# Contact dynamics via implicit complementarity

Emo Todorov

University of Washington

ICRA 2010

## Multi-joint dynamics with contact

- Continuous-time dynamics (may not have solution):

$$M(\mathbf{q})\,d\mathbf{w} = \mathbf{n}(\mathbf{q},\mathbf{w})\,dt + K(\mathbf{q})^T\,\mathbf{f}$$
$$K(\mathbf{q})\,\mathbf{w} = \mathbf{v}$$

| | |
|---|---|
| $\mathbf{q}, \mathbf{w}$ | joint position and velocity |
| $\mathbf{f}, \mathbf{v}$ | contact impulse and velocity |
| $M$ | joint-space inertia matrix |
| $\mathbf{n}$ | Coriolis, centripetal, gravitational, applied forces |
| $K$ | contact Jacobian |

$\mathbf{f}, \mathbf{v}$ are related through (an approximation to) the laws of contact and friction

- Discrete-time dynamics (always have solution):

Euler discretization with time step $\Delta$ yields $d\mathbf{w} \approx \mathbf{w}_{t+\Delta} - \mathbf{w}_t$,

$$M_t\mathbf{w}_{t+\Delta} = M_t\mathbf{w}_t + \Delta\mathbf{n}_t + K_t^T\mathbf{f}_{t+\Delta}$$
$$K_t\mathbf{w}_{t+\Delta} = \mathbf{v}_{t+\Delta}$$

## LCP formulation of contact

We need to solve

$$M\mathbf{w} = \mathbf{b} + K^T\mathbf{f}$$
$$K\mathbf{w} = \mathbf{v}$$

Find $\mathbf{f}, \mathbf{v}$ by solving

$$\boxed{A\mathbf{f} + \mathbf{c} = \mathbf{v}}$$

$A = KM^{-1}K^T$ : inverse inertia in contact space
$\mathbf{c} = KM^{-1}\mathbf{b}$ : contact velocity in the absence of contact force

Compute $\mathbf{w}$ as

$$\mathbf{w} = M^{-1}\left(\mathbf{b} + K^T\mathbf{f}\right)$$

$\mathbf{f} = \left[f^N; \mathbf{f}^F\right]$ and $\mathbf{v} = \left[v^N; \mathbf{v}^F\right]$ should satisfy the constraints

**Complementarity**

$$f^N \geq 0, \quad v^N \geq 0, \quad f^N v^N = 0$$
$\mathbf{v}^F$ parallel to $\mathbf{f}^F$
$$\left\langle \mathbf{v}^F, \mathbf{f}^F \right\rangle \leq 0, \quad \left\|\mathbf{f}^F\right\| \leq \mu f^N$$

The latter constraints are nonlinear, however the friction cone can be approximated with a *n*-sided pyramid, yielding a linear complementarity problem (LCP).

Widely used: ODE, PhysX, Havoc...

Problems with LCP and motivation for our method

- The approximation to the friction cone is inaccurate for small $n$
- Large $n$ results in too many auxiliary variables that slow down the solver
- Available algorithms are either slow (Lemke) or introduce additional approximations often resulting in spring-damper-like behavior

- The best general-purpose algorithm for solving LCPs is the PATH algorithm, which replaces the LCP with a nonlinear equation (and solves it using a non-smooth Newton method)
- If we are going to replace the LCP with a nonlinear equation, do we need the LCP in the first place? Or can we construct a nonlinear equation directly, without approximating the friction cone and introducing auxiliary variables? Yes we can.

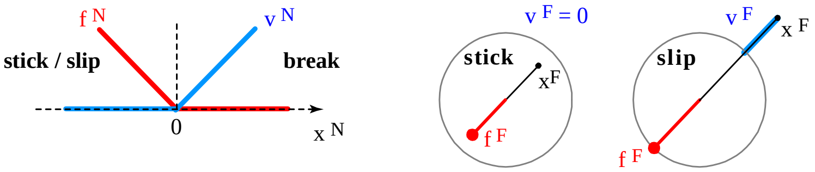
Implicit nonlinear complementarity

Instead of solving $A\mathbf{f} + \mathbf{c} = \mathbf{v}$ under complementarity constraints on $\mathbf{f}, \mathbf{v}$, we design functions $\mathbf{f}(\mathbf{x}), \mathbf{v}(\mathbf{x})$ such that the constraints are satisfied for all $\mathbf{x}$, and then solve the (unconstrained) nonlinear equation

$$A\mathbf{f}(\mathbf{x}) + \mathbf{c} = \mathbf{v}(\mathbf{x})$$

$\mathbf{x}$ is a hybrid variable encoding both contact velocities and contact forces.

normal forces and velocities: friction forces and velocities:




$\mathbf{v}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{x}$, thus the (non-smooth) equation becomes

$$\mathbf{r}(\mathbf{x}) \triangleq (A - I)\,\mathbf{f}(\mathbf{x}) - \mathbf{x} + \mathbf{c} = 0$$

# The functions f and v

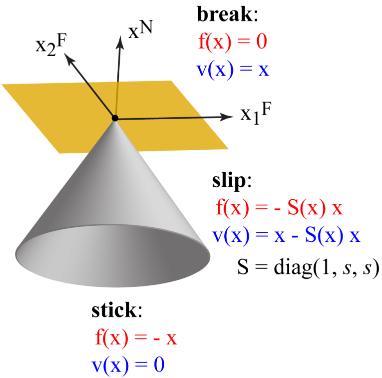
normal forces and velocities:

$$f^N(\mathbf{x}) = \max\left(0,\ -x^N\right)$$
$$v^N(\mathbf{x}) = \max\left(0,\ x^N\right)$$

friction forces and velocities:

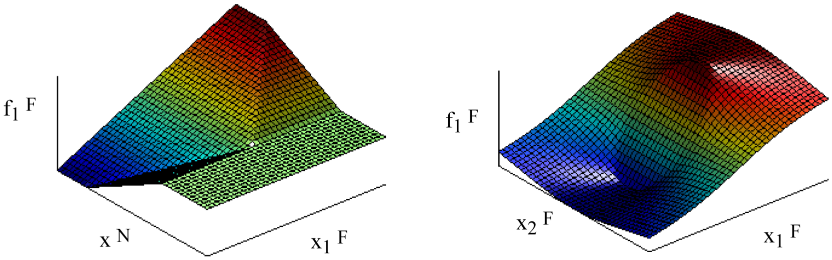
$$s(\mathbf{x}) \triangleq \min\left(1,\ \frac{\mu f^N(\mathbf{x})}{\|\mathbf{x}^F\|}\right)$$
$$\mathbf{f}^F(\mathbf{x}) = -s(\mathbf{x})\,\mathbf{x}^F$$
$$\mathbf{v}^F(\mathbf{x}) = \mathbf{x}^F - s(\mathbf{x})\,\mathbf{x}^F$$

3D forces and velocities:



Shape of the function f

$$\mathbf{f}^F(\mathbf{x}) = -\min\left(1, \frac{\mu \max(0, -x^N)}{\|\mathbf{x}^F\|}\right)\mathbf{x}^F$$



Root-finding via optimization

Solving $\mathbf{r}(\mathbf{x}) = 0$ is equivalent to minimizing the objective function

$$\ell(\mathbf{x}) = \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|^2$$

This is a non-linear least squares problem, which (in principle) can be handled by a Gauss-Newton method:

$$J(\mathbf{x}) = \frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \quad \text{Jacobian/subdifferential of } \mathbf{r}(\mathbf{x})$$

$$J(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \quad \text{gradient of } \ell(\mathbf{x})$$

$$J(\mathbf{x})^T J(\mathbf{x}) \quad \text{approximate Hessian of } \ell(\mathbf{x})$$

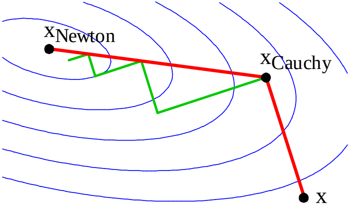
The (stabilized) Newton iteration is

$$\mathbf{x} \leftarrow \mathbf{x} - \left(J(\mathbf{x})^T J(\mathbf{x}) + \lambda I\right)^{-1} J(\mathbf{x})^T \mathbf{r}(\mathbf{x})$$

$\lambda$ is adapted online in Levenberg-Marquardt fashion.

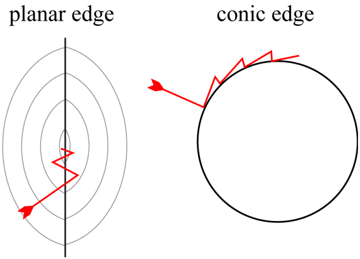
# Optimization with edge-aware linesearch

Second-order methods avoid the chattering characteristic of first-order methods (green):




Here we use the "dogleg" method which involves two linesearhes (red). The Cauchy point is the minimum along the gradient.

Non-smoothness can still cause chattering:

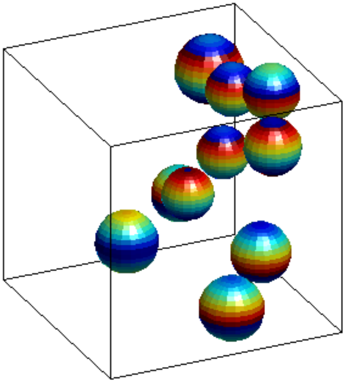



We explicitly consider the intersections of the red lines with the edges where $\mathbf{r}(\mathbf{x})$ is non-smooth (planes and cones).

## Numerical results

With $\mu = 1$ and $n_c = 16$ contacts the algorithm takes 5 iterations per time step (no warm start) and accepts the Newton point (without linesearch) on 70% of iterations.

Each iteration here is faster than an iteration of an LCP solver because we are factorizing smaller matrices: $3n_c$ -by- $3n_c$ as opposed to, say, $10n_c$ -by- $10n_c$.



| $\mu$ | $n_b = 5$, $n_c = 7$ | $n_b = 10$, $n_c = 16$ | $n_b = 15$, $n_c = 27$ |
|---|---|---|---|
| 0.1 | 3.8<br>99 % | 4.8<br>98 % | 6.5<br>90 % |
| 0.5 | 2.6<br>95 % | 5.3<br>85 % | 7.5<br>73 % |
| 1.0 | 2.8<br>90 % | 4.9<br>71 % | 10.2<br>60 % |
| 2.0 | 2.9<br>88 % | 4.6<br>71 % | 16.2<br>55 % |

# A convex, smooth and invertible contact model for trajectory optimization

Emo Todorov
University of Washington

ICRA 2011

# Trajectory optimization

**Trajectory optimization via forward dynamics** $\dot{q}_{t+1} = f\left(q_t, \dot{q}_t, u_t\right)$

jointly optimize $q_1 \ldots q_T$ and $u_0 \ldots u_{T-1}$ subject to the dynamics constraints

- double the dimensionality
- non-linear equality constraints

optimize $u_0 \ldots u_{T-1}$, define $q_1 \ldots q_T$ by integrating the forward dynamics

- dense Hessian (each control affects all future positions)
- small time step needed (for stable integration)

**Trajectory optimization via inverse dynamics** $u_t = f^{-1}\left(q_t, \dot{q}_t, \dot{q}_{t+1}\right)$

optimize $q_1 \ldots q_T$, define $u_0 \ldots u_{T-1}$ from the inverse dynamics

- minimal representation (no equality constraints)
- large time steps can be used (no stability concerns)
- the Hessian of the trajectory cost is sparse

$$J\left(q_0 \ldots q_T\right) = \sum_{t=1}^{T-1} \ell\left(q_t,\ \dot{q}_t\left(q_{t-1}, q_t\right),\ u_t\left(q_{t-1}, q_t, q_{t+1}\right)\right)$$

# Inverse dynamics with contact

The inverse is not well-defined because contact makes it possible to apply forces that do not affect the position (e.g. pushing against the ground)

Resolving this indeterminacy in an adhoc way generally results in inverse dynamics that are discontinuous, and thus unsuitable for optimization

Contacts could be modeled with springs (which are invertible), but springs are a poor model of contact and have been largely abandoned in favor of implicit solvers (e.g. LCP)

None of the existing implicit solvers are invertible

Here we develop the first implicit solver that is invertible (and furthermore convex).

An inverse implies that any force causes displacement (resembling a spring), nevertheless the resulting contacts are "hard", i.e. there is no penetration

# Review of implicit contact modeling

| | |
|---|---|
| $M$ | inertia matrix |
| $n$ | passive forces |
| $B$ | actuation matrix |
| $h$ | time step |
| $K$ | contact Jacobian |
| $f$ | contact impulse |
| $v$ | contact velocity |
| $A$ | contact inverse inertia |
| $c$ | contact bias velocity |

$$M(q_t)\frac{\dot{q}_{t+1}-\dot{q}_t}{h}+n(q_t,\dot{q}_t)=Bu_t+K(q_t)^T\frac{f_t}{h}$$

$$M_t\dot{q}_{t+1}=M_t\dot{q}_t+h(Bu_t-n_t)+K_t^{\,T}f_t$$

$$K_t\dot{q}_{t+1}=v_{t+1}$$

$$A_t=K_tM_t^{-1}K_t^{\,T}$$

$$c_t=K_t\dot{q}_t+hK_tM_t^{-1}(Bu_t-n_t)$$

$$A_tf_t+c_t=v_{t+1}$$

forward dynamics: $(A_t,c_t)\rightarrow(f_t,v_{t+1})$

inverse dynamics: $(A_t,v_{t+1})\rightarrow(f_t,c_t)$

$A\,f+c=v$ has twice as many unknowns as equality constraints.

However it can be solved by taking into account additional constraints on $f$ and $v$.

$$f_N\geq 0,\quad v_N\geq 0$$

$$\mu\, f_N\geq\|f_T\|$$

Here we will relax the complementarity constraint

$$f_N\,v_N=0$$

# Our forward dynamics model

The contact impulse $f$ minimizes kinetic energy in contact space:

$$\frac{1}{2}v^T A^{-1} v = \frac{1}{2}(Af+c)^T A^{-1}(Af+c)^T = \frac{1}{2} f^T A f + f^T c + const$$

subject to $\quad f_N \geq 0, \quad v_N \geq 0, \quad \mu f_N - \|f_T\| \geq 0 \quad$ for each contact

We define $f$ as the solution obtained by a primal interior-point method, which finds the (unconstrained) minimum of the convex function

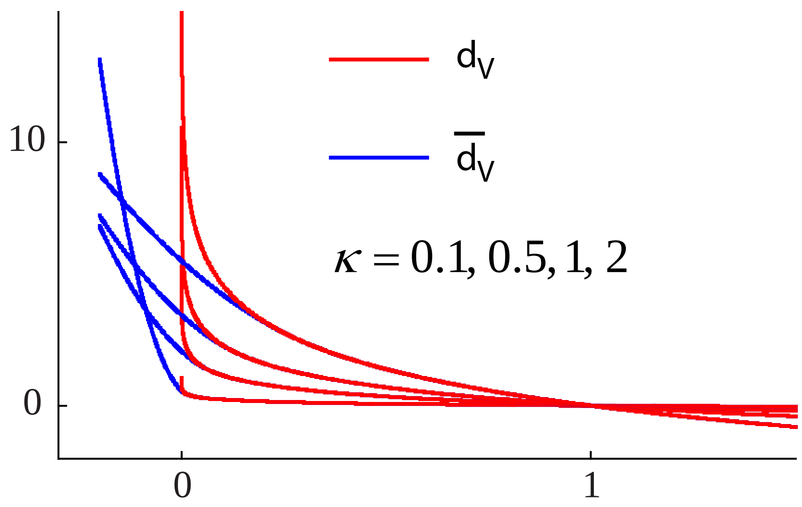
$$\frac{1}{2} f^T (A+R) f + f^T c + d_F(f) + d_V(v)$$

where $d_V(v) = -\kappa \sum_{contacts} \log v_N$

$$d_F(f) = -\kappa \sum_{contacts} \log f_N + \log\left(\mu^2 f_N^2 - \|f_T\|^2\right)$$

The solution depends on $R$, $\kappa$

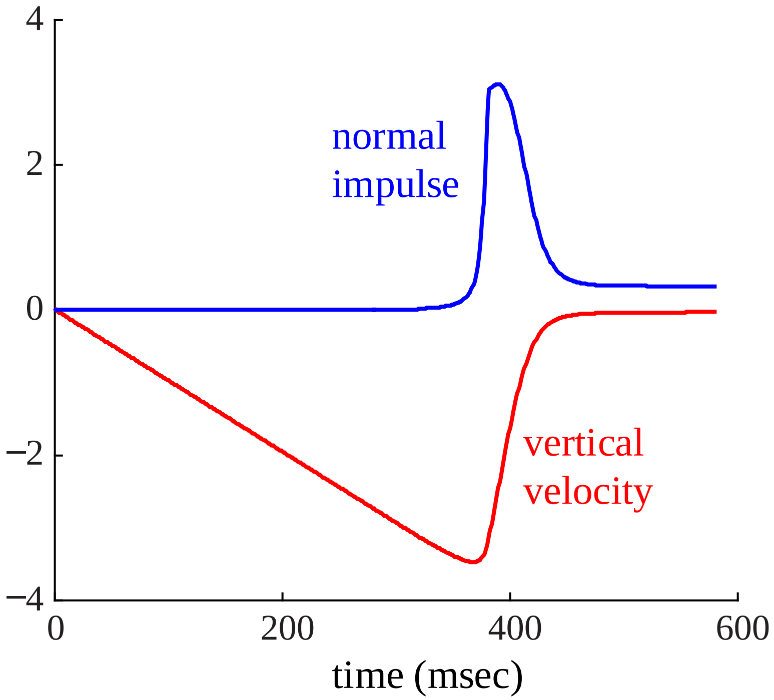
To find a feasible solution, start with a quadratic in $v$ instead of log-barrier.



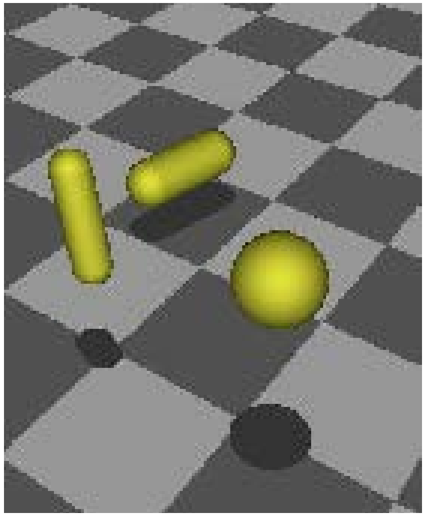
# The forward dynamics are sensible

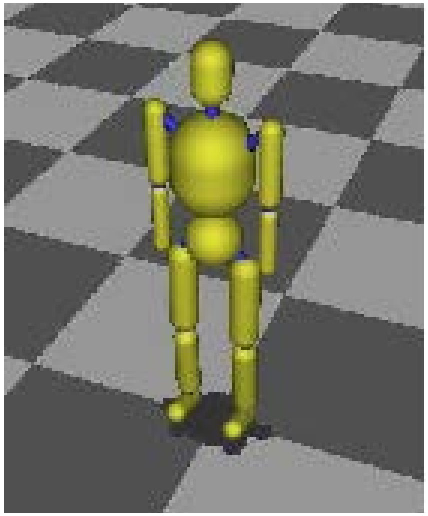
The complementarity constraint (which we ignored) is softly enforced by the regularized kinetic energy cost, because violating complementarity tends to increase both the kinetic energy and the ($R$-norm of the) contact force.

Ball-drop test:




Interactive test:





# Our inverse dynamics model

The forward dynamics solution $(f, v)$ is at the minimum of the convex function

$$\text{forward:} \qquad \frac{1}{2} f^T (A+R) f + f^T c + d_F(f) + d_V(v)$$

We will recover $(f, c)$ from $(A, v)$ using the fact that the gradient at the minimum is 0:

$$(A+R) f + c + \frac{\partial d_F(f)}{\partial f} + \frac{\partial v}{\partial f} \frac{\partial d_V(v)}{\partial v} = 0$$

Using $A f + c = v$ yields

$$v + R f + \frac{\partial d_F(f)}{\partial f} + A \frac{\partial d_V(v)}{\partial v} = 0$$

Key observation: the unknown $c$ vanished (it was absorbed in the known $v$)

The above equation can be solved by minimizing the convex function

$$\text{inverse:} \qquad \frac{1}{2} f^T R f + f^T r + d_F(f)$$

where the new constant is $\quad r = v + A \frac{\partial d_V(v)}{\partial v}$

# Performance of the inverse contact solver

*3 GHz Intel processor, single thread, 18-dof humanoid*

| # contacts | inverse evaluations per second |
|---|---|
| 2 | 241, 774 |
| 5 | 54, 190 |
| 10 | 12, 991 |
| 20 | 2, 361 |