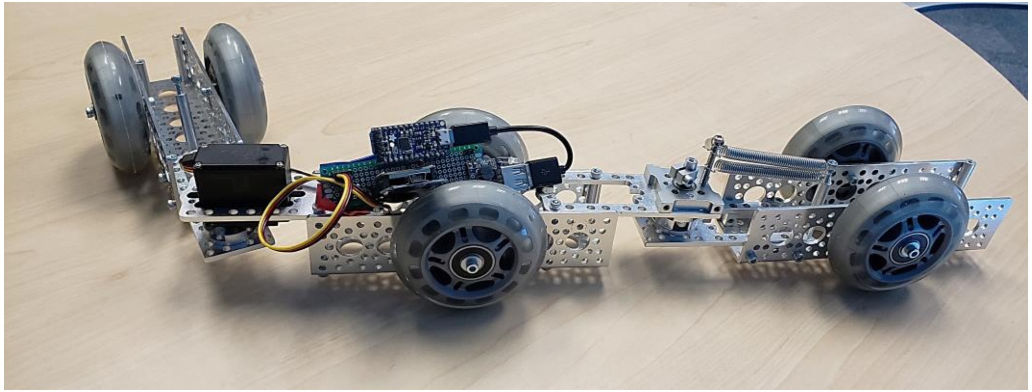
COMS W4733: Computational Aspects of Robotics

Lecture 28: Applications and State of the Art



Instructor: Tony Dear

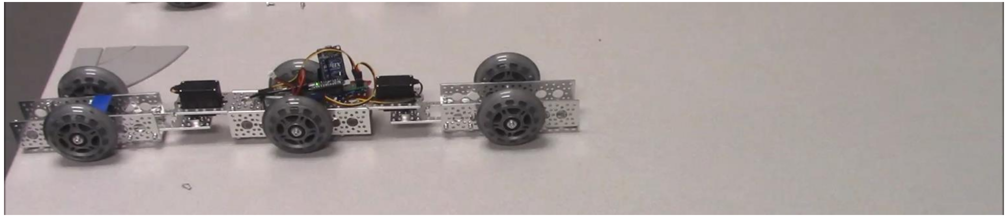
The Field of Robotics

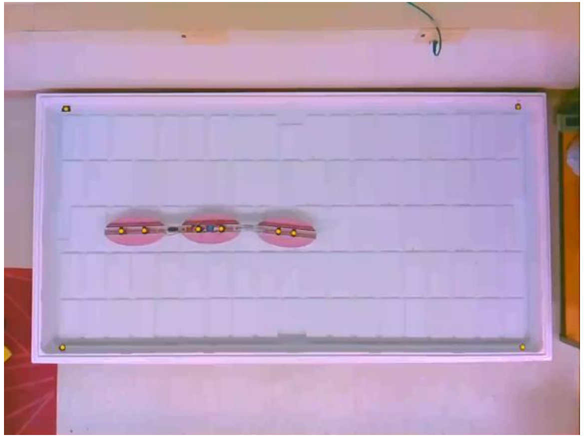
- Robotics is a very *interdisciplinary* area that only partially encompasses CS
- This class: Covered “computational” aspects of many common tasks and goals

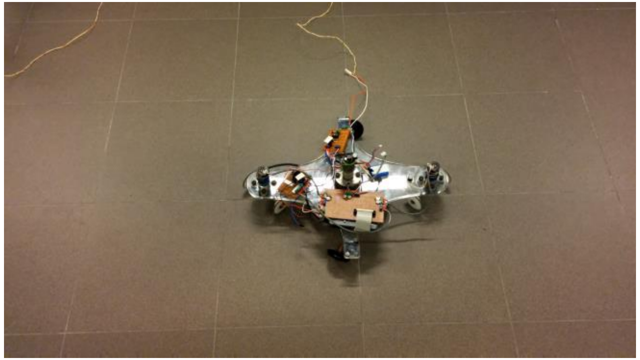
- In a sense, all of robotics is computational!
- We’ve only barely scratched the surface in both methods and applications

- Case study: Robot locomotion (my own research)

Locomotion Geometry



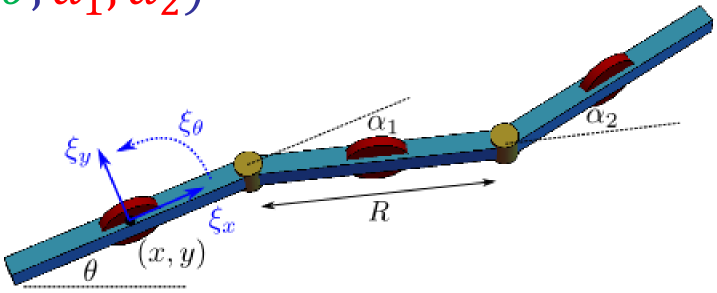




Kinematic Structure

- Configuration variables: $\boldsymbol{q} = (\boldsymbol{g}, \boldsymbol{\alpha})^T = (x, y, \theta, \alpha_1, \alpha_2)$

- Body velocities: $\boldsymbol{\xi} = \left(\xi_x, \xi_y, \xi_\theta\right)^T = \boldsymbol{R}(\theta)\dot{\boldsymbol{g}}$



- Nonholonomic constraints: $\boldsymbol{\omega}_\xi(\boldsymbol{\alpha})\boldsymbol{\xi} + \boldsymbol{\omega}_\alpha(\boldsymbol{\alpha})\dot{\boldsymbol{\alpha}} = 0$

Position variables ("outputs")

$$\boxed{\boldsymbol{\xi} = -\boldsymbol{A}(\boldsymbol{\alpha})\dot{\boldsymbol{\alpha}}}$$

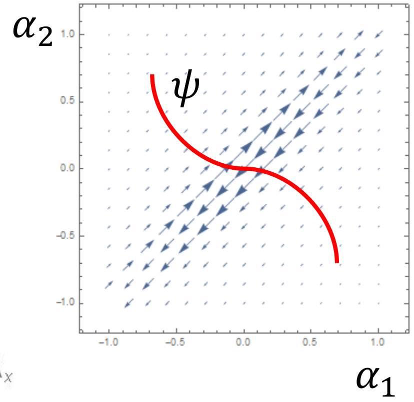
Shape variables ("inputs")

Connection mapping

Geometric Motion Planning

$\xi = -A(\boldsymbol{\alpha})\dot{\boldsymbol{\alpha}}$

Input trajectory: $\psi: [0, t] \to \boldsymbol{\alpha}$



Integrate velocity -> odometry:

$$-\int A_i(\boldsymbol{\alpha}(t))\dot{\boldsymbol{\alpha}}(t)\,dt = -\int_\psi A_i(\boldsymbol{\alpha})\,d\boldsymbol{\alpha}$$

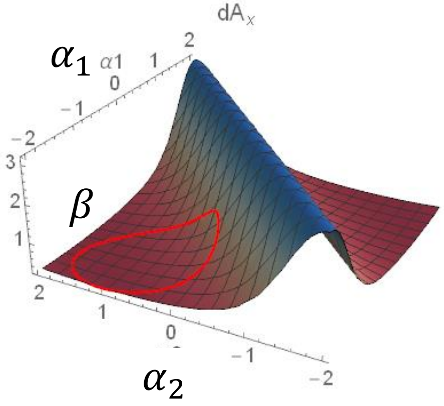
Stokes' theorem

ψ is periodic -> closed curve in shape space

$$-\int_\beta dA_i(\boldsymbol{\alpha})$$

Measure of displacement

exterior derivative

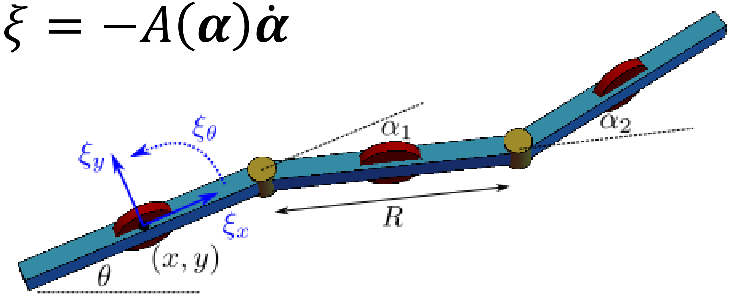


Snake Robot Connection

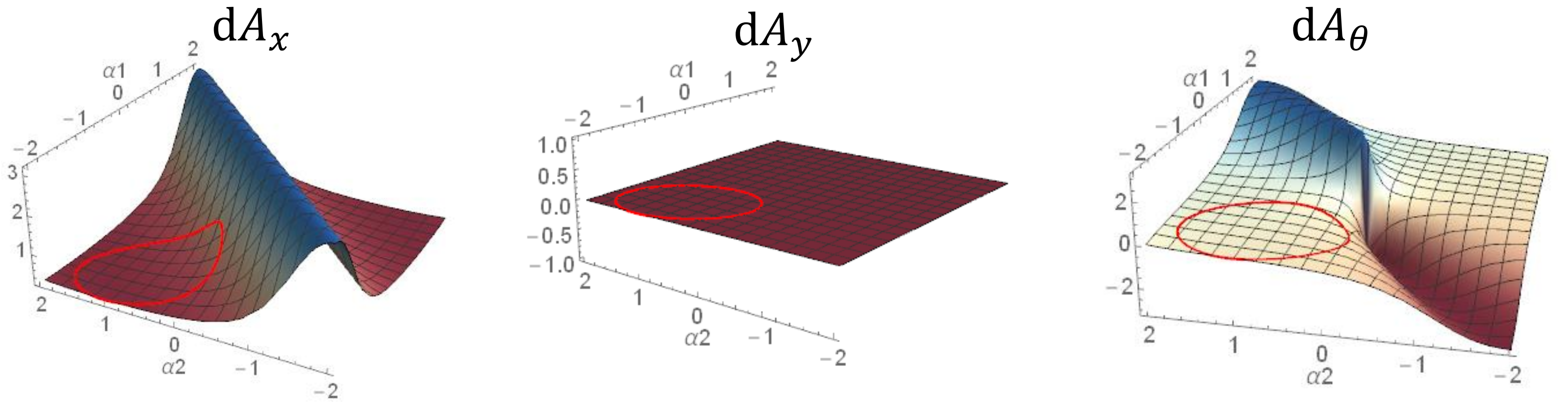
$$A(\boldsymbol{\alpha}) = \frac{1}{D}\begin{pmatrix} \frac{R}{2}(\cos\alpha_1 + \cos(\alpha_1-\alpha_2)) & \frac{R}{2}(1+\cos\alpha_1) \\ 0 & 0 \\ \sin\alpha_1 + \sin(\alpha_1-\alpha_2) & \sin\alpha_1 \end{pmatrix}$$

$$D = \sin\alpha_1 + \sin(\alpha_1-\alpha_2) - \sin\alpha_2$$

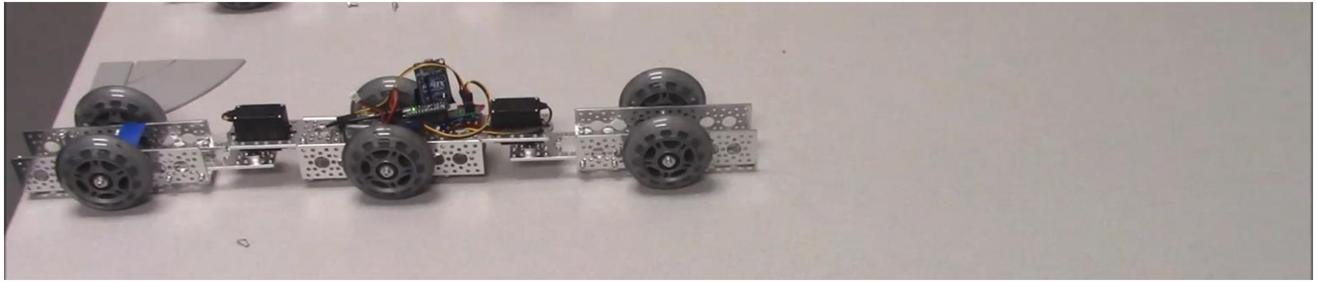
$$\xi = -A(\boldsymbol{\alpha})\dot{\boldsymbol{\alpha}}$$

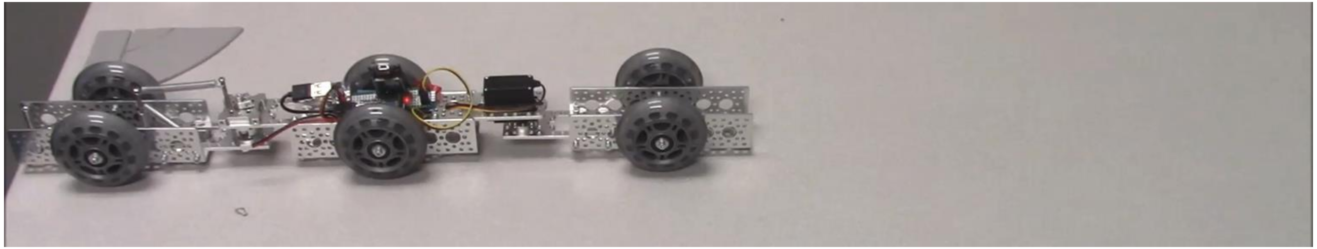


Gaits are closed loops
in shape space



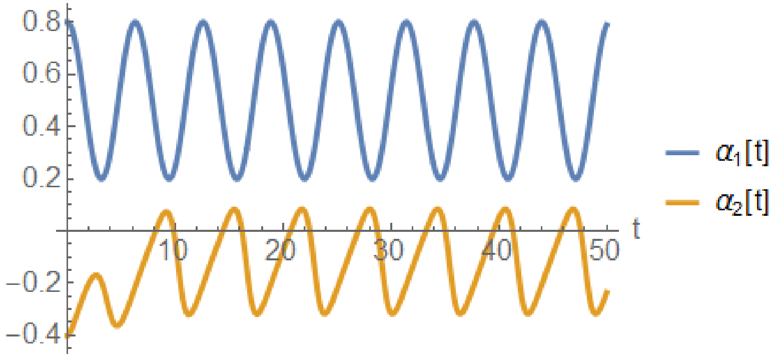
A “Small” Modification: Motor to Spring

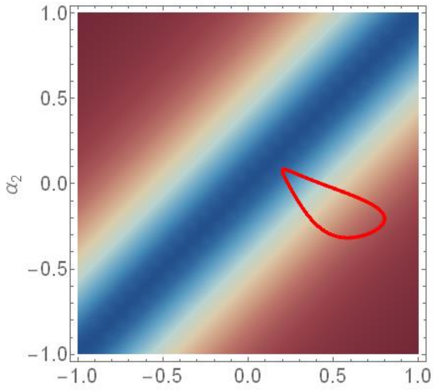




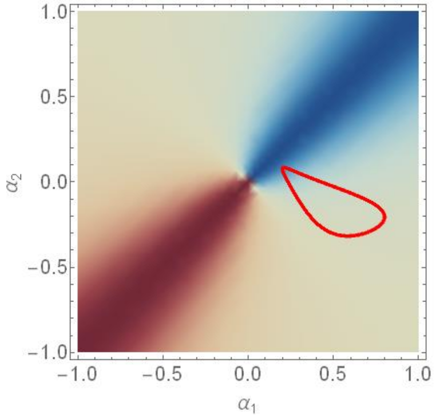
Non-Singular Gaits

- Both joints start far away from each other
- Damping slows the compliant joint more than spring pushing it





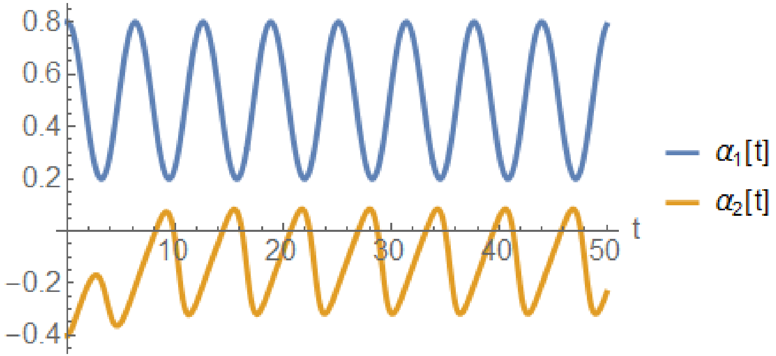
dA_x

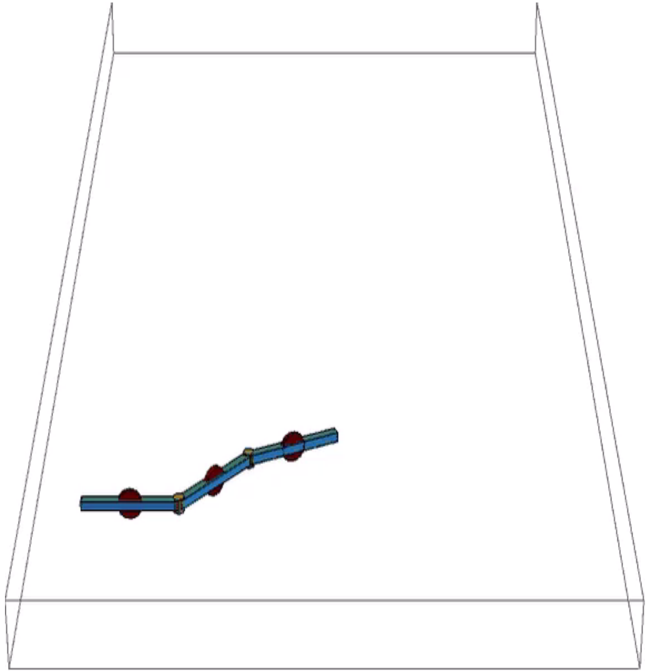


dA_θ

Non-Singular Gaits

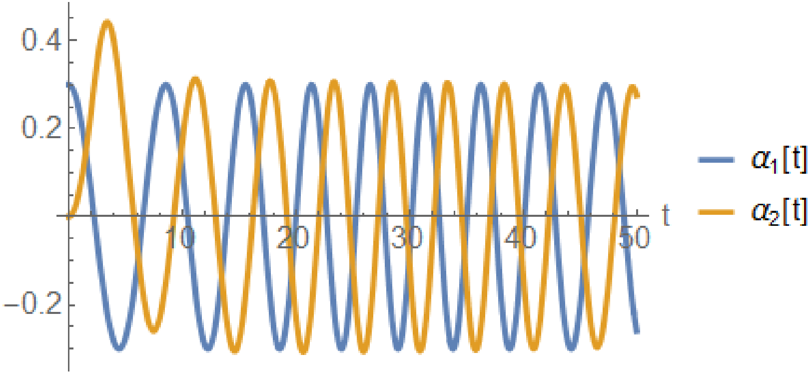
- Both joints start far away from each other
- Damping slows the compliant joint more than spring pushing it

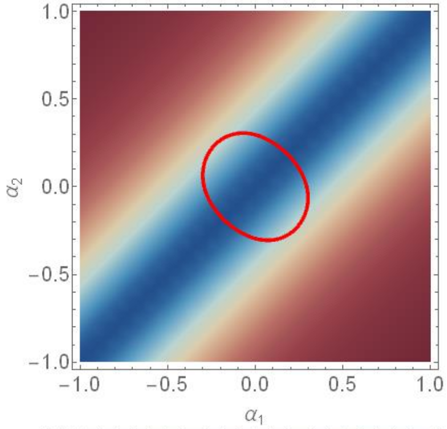




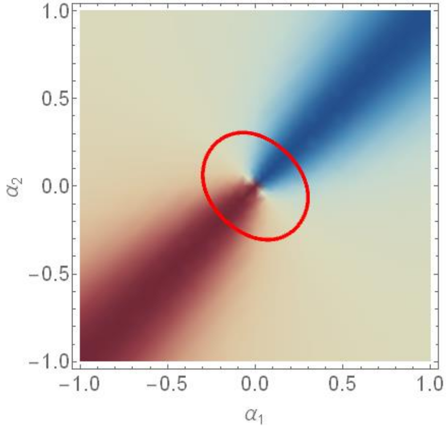
Singular Gaits

- Joints start near one another
- Insufficient damping to stop compliant joint crossing α_1





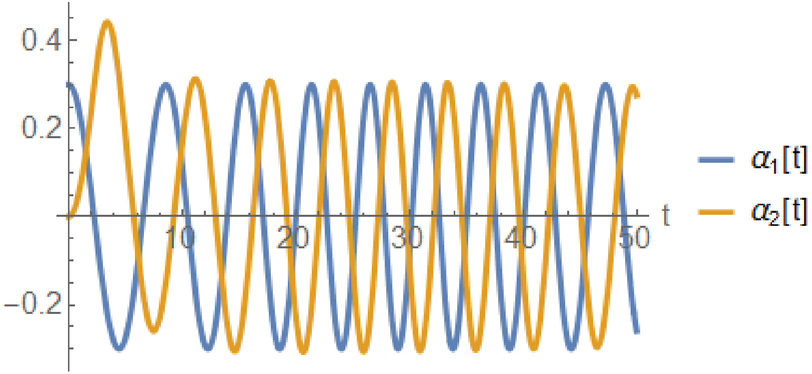
$\mathrm{d}A_x$

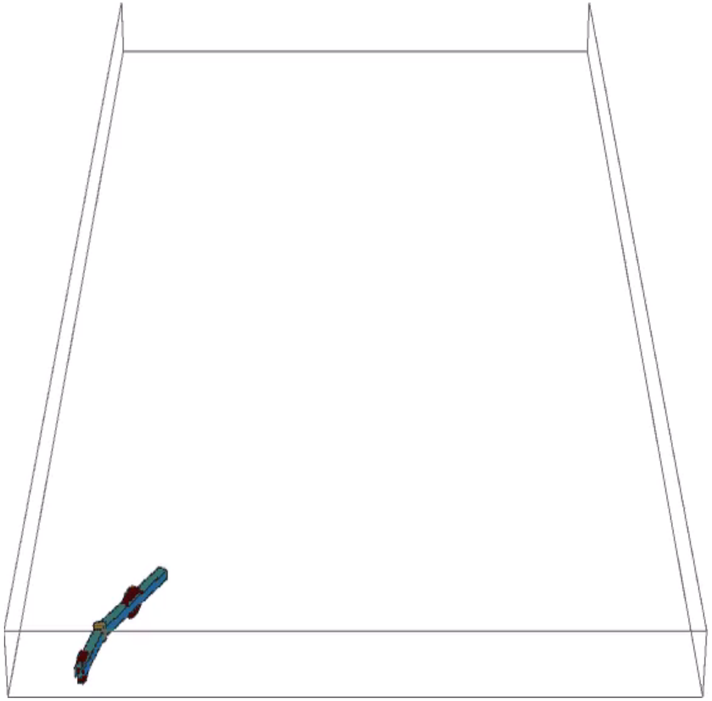


$\mathrm{d}A_\theta$

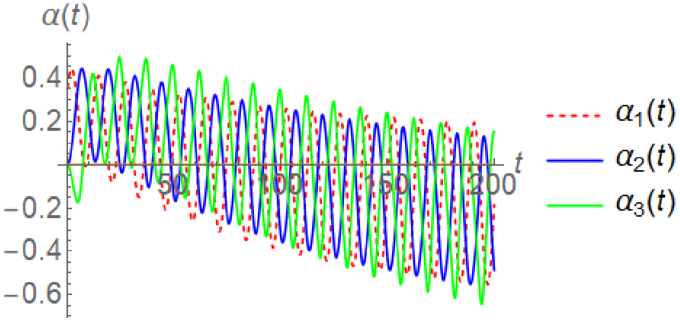
Singular Gaits

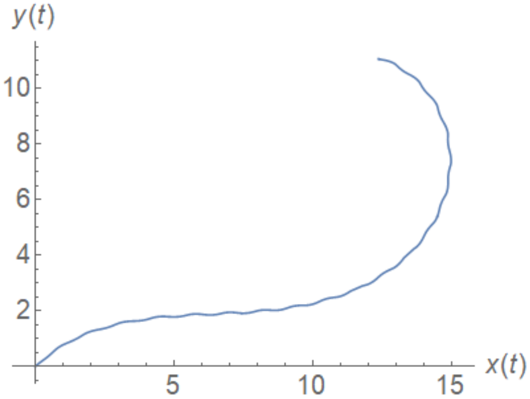
- Joints start near one another
- Insufficient damping to stop compliant joint crossing α_1

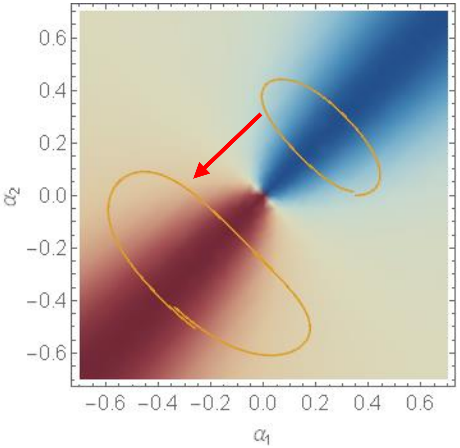




Stitching Gaits Together

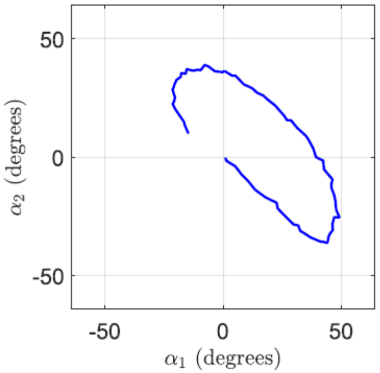


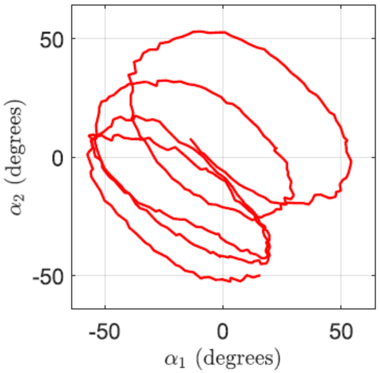


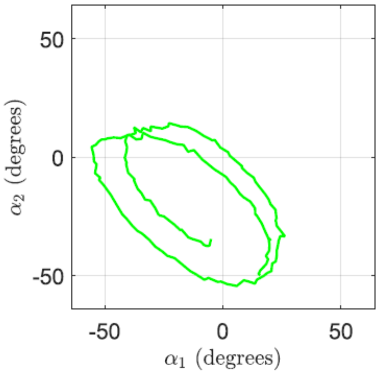


$\mathrm{d}A_\theta$

Navigation and Obstacle Avoidance

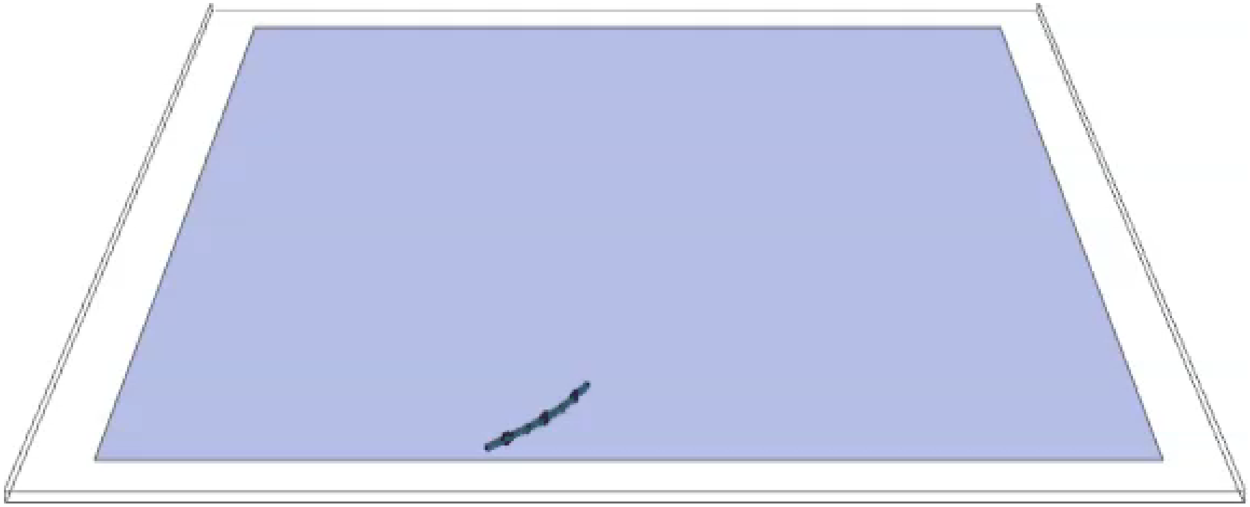






Adding External Actuation

- The robot system can be arbitrarily complex as long as we can model (or approximate the model) as kinematic—visual tools still apply!
- Ex: Passive snake on a moving plane



The Field of Robotics

- Robotics is a very *interdisciplinary* area that only partially encompasses CS
- This class: Covered "computational" aspects of many common tasks and goals

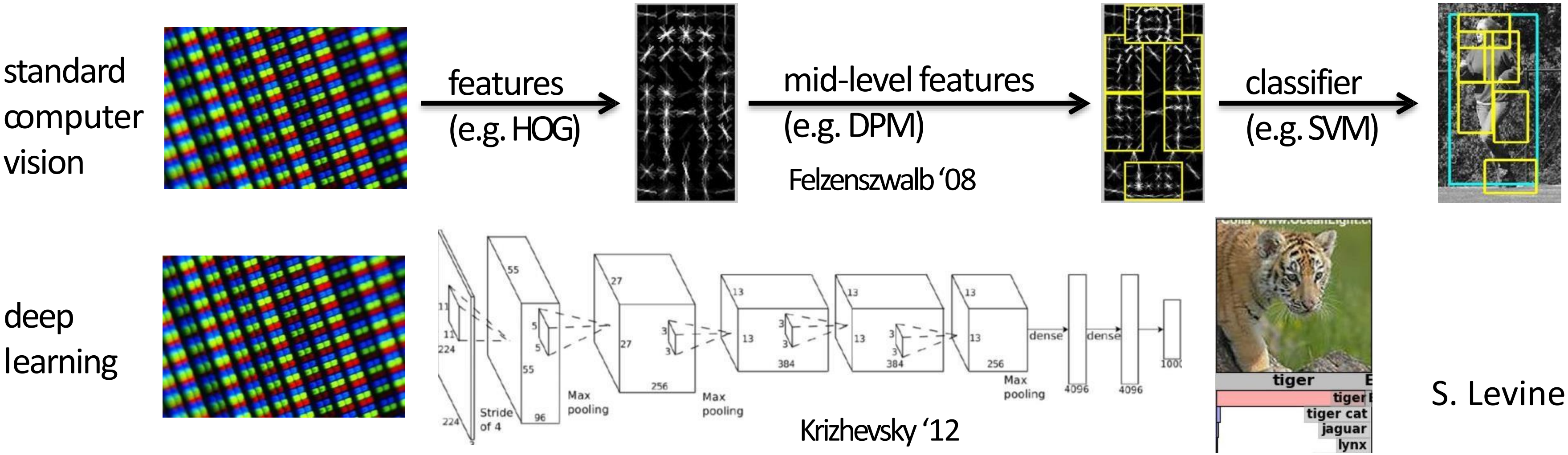
- In a sense, all of robotics is computational!
- We've only barely scratched the surface in both methods and applications

- Many other areas: Vision, learning, manipulation, planning, dynamics and control, multi-agent, human-robot interactions, …
- Applications: industrial, medical, field, aquatic, bio-inspired, household, education, entertainment, autonomous vehicles, …

Case Study: Deep Learning and Robotics

- Many sub-areas and applications in robotics share tools and advances
- Recent successes in **deep learning** have contributed to multiple areas

- Deep learning in computer vision:

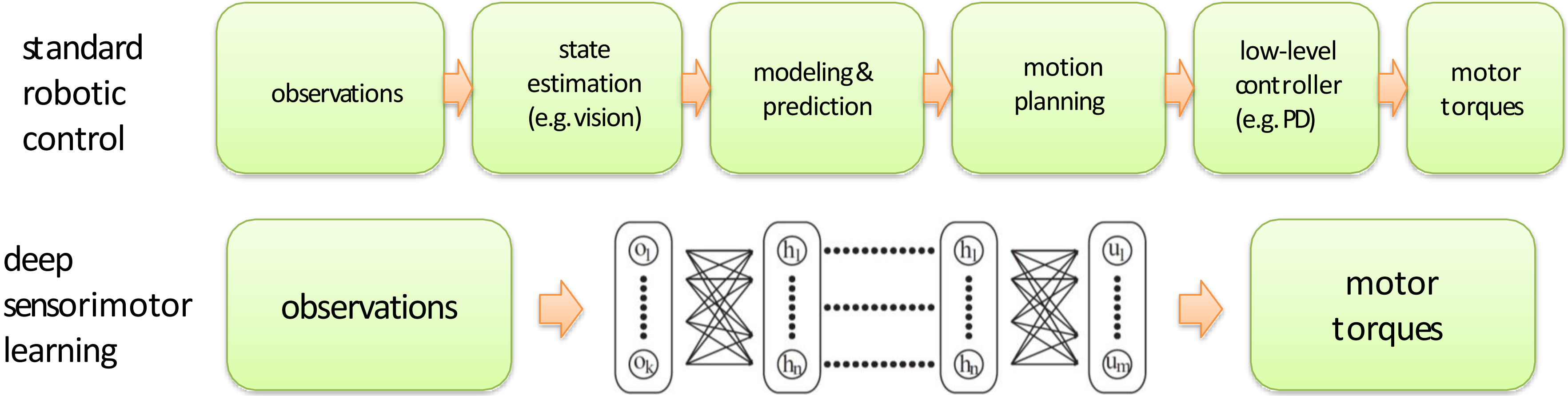


S. Levine

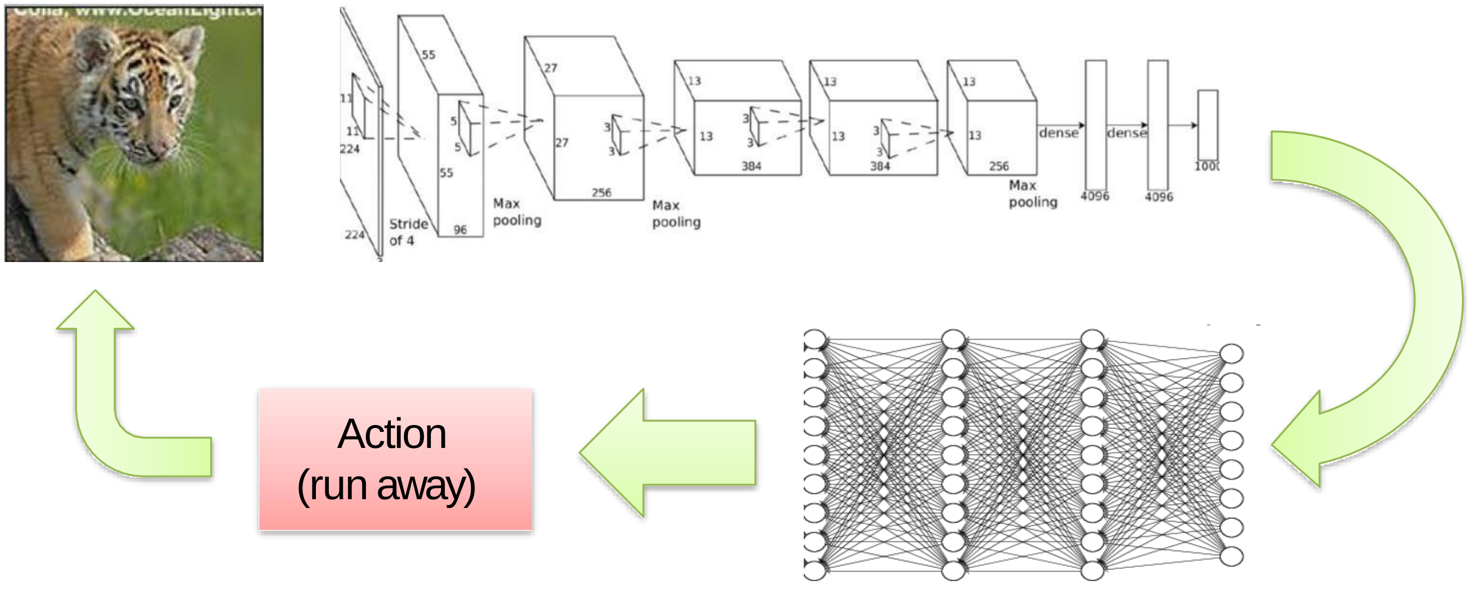
Deep Learning for Robot Control

- Deep neural networks are really just fancy results of a regression
- We saw that many RL tasks can also be implemented using deep learning

- Ex: Robot control instead of vision

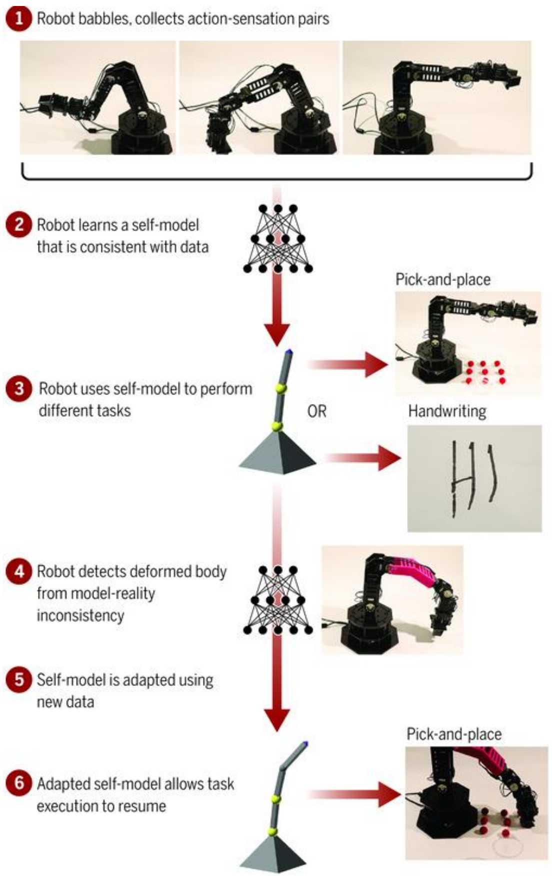


Deep Sensorimotor Loop

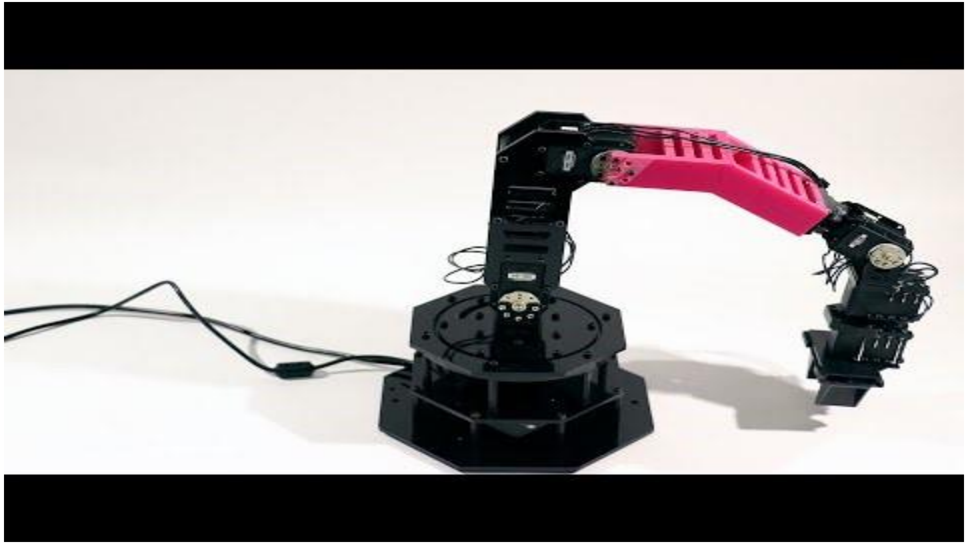


Deep Self Modeling

- R. Kwiatkowski and H. Lipson, 2019
- Instead of specifying a physical model, learn it entirely via RL and represent it with a DNN
- Model is not fixed; robot can continue learning and updating it as it gets new info
- Maybe closer to human control?



Deep Self Modeling



Deep Learning and Manipulation

- *Robot manipulation* is a highly complex task that combines (visual) perception, high- and low-level planning, and kinematic control
- Integrated sensorimotor loop is a natural candidate for a DNN representation
- Gathering data at Google Brain (S. Levine et al.):

Deep Learning for Pose Estimation

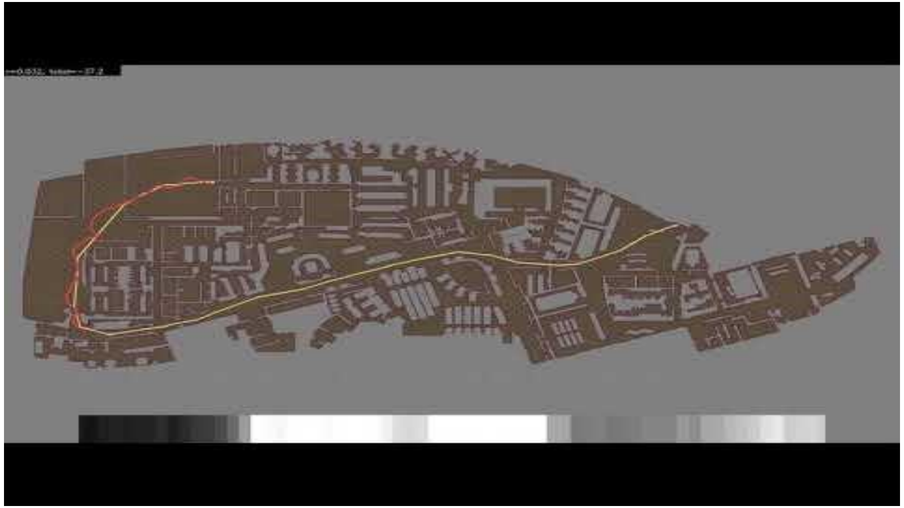
- Manipulation requires that a robot can recognize the pose of an object
- If using deep learning, large amounts of data are required
- NVIDIA: Successfully trained a DNN using synthetic (i.e., unlimited) data

Deep Learning and Assistive Robots

- Robots have huge potential to help people who need assistance in different ways, whether due to age, disability, injury, etc.
- E.g.: Fetching objects, preparing food/drink, dressing people
- Z. Erickson et al. (GA Tech): Train a DNN using forces instead of vision

Deep RL for Motion Planning

- Recall PRMs: Construct a graph by sampling and connecting points in C-space
- Determining connectivity can be expensive and/or wasteful!
- PRM-RL: Use RL to train an agent to determine point-point connectivity



Deep RL and Locomotion

- Locomotion can become a different control task for each possible robot morphology
- Traditional research specializes on particular robot classes: humanoids, quadrupeds, hexapods, snakes, fish, etc.

Bonus: Advances in Simulation

- While deep RL requires a lot of training data, another bottleneck for robotics often comes from simulation of complex physics
- NVIDIA: Use GPUs to simulate hundreds of robots to quickly learn a task
- Agents can even interact with one another

Other Deep Learning Applications

- Learning complex system dynamics in face of uncertainties, changing environments, partial state information, and high dimensionality
- Learning control policies as above

- Complex manipulation—deformable objects, tools, different geometries

- Advanced object detection, interpreting human actions and intentions
- High-level (human-level) task planning

Deep Learning Video Links

- Deep self modeling at Columbia
- Data collection at Google Brain
- Pose estimation at NVIDIA
- Dressing robot at Georgia Tech
- PRM-RL at Google Brain
- Quadruped locomotion at Google Brain
- Robotic simulation at NVIDIA

THANK YOU!

- Feel free to keep in touch if you’re around next year
- Consider applying to be a TA for 4701 (AI) ☺
- Good luck on your finals and have a great summer!