Online optimization

Emo Todorov

Applied Mathematics
Computer Science and Engineering

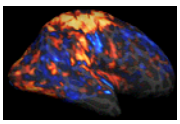
University of Washington

The case for online optimization

complex “hardwired” behaviors can be generated with few neurons

we often think of learning as setting up a similar neural machine,
which is responsible for online generation of behavior

yet most neurons in your head are doing something when you move



online optimization / model-predictive control:

- works great for smooth slow dynamics (e.g. chemical process control)

- scaling it to more complex dynamics requires fast simulation (**internal models**)
  combined with efficient **optimization/learning**

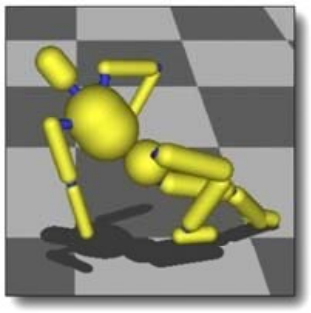
MuJoCo: A physics engine for control

fast recursive algorithms for smooth dynamics

new algorithms for **contact dynamics**

parallel evaluation of trajectory costs, gradients and Hessians (soon to run on GPUs)

12,000 lines of C/C++ code, still growing



Timing tests on 18-dof 3D walker with 6 contacts:

- 400,000 evaluations per sec on 12-core PC; 1.3 M without contacts

- 100 quasi-Newton trajectory optimization steps per second

- 4,000 times faster than real time at 100 fps

Online optimization

At time step t, solve a trajectory optimization problem of the form

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

Larger horizon N results in better performance but requires more computation

The final cost $h(x)$ should approximate all future costs incurred after time $t+N$, i.e. the optimal cost-to-go function (or value function)

The running cost $\ell(x,u)$ usually penalizes task errors and control energy

At each time step:

- optimize the trajectory (initializing from the previous time step)
- apply the first control u_t
- observe/estimate the next state x_{t+1}

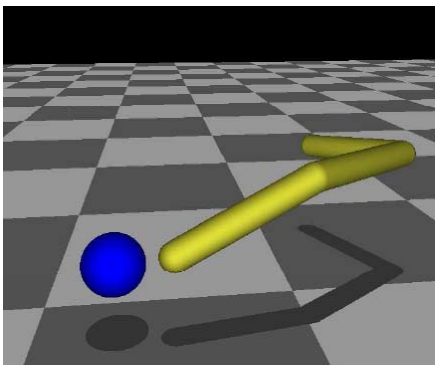
There is always a plan, but the plan changes all the time, and only the initial portion is ever executed.

Application to swimming

(Tassa, Erez and Todorov)

method 1: optimize only the running cost, ignore the final cost / cost-to-go

this works well when a lot of progress can be made within the planning horizon

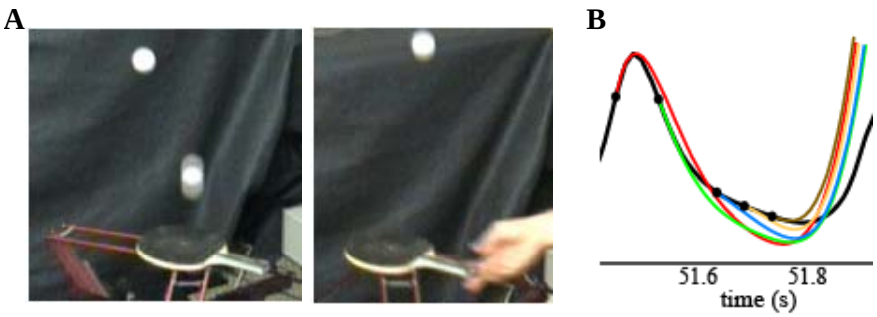


Application to ball bouncing

(Kulchenko and Todorov)

method 2: use some intuitive approximation to the optimal cost-to-go

similar to evaluation functions in chess; rough approximations are often sufficient



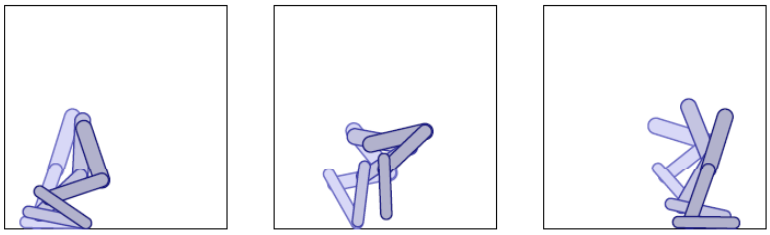
343,000 views on YouTube in 4 months

Application to hopping

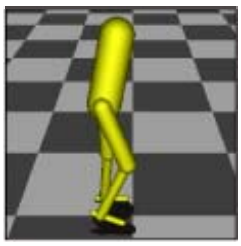
(Erez, Tassa and Todorov)

method 3: use offline optimization to model the optimal cost-to-go (at least locally)

this approach is the most powerful, and most likely to be a good model of the brain



next project: 3D walking
(offline optimization for now)



If the brain did this, what would we expect to see?

behavioral level:

intelligent responses in unusual circumstances (e.g. large unexpected perturbations)

goal achievement in the presence of substantial variability

rapid emergence of new behaviors (e.g. limping)

neural level:

a lot more online computation than what seems necessary to run an existing controller

representation of movement features at multiple times into the future

extensive use of internal models (the cerebellum has a huge number of neurons)

Experimental challenge

we know how to design and interpret experiments that involve many repetitions of the same movement

however there is limited role for online optimization in that context

instead we need experiments where subjects are required to come up with new movements all the time

how can we get experimenters to do such experiments?

show cool movies of robots doing cool things,
and hopefully get the experimenters excited ☺

Advanced robots we are starting to use