Last Time:

- LQR with Quaternions
- Contact Intro

Today:

- Contact modeling with hybrid systems
- Traj Opt for legged systems

---

* Falling Brick Two Ways:

1) Time-Stepping Method

$$m\dot{v} = -mg + J^T\lambda \quad , \quad g = \begin{bmatrix} 0 \\ 9.81 \end{bmatrix} \quad x = \begin{bmatrix} q \\ v \end{bmatrix}$$

contact Jacobian

contact force

$$\phi(q) = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{J} \begin{bmatrix} q_x \\ q_y \end{bmatrix} \geq 0$$

"signed distance function"

"Interpenetration Constraint"

$\Downarrow$ Backward Euler

$$m\left(\frac{v_{n+1} - v_n}{h}\right) = -mg + J^T\lambda_n$$

$$q_{n+1} = q_n + h v_{n+1}$$

$$\phi(q_{n+1}) \geq 0$$

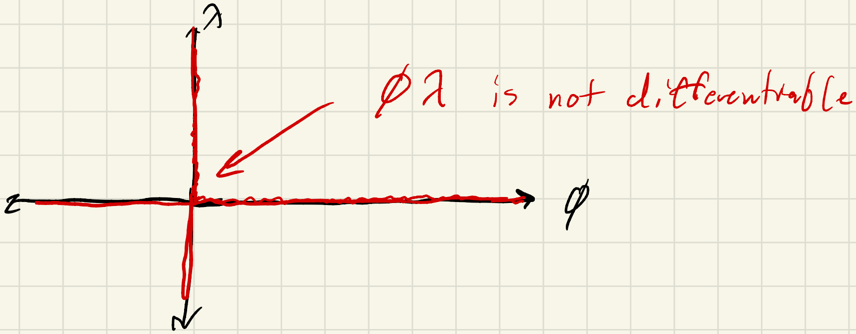
$$\lambda_n \geq 0$$ only positive forces (pushing)

$$\phi(q_{n+1})\, \lambda_n = 0$$ no force unless brick is in contact

- This is a QP in disguise! (KKT conditions):

$$\min_{v_{n+1}} \ \tfrac{1}{2} m v_{n+1}^T v_{n+1} + m v_{n+1}^T (hg - v_n)$$
$$\text{s.t.} \quad J(q_n + h v_{n+1}) \geq 0$$

- Exact impact time is not resolved (only time step)
- Contact forces (λ_n) are explicitly computed
- Doesn't generalize to higher-order integration (e.g. RK4). Often need to take very small steps.
- Widely used in robotics simulators: Bullet, Dart, Gazebo. (also kinda MuJoCo)
- Key problem for traj opt: complementarity condition:



2) Hybrid Method

$$\dot{x} = f(x) = \begin{bmatrix} v \\ -g \end{bmatrix}$$ "smooth vector field/dynamics"

$$\phi(x) \geq 0$$ "guard function"

$$x' = g(x) = \begin{bmatrix} q_x \\ q_y \\ v_x \\ 0 \end{bmatrix}$$ "jump map"

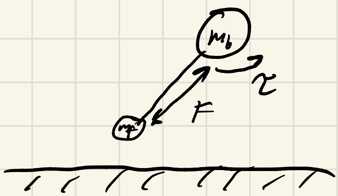
zero out vertical velocity
"inelastic collision"

$$
\begin{aligned}
&\text{while } t < t_{final} \\
&\quad \text{if } \phi(x) \geq 0 \\
&\qquad \dot{x} = f(x) && \leftarrow \text{use any integrator you want (e.g. RK4)} \\
&\quad \text{else } (\phi = 0) && \leftarrow \text{guard function detects impact} \\
&\qquad x' = g(x) && \leftarrow \text{jump map simulates non-smooth impact event} \\
&\quad \text{end} \\
&\text{end}
\end{aligned}
$$

- Precise impact time
- Contact forces are not explicitly computed
- Can use high-accuracy integrators
- Widely used for Traj Opt
- Key insight for Traj Opt! if we know (or specify) impact times a-priori, we don't need the guard function and can just deal with $f(x)$ and $g(x)$, which are differentiable.

---

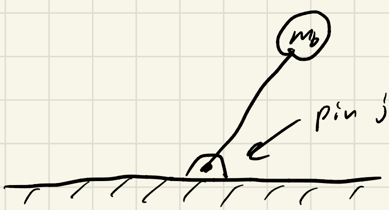
* Hybrid Trajectory Optimization for Legged Robots
  - One-legged hopping robot:

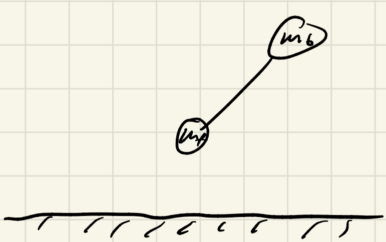


$$x = \begin{bmatrix} r_b \\ r_f \\ v_b \\ v_f \end{bmatrix} \in \mathbb{R}^8 \qquad u = \begin{bmatrix} F \\ \tau \end{bmatrix} \in \mathbb{R}^2$$

- Define smooth dynamics models for each mode (stance vs. flight).

$$\dot{x} = f_1(x,u) \qquad\qquad \dot{x} = f_2(x,u)$$





- Define jump map to transition between modes

$$x' = g_{21}(x) = \begin{bmatrix} r_b \\ r_f \\ v_b \\ 0 \end{bmatrix} \leftarrow \text{zero out foot velocity when it impacts}$$

$(g_{12}(x) = x$ for this problem$)$

- Assign modes to alternating groups of knot points by enforcing appropriate constraints:

for $k = 1 : N_1$
$\quad x_{n+1} = f_1(x_n, u_n)$
$\quad \phi(x_n) = 0$
end
for $n = (N_1 + 1) : N_2$
$\quad x_{n+1} = f_2(x_n, u_n)$
end
$\quad x_{N2} = g_{21}(x_{12})$
$\quad \phi(x_{N2}) = 0$

* Example: Hopper

- Mode specification is easy for this system/task

- Gets much harder for eg. quadruped: must pre-specify gait (trot vs. gallop)

- Most state-of-the art controllers (e.g. Cheetah) use carefully tuned heuristics for gait generation and/or footstep planning in an outer loop.