MuJoCo:
A physics engine for model-based control

Yuval Tassa, Emo Todorov
University of Washington

Movement
Control
Laboratory
motivation

• Control real robots in real time.

• Model-based optimization requires fast physics simulation. Much faster than real-time!

• Our ballpark: 10-100 DoFs, up to 1000 impulses.

• “MuJoCo” – a physics engine for control.
MuJoCo

- **Multi Joint** dynamics with **Contact**
- C/C++, multithreaded, no runtime allocation
- Minimal representation (generalized coordinates)
- Equality constraints (for loop topologies)
- Tendons & wrapping objects, actuator models
- Velocity-stepping impulse solver using convex optimization
- Joint limits & friction, contact, sliding, torsional, rolling friction
- Fast, differentiable, analytically invertible
MuJoCo
smooth dynamics

\[ M(q) \dot{v} = c(q, v) + u \]

<table>
<thead>
<tr>
<th>$q$</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>velocity</td>
</tr>
<tr>
<td>$u$</td>
<td>applied force</td>
</tr>
<tr>
<td>$M(q)$</td>
<td>inertia matrix</td>
</tr>
<tr>
<td>$c(q, v)$</td>
<td>springs, dampers, gravity, Coriolis, centripetal.</td>
</tr>
</tbody>
</table>
contact dynamics

<table>
<thead>
<tr>
<th>$q$</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>velocity</td>
</tr>
<tr>
<td>$u$</td>
<td>applied force</td>
</tr>
<tr>
<td>$M(q)$</td>
<td>inertia matrix</td>
</tr>
<tr>
<td>$c(q, v)$</td>
<td>springs, dampers, gravity, Coriolis, centripetal.</td>
</tr>
<tr>
<td>$h$</td>
<td>time step</td>
</tr>
<tr>
<td>$J(q)$</td>
<td>contact Jacobian</td>
</tr>
<tr>
<td>$f(q, v, u)$</td>
<td>contact impulse</td>
</tr>
</tbody>
</table>

\[
M \frac{dv}{dt} = (c + u)dt + J^T f
\]

**discretize** \[\downarrow\]

\[
M(v' - v) = (c + u)h + J^T f
\]

**forward:** \((q, v, u) \rightarrow (f, v')\)

**inverse:** \((q, v, v') \rightarrow (f, u)\)
forward dynamics

\[ Jv' = J(v + M^{-1}(c + u)h) + JM^{-1}J^T f \]

\[ v^+ = v^- + Af \]

\[
\begin{array}{|c|c|}
\hline
q & \text{position} \\
\hline
v & \text{velocity} \\
\hline
u & \text{applied force} \\
\hline
h & \text{time step} \\
\hline
M(q) & \text{inertia matrix} \\
\hline
c(q, v) & \text{Gravity, Coriolis, centripetal, springs, dampers etc.} \\
\hline
J(q) & \text{contact Jacobian} \\
\hline
f(q, v, u) & \text{contact impulse} \\
\hline
A & \text{c-space inverse-inertia} \\
\hline
v^- & \text{pre-impulse velocity} \\
\hline
v^+ & \text{post-impulse velocity} \\
\hline
\end{array}
\]
constraints on $f$

- joint limit: $f_i \geq 0$
- joint friction: $\eta(i) \pm f_i \geq 0$
- contact: $f_i \geq 0$
- friction: $f_i^2 - \sum_{j=1}^{d(i)} \left( \frac{f_{i+j}}{\mu_j(i)} \right)^2 \geq 0$
  \[d \in \{2, 3, 5\}\]

$\implies \phi(f) \geq 0$

convex!

d\in\{2, 3, 5\}

tangential
torsional
rolling
complementarity?

• NP-hard, not an optimization.

• Holds only for rigid contacts. (which we will presently soften)

• Unrealistic? [Chatterjee et al.]

• In practice, unnecessary.
energy dissipation

“Gauss principle”

contact-space kinetic energy:
\[ \frac{1}{2} v^+^T A^{-1} v^+ = \frac{1}{2} f^T A f + f^T v^- \]

(reminder:
\[ v^+ = v^- + A f \])

convex optimization:

minimize \( f \)
\[ \frac{1}{2} f^T A f + f^T v^- \]
subject to \( \phi(f) \geq 0 \)

but

Semi-definite A:

Existing penetrations:

<table>
<thead>
<tr>
<th>A</th>
<th>c-space inverse-inertia</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v^- )</td>
<td>pre-impulse velocity</td>
</tr>
<tr>
<td>( v^+ )</td>
<td>post-impulse velocity</td>
</tr>
</tbody>
</table>
**energy dissipation**

“Gauss principle”

**convex optimization:**

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} f^T (A + R) f + f^T (v^- - v^*) \\
\text{subject to} & \quad \phi(f) \geq 0
\end{align*}
\]

- Positive-definite regularizer \( R \) makes the solution
  - unique
  - differentiable
  - invertible
- Offset velocity \( v^* \) for constraint stabilization.

<table>
<thead>
<tr>
<th>( A )</th>
<th>c-space inverse-inertia</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v^- )</td>
<td>pre-impulse velocity</td>
</tr>
<tr>
<td>( v^+ )</td>
<td>post-impulse velocity</td>
</tr>
<tr>
<td>( R )</td>
<td>impulse regularizer</td>
</tr>
<tr>
<td>( v^* )</td>
<td>offset velocity</td>
</tr>
</tbody>
</table>
contact softening

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} f^T (A + R) f + f^T (v^- - v^*) \\
\text{subject to} & \quad \phi(f) \geq 0
\end{align*}
\]

How to choose \( R \) and \( v^* \)?

- Diagonal regularizer: \( R_{ii} = \epsilon A_{ii} \)

- \( v^* \) fixes penetrations exponentially with time-constant \( \kappa \).

- Objects at rest on the ground will penetrate to depth \( \approx \frac{\kappa^2 g \epsilon}{1 + \epsilon} \) independent of mass.
local Dynamic Programming

• **Forward pass:**
  Integrate the trajectory using the nominal controls.

• **Backward pass:**
  Approximate the Value along the current trajectory, improve the control sequence.

\[
x(t + 1) = f(x(t), u(t))
\]

\[
v(t, x) = \min_u [\ell(x, u) + v(t + 1, f(x, u))]
\]
Model Predictive Control
(online trajectory optimization)

Repeat:

• measure the current state $x_1$

• solve the finite horizon problem quickly

\[
\arg\min_{u_1, u_2, \ldots, u_{N-1}} \left[ \sum_{i=1}^{N-1} \ell(x_i, u_i) + \ell_f(x_N) \middle| x_1 \right]
\]

• (meanwhile) apply the initial controls $u_1$, $u_2$ ... to the system until a new policy arrives
Model Predictive Control

26ms / iter
4-core laptop
24 timesteps
400ms horizon
27 DoFs
21 actuators
~40 impulses

work with Tom Erez
is inversion possible?

$$(q, v, v') \rightarrow (f, u)$$

• An ill-defined problem for rigid bodies, forces do not always affect the state.

• Possible with spring-contacts, but these have other problems (tiny timesteps, energy injection).

• We present an invertible velocity-stepping method with soft contact.

what are the forces?
inverse dynamics

forward: minimize \( f: \phi(f) \geq 0 \) \( \frac{1}{2} f^T (A + R) f + f^T (v^- - v^*) \)

\[ \iff \]

inverse: minimize \( f: \phi(f) \geq 0 \) \( \frac{1}{2} f^T R f + f^T (v^+ - v^*) \)

\[ A \text{ is not required!} \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>c-space inverse-inertia</td>
</tr>
<tr>
<td>( v^- )</td>
<td>pre-impulse velocity</td>
</tr>
<tr>
<td>( v^+ )</td>
<td>post-impulse velocity</td>
</tr>
<tr>
<td>( R )</td>
<td>impulse regularizer</td>
</tr>
<tr>
<td>( v^* )</td>
<td>offset velocity</td>
</tr>
</tbody>
</table>

Key observation:
Since \( R \) is diagonal, this problem decomposes into separate, small problems, one per contact.
analytic solution

• Rewrite the inverse problem in least-squares form:

\[
\min_{f: \phi(f) \geq 0} \frac{1}{2} (f - y)^T R (f - y) \quad y \equiv R^{-1} (v^* - v^+) \]

• Minimize a quadratic function inside an elliptical cone:

\[
\min_{x: x \in \mathcal{C}} \sum_{k=0}^{d} r_k (x_k - y_k)^2 \quad \mathcal{C} \equiv \left\{ x \in \mathbb{R}^{d+1} : x_0 \geq 0, \ x_0^2 \geq \sum_{j=1}^{d} \mu_j^{-2} x_j^2 \right\}
\]

• See details in [Todorov, ICRA 2014], available online.
inverse dynamics
direct optimization

\[
\arg\min_{q_1, q_2, \ldots, q_N} \left[ \sum_{i=1}^{N} \ell(q_i, v_i, u_i, f_i) \right] \quad \text{unconstrained optimization!}
\]

\( \{u_i, f_i\} = \text{inverse}(q_i, v_i, v_{i+1}) \)

\( v_i = (q_i - q_{i-1})/h \)

Optimization space: positions
traitory estimation

A sequence of past positions: \( z_t = (q_{t-t_E}, \ldots, q_t) \)

\[- \log \hat{p}(z_t) = \text{const} - \log \bar{p}(z_t) - \]

\[
\sum_{k=t-t_E+1}^{t-1} \log p_k(y_k|q_k, v_k, \dot{v}_k) + \log p(\tau_k - u_k)
\]

| \( q \) | position |
| \( v \) | velocity |
| \( z \) | past positions |
| \( u \) | applied force |
| \( \tau \) | predicted force |
| \( y \) | sensor measurement |
Darwin robot:

State estimate:

contact detail
cross validation
THANKS!

To join the BETA program:
tassa@cs.washington.edu