Model-based robotics, without the simplifications

Emo Todorov

Applied Mathematics, Computer Science & Engineering
University of Washington

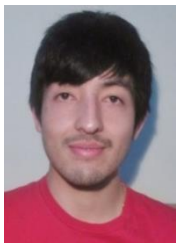
Contributions from:


Igor Mordatch


Vikash Kumar


Yuval Tassa


Tom Erez


Kendall Lowrey


Galen Andrew


Paul Kulchenko


Svet Kolev

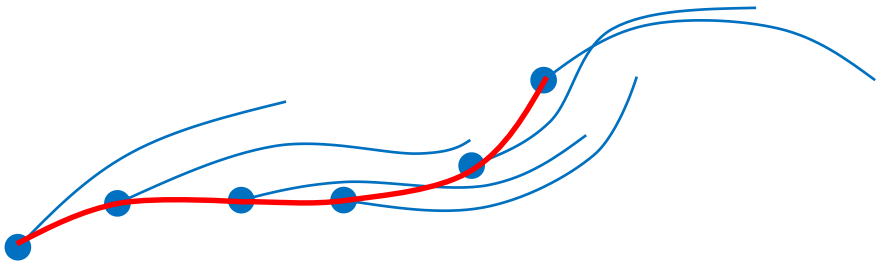

Joseph Xu

Funding: NSF, NIH, DARPA

Model-predictive control (MPC)

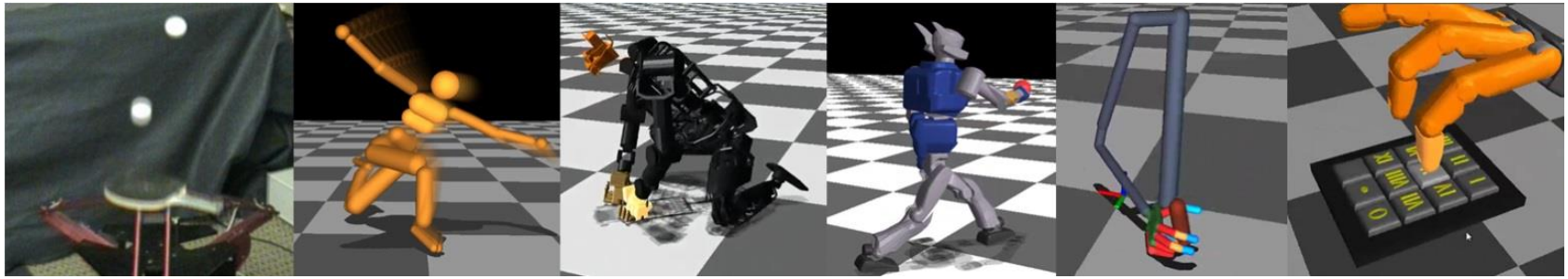
At each time step t, do trajectory optimization with warm-start from the previous time step:

$$\min\ h(x_{t+N}) + \sum_{k=t}^{t+N-1} \ell(x_k, u_k)$$



We have been able to apply MPC to the full robot dynamics thanks to:

- improved models of contact dynamics;
- efficient physics simulator (MuJoCo);
- efficient optimization algorithm (iLQG);
- selection of cost functions that are realistic yet easier to optimize.



Tassa, Erez and Todorov, *IROS* 2012
Erez et al, *Humanoids* 2013
Tassa et al; Kumar et al; Erez et al; *ICRA* 2014

Cost function automation

$$\text{cost}(t) = \sum_i \text{coef}_i * \text{norm}_i\left(\text{feature}_i\left(\text{state}(t), \text{control}(t)\right) - \text{reference}_i(t)\right)$$

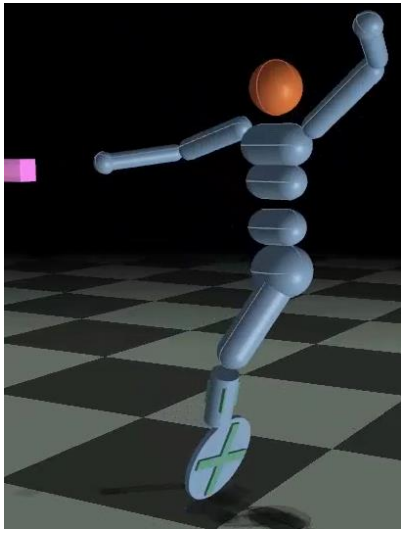
XML file format for specifying cost functions:

```
<feature name='torques'>
    <data field='ctrl' item='all'/>
</feature>

<feature name='height'>
    <op type='sum' ref='.7'>
        <data field='xipos' item='pelvis' entry='z' coef='1'/>
        <data field='xipos' item='wheel' entry='z' coef='-1'/>
    </op>
</feature>

<feature name='upright'>
    <data field='xmat' item='torso' entry='zz zx zy' ref='1 0 0'/>
</feature>

<cost>
    <term feature='torques'  norm='quadratic'   coef='.1'   />
    <term feature='balance'  norm='L2'          coef='.03'  />
    <term feature='height'   norm='smooth_abs2' coef='1'    />
    <term feature='upright'  norm='quadratic'   coef='1'    />
    <term feature='reach'    norm='L2'          coef='.07'  />
    <term feature='com vel'  norm='power'       coef='.0002'/>
    <term feature='ang mom'  norm='quadratic'   coef='.0001'/>
</cost>
```



torques	0.1
balance	0.03
height	1
upright	1
reach	0.07
com vel	0.0002
ang mom	0.0001

Tassa, Erez and Todorov (2014)

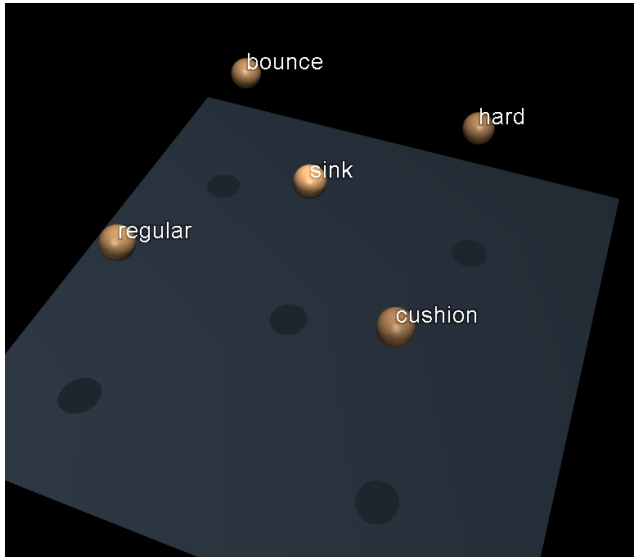
Contact dynamics with adjustable softness

We either have to start with a soft model and use continuation to make it harder, or in the case of MPC, optimize through a soft model all the time and apply the controls to the nominal hard model.

This requires a contact model that has adjustable softness, is sufficiently smooth to be used within an optimization loop, and can be evaluated quickly.

We have developed such a contact model and implemented it in MuJoCo.
Forward dynamics become a convex optimization problem.
Inverse dynamics are uniquely-defined and can be computed analytically.

NP-hard complementarity constraints are avoided.
Continuous-time formulation, amenable to RK4 etc.
Rich parameterization that facilitates system id.

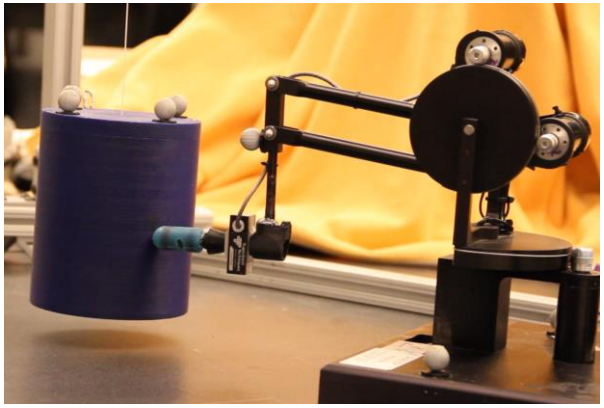


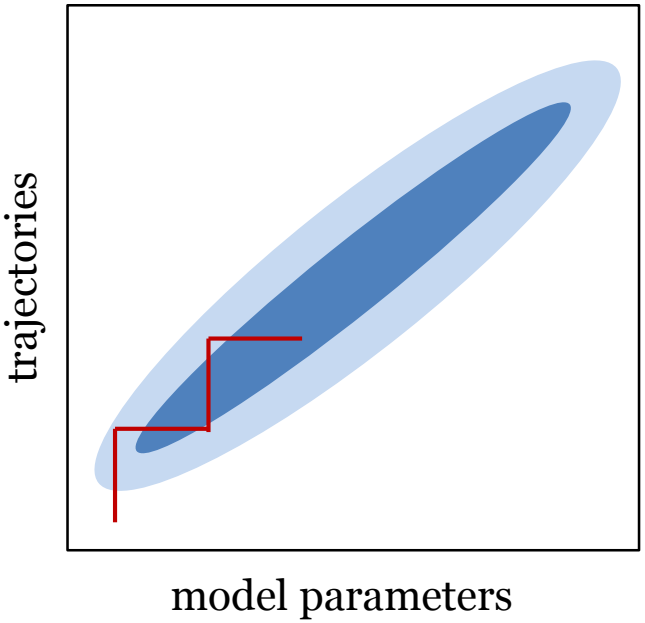
Todorov, *ICRA* 2010, 2011, 2014
Todorov, 2015, in preparation

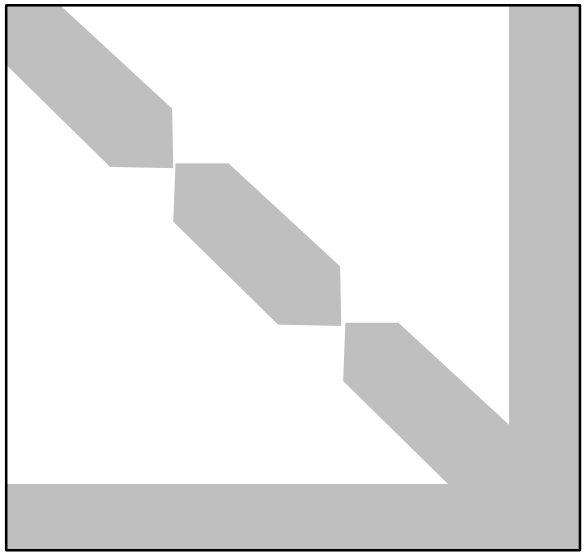
System id and state estimation with contacts

Given noisy sensor data about
- trajectory kinematics
- contact forces

Estimate (using the simulator as a physics prior)
- trajectory kinematics
- contact forces
- model parameters

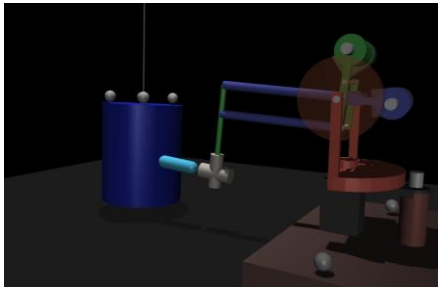




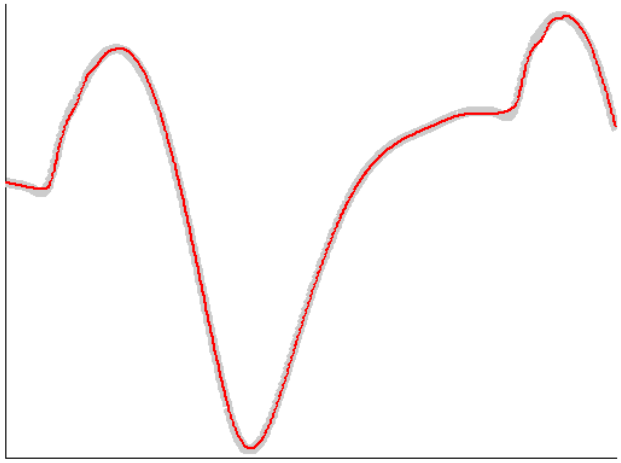


sparse Hessian,
sparse factorization

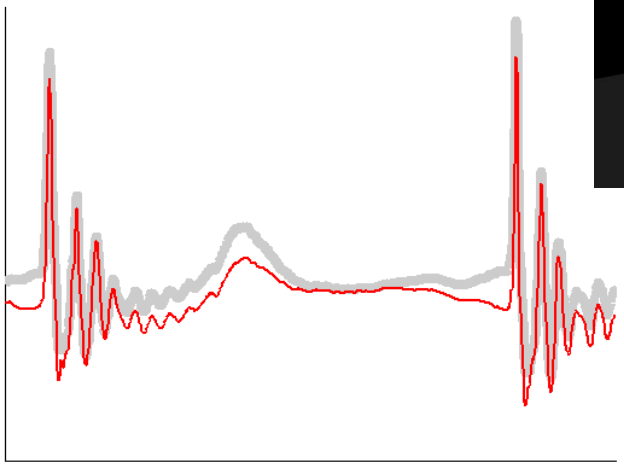
Predicted vs. measured sensor data



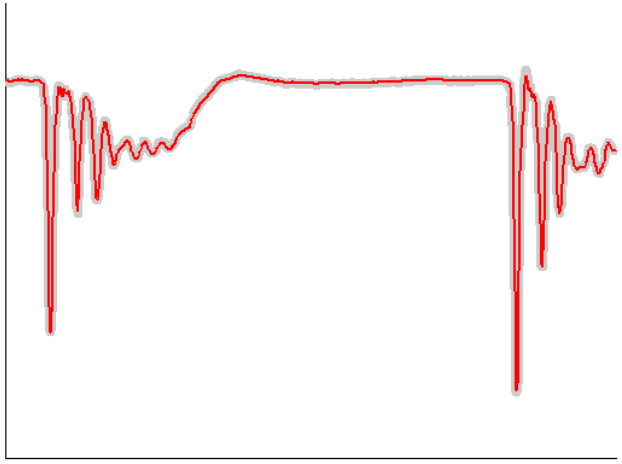
joint angle



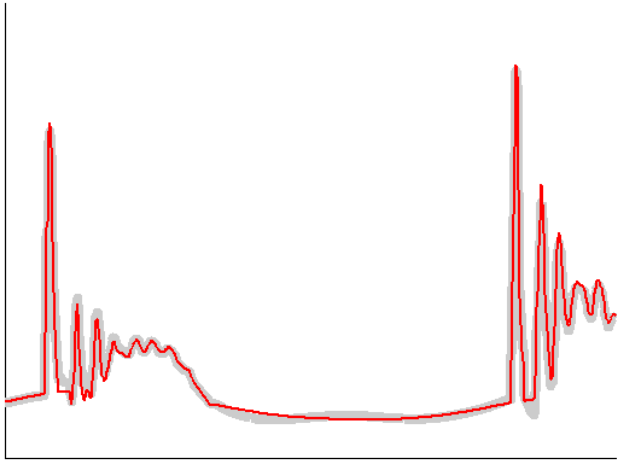
joint acceleration



contact force



object acceleration

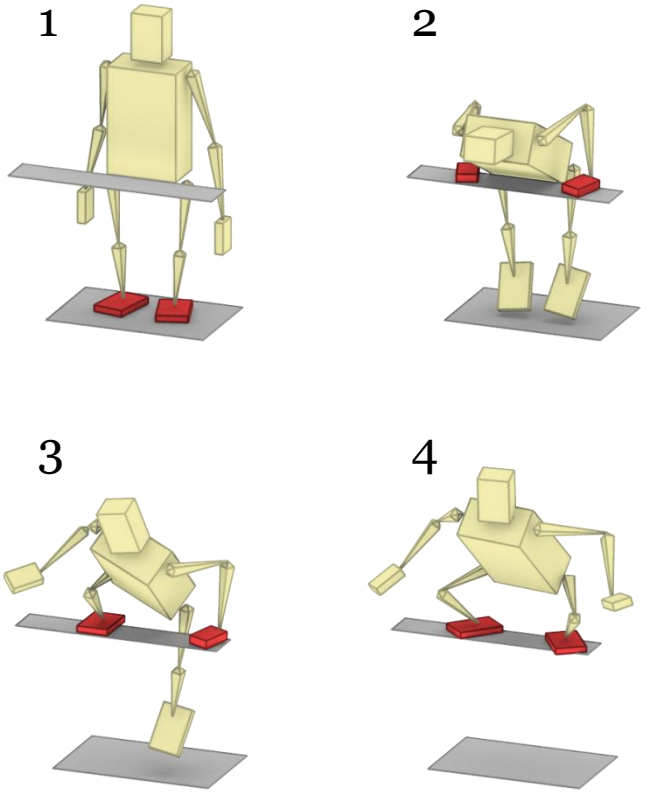


1 sec

Contact-invariant optimization (CIO)

Contacts make the dynamics discontinuous,
and may seem to require combinatorial search.

CIO is a domain-specific relaxation method
designed to avoid such combinatorial search.



Configuration: q
Contact force: f
Applied force: $u = M(q)\ddot{q} + c(q,\dot{q}) - J(q)^T f$

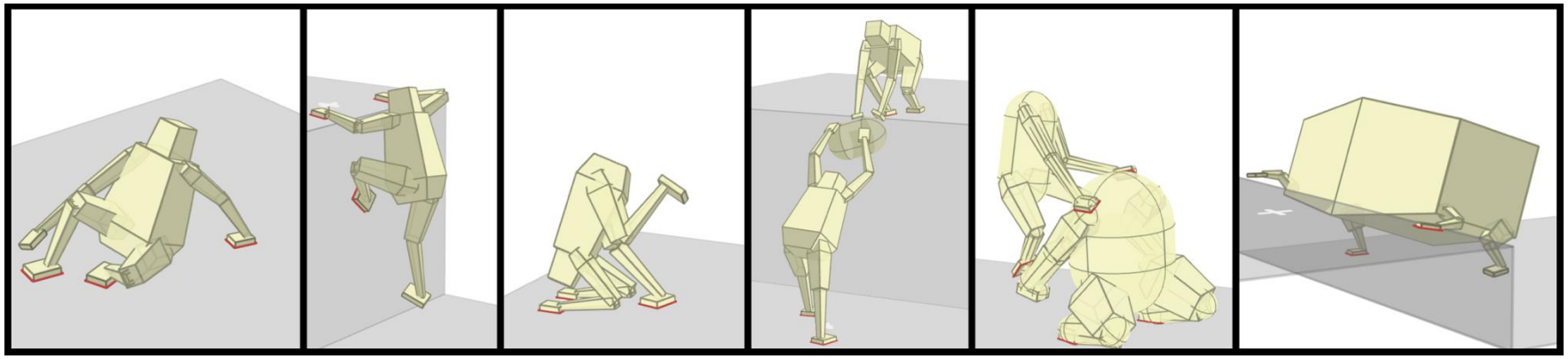
Min over trajectories $q(t)$, $f(t)$:

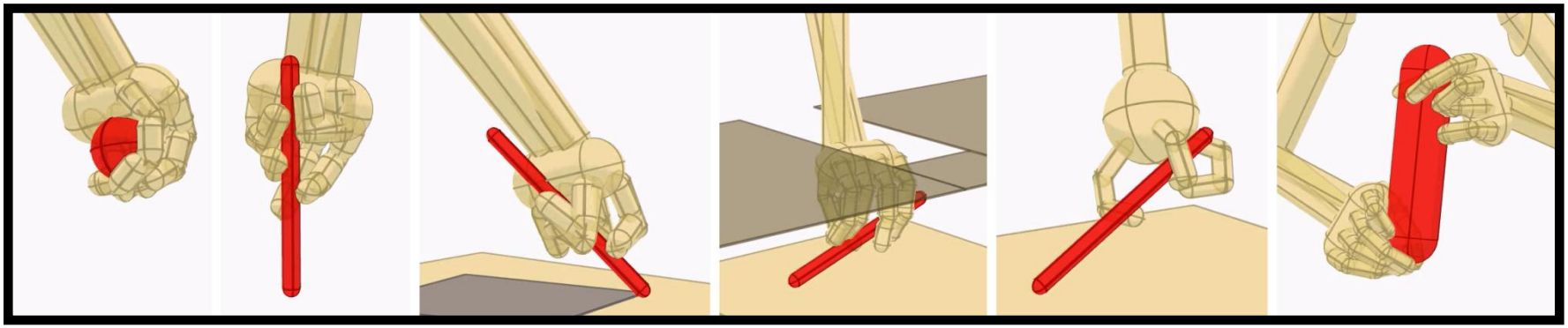
$$\text{TaskCost}(q, f) +$$
$$\text{ControlCost}(u(q, f)) +$$
$$\text{ContactViolation}(q) +$$
$$\text{FrictionConeViolation}(f) +$$
$$f_{\text{normal}} * \text{ContactDistance}(q) * \textit{continuation}$$

Unconstrained problem:
BFGS, CG, Gauss-Newton
~ 5 min, ~1000 iterations

Mordatch, Todorov and Popovic, *SIGGRAPH* 2012
Mordatch, Popovic and Todorov, *SCA* 2012

Results

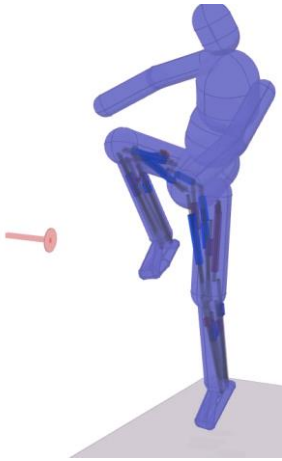




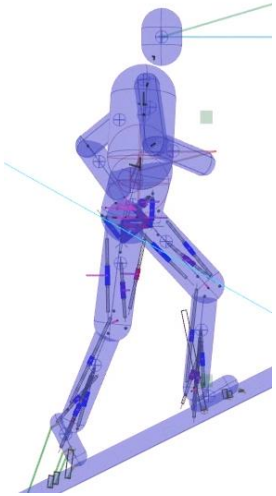
Extension to musculo-skeletal dynamics

jumping

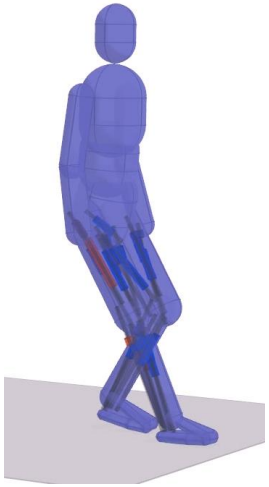
kicking



slope



stopping



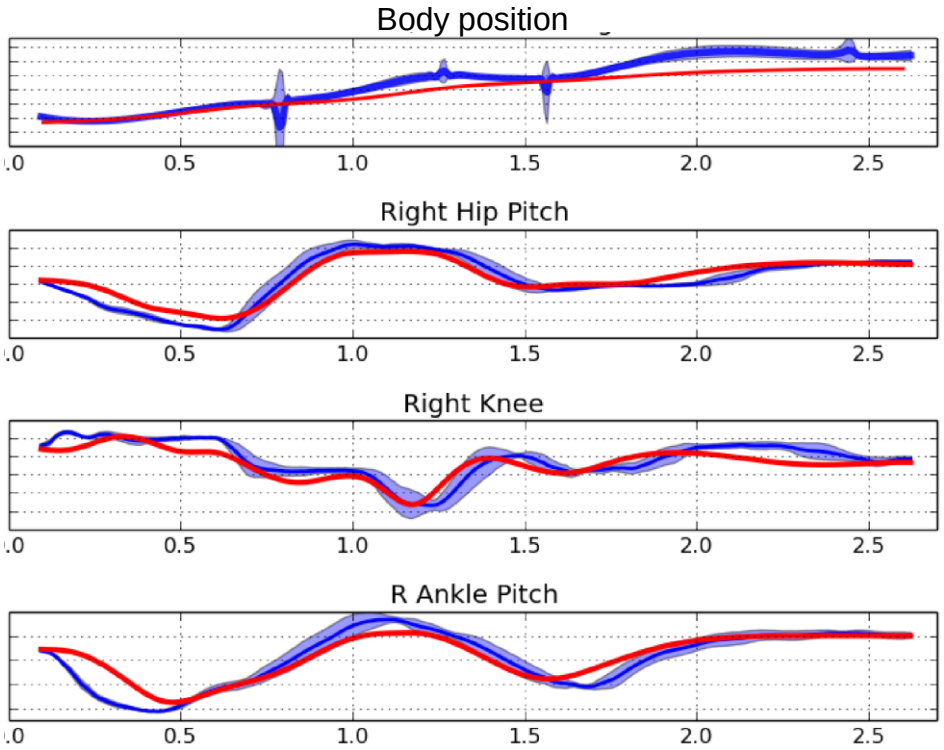
moon gravity

Mordatch, Wang, Todorov and Koltun, *SIGGRAPH ASIA* 2013

Ensemble-CIO

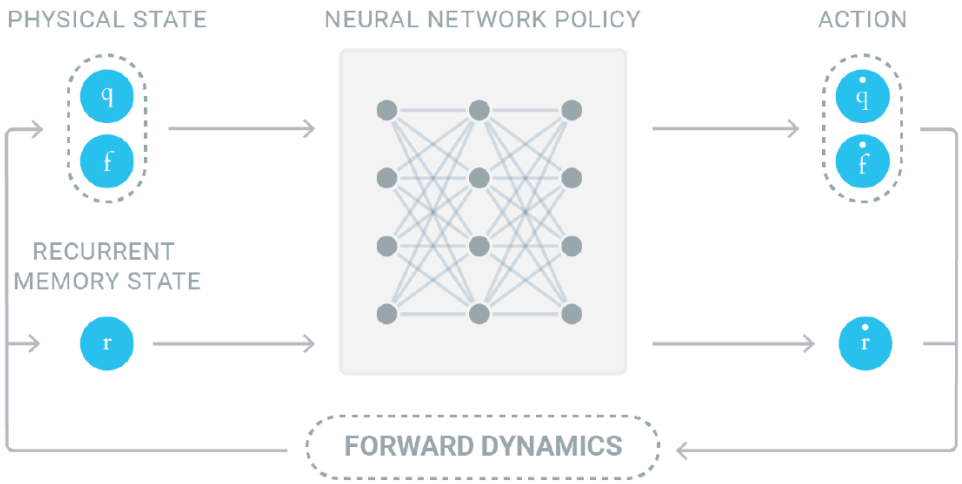
To improve robustness, we generate ~10 perturbed models around the nominal one, and optimize a single trajectory that must be feasible for all perturbed models.

We then execute this trajectory on the robot, using the locally-optimal feedback gains obtained from the trajectory optimizer.



Mordatch, Lowrey and Todorov (2015)

Training neural networks with trajectory optimization



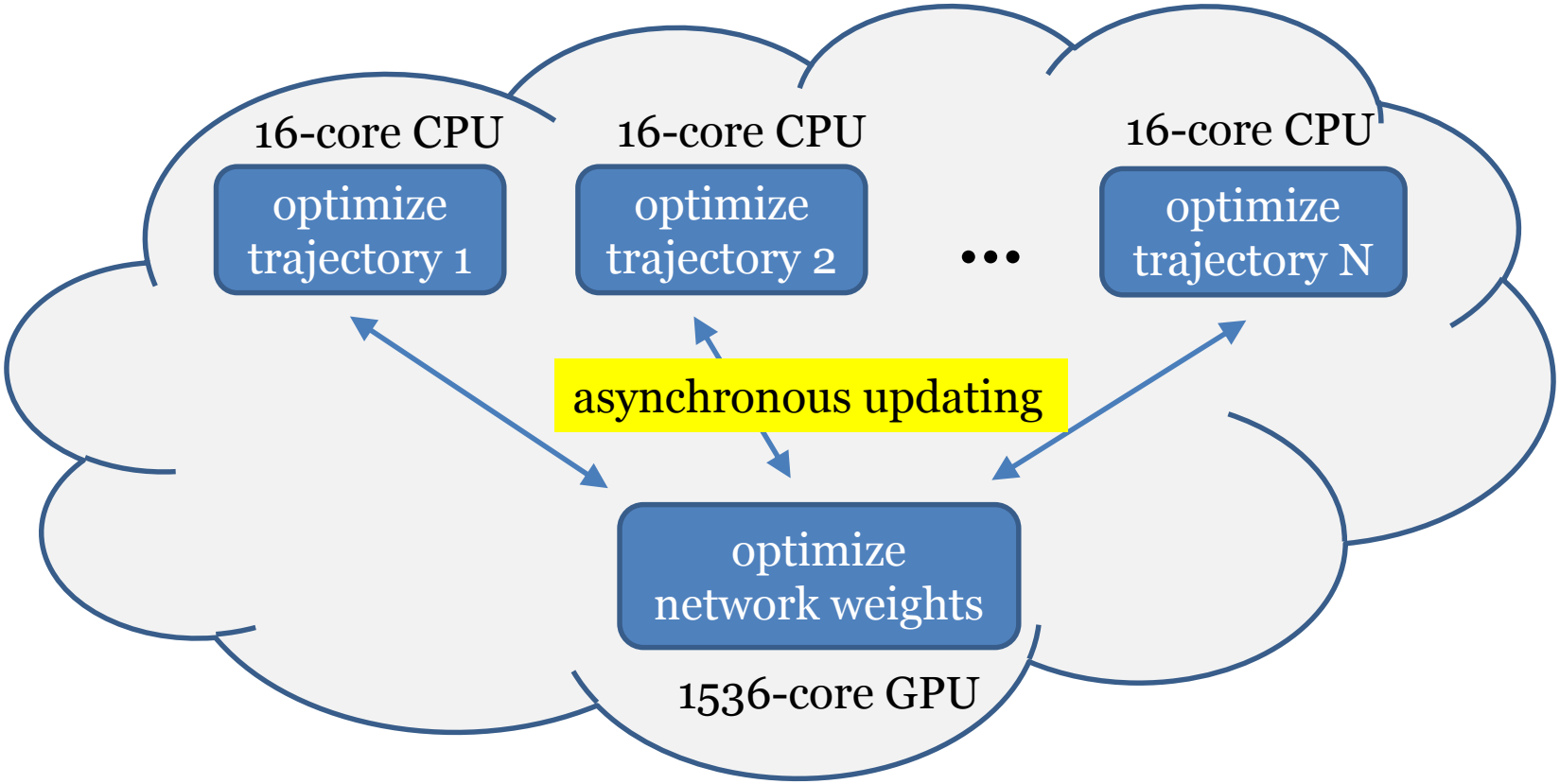
Trajectory optimization:
optimize the sequence of **body and brain** states, under a mixed cost that includes movement costs as well as differences from the network output (so as to keep the trajectories and the network close throughout training)

Network training:
supervised learning from the optimized trajectories

The "action" is the change in state. Thus the network is not learning a control law, but rather the dynamics of the system under an implicit control law.

Mordatch, Andrew, Lowrey, Popovic and Todorov (2015)

Training in the cloud

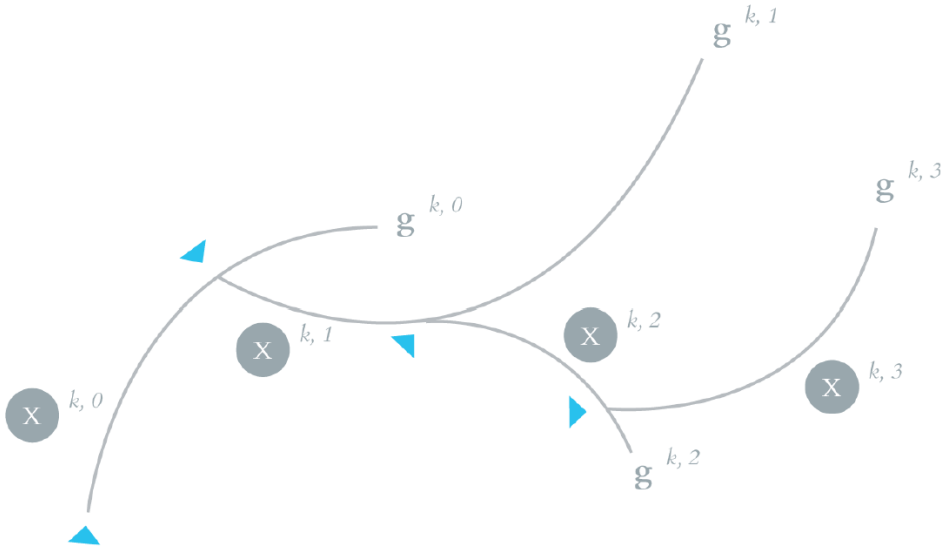


training a network with 5 layers of 250 units (250,000 weights) takes 2.5 hours
this includes 200 steps of (re) optimization for 1,000 trajectory pieces

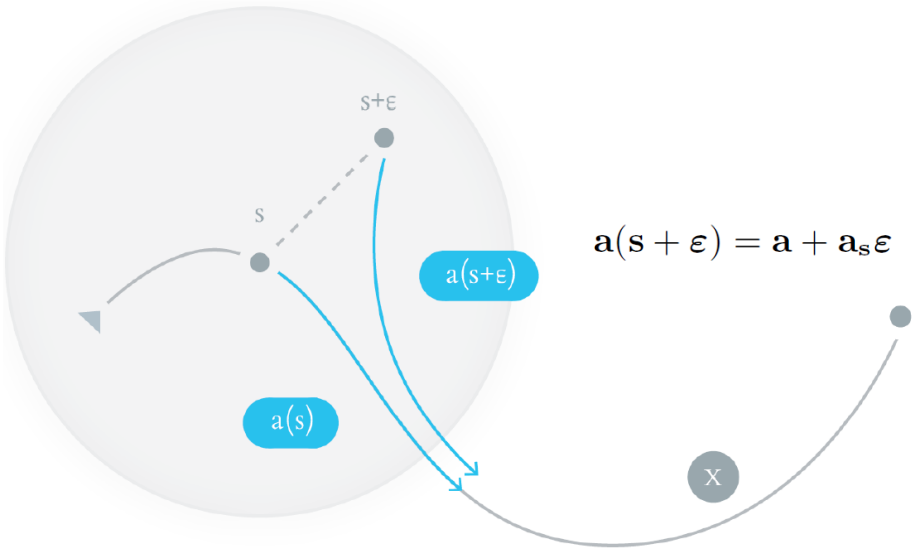
cloud computing bill from Amazon for this project: $6,000

Making the training set diverse

The movement cost (i.e. the spatial goal) change often, so as to generate a tree of trajectories that span a large portion of the relevant behavioral space.



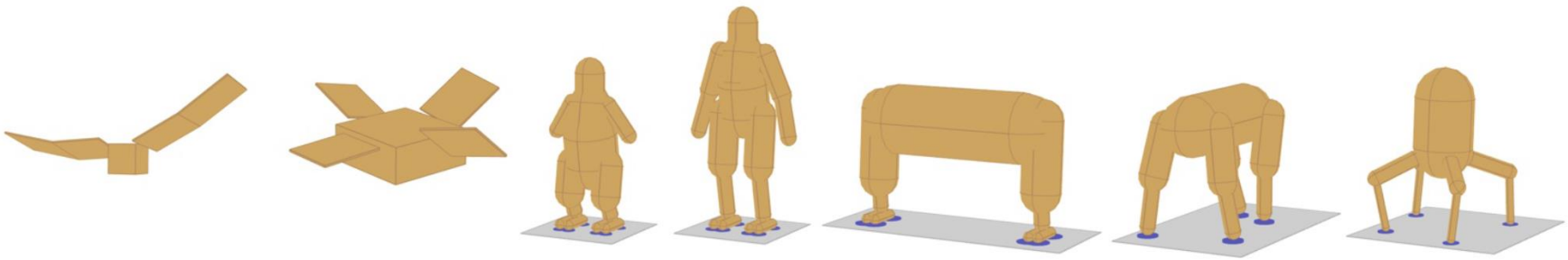
Noise is injected during training, forcing the network to learn not only the nominal response but also the corrections around it. This resembles an old method called Tangent Propagation (Simard et al 98).



Results

The same training method works for a variety of motor tasks and body morphologies:

- flying
- swimming
- bipedal walking
- quadrupedal walking



MuJoCo: www.mujoco.org

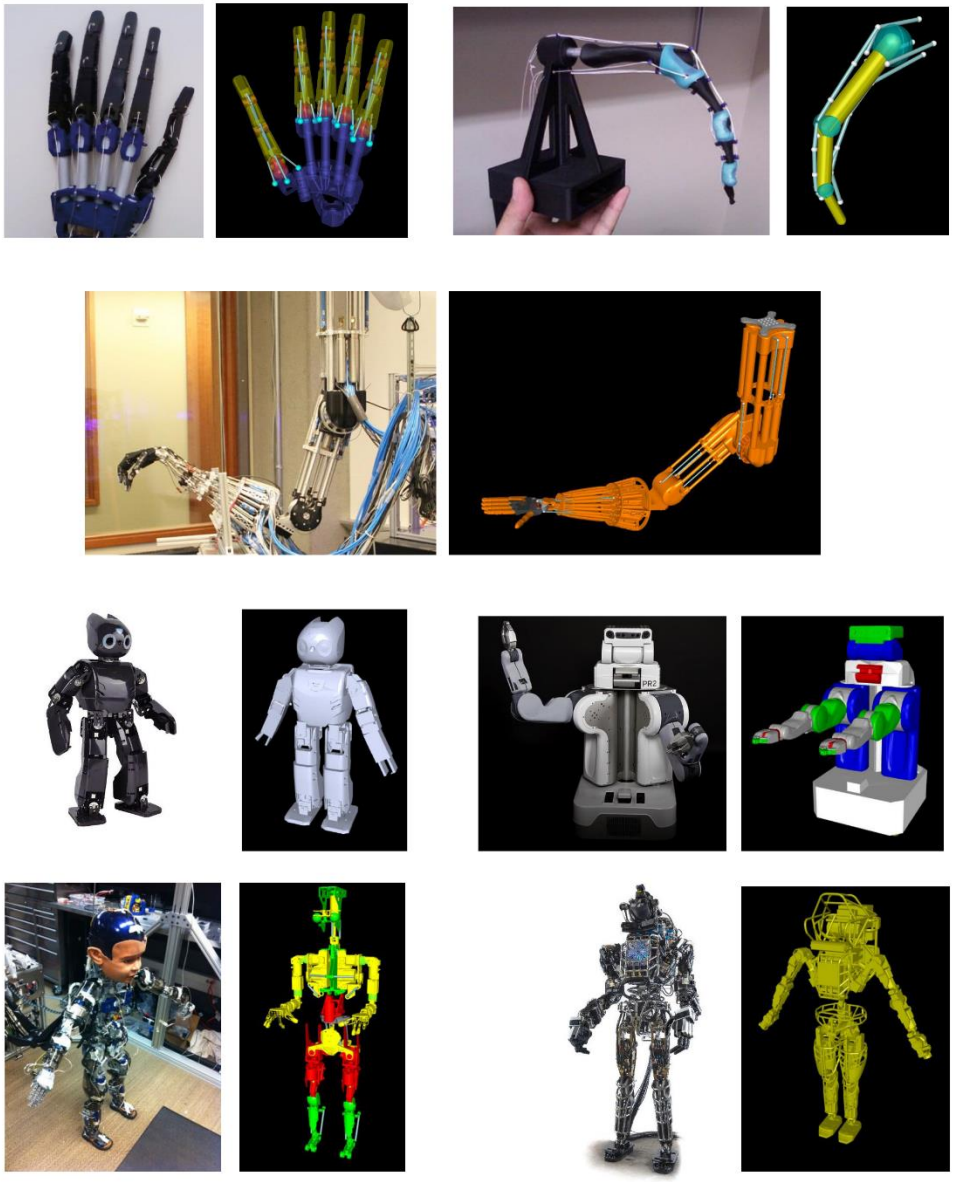
MuJoCo Pro
SDK for researchers;
invitation-only preview now;
commercial product later.

MuJoCo Sim
alternative to Gazebo;
coming soon.

MuJoCo HAPTIX
Sim + motion capture + VR;
developed for DARPA;
available now.

MuJoCo is just the simulator.

The next step is to implement a universal optimizer on top of it.



Some robots are hard to model…

Xu and Todorov (2015)