Machine Learning for Control: Experiments with Agile Quadrupeds and Perching UAVs

Russ Tedrake
Assistant Professor
MIT Computer Science and Artificial Intelligence Lab
The State-of-the-Art in Bipedal Walking Robots

Honda’s ASIMO

Russ Tedrake, MIT

Machine Learning for Control
Performance of Honda’s ASIMO Control

- Works well on flat terrain, and even up stairs
- Trajectories are constrained by an overly restrictive measure of dynamic balance

- Cannot compete with humans in terms of:
  - **Speed** (0.83 m/s top speed)
  - **Efficiency** (uses roughly 20x energy, scaled, as a human)
  - **Robustness** (no examples on uneven or unmodelled terrain)
The Challenge: Underactuated Systems
Walking is underactuated

- Consider a 7 link planar biped:
  - 6 actuators (one at each joint)
  - Want to control 7+ degrees of freedom.

- Note: “Fully”-actuated if we assume that one foot is bolted to the ground (walking robotic arms)
Honda’s ASIMO Control

- Honda’s solution: constrain the dynamics
  - Keep foot flat on the ground (fully actuated)
  - Estimate danger of foot roll by measuring ground reaction forces
  - Carefully design desired trajectories
  - Keep knees bent (avoid singularities)
  - High-gain PD control

- Same approach used by a large number of “ZMP walkers”
Passive Dynamic Walking

[Collins et al., 2001]
Robust and Efficient Bipeds

- To achieve high performance, we must *relinquish* control!
- High-gain feedback is not necessary
- Passive walker only works downhill, initialized by a “skilled hand”
- Challenge: How do you design a *minimal* control system to push and pull the natural dynamics?
A Machine Learning Approach

- Reinforcement learning

Every time the robot runs, give it a score (cost function). A "Learning" algorithm on robot associates control actions with rewards. Through trial and error, the robot can learn very advanced skills. Improved algorithms allow the robot to learn more skills in less trials. Reinforcement learning is also known as approximate optimal control.
A Machine Learning Approach

- Reinforcement learning
  - Every time the robot runs, give it a score (cost function)
A Machine Learning Approach

- Reinforcement learning
  - Every time the robot runs, give it a score (cost function)
A Machine Learning Approach

- Reinforcement learning
  - Every time the robot runs, give it a score (cost function)
  - “Learning” algorithm on robot associates control actions with rewards
A Machine Learning Approach

- Reinforcement learning
  - Every time the robot runs, give it a score (cost function)
  - “Learning” algorithm on robot associates control actions with rewards
  - Through trial and error, the robot can learn very advanced skills
Reinforcement learning

- Every time the robot runs, give it a score (cost function)
- “Learning” algorithm on robot associates control actions with rewards
- Through trial and error, the robot can learn very advanced skills
- Improved algorithms allow the robot to learn more skills in less trials
A Machine Learning Approach

- Reinforcement learning
  - Every time the robot runs, give it a score (cost function)
  - “Learning” algorithm on robot associates control actions with rewards
  - Through trial and error, the robot can learn very advanced skills
  - Improved algorithms allow the robot to learn more skills in less trials

- Reinforcement learning is also known as \textit{approximate} optimal control
Reinforcement learning example
Learning To Walk in 20 minutes

Science, 2005
Learning Results

- Learning only on the robot (no simulation)
Learning Results

- Learning only on the robot (no simulation)
- Algorithm comes with theoretical convergence guarantees
  - Stochastic gradient descent on the average reward, despite noise and disturbances
  - Converges to local minima

Very fast and robust convergence
After initial learning, adapts to changes in a few steps
Always learning, always adapting to the terrain as it walks
Learning Results

- Learning only on the robot (no simulation)
- Algorithm comes with theoretical convergence guarantees
  - Stochastic gradient descent on the average reward, despite noise and disturbances
  - Converges to local minima
- Very fast and robust convergence
  - Learns from a “blank slate” in < 20 minutes
  - After initial learning, adapts to changes in a few steps
Learning Results

- Learning only on the robot (no simulation)
- Algorithm comes with theoretical convergence guarantees
  - Stochastic gradient descent on the average reward, despite noise and disturbances
  - Converges to local minima
- Very fast and robust convergence
  - Learns from a “blank slate” in < 20 minutes
  - After initial learning, adapts to changes in a few steps
- Always learning, always adapting to the terrain as it walks
How far can we take this learning idea?
Simple Bipeds on Rough Terrain

Russ Tedrake, MIT

Machine Learning for Control
Quadrupeds on rough terrain (motion planning)
High-dimensional underactuated motion planning

- Formulate underactuated control as a search problem (aka. kino-dynamic motion planning)
  - Current real-time methods limited to relatively low-dimensional problems (simple robots)
  - Underactuated systems are notoriously difficult (tunnels and tubes)
- Big idea: Exploit structure in the eqs of motion
  \[
  H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B\tau.
  \]

- Example: Dimensionality reduction for motion planning
  - Make a high-dimensional underactuated system act like a low-dimensional fully-actuated system
  - Method either produces the correct torques, or says “can’t do that”
  - Perform a low-dimensional search
Little Dog highlights

Russ Tedrake, MIT

Machine Learning for Control
Flapping-Winged Flight
Flapping-Winged Flight
Autonomous Flapping-Winged Flight

Russ Tedrake, MIT

Machine Learning for Control
Autonomous Flapping-Winged Flight
Control in unsteady fluid dynamics
Two Goals for Flapping Flight

- Probably won’t beat a propeller for efficient forward flight
- Two goals for outperforming fixed-wing aircraft:
  1. Aggressive aerial maneuvers (e.g. landing on a perch)
  2. “Harvesting” energy from the air
The Heaving Foil (work with Jun Zhang, NYU)

- Rigid, symmetric wing
- Driven vertically
- Free to rotate horizontally

[Vandenberghe et al., 2006]
Symmetry breaking leads to forward flight
Flow visualization
Prospects for optimization

- Previous work only consider sinusoidal trajectories
- Control problem: optimize stroke form to maximize efficiency
- CFD model [Alben and Shelley, 2005]
Prospects for optimization

- Previous work only consider sinusoidal trajectories
- Control problem: optimize stroke form to maximize efficiency
- CFD model [Alben and Shelley, 2005]

- Takes approximately 36 hours to simulate 30 flaps
Prospects for optimization

- Previous work only consider sinusoidal trajectories
- Control problem: optimize stroke form to maximize efficiency
- CFD model [Alben and Shelley, 2005]

Takes approximately 36 hours to simulate 30 flaps
Can we perform the optimization directly in the fluid?
Learning results

- Parameterized spline trajectory
- Policy gradient learning
- Learning improves efficiency 3x in just 15 minutes
- Dynamic explanation of optima

Implications:
- Optimization in experimental fluid dynamics
- Control for the birds
An airplane that can land on a perch

- Conventional aircraft use high-gain feedback and avoid complicated nonlinear dynamics regimes (just like ASIMO).
- Higher performance in maneuverability and efficiency if you exploit the nonlinear, unsteady fluid dynamics (just like walking).
- A benchmark problem: landing on a perch.
Experiment Design

- Glider (no propellor)
- Dihedral (passive roll stability)
- Offboard sensing and control
System Identification

- Real flight data
  - Very high angle-of-attack regimes
  - Surprisingly good match to theory
  - Vortex shedding

![Lift Coefficient](image1)

![Drag Coefficient](image2)

Russ Tedrake, MIT

Machine Learning for Control
Glider Perching

- Enters motion capture @ 6 m/s.
- Perch is < 3.5 m away.
- Entire trajectory @ 1 second.

Requires Separation!
Conclusions

- New tools from machine learning can help solve underactuated control problems, such as the control of locomotion, but we must take advantage of:
  - Mechanical design and plant dynamics
  - Classical control

- Approximate optimal control solutions can exploit the dynamics of walking machines, to produce efficient and robust gaits.

- Initial evidence suggests that designing control policies for fluid systems (at intermediate Reynolds numbers) might be easier than completely describing their dynamics (birds don’t solve Navier–Stokes)


*Physics of Fluids, 18.*