Physics-Based Animation

09 — Constraints and collisions

October 4, 2016
Assignment news
Constrained dynamics
wrap-up
Quick recap: Constrained dynamics

- Take a generic physical system: \( x, v, M, f \)
  \[
  \begin{align*}
  \dot{x} &= v \\
  M\dot{v} &= f(x, v)
  \end{align*}
  \]

- Impose (equality) constraints \( c(x) = 0 \)

- Constraint forces are of the form \( f_{\text{con}} = J_c^T\lambda \)
  \[
  \begin{align*}
  \dot{x} &= v \\
  M\dot{v} &= f(x, v) + J_c(x)^T\lambda \\
  c(x) &= 0 \iff J_c(x)v = 0 \iff \ldots
  \end{align*}
  \]

\begin{itemize}
  \item differential-algebraic equation (DAE)
\end{itemize}
Quick recap: Constrained dynamics

\[
\begin{align*}
\dot{x} &= v \\
M \dot{v} &= f(x, v) + J_c(x)^T \lambda \\
c(x) = 0 &\iff J_c(x)v = 0 \\
\end{align*}
\]

In the Pixar notes:

• Differentiate \( J_c v = 0 \) again to get \( \dot{J}_c v + J_c \dot{v} = 0 \)

• Solve for \( \lambda \) and \( \dot{v} \) (work this out)

Benefit: eliminates \( \lambda \), reducing DAE to ODE again

Problems: drift, tuning, need to compute \( \dot{J}_c \)
**Implicit projection**

Discretize using implicit integration (e.g. backward Euler), impose constraint directly on $x^{n+1}$

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

\[
Mv^{n+1} = Mv^n + (f(x^{n+1}, v^{n+1}) + J_c(x^{n+1})^T\lambda)\Delta t
\]

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

$2N + M$ nonlinear equations in $2N + M$ unknowns ($x^{n+1}, v^{n+1}, \lambda$)

Apply Newton’s method:
linearize about $x^n, v^n$, assume $J_c$ is constant, solve for $\lambda$
Explicit integration + constraint projection

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

1. First integrate only external forces:

\[M\dot{v} = f\]

2. Then integrate constraint forces:

\[
\dot{x} = v \\
M\dot{v} = J_c(x)^T \lambda \\
c(x) = 0
\]
Explicit integration
+ constraint projection

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

1. First integrate only external forces:

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

2. Then integrate constraint forces:

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

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

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

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 \( 0 \).

Again, linearize about \( x^n \), assume \( J_c \) is constant, solve for \( \lambda \)

(Again, the matrix \( J_c M^{-1} J_c^T \)…

\[
\begin{align*}
\text{c(x)} &= 0 \\
x^n + \tilde{v} \Delta t \\
J_c^T
\end{align*}
\]
Velocity stabilization

In continuous time,

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

With discrete time steps, this is not true in general.

One can add another projection step for \( J_c(x^{n+1})v^{n+1} = 0 \):

\[
\frac{d}{dt} \begin{bmatrix} x \\ Mv \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix} + \begin{bmatrix} v \\ J^T_c \lambda_x \end{bmatrix} + \begin{bmatrix} 0 \\ J^T_c \lambda_v \end{bmatrix}
\]

\( c(x) = 0 \)

\( J_c v = 0 \)
Explicit integration + constraint projection

1. Compute a candidate velocity:

\[ \ddot{v} = v^n + M^{-1}f^n \Delta t \]

2. Modify velocity to project \( x^{n+1} \) onto constraint manifold, \( c(x^{n+1}) = 0 \)

3. (Optional) Project velocity to constraint tangent space, \( J_c(x^{n+1})v^{n+1} = 0 \)

“Velocity filters”
Inequality constraints and collisions
Inequality constraints

\[ \| x_1 \| - \ell_1 = 0 \]
\[ \| x_1 \| - \ell_1 \leq 0 \]

\[ c(x) = 0 \]
\[ J_c v = 0, \quad f_{\text{con}} = J_c^T \lambda \]

\[ c(x) \leq 0 \]
\[ \text{If } c(x) = 0 \text{ then } J_c v \leq 0, \quad f_{\text{con}} = J_c^T \lambda \]
\[ \text{If } c(x) < 0 \text{ then } J_c v = \text{anything, } f_{\text{con}} = 0 \]
Inequality constraints

\[ 0^\circ \leq \theta \leq 150^\circ \]

\[ y \geq 0 \]
Collisions as constraints

Consider a mass-spring system colliding with a static object. Give the object a signed distance field $\varphi$:

$$\varphi(x) = \pm \text{(distance from } x \text{ to surface of object)},$$

negative if inside, positive if outside

- Non-penetration constraint on particle $i$:
  $$\varphi(x_i) \geq 0$$

- Jacobian = surface normal $n$

- If constraint is active (particle is on object surface),
  $$n \cdot v_i \geq 0,$$
  $$f_{\text{con}} = \lambda n$$
Penalty forces

\[ U_{\text{con}}(x) = \begin{cases} 
0 & c(x) \leq 0 \\
\frac{1}{2} k c(x)^2 & c(x) > 0 
\end{cases} \]

\[ f_{\text{con}}(x) = \begin{cases} 
0 & c(x) \leq 0 \\
-k c(x) \nabla c(x) & c(x) > 0 
\end{cases} \]

Check if current state violates constraint.
If so, apply force proportional to constraint violation.
Constraint projection for inequality constraints

1. Compute a candidate step

2. *Check if it leaves the allowed region*  
   (collision detection)

3. If so, apply constraint forces to fix it  
   (collision response)
An Illegal State $X$

$X(t_0 + 2\Delta t)$

$X(t_0 + \Delta t)$

$X(t_0)$

$X(t_0 + 3\Delta t)$

illegal state

Strategy?
Plan 2: Backstep

\[ X(t_0 + 2\Delta t) \]

\[ X(t_0 + \Delta t) \]

\[ X(t_0) \]
Plan 3: Just Lie About It!

\[ \mathbf{X}(t_0 + 2\Delta t) \]

\[ \mathbf{X}(t_0 + \Delta t) \]

\[ \mathbf{X}(t_0) \]
Next class

- *Collision detection and response*

- **Reading:** Witkin and Baraff notes, “Rigid Body Dynamics” only Ch. 6 “Problems of Nonpenetration Constraints” and Ch. 7 “Collision Detection”