Assignment news

• Assignment due Monday next week

• Deliverables: all the code files

• Optional “pre-submission” by Friday:
  Upload the code to Moodle and email me if you want me to take a look at it and give you advice.
  You can do this once per assignment.

• `scons -Q debug=1` to compile with debugging symbols

• SCons can also generate Visual Studio projects (though I haven’t tried this myself), see http://scons.org/doc/production/HTML/scons-user.html#b-MSVSPProject
Constraints

Mechanisms  Articulated bodies  Inextensible ropes
Test case: Double pendulum
Maximal coordinates

Keep all the degrees of freedom: $x_1, x_2$

Add constraints as equations:

\[
c_1(x) := \|x_1\| - \ell_1 = 0
\]
\[
c_2(x) := \|x_2 - x_1\| - \ell_2 = 0
\]

\[
c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \end{bmatrix} = 0
\]
Penalty forces

Replace hard constraints with “soft” ones, i.e. springs of finite stiffness

\[ U_{\text{con}}(\mathbf{x}) = \frac{1}{2} k \| \mathbf{c}(\mathbf{x}) \|^2 \]
\[ = \frac{1}{2} k (c_1(\mathbf{x})^2 + c_2(\mathbf{x})^2 + \cdots) \]

\[ \mathbf{f}_{\text{con}}(\mathbf{x}) = -k (c_1 \nabla c_1 + c_2 \nabla c_2 + \cdots) \]
\[ = -k \mathbf{J}_c^T \mathbf{c} \]

• Force parallel to constraint gradient
• Magnitude proportional to constraint violation
  (usually add some damping forces too)

Problems?
Ideal constraint forces

\[ f_{\text{con}}(x) = -k c_1 \nabla c_1 + \cdots \]

What can we say about \( f \) as \( k \to \infty \)?

\[ f_{\text{con}}(x) = \lambda_1 \nabla c_1 + \cdots \]

What can we say about position and velocity?

\[ c_1(x) = 0, \ldots \]

\[ \nabla c_1 \cdot v = 0, \ldots \]
Ideal constraint forces

\[ f_{\text{con}}(x) = -k c_1 \nabla c_1 + \cdots 
= -k J_c^T c \]

What can we say about \( f \) as \( k \to \infty \)?

\[ f_{\text{con}}(x) = \lambda_1 \nabla c_1 + \cdots 
= J_c^T \lambda \]

What can we say about position and velocity?

\[ c_1(x) = 0, \ldots \iff c(x) = 0 \]

\[ \nabla c_1 \cdot \mathbf{v} = 0, \ldots \iff J_c \mathbf{v} = 0 \]

Exercise: verify these equations.
Constrained dynamics

\[ \dot{x} = v \]
\[ M \ddot{v} = f(x, v) + J_c(x)^T \lambda \]
\[ c(x) = 0 \iff J_c(x)v = 0 \iff \ldots \]
The constraint Jacobian

\[ J_c = \frac{dc}{dx} \]

\(N\) dofs, \(M\) constraints: \(J_c\) is \(M \times N\)

\(J_c\) maps from “dof space” to “constraint space”

- \(v\) = node velocities, \(J_c v\) = how fast each constraint is being violated

\(J_c^T\) maps from “constraint space” to “dof space”

- \(\lambda\) = response of each constraint, \(J_c^T \lambda\) = forces on nodes

Examples
Constrained dynamics

\[ \dot{x} = v \]

\[ M\dot{v} = f(x, v) + J_c(x)^T\lambda \]

\[ c(x) = 0 \iff J_c(x)v = 0 \iff \ldots \]

For time integration, we want to go from \( x^n, v^n \) to \( x^{n+1}, v^{n+1} \) such that

\[ c(x^{n+1}) = 0 \]

\[ J_c(x^{n+1})v^{n+1} = 0 \]
Implicit-explicit methods

Suppose you have an ODE

\[ y' = f(y) = f_1(y) + f_2(y) \]

- Forward Euler: \[ y^{n+1} = y^n + (f_1(y^n) + f_2(y^n))\Delta t \]
- Backward Euler: \[ y^{n+1} = y^n + (f_1(y^{n+1}) + f_2(y^{n+1}))\Delta t \]
- Consider instead:

\[ y^{n+1} = y^n + (f_1(y^n) + f_2(y^{n+1}))\Delta t \]
\[ = \tilde{y} + f_2(y^{n+1})\Delta t \]

\[ y^n + f_1(y^n)\Delta t \]
Implicit-explicit methods

\[
\frac{d}{dt} \begin{bmatrix} x \\ Mv \end{bmatrix} = \begin{bmatrix} v \\ f + J_c^T \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix} + \begin{bmatrix} v \\ J_c^T \lambda \end{bmatrix}
\]

Explicit integration of other forces

\[
M\ddot{v} = Mv^n + f^n \Delta t
\]

Implicit projection to constraint set

\[
x^{n+1} = x^n + v^{n+1} \Delta t
\]

\[
Mv^{n+1} = M\ddot{v} + (J_c^{n+1})^T \lambda \Delta t
\]

\[
c^{n+1} = 0
\]

If there are no constraints, does this reduce to forward Euler?
Constraint projection

\[ x^{n+1} = x^n + v^{n+1} \Delta t \]
\[ Mv^{n+1} = M\ddot{v} + (J_{c}^{n+1})^T \lambda \Delta t \]
\[ c^{n+1} = 0 \]

Changing \( \lambda \) affects \( v^{n+1} \), which in turn affects \( x^{n+1} \).

We want to find the value of \( \lambda \) so that \( c(x^{n+1}) \) becomes 0.