3B3-U

Previously: Interesting Problems in Multi-Robots

- Planning
- Task Allocation
- Consensus
- Flocking

All of these problems are extensions on problems we have looked at in class. The main difference is communication between agents.



Strategy: Balance matching heading and distance from neighbors

Boids, Stanley & Stella Breaking the Ice, 1987
https://www.youtube.com/watch?v=3bTqWsVqyzE

Final Project Reminders

- Presentations next week (3 min/group)
  - 1 min problem definition
  - 1 min results
  - 1 min lessons learned / remaining challenges
- Schedule posted online – submit your slides/demos ahead of time
- Final reports due 12/12
  - Remember to include an intro that defines the problem you are solving (there is no lab handout for a final project!)

What we covered this semester

Position/Orientation

Planning

Lecture	Topic
1	Introduction
2	Background and Definitions
3	Rotations in 2D and 3D
4	Homogeneous Transformations
5	Forward Kinematics of a Serial Manipulator
6	Denavit-Hartenberg Parameters
7	Inverse Position Kinematics
8	Inverse Orientation Kinematics
9	Quaternions
10	Trajectory Planning in Joint Space
11	Trajectory Planning in Configuration Space
12	Probabilistic Trajectory Planning
13	Planning on Other Robot Types
14	Velocity Kinematics

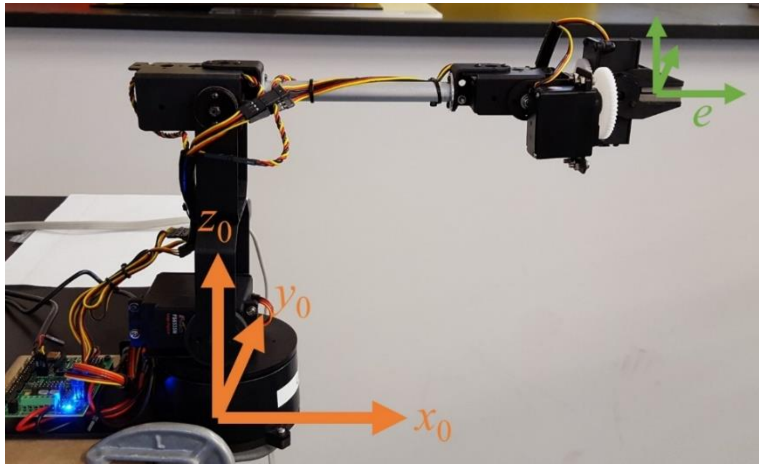
Lecture	Topic
15	More Velocity Kinematics
16	Inverse Velocity Kinematics
17	Guest: Medical Robotics
18	Jacobians and Statics
19	Trajectory Planning with Potential Fields
20	Guest: Legged Robotics
21	Joint Space Dynamics
22	More Joint Space Dynamics
23	Control and Actuation
24	Modern Planning and Control
25	Guest: Multi-Robot Systems
26	Design
27	Final Presentations
28	Final Presentations

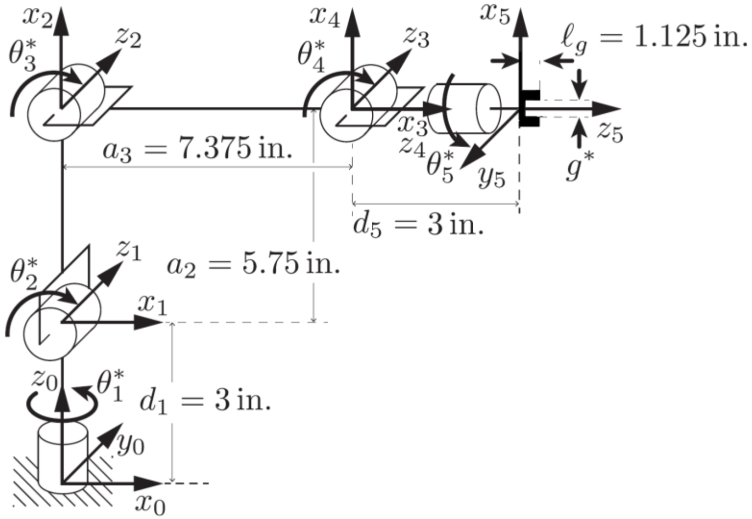
Velocity

Statics and Dynamics

Extensions

Forward Kinematics (Lab 1)

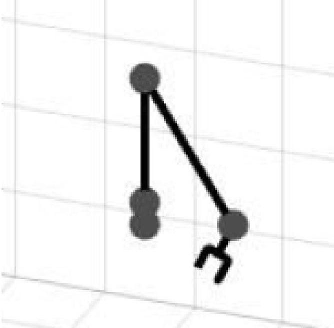




1) For any serial manipulator, you can draw a sequence of coordinate frames

2) With these frames, you can define **Denavit-Hartenberg parameters…**

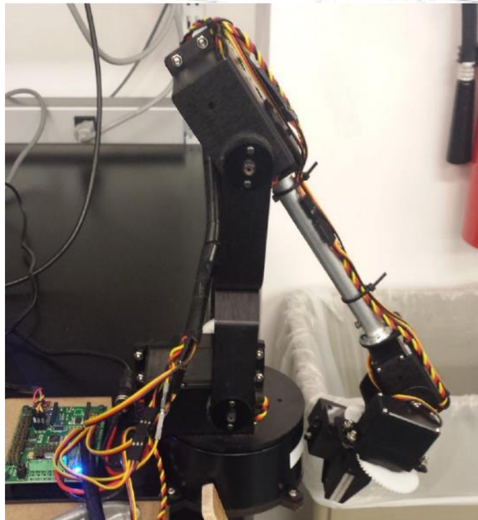
x step	a_i Link Length	distance between z_{i-1} and z_i, measured along x_i
	α_i Link Twist	angle between z_{i-1} and z_i, measured in the plane normal to x_i (right hand rule)
z step	d_i Link Offset	distance between x_{i-1} and x_i, measured along z_{i-1}
	θ_i Joint Angle	angle between x_{i-1} and x_i, measured in the plane normal to z_{i-1} (right hand rule)



3) …that give you transformation matrices for the manipulator FK

$$\mathbf{T}_n^0 = A_1(q_1) \cdots A_n(q_n)$$

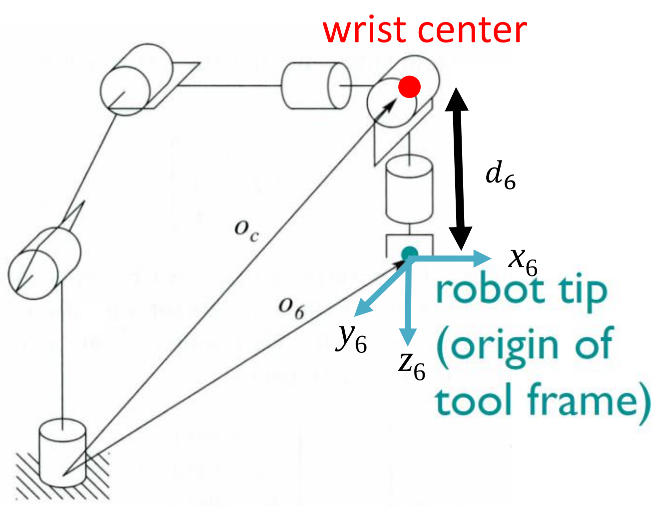
$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Inverse Kinematics (Lab 2)

Given $\mathbf{H} = \begin{bmatrix} \mathbf{R} & o \\ 0 & 1 \end{bmatrix}$ and a certain manipulator with *n* joints,
find $q_1,\ldots,q_n$ such that $\mathbf{T}_n^0(q_1, \ldots, q_n) = \mathbf{H}$

1) Kinematic decoupling allows you to separate position IK from orientation IK



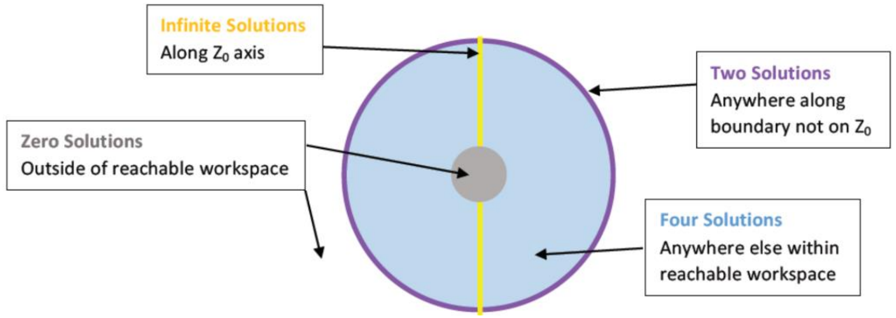
$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - d_6 r_{13} \\ o_y - d_6 r_{23} \\ o_z - d_6 r_{33} \end{bmatrix}$$

position

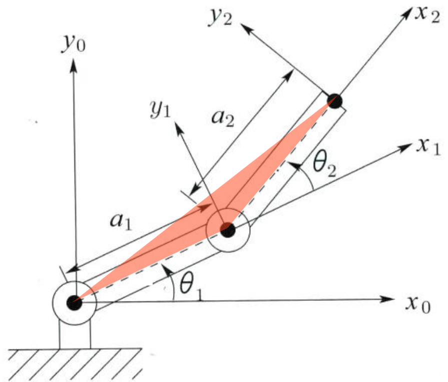
$$\mathbf{R}_6^3 = (\mathbf{R}_3^0)^{-1}\mathbf{R} = (\mathbf{R}_3^0)^{\mathrm{T}}\mathbf{R}$$

orientation

2) There are often multiple solutions to the IK problem



3) Algebraic and geometric techniques are useful for finding these solutions

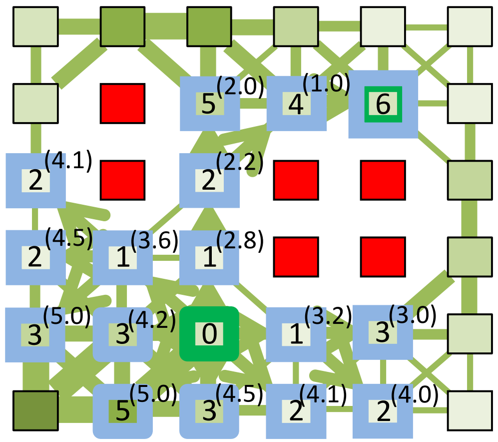


$$\theta_1 = \operatorname{atan2}\left(\frac{o_y}{o_x}\right) - \operatorname{atan2}\left(\frac{a_2 \sin\theta_2}{a_1 + a_2\cos\theta_2}\right)$$

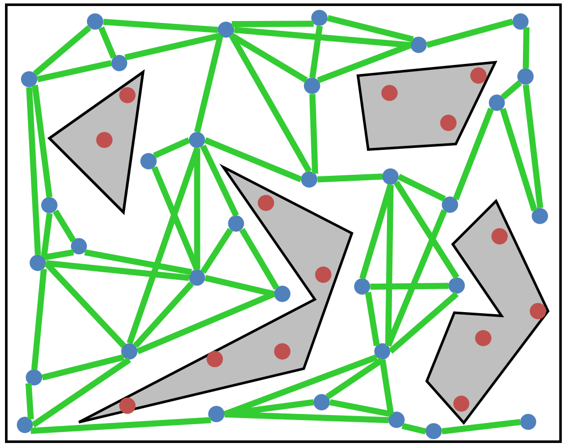
$$\theta_2 = \cos^{-1}\left(\frac{o_x^2 + o_y^2 - a_1^2 - a_2^2}{2a_1a_2}\right)$$

Trajectory Planning (Labs 3 and 5)

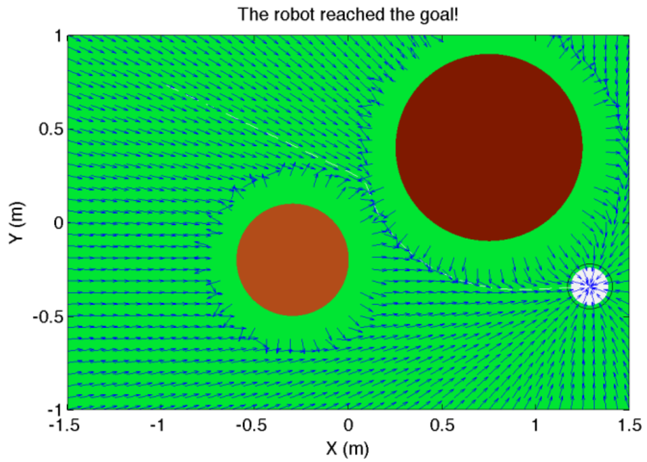
1) Grid-based search is resolution-complete but computationally expensive



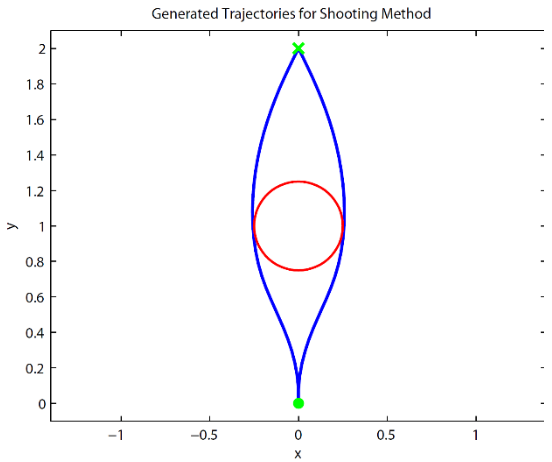
2) Sampling-based planners are probabilistically complete



3) Potential fields are computationally cheap but may have local minima



4) Kinodynamic planners incorporate dynamics



Jacobians (Labs 4 and 5)

1) The velocity of a point on a manipulator can be described with using the manipulator Jacobian

linear velocity

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$$

(6 x n) Jacobian

angular velocity

$$\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \dot{q}$$

2) Velocity FK and IK involve manipulating a matrix equation

$$v_n^0 = J_v \dot{q}$$

forward velocity kinematics

$$\dot{q} = J_v^{-1} v_n^0$$

inverse velocity kinematics

3) Singularities occur whenever a robot loses the ability to move its end effector in a certain direction (J loses rank)

4) When a robot is static, endpoint forces and torques can be computed using the transpose of the Jacobian

(n x 1) (n x 6) (6 x 1)

$$\vec{\tau} = J^T(\vec{q})\vec{F}$$

joint forces and torques

Jacobian matrix transpose

endpoint forces and torques

Application: Potential Fields

Dynamics and Control

1) **Euler-Lagrange** : For small DOF, closed-form manipulator dynamics can be described using the manipulator equation

$$\tau = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q)$$

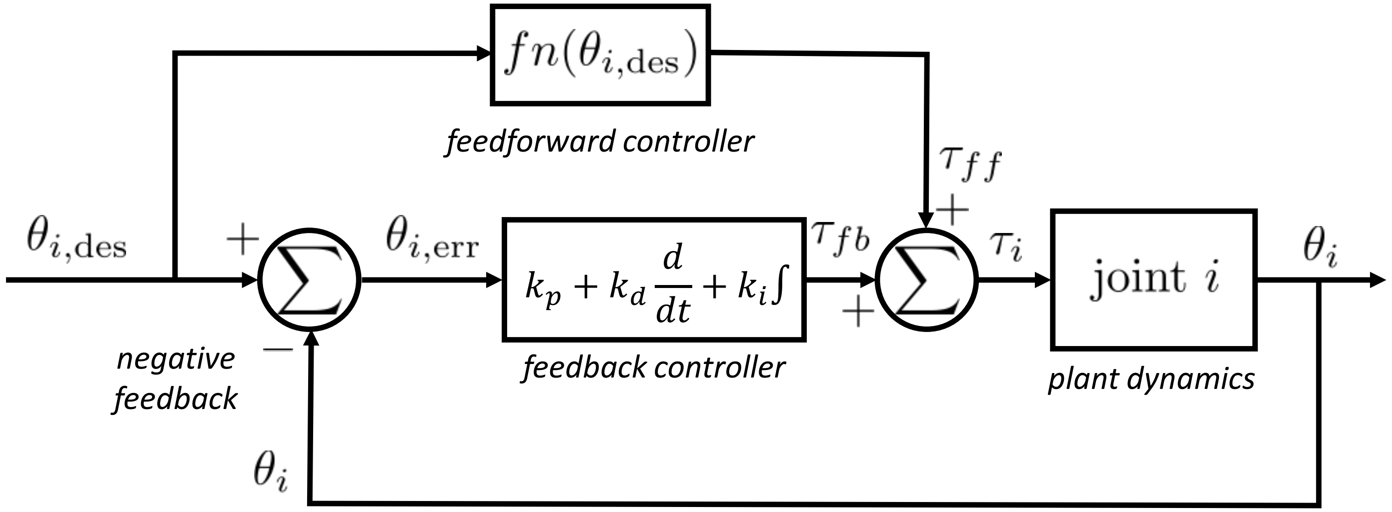
$$D = \sum_{i=1}^{N} \left(m_i J_{vci}^{\top} J_{vci} + J_{\omega i}^{\top} R_i I_i R_i^{\top} J_{\omega i} \right) \qquad g = \frac{\partial}{\partial q} \sum_{i=1}^{N} m_i \vec{g} \cdot \vec{r}_i$$

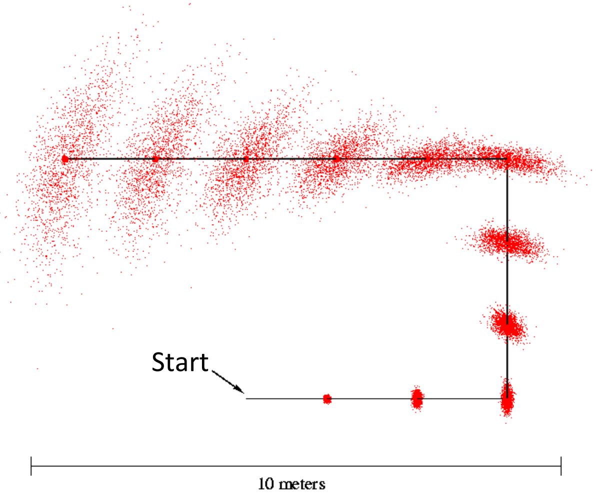
$$(C\dot{q})_k = \sum_{i,j} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j$$

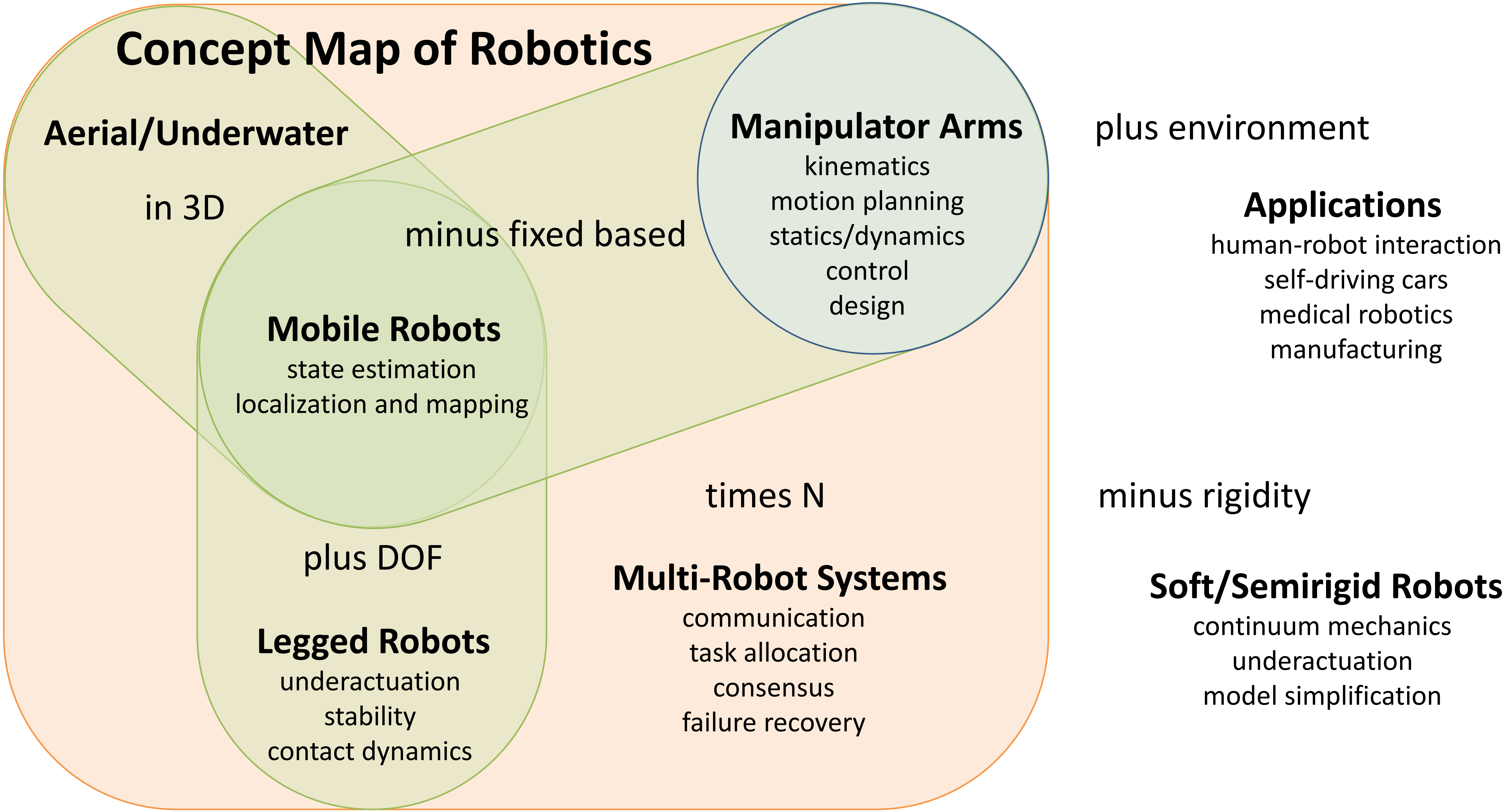
2) **Newton-Euler** : For large DOF, iterative approaches provide force info for a particular time evolution

1. Start with $\omega_0 = 0, \alpha_0 = 0, a_{c,0} = 0, a_{e,0} = 0$
2. Solve kinematic constraints for i from 1 to n
3. Start with $f_{n+1} = 0, \tau_{n+1} = 0$
4. Solve force/moments for i from n to 1

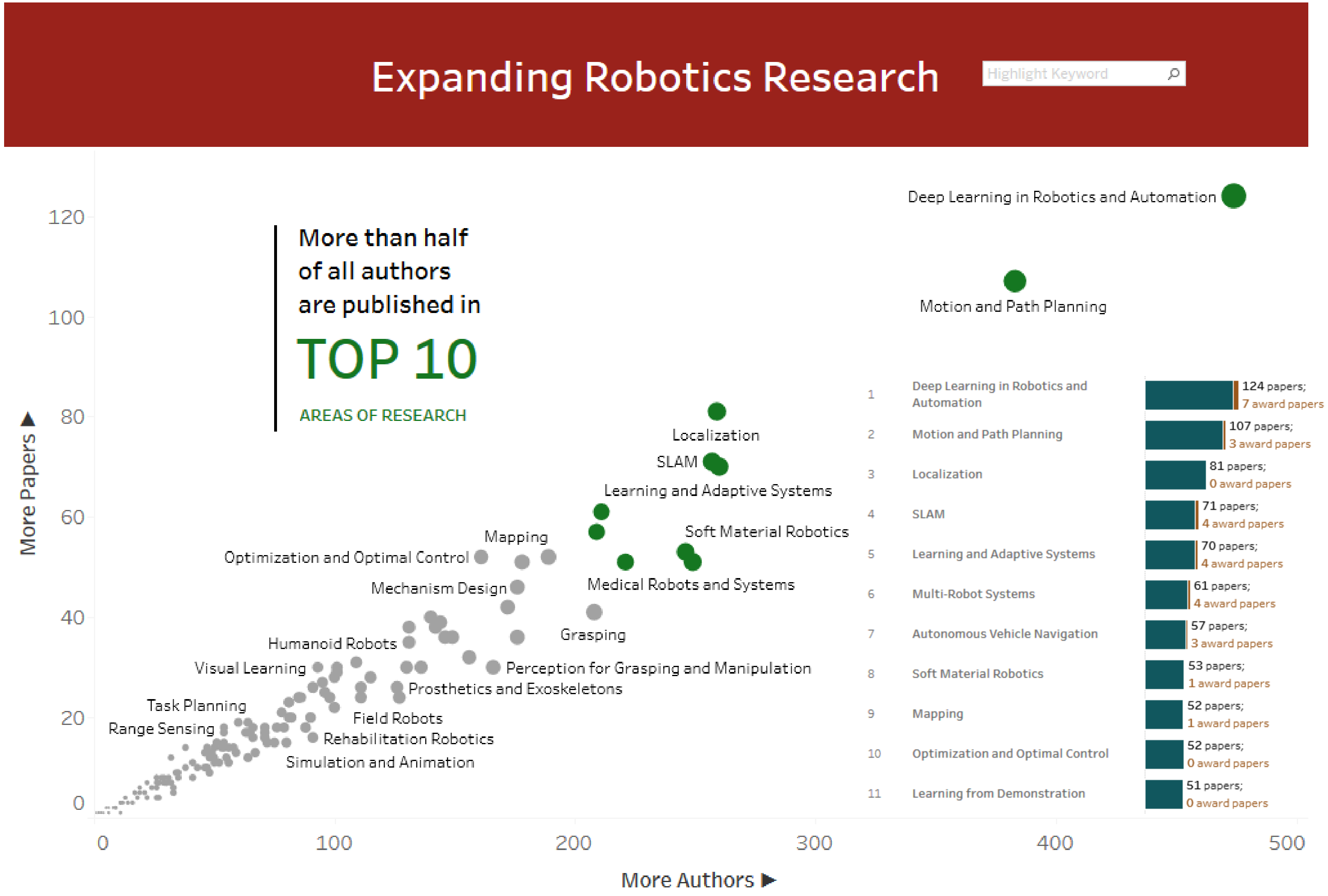
3) Given these dynamics, we can generate feedback and feedforward controllers to follow desired trajectories and estimate state






Concept Map of Robotics
Aerial/Underwater
in 3D
minus fixed based
Manipulator Arms
kinematics
motion planning
statics/dynamics
control
design
plus environment
Applications
human-robot interaction
self-driving cars
medical robotics
manufacturing
Mobile Robots
state estimation
localization and mapping
times N
minus rigidity
plus DOF
Multi-Robot Systems
communication
task allocation
consensus
failure recovery
Soft/Semirigid Robots
continuum mechanics
underactuation
model simplification
Legged Robots
underactuation
stability
contact dynamics

What's on the horizon?



https://public.tableau.com/views/ICRA2018/Dashboard1?:embed=y&:display_count=yes&:showVizHome=no

https://youtu.be/HSA5Bq-1fU4

Amazon Warehouse

Multiple robots in an **structured space** with **global tracking** and **no human** interference.

Solved

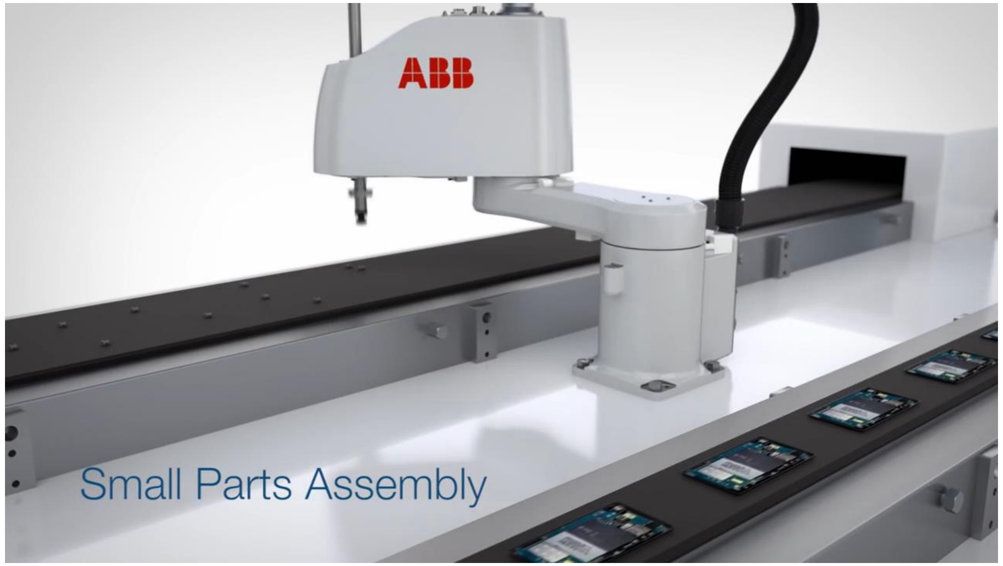
https://youtu.be/aaOB-ErYq6Y

Waymo Self-Driving Car

Multiple robots in an **structured space** with **local tracking** and **human** interference.

Open questions:

- **Deep learning** – for car/pedestrian detection, behavior prediction
- **Motion planning** – long range vs short range
- **Localization/SLAM** – when driving on/off the map
- **Multi-robot systems** – for multi-car communications



https://youtu.be/97KX-j8Onu0

Manufacturing Line

Robot in an **structured space** manipulating **identical parts** with **no human** interference.

Solved

© Authors of ICRA 2018 Paper 2592 Wed PM Pod E.3 & Award Session Tue 14:30 - 15:15

https://youtu.be/zLXvzitRSCQ

Picking Challenge

Multiple objects of different **shapes** with **inaccurate tracking**.

Open questions:

- **Deep learning** – for grasp planning
- **Motion planning/Localization** – in the presence of uncertainty
- **Soft material robotics** – for robust grasping

How do we do kinematics with non-rigid robots?

The **Denavit-Hartenberg transform** results from successive rotations and translations via the four DH parameters

a parameterization for homogeneous transformations

The transform from i to i-1 is

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Three DH parameters will be **constant** for each joint’s transformation, and one will **vary**.

Plug DH parameters into the above formula to find each joint’s transformation matrix.

The final transformation matrix from tip to base is

$$\mathbf{T}_n^0 = A_1(q_1) \cdots A_n(q_n)$$

How do we do kinematics with non-rigid robots?

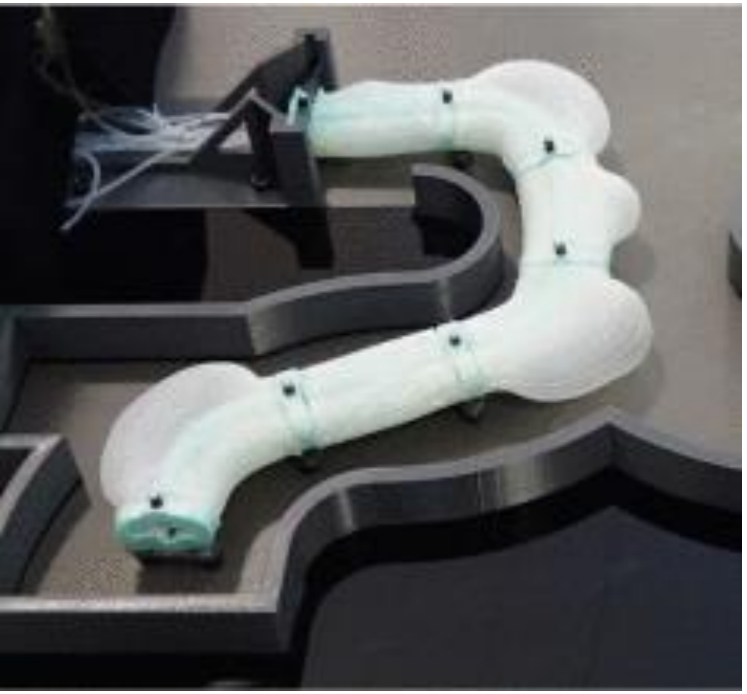
The **Denavit-Hartenberg transform** results from successive rotations and translations via the four DH parameters

a parameterization for homogeneous transformations

The transform from i to i-1 is

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where do we put the links and joints?



Marchese et al. SoRo 2015

Three DH parameters will be **constant** for each joint's transformation, and one will **vary**.

Plug DH parameters into the above formula to find each joint's transformation matrix.

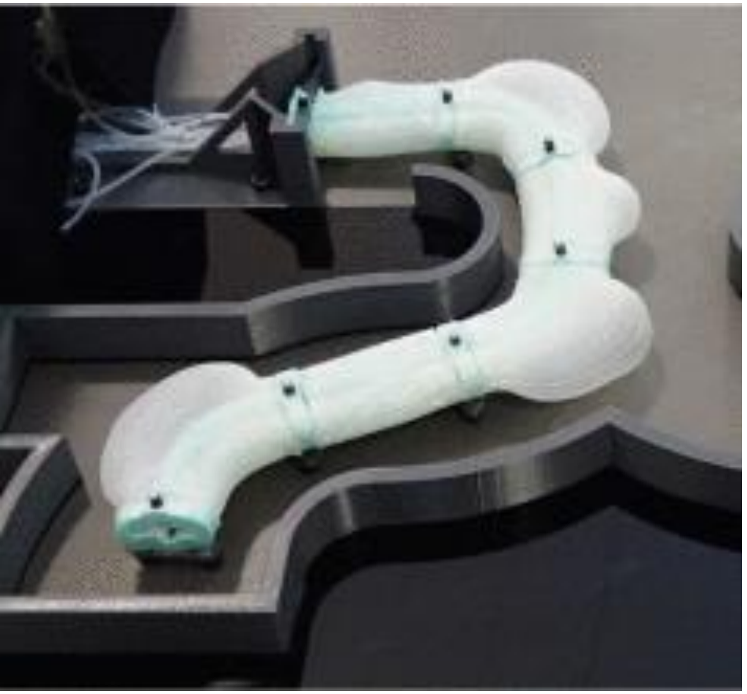
The final transformation matrix from tip to base is

$$\mathbf{T}_n^0 = A_1(q_1) \cdots A_n(q_n)$$

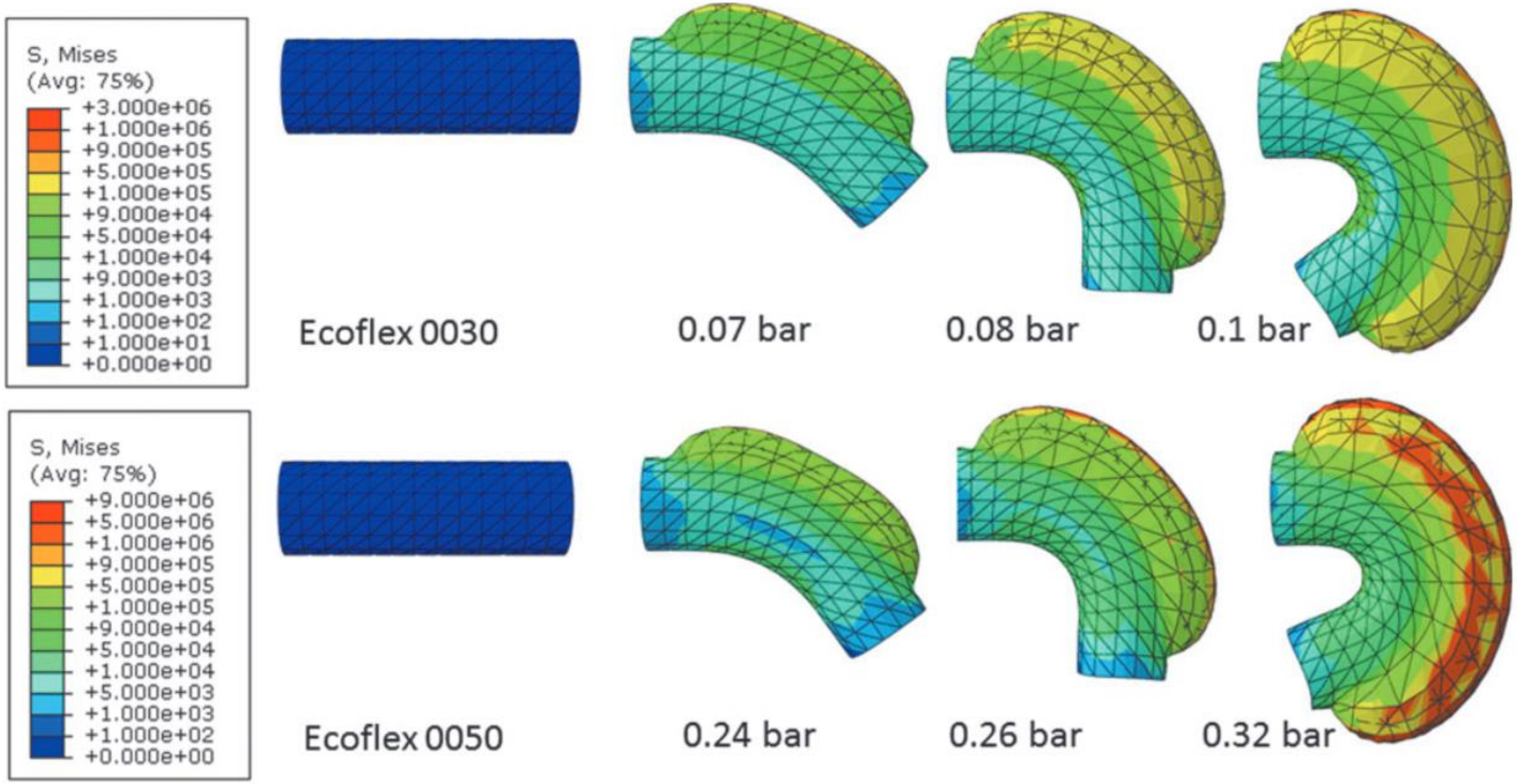
How do we do kinematics with non-rigid robots?

These robots actually have an infinite number of DOF!

Where do we put the links and joints?



Marchese et al. SoRo 2015

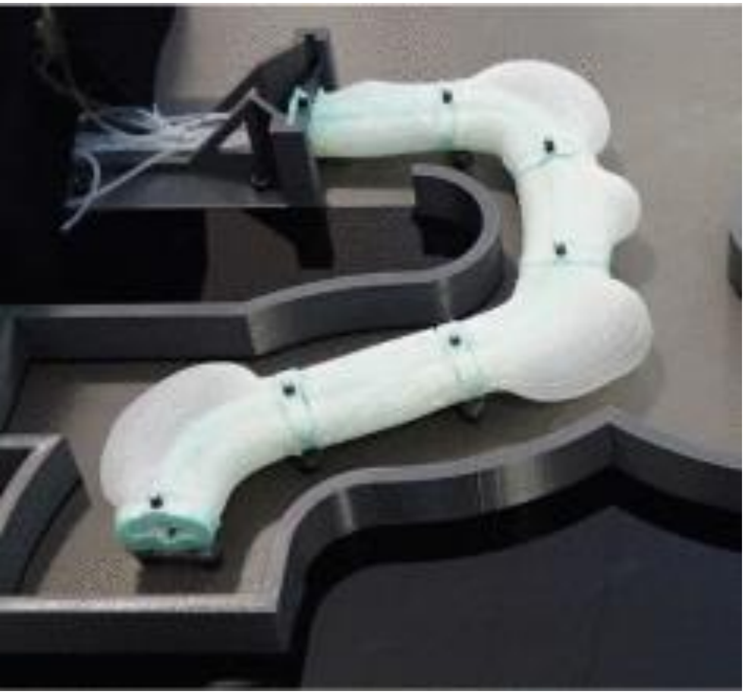


Elsayed et al. SoRo 2014.

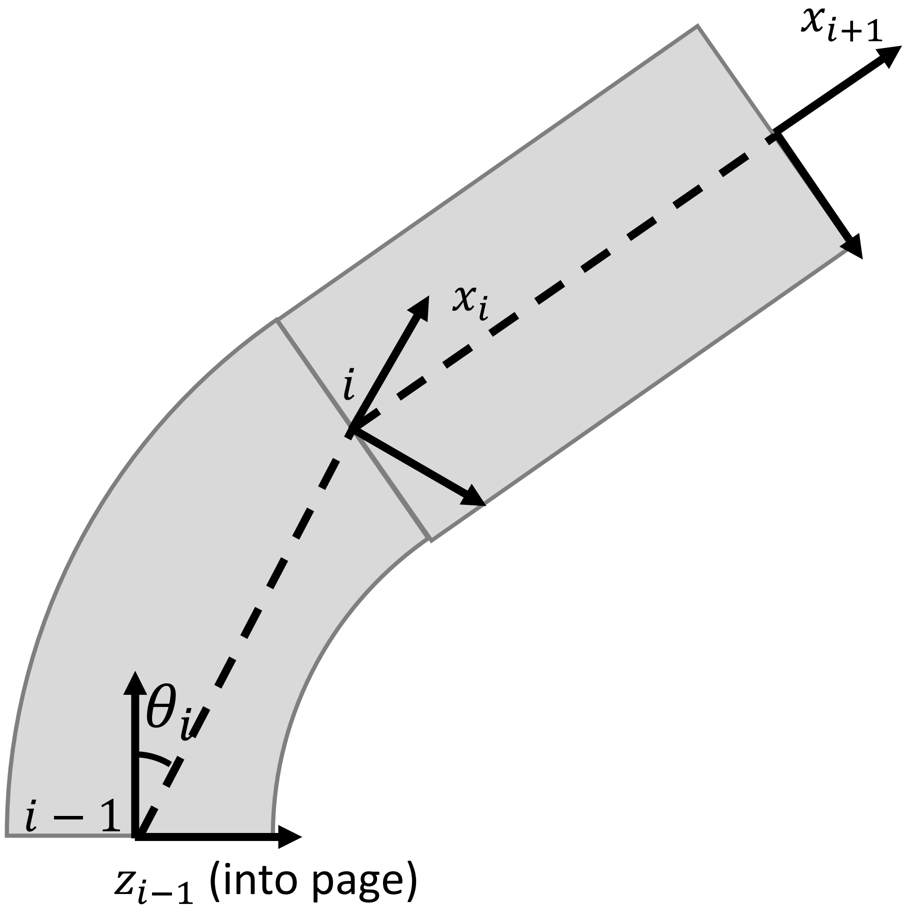
How do we do kinematics with non-rigid robots?

The **Denavit-Hartenberg transform** results from successive rotations and translations via the four DH parameters

Where do we put the links and joints?



Marchese et al. SoRo 2015



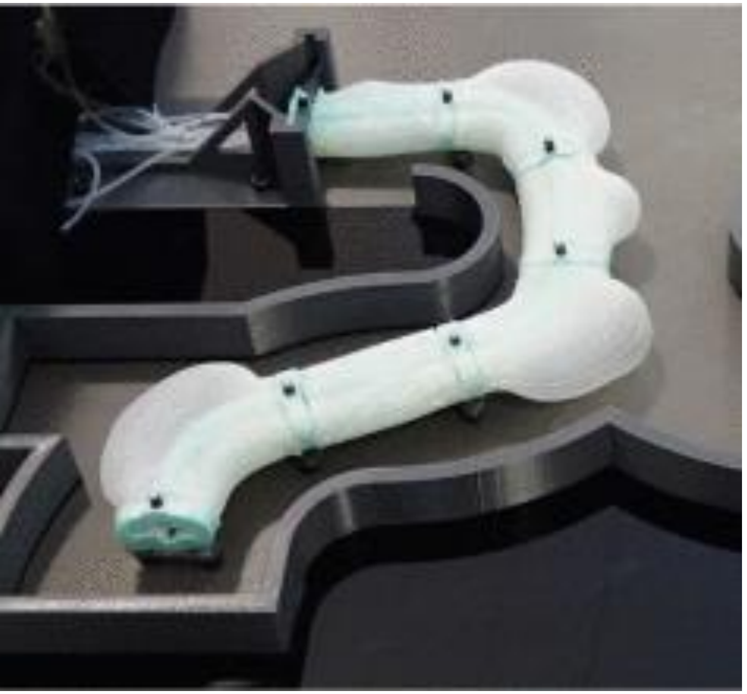
A_{i+1} matrix depends on θ_i!

a_i Link Length	distance between z_{i-1} and z_i, measured along x_i
α_i Link Twist	angle between z_{i-1} and z_i, measured in the plane normal to x_i (RHR)
d_i Link Offset	distance between x_{i-1} and x_i, measured along z_{i-1}
θ_i Joint Angle	angle between x_{i-1} and x_i, measured in the plane normal to z_{i-1} (RHR)

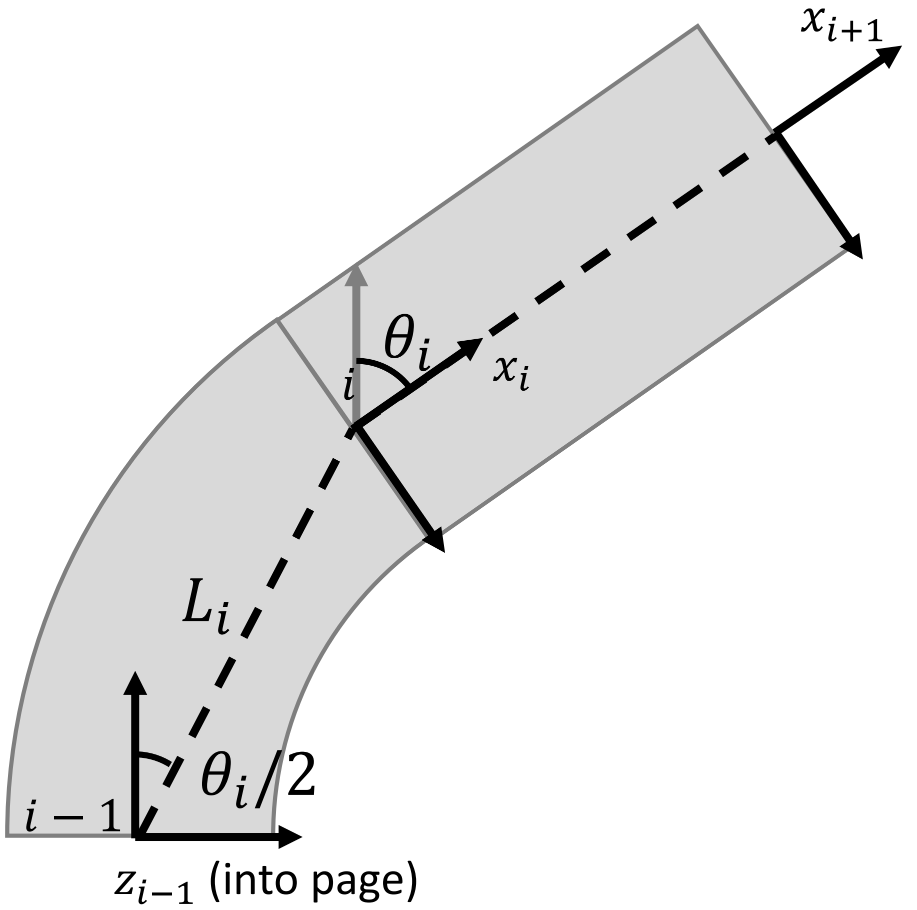
How do we do kinematics with non-rigid robots?

The **Denavit-Hartenberg transform** results from successive rotations and translations via the four DH parameters

Where do we put the links and joints?



Marchese et al. SoRo 2015

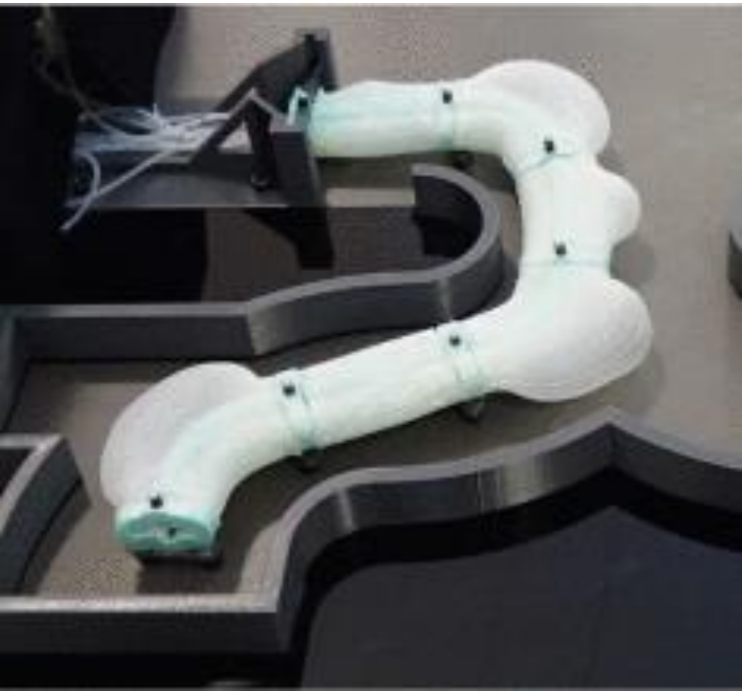


$$A_i = \begin{bmatrix} R(z_{i-1},\theta_i) & \begin{matrix} L_i\cos(\theta_i/2) \\ L_i\sin(\theta_i/2) \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

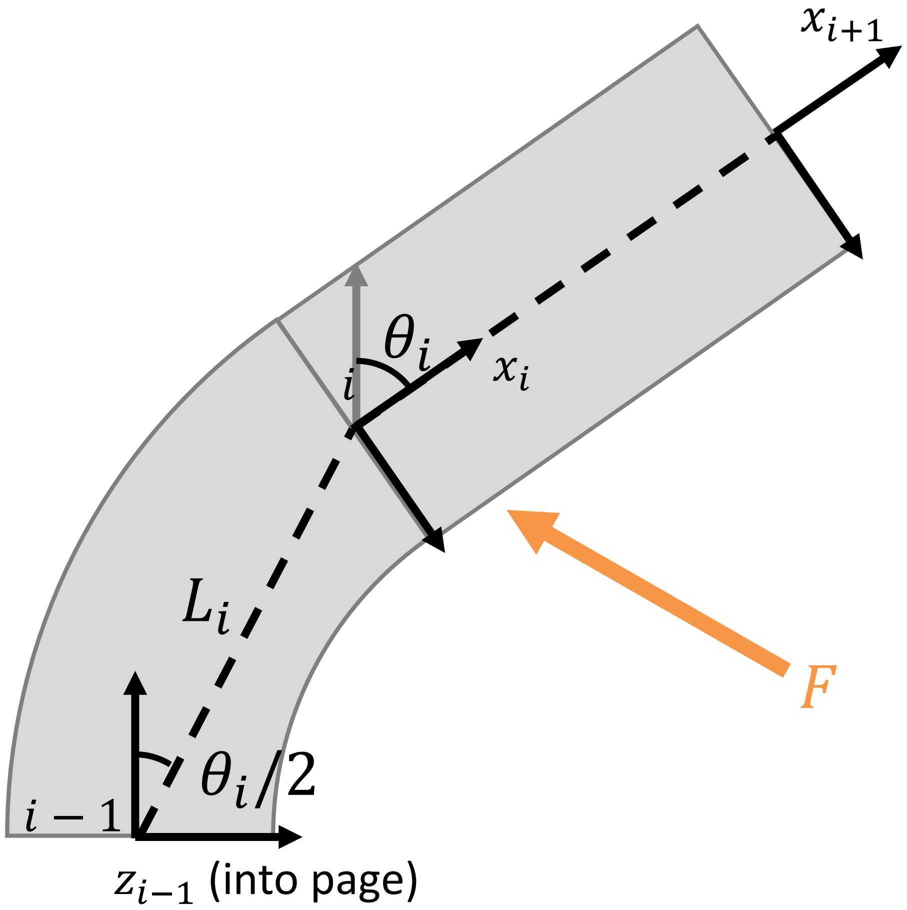
a_i Link Length	distance between z_{i-1} and z_i, measured along x_i
α_i Link Twist	angle between z_{i-1} and z_i, measured in the plane normal to x_i (RHR)
d_i Link Offset	distance between x_{i-1} and x_i, measured along z_{i-1}
θ_i Joint Angle	angle between x_{i-1} and x_i, measured in the plane normal to z_{i-1} (RHR)

How do we do kinematics with non-rigid robots?

Where do we put the links and joints?



Marchese et al. SoRo 2015

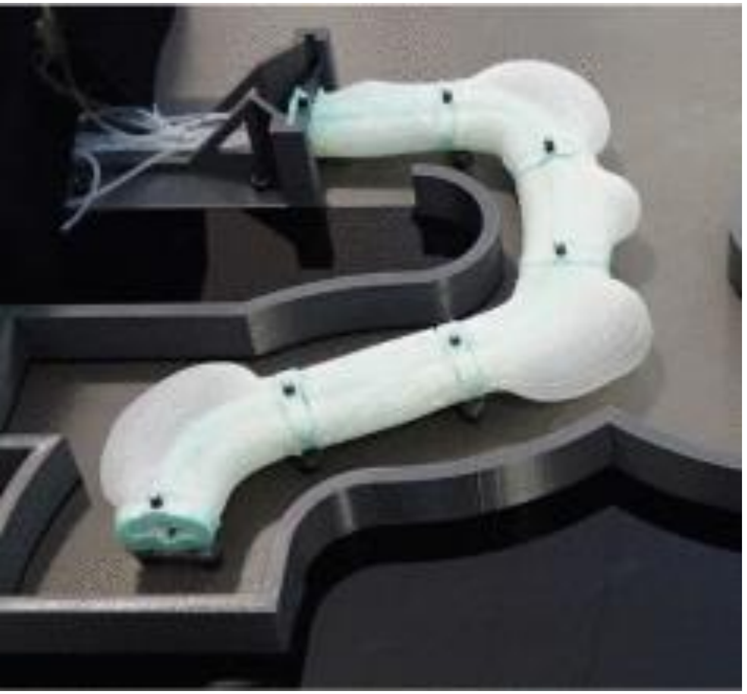


$$A_i = \begin{bmatrix} R(z_{i-1}, \theta_i) & \begin{matrix} L_i\cos(\theta_i/2) \\ L_i\sin(\theta_i/2) \\ 0 \end{matrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

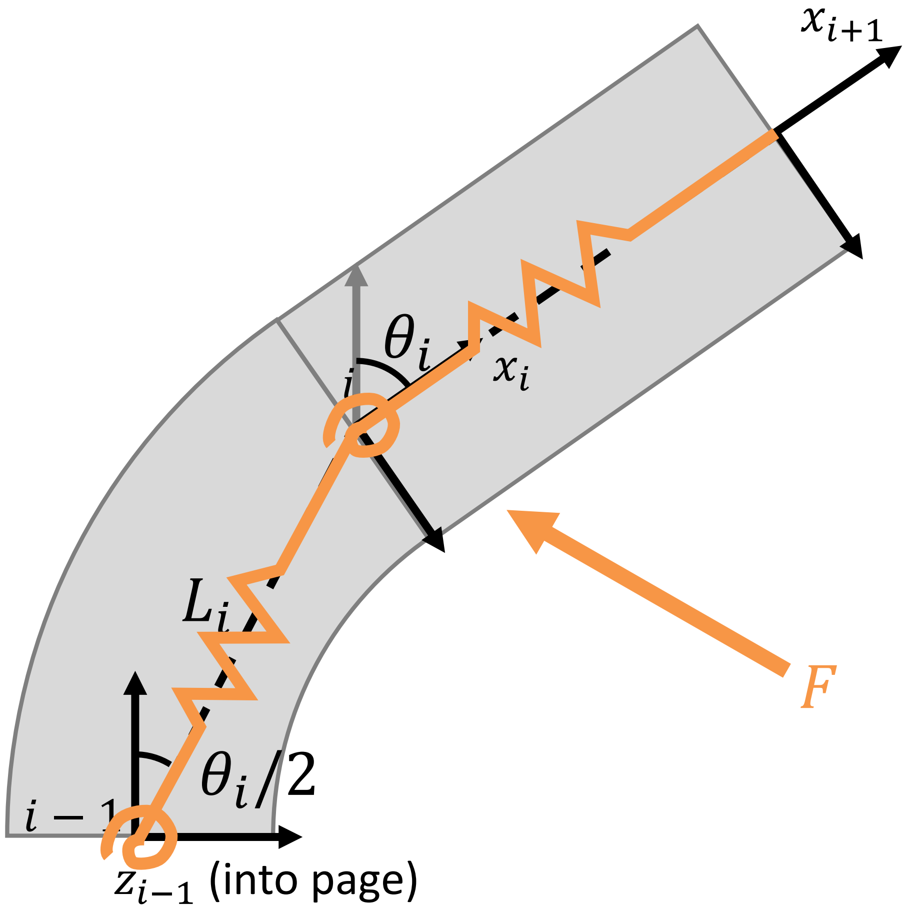
Rigid robots can resist external forces using motors and material stiffness.

How do we do kinematics with non-rigid robots?

Where do we put the links and joints?

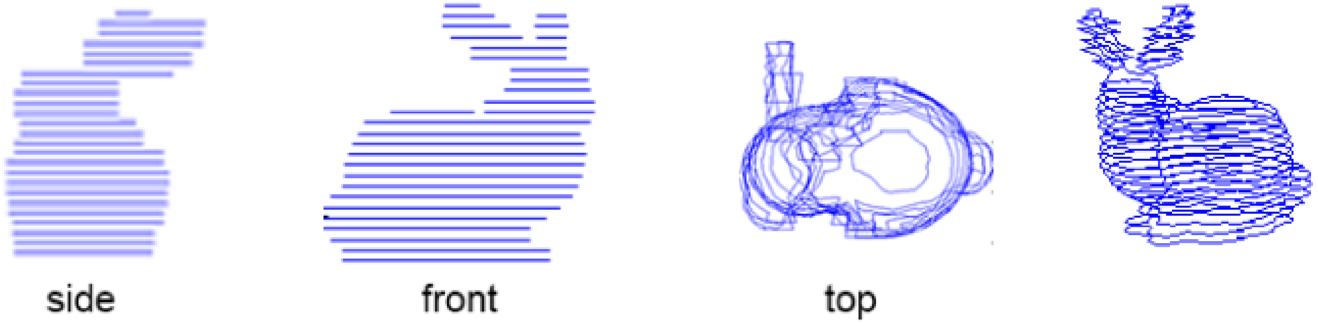


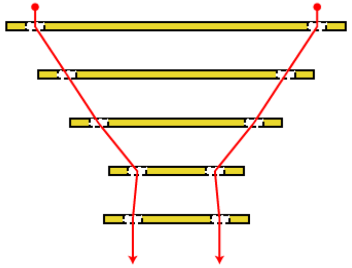
Marchese et al. SoRo 2015

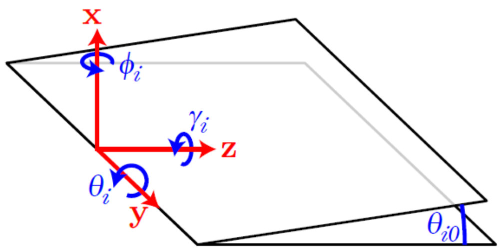


$$A_i = \begin{bmatrix} & & & L_i\cos(\theta_i/2) \\ & R(z_{i-1},\theta_i) & & L_i\sin(\theta_i/2) \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

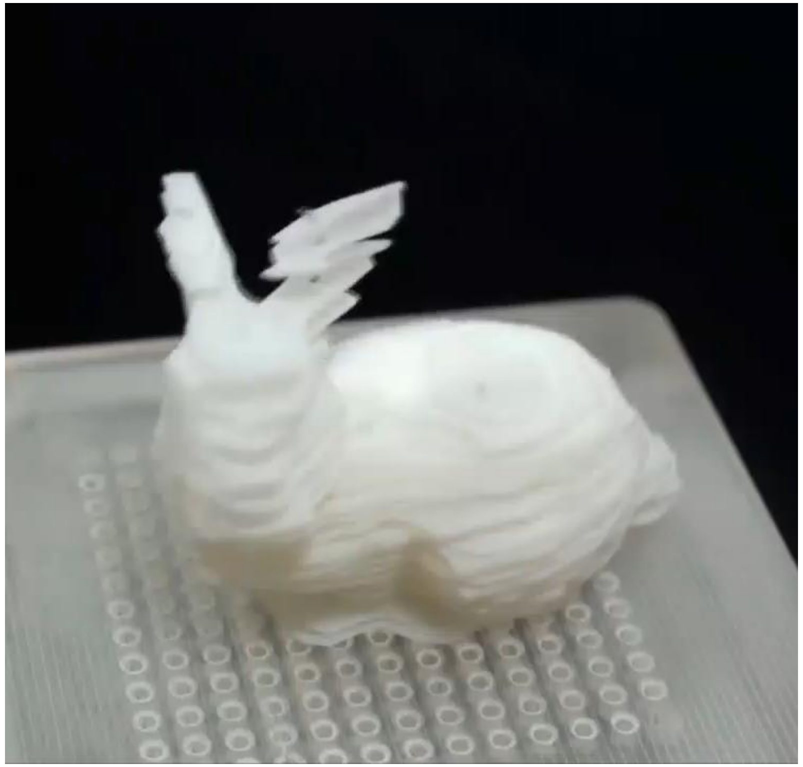
Reduced Parameter Models for Compliant Structures

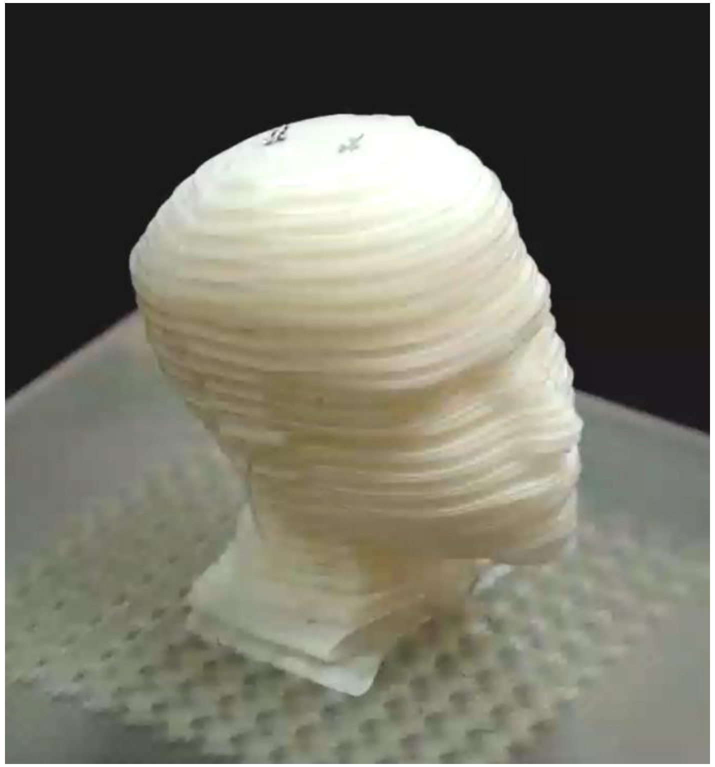






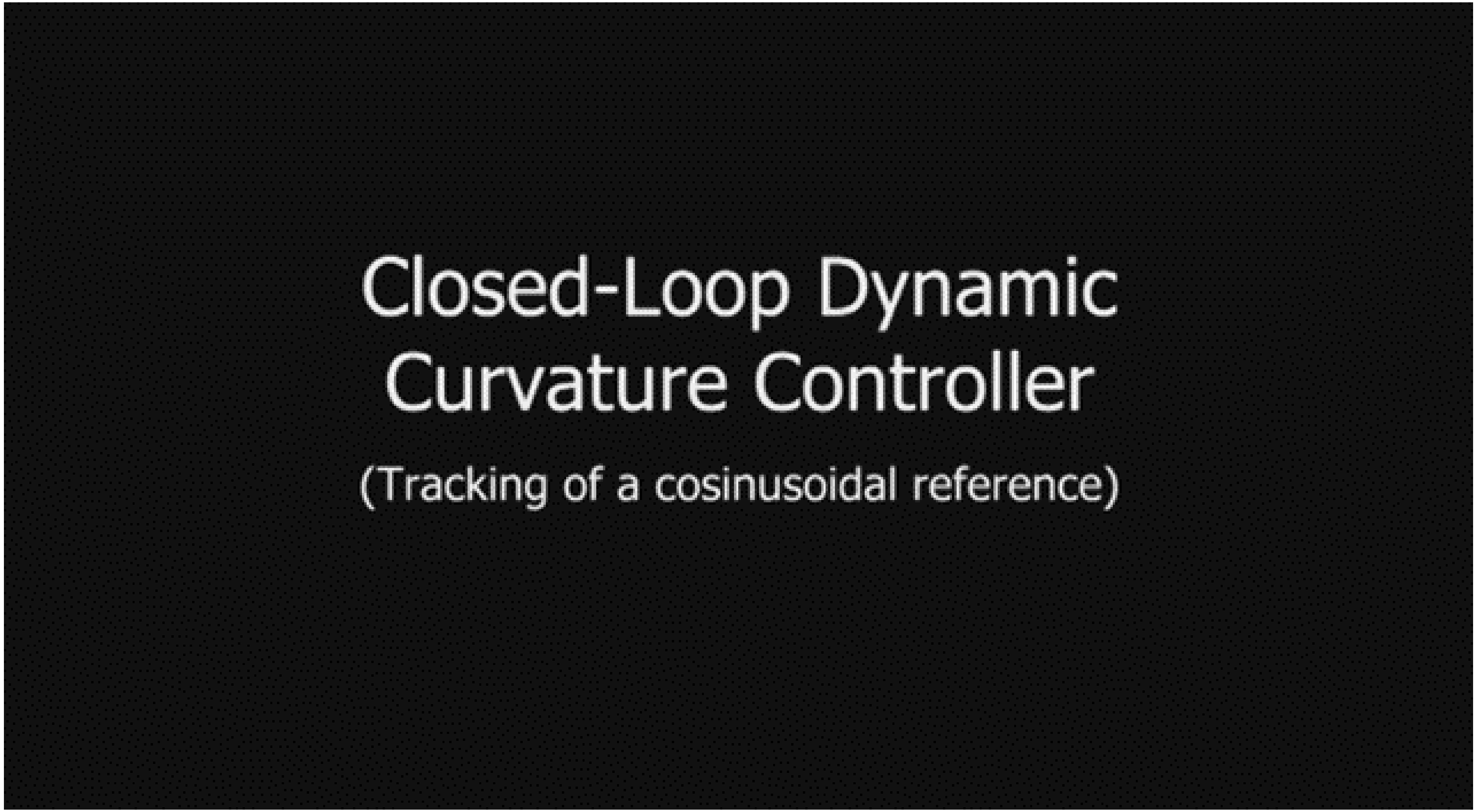
$$E = \sum_{i}^{N_f} \left[\frac{k_\theta}{2} (\theta_i - \theta_{i0})^2 + \frac{k_\phi}{2} \phi_i^2 + \frac{k_\gamma}{2} \gamma_i^2 \right]$$





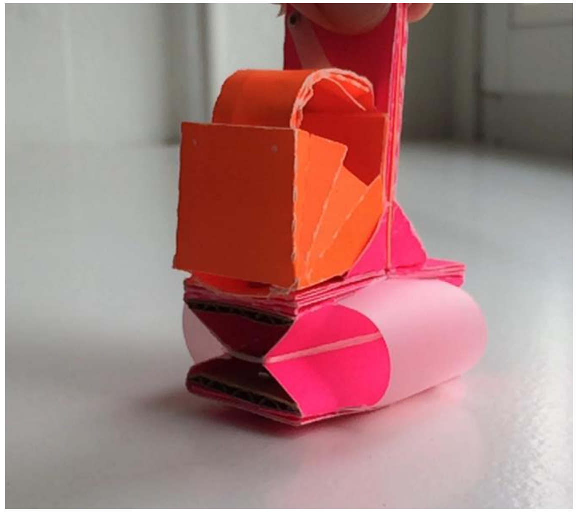
Yim, Sung, Miyashita, Rus, Kim, *IJRR* 37.6 (2018): 611-628.

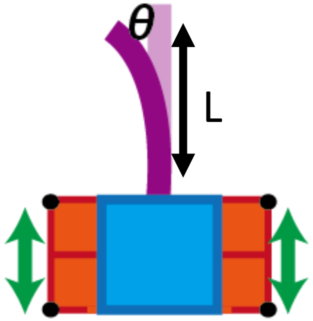
$$\tau + J^T f_{ext} = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + Kq$$



https://youtu.be/2tPLJQppJig, Santina et al. ICSR 2018

Compliant Origami Legs

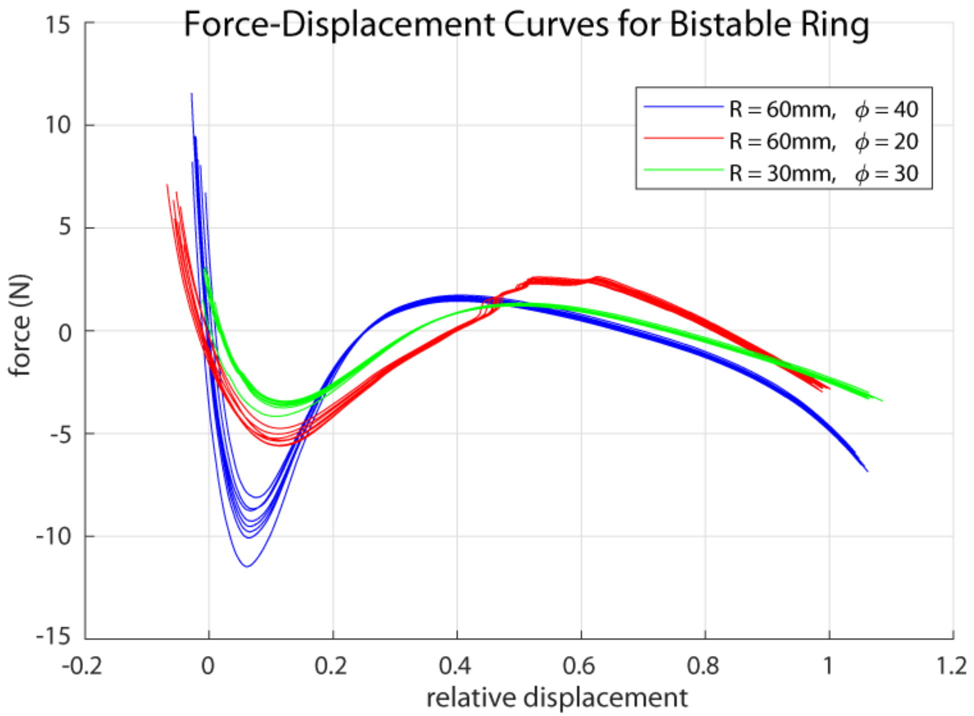


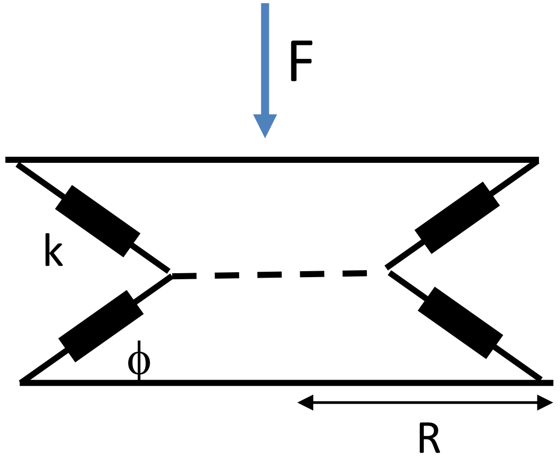


$$F_{hip} = \frac{2\theta_1 EI}{L^2}$$

Deng, Sung, *JOSME* (2018): 965-980.

Control Mechanics using Geometry



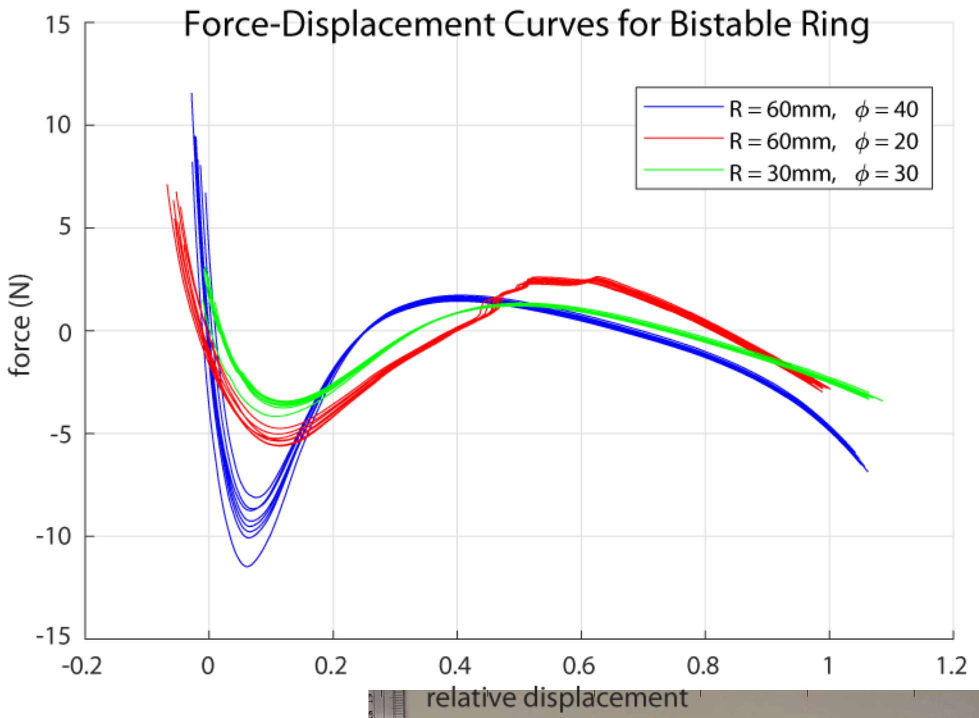


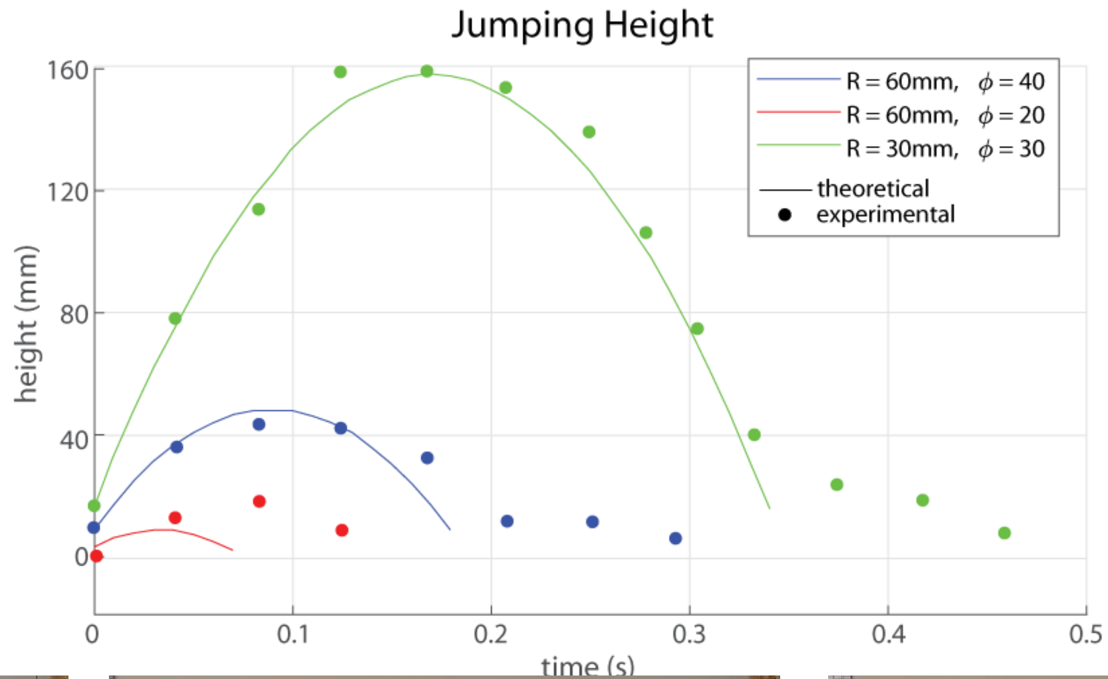
Deyuan Chen

Prof. James Pikul

Jaimie Carlson

Control Mechanics using Geometry

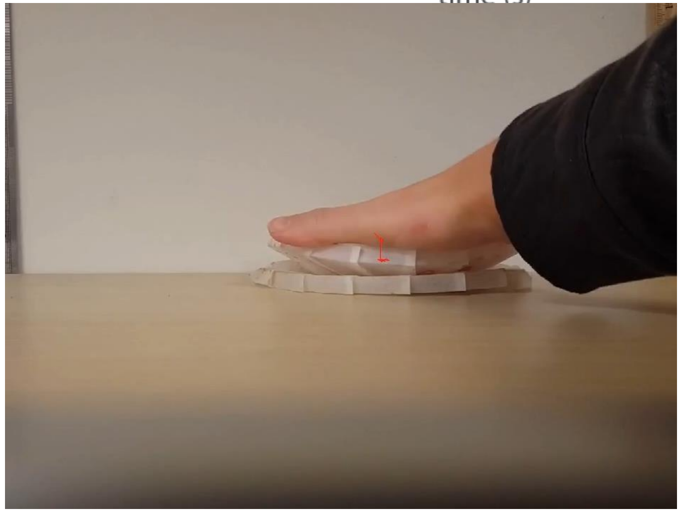


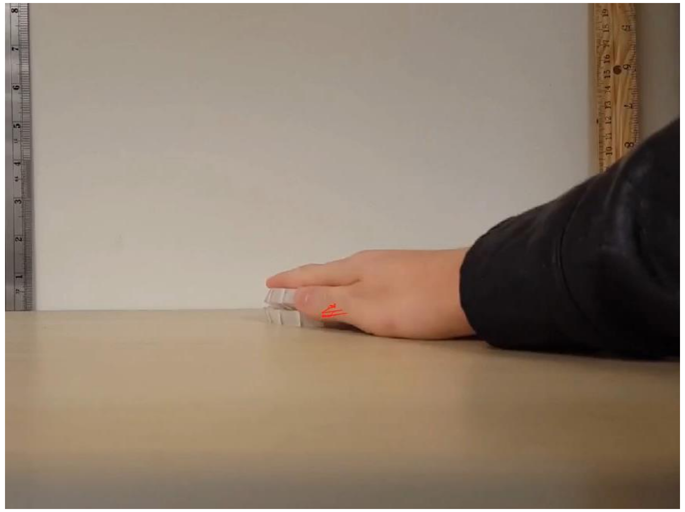


Deyuan Chen

Prof. James Pikul

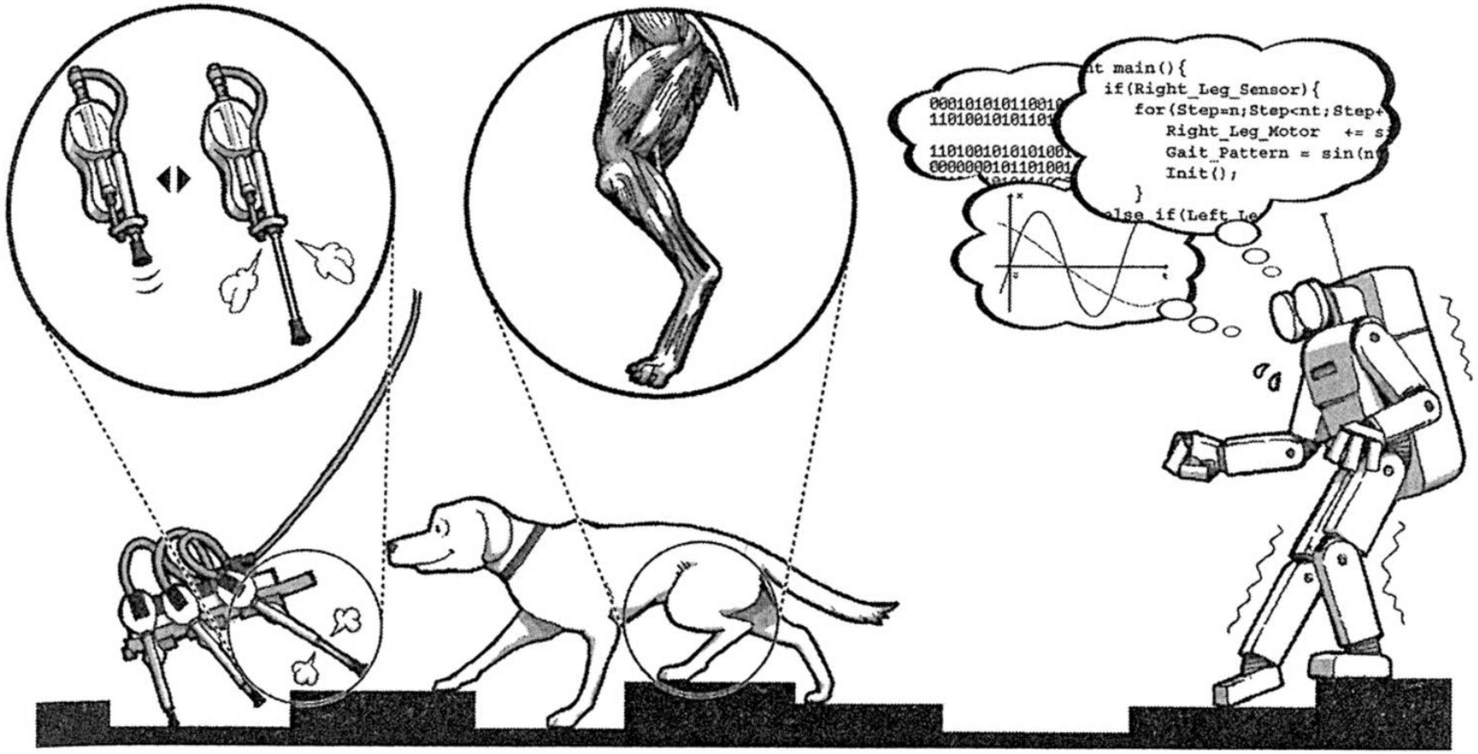
Jaimie Carlson





Morphological Computation

Appropriate use of the body morphology leads to a reduction of the total amount of computation that is required to complete the task.



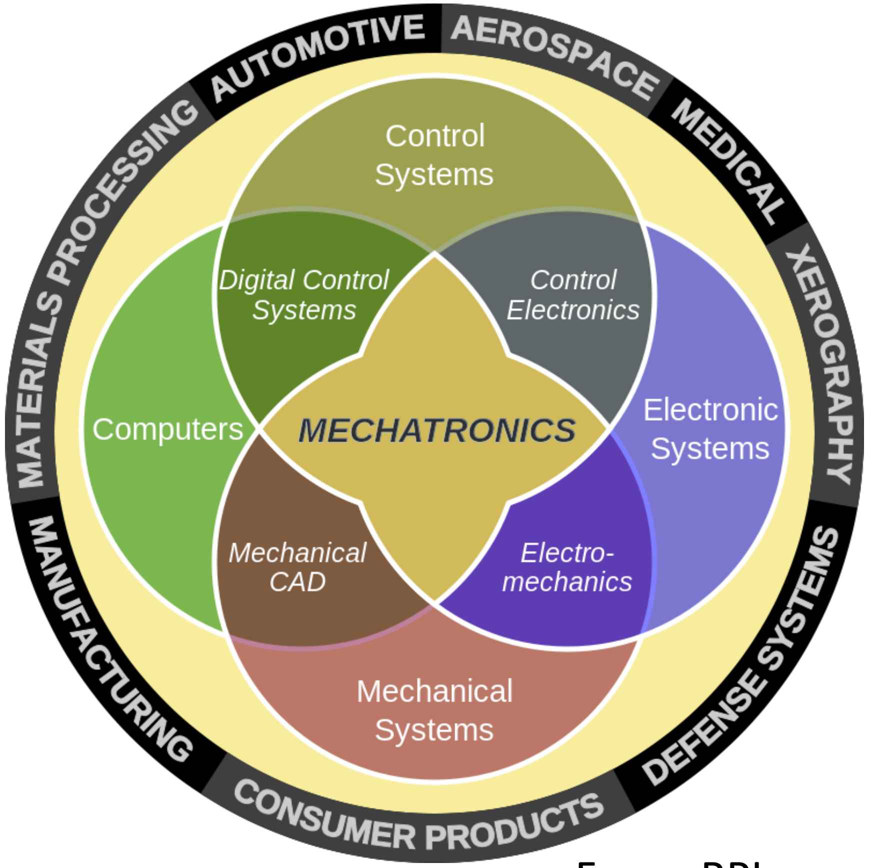
(a) (b) (c)

Müller & Hoffman, Alife 2017.

What is robotics?

Robotics is a subset of Mechatronics, the synergistic integration of mechanics, electronics, controls, and computer science.

Preface of SHV



From: RPI

Subfields of robotics (non-exhaustive)

DESIGN
mechanisms
actuators
kinematics/dynamics
bioinspired design
manufacturing

TYPES/APPLICATIONS
legged
mobile
aerial
underwater
micro/nano
manipulators and graspers
parallel robots
soft robots

SENSING
force and tactile sensing
perception
computer vision
range sensing
sensor fusion

ALGORITHMS
learning
motion planning
navigation
localization and mapping
failure recovery
robot networks
multi-robot coordination
scheduling

CONTROL
PID control
adaptive control
optimization and optimal control
collision avoidance
distributed robotics
grasping and manipulation
human-robot interaction
underactuated robotics

Next time: Final Project Presentations

Dec. 4

10: Ma, Wang
23: Lyu, Peng, Zhang
24: Li, Misra
9: Chang, Li
32: Bernstein, Schwartz, Wang, Winograd
15: Remba, Walsh
29: Shen, Wang
33: Brink, Glen
27: Blumenstein, Ingerman, Moberg
17: Li, Yang
25: Gollapudi, Patel, Wang
6: Hu, Yang
41: Zhao
13: Chari, Shinkle
31: Lan, Xie
36: Kopli
44: Shur
2: Gelb, Mitchnik, Raizen
35: Huang
1: Kao, Shatalin
11: Cao, Zhang, Zheng
18: Friberg
22: Kulkarni, Sharma

Dec. 6

39: Sorna
49: Chordia, Shirsath
38: Stabile
14: Beser, Gui, Kustikova, Salam
34: Dcunha
3: Gu, Zhao
8: Brundage, Grey, Hawkes, van Hoffelen
7: Chun, Hussey, Mok, Zhao
19: Jia, Zhou
4: Collins, Fine, Tu
16: Hussein, Wain
43: Mohan
5: Wang, Yu, Zhang
42: Huang
28: He
37: Ning
26: Sun, Zheng
20: Arcot, Vasudevan
40: Sekar
30: Bhat, Kalluraya, Yadav
21: Kaufman, Savant
12: Hsu, Scheuer