Last Time:

- LQR with Quaternions

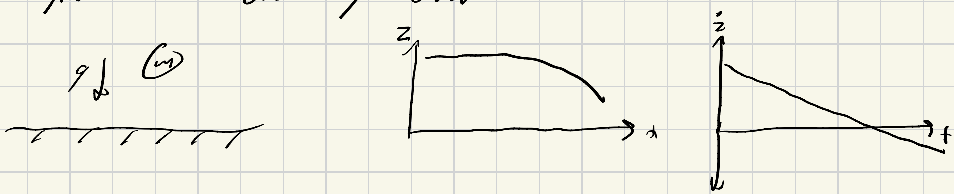
Today:

- Contact dynamics
- Hybrid systems modeling
- TrajOpt for legged systems

---

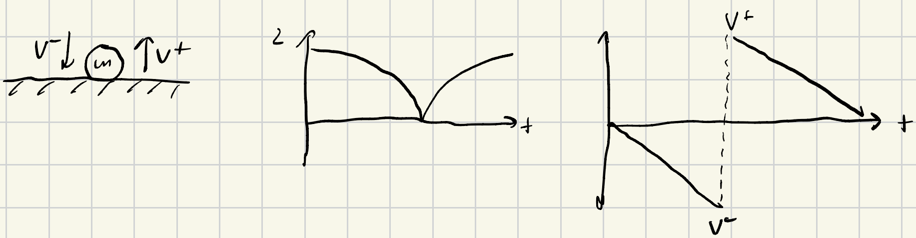
* Contact Dynamics

- Imagine a bouncing ball



- In the air, dynamics described by smooth ODE $(m\ddot{z} = -mg)$

- When the ball hits the ground



- Because of discontinuities, can't write down dynamics around impact as an ODE

* TWO Options:

1) Event-based / hybrid: Integrate ODE while checking for contact events using a "guard function" (e.g. $z \geq 0$). When contact happens, execute "jump map" that models the discontinuity, then continue integrating ODE.

2) Time-stepping / contact-implicit: Solve a constrained optimization problem at each time step that enforces no interpenetration between objects ($\phi(x) \geq 0$) by solving for contact forces jointly with state (HW1 brick problem)

- Both are widely used and have pros/cons

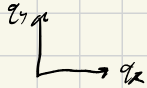
- In control, hybrid formulation is easy to implement with standard algorithms (e.g. DIRCOL)

- Downsides: requires enumeration of all possible "contact modes" and/or pre-specification of "mode sequence"

- Very successful in locomotion

- Contact-implicit method doesn't need mode sequence pre-specified, but the optimization problems are much harder.

* Falling Brick Two Ways:



1) Time-Stepping Method:

$$m\dot{v} = -mg + J^T\lambda$$

← contact force

Contact Jacobian

$$g = \begin{bmatrix} 0 \\ 9.8 \end{bmatrix}, \qquad x = \begin{bmatrix} q \\ v \end{bmatrix}$$

$$\underbrace{\phi(q)}_{\text{signed-distance function}} = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{J} \begin{bmatrix} q_x \\ q_y \end{bmatrix}$$

⇓ 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 pushing (no pulling)

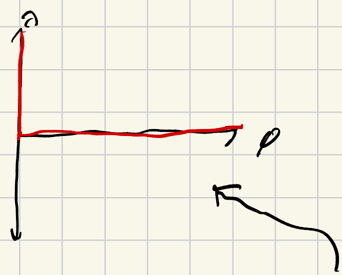
$$\phi(q_{n+1})\,\lambda_n = 0$$ ← no force unless you're in contact

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

$$\min_{v_{n+1}} \frac{1}{2} m v_{n+1}^T v_{n+1} + m v_{n+1}^T (hg - v_n)$$

$$\text{s.t.} \quad J(q_n + h v_{n+1}) \geq 0$$

- Exact impact time is not not resolved (only step)
- Contact forces (λ_n) are explicitly computed
- Doesn't generalize to higher-order integration (e.g. RK4) need to take small steps
- Widely used: PyBullet, Dart, Gazebo, etc.



- key issue for traj opt: complementarity condition is non-smooth

2) Hybrid Method:

$$\dot{x} = f(x) \Rightarrow \begin{bmatrix} \dot{q} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -v \\ -g \end{bmatrix}$$ ← "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

while $t < t_{final}$

if $\phi(x) \geq 0$

$\dot{x} = f(x)$ ← e.g. RK4

else ($\phi = 0$)

$x' = g(x)$ ← jump map simulates non-smooth impact

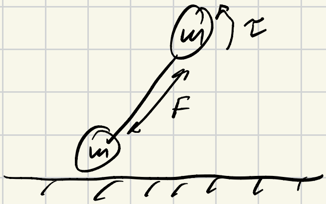
end

end

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

* Hybrid Traj Opt for Legged Robots:

- One-legged hopper:



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

- Define jump map to transition between modes:

$$x' = g_{21}(x) = \begin{bmatrix} r_b \\ r_f \\ v_b \\ 0 \end{bmatrix}$$ (impact)

← zero out foot velocity at impact

$$x' = g_{12}(x) = x$$ for this problem (lift-off)

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

for $k = 1 : N_1$

$$x_{n+1} = f_1(x_n, u_n)$$
$$\phi(x_n) = 0$$

end

for $k = (N_1 + 1) : N_2$

$$x_{n+1} = f_2(x_n, u_n)$$

$$\phi(x_n) \geq 0$$

end

$$x_{N_2} = g_{21}(x_{N_2-1})$$

$$\phi(x_{N_2}) = 0$$

(repeate) ...