Model-Based Optimization for Intelligent Robot Control

Tom Erez

Motion Control Lab
Computer Science
University of Washington

Goal:
optimal control of humanoid robots

- Quick, autonomous adaptation to novel circumstances.
- Unified framework for different tasks (e.g. grasping, walking).
- User guidance should be easy and intuitive.

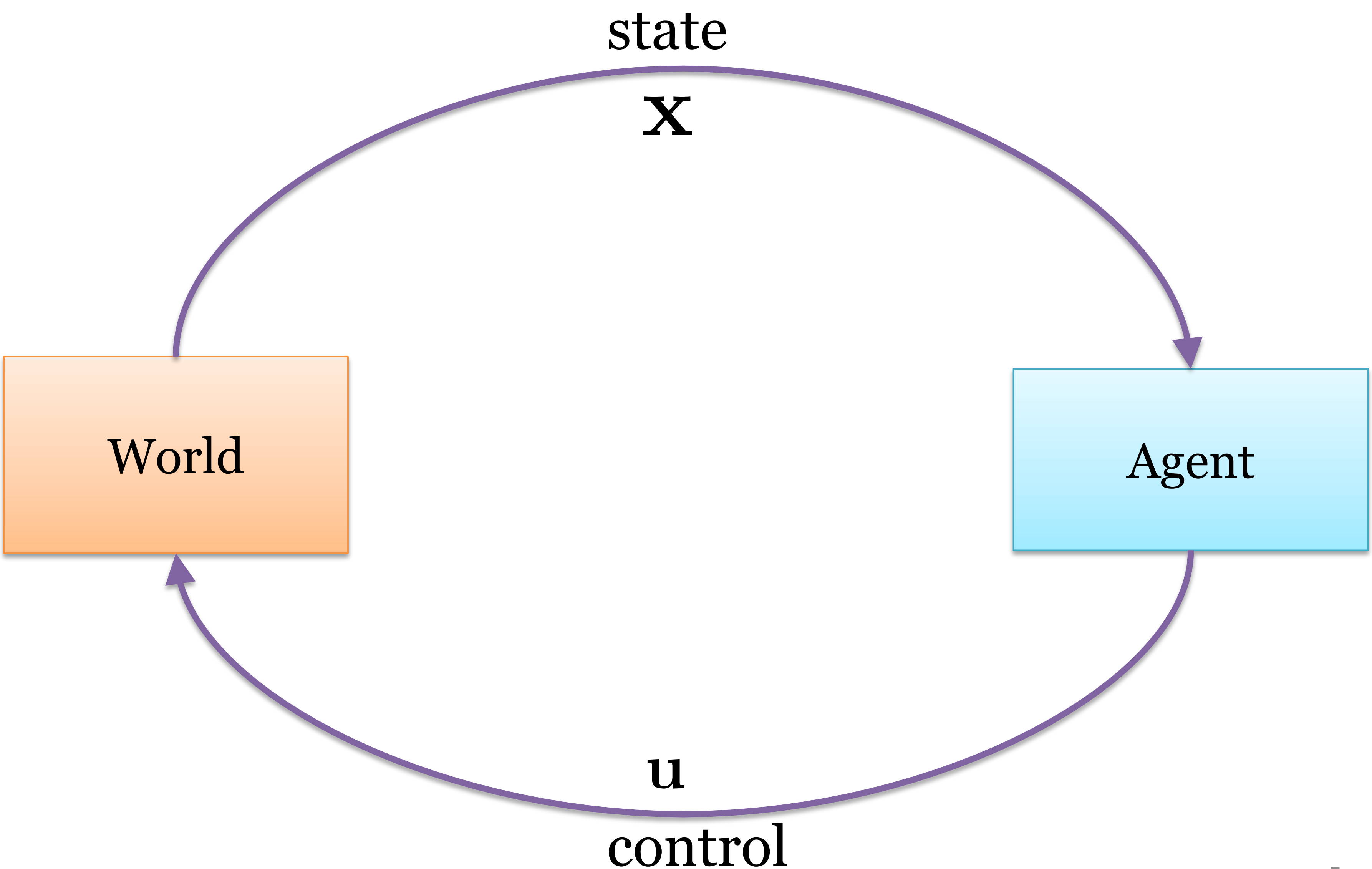
Challenges

- **Simulation**: torques, friction, contact.
- **Optimization**: nonlinear, nonconvex, high-DoF.
- **Systems**: estimation, integration, feedback control.
- **Human operator**: flexibility, visualization, interfaces.

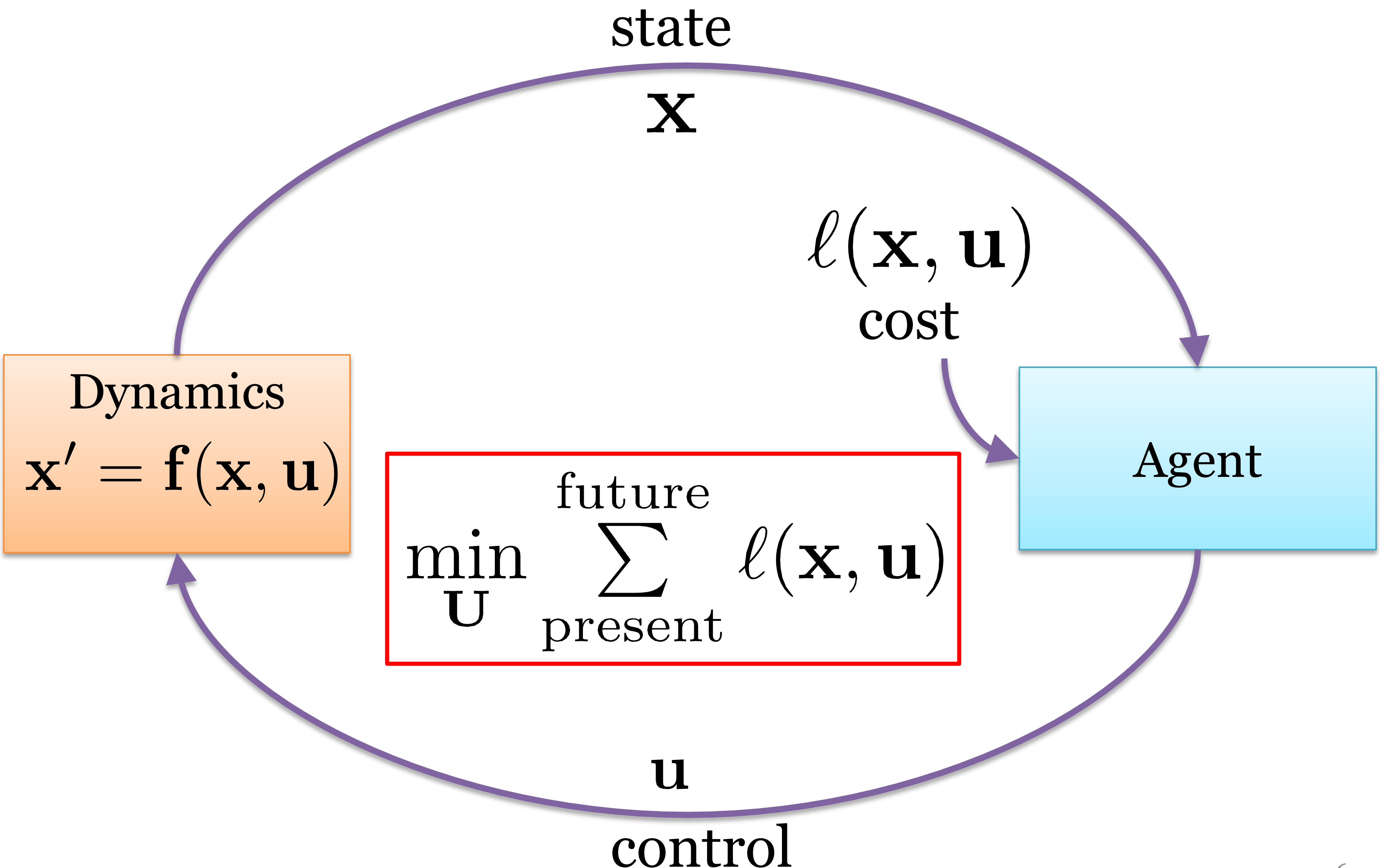
MPC in MuJoCo

- A new physics engine: MuJoCo (Multi-Joint simulation with Contacts).
- An optimization framework around MuJoCo that can operate in real-time.
- An integrated system for controlling articulated robots.

Optimal Control



Optimal Control



Model-Based Optimization

Physics

$\mathbf{f}(\mathbf{x}, \mathbf{u})$

Task

$\ell(\mathbf{x}, \mathbf{u})$

Optimization

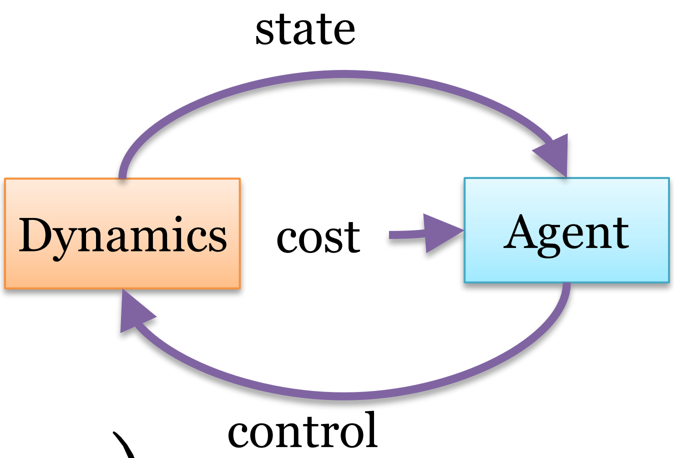
$$\min_{\mathbf{U}} \sum_{\text{present}}^{\text{future}} \ell(\mathbf{x}, \mathbf{u})$$

Behavior

Trajectory Optimization

Trajectory Optimization

State $\mathbf{x}$, control $\mathbf{u}$



Dynamics: $\mathbf{x}' = \mathbf{f}(\mathbf{x}, \mathbf{u})$

Cost: $\ell(\mathbf{x}, \mathbf{u}), \Omega(\mathbf{x}_{\mathrm{final}})$

Trajectory Optimization

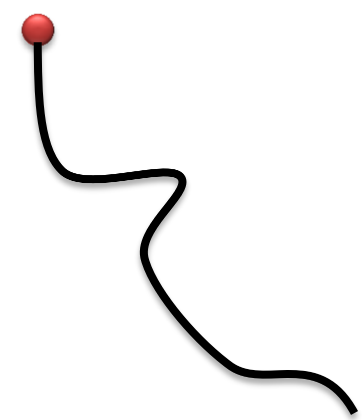
State $\mathbf{x}$, control $\mathbf{u}$

Dynamics: $\mathbf{x}' = \mathbf{f}(\mathbf{x}, \mathbf{u})$

Cost: $\ell(\mathbf{x}, \mathbf{u}), \Omega(\mathbf{x}_{\text{final}})$

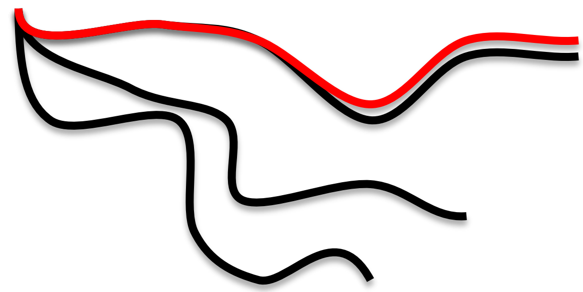
Initial state: $\mathbf{x}_0$

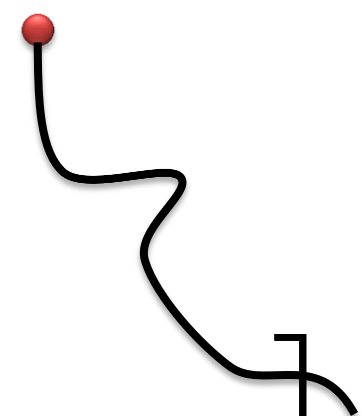
Control sequence: $\mathbf{u}_1, \mathbf{u}_2 \ldots \mathbf{u}_N$



(Single shooting / indirect optimization)

Trajectory Optimization



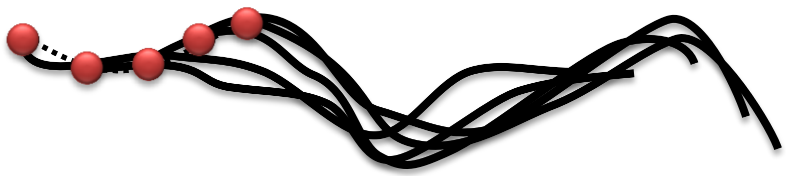


$$\min_{\mathbf{u}_1 \ldots \mathbf{u}_N} \left[\sum_{k=1}^{N} \ell(\mathbf{x}_{k-1}, \mathbf{u}_k) + \Omega(\mathbf{x}_N) \right]$$

One Plan is not Enough

- Modeling errors
- Estimation errors
- Dynamic environment

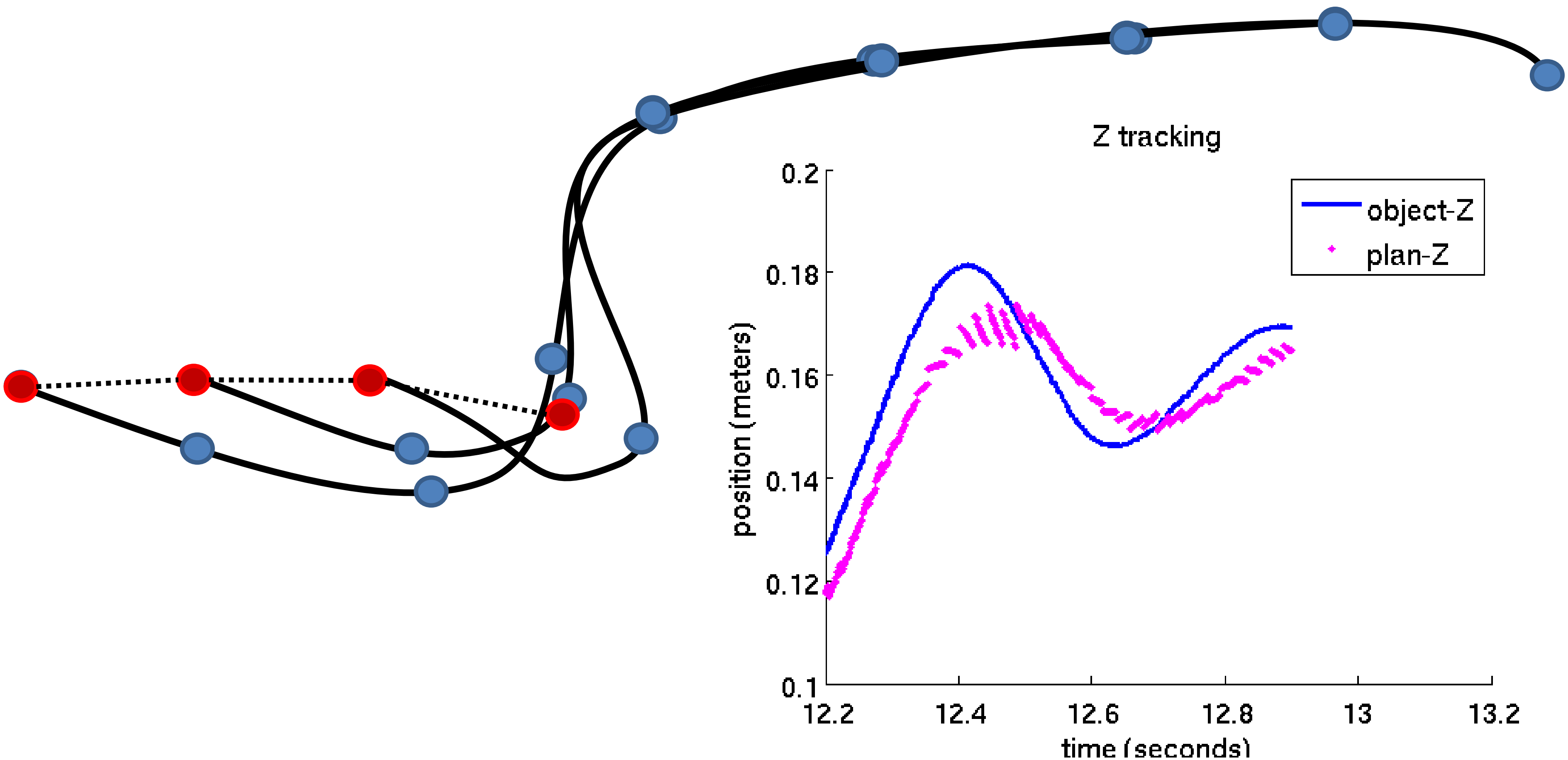
Model-Predictive Control



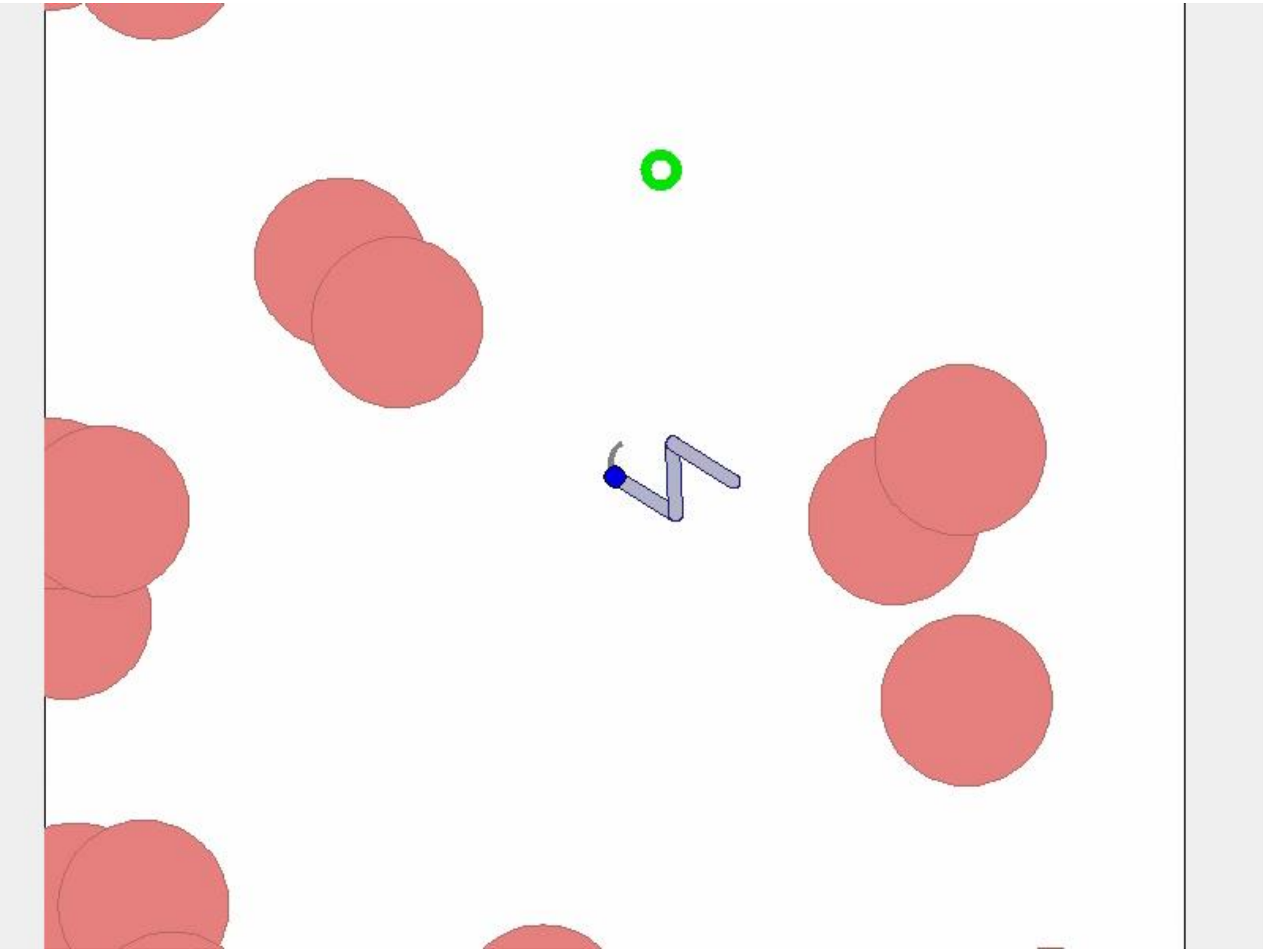
- There is always a plan.
- The plan is constantly updated.
- Only the initial portion is ever executed.

Tassa, Erez and Smart, NIPS 2008
Tassa, Erez and Todorov, IROS 2012

Modeling Errors

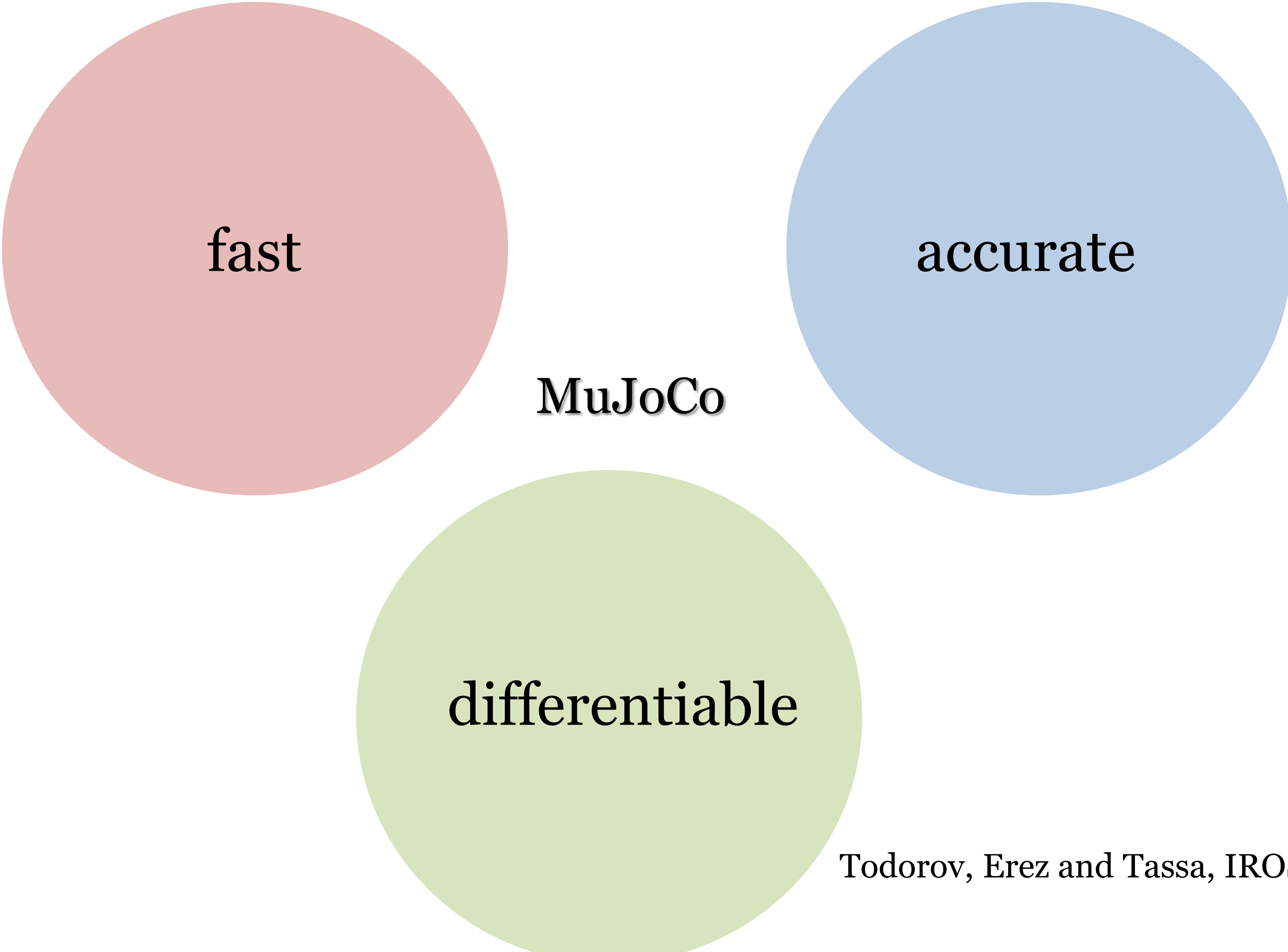


The Swimmer



Tassa, Erez and Smart, NIPS 2008

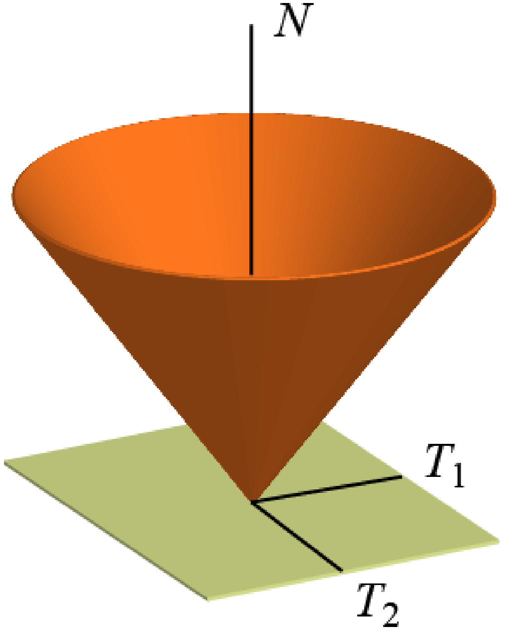
Physics Simulation



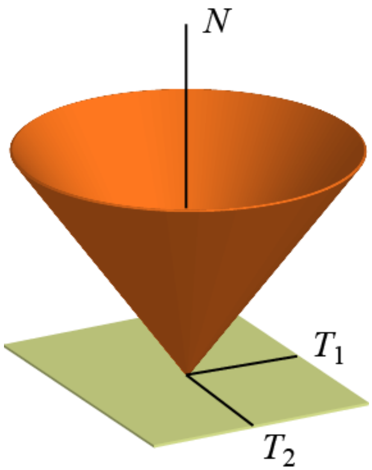
Todorov, Erez and Tassa, IROS 2012

Contact Modeling

- Non-convex
- Non-smooth
- NP-hard



Contact dynamics



Coulomb friction (NP hard)

$$v = Af + b$$
$$v_N \perp f_N$$
$$\|v_T\| \perp \mu f_N - \|f_T\|$$
$$v_T = \alpha f_T, \ \alpha \leq 0$$

$x \perp y$	complementarity: $x \geq 0, \ y \geq 0, \ xy = 0$
A	contact inverse inertia: $A = JM^{-1}J^T$
b	contact velocity before impulse
$\boldsymbol{f}$	**contact impulse**
$\boldsymbol{v}$	**contact velocity after impulse**

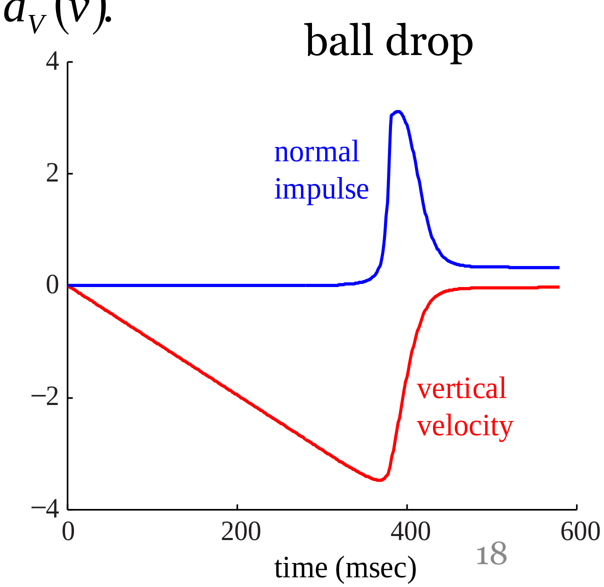
Define the contact impulse by minimizing contact-space kinetic energy $\frac{1}{2}v^T A^{-1} v$,
subject to $v_N \geq 0, \quad f_N \geq 0, \quad \mu f_N \geq \|f_T\|$ for each contact.
Replace the penetration constraint with a penalty function $d_V(v)$.

Forward contact dynamics: $(A, b) \rightarrow (f, v = Af + b)$

$$f^* = \arg\min_f \frac{1}{2} f^T A f + f^T b + d_V(Af + b)$$

Inverse contact dynamics: $(A, v) \rightarrow (f, b = v - Af)$

$$f^* = \arg\min_f \ f^T (v + A \nabla d_V(v))$$



Todorov, *ICRA* 2011

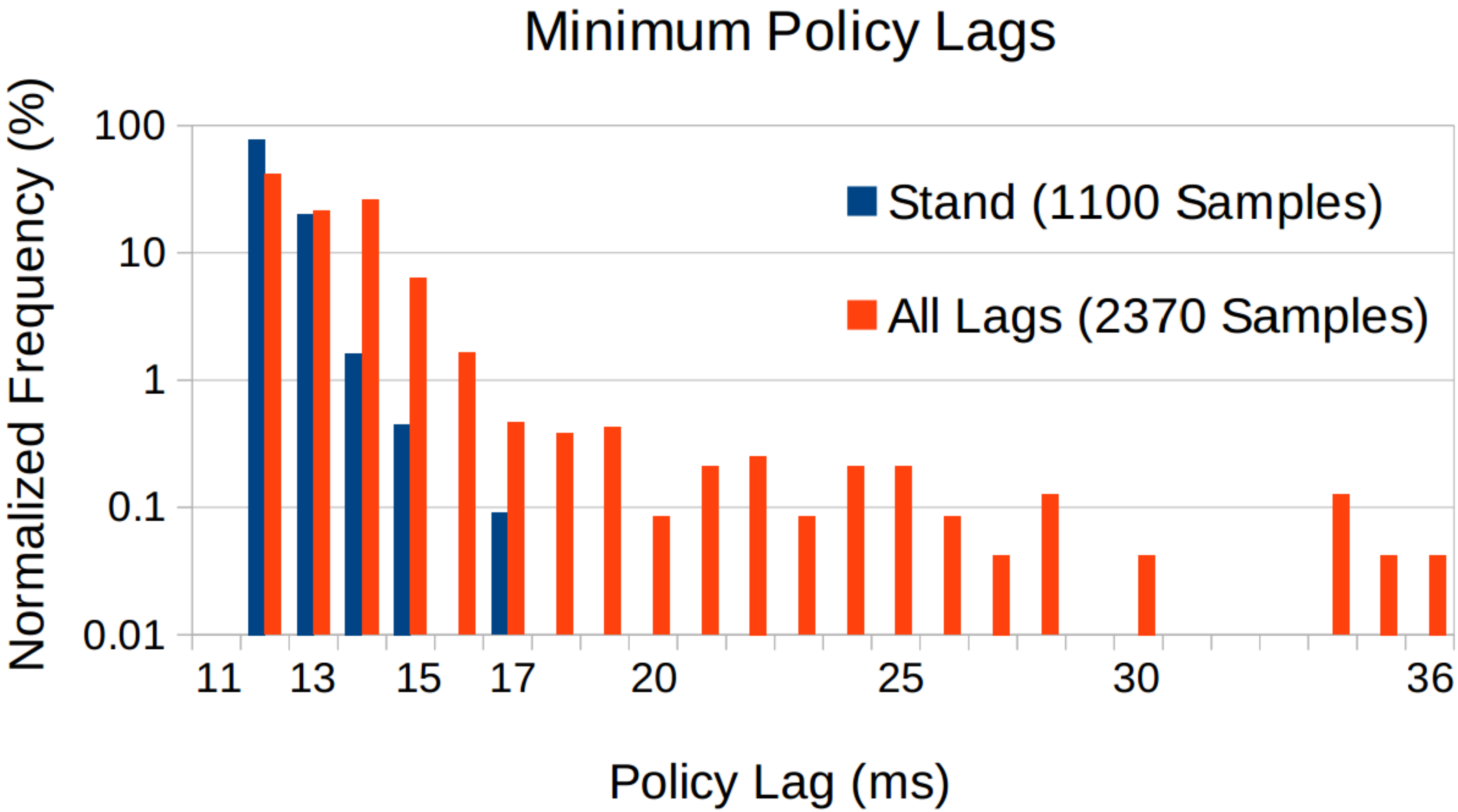
Timing (Single Core)

One dynamics step (35 DOF, 8 contacts)	0.01 ms = 100,000/sec

Harnessing Parallelization

One optimization step

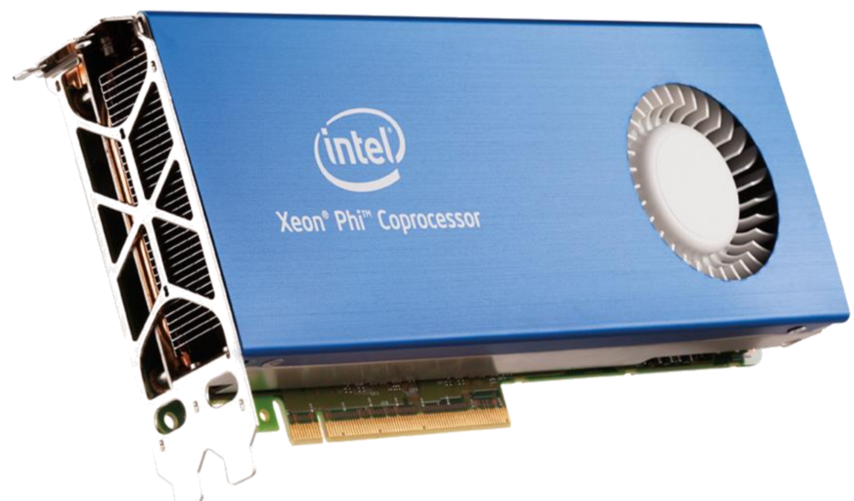
Single i7 core:	9 updates/second
Quad-core i7:	33 updates/second
8-core Xeon (Amazon EC2):	50 updates/second



Minimum Policy Lags
Normalized Frequency (%)
100
10
1
0.1
0.01
Stand (1100 Samples)
All Lags (2370 Samples)
11
13
15
17
20
25
30
36
Policy Lag (ms)

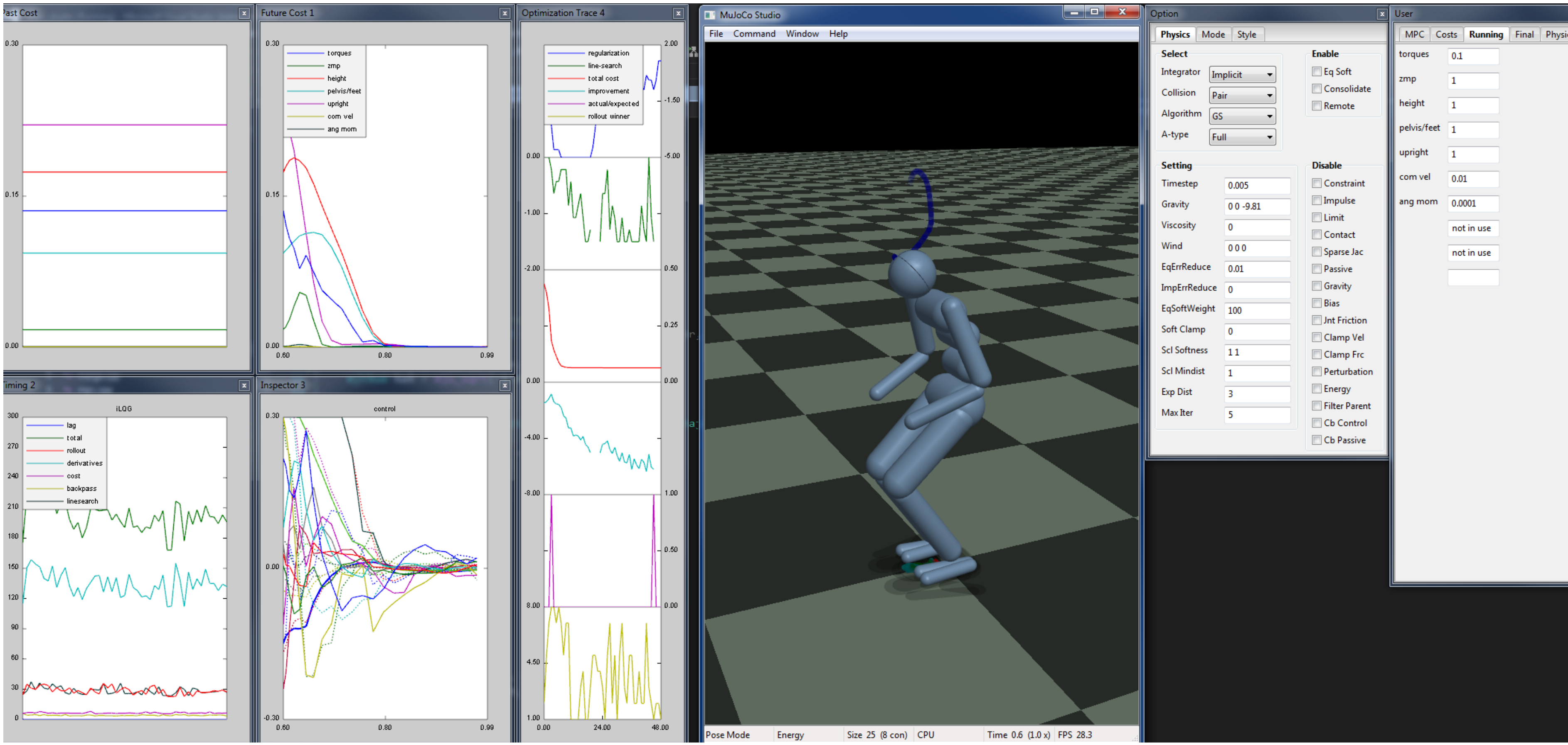
Harnessing Parallelization

- GPGPU
- Intel’s Xeon Phi co-processor



System integration

- Specification language for models and cost functions, transitions and alterations.
- Visualization for monitoring the system's state – timing, estimation, prediction errors, optimization status.
- GUI – switching between tasks, adjusting weights.


Past Cost
Future Cost 1
torques
zmp
height
pelvis/feet
upright
com vel
ang mom
Optimization Trace 4
regularization
line-search
total cost
improvement
actual/expected
rollout winner
Timing 2
iLQG
lag
total
rollout
derivatives
cost
backpass
linesearch
Inspector 3
control
MuJoCo Studio
File Command Window Help
Pose Mode Energy Size 25 (8 con) CPU Time 0.6 (1.0 x) FPS 28.3
Option
Physics Mode Style
Select
Integrator Implicit
Collision Pair
Algorithm GS
A-type Full
Enable
Eq Soft
Consolidate
Remote
Setting
Timestep 0.005
Gravity 0 0 -9.81
Viscosity 0
Wind 0 0 0
EqErrReduce 0.01
ImpErrReduce 0
EqSoftWeight 100
Soft Clamp 0
Scl Softness 1 1
Scl Mindist 1
Exp Dist 3
Max Iter 5
Disable
Constraint
Impulse
Limit
Contact
Sparse Jac
Passive
Gravity
Bias
Jnt Friction
Clamp Vel
Clamp Frc
Perturbation
Energy
Filter Parent
Cb Control
Cb Passive
User
MPC Costs Running Final Physics
torques 0.1
zmp 1
height 1
pelvis/feet 1
upright 1
com vel 0.01
ang mom 0.0001
not in use
not in use

Future work

- Estimation with contact
- Combining direct optimization to minimize cost tweaking
- Learning from experience
- “Instincts” – low-level overriding controllers
- Stiff, geared robots
- Pneumatic systems

Emo Todorov

Thank you!

Yuval Tassa