Last Time:

- Convex Optimization Overview
- Convex MPC

Today:

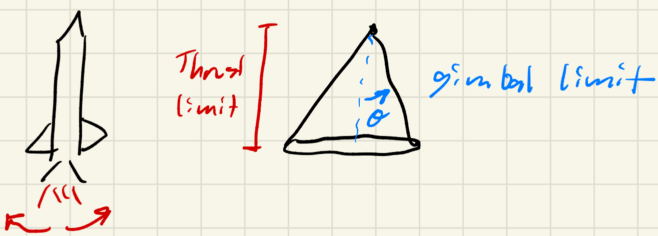
- Convex MPC Examples
- Nonlinear Trajectory Optimization
- Differential Dynamic Programming

---

* Other MPC Examples

  - Rocket Landing

    - Thrust vector constraints are conic:



  - SpaceX + JPL Mars landing use SOCP-based MPC with linearized dynamics.

- Legged Robots

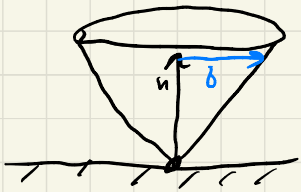
  - Contact forces must obey "friction cone" constraint

$$\|b\|_2 \le \mu n$$

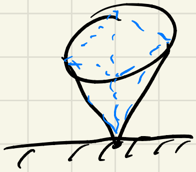
normal force $\in \mathbb{R}$

friction force $\in \mathbb{R}^2$

friction coefficient $\in \mathbb{R}$



- Cone is often approximated as a pyramid so the constraint is linear ($Ax \le b$)



- MIT Cheetah and other quadrupeds use QP-based MPC with linearized dynamics.

* What about Nonlinear Dynamics?

  - Linear stuff often works well, so use it if you can

  - Nonlinear dynamics make the MPC problem non-convex
    $\Rightarrow$ no convergence guarantees

  - Can work well in practice with effort

* Nonlinear Traj Opt Problem:

$$\min_{\substack{x_{1:N}\\u_{1:N-1}}} \quad J = \underbrace{\sum_{n=1}^{N-1} l_n(x_n, u_n) + l_N(x_N)}_{\text{non-convex cost}}$$

$$\text{s.t.} \quad x_{n+1} = f(x_n, u_n) \leftarrow \text{nonlinear dynamics}$$

$$\left.\begin{aligned} x_n &\in \mathcal{X}_n \\ u_n &\in \mathcal{U}_n \end{aligned}\right\} \text{non-convex constraints}$$

- Usually assume costs and constraints are C^2 (continuous 2nd derivatives)

---

* Differential Dynamic Programming (DDP)

- Nonlinear traj opt method based on approximate DP
- Use 2nd-order Taylor approximations of cost-to-go in DP and compute Newton steps.
- Very fast convergence possible

- cost-to-go expansion:

$$V_n(x+\Delta x) \approx V_n(x) + \underbrace{p_n^T}_{\text{gradient}} \Delta x + \frac{1}{2} \Delta x^T \underbrace{P_n}_{\text{Hessian}} \Delta x$$

$$p_N = \nabla_x l_N(x), \quad P_N = \nabla_{xx}^2 l_N(x)$$

- Action-Value function expansion :

$$S_n(x+\Delta x, u+\Delta u) \approx S_n(x,u) + \begin{bmatrix} g_x \\ g_u \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix}^T \begin{bmatrix} G_{xx} & G_{xu} \\ G_{ux} & G_{uu} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix}$$

$(G_{ux} = G_{xu}^T)$

- Bellman Equation

$$V_{N-1}(x) = \min_{\Delta u} \Big[S_{N-1}(x,u) + g_x^T \Delta x + g_u^T \Delta u + \frac{1}{2}\Delta x^T G_{xx} \Delta x + \frac{1}{2}\Delta u^T G_{uu} \Delta u + \frac{1}{2}\Delta x^T G_{xu} \Delta u + \frac{1}{2}\Delta u^T G_{ux} \Delta x \Big]$$

$$\nabla_{\Delta u}[\quad] = g_u + G_{uu}\Delta u + G_{ux}\Delta x = 0$$

$$\Rightarrow \boxed{\Delta u_{N-1} = -\underbrace{G_{uu}^{-1} g_u}_{d_{N-1}} - \underbrace{G_{uu}^{-1} G_{ux}}_{K_{N-1}} \Delta x = \underbrace{-d_{N-1}}_{\text{"Feed Forward term"}} - \underbrace{K_{N-1}\Delta x}_{\text{"Feedback term"}}}$$

- Plug δu back into S_{N-1} to get $V_{N-1}(x+\delta x)$

$$\Rightarrow V_{N-1}(x+\delta x) \approx V_{N-1}(x) + g_x^T \delta x + g_u^T(-d_{N-1} - K_{N-1}\delta x)$$
$$+ \tfrac{1}{2}\delta x^T G_{xx}\delta x + (d_{N-1} + K_{N-1}\delta x)^T G_{uu}(d_{N-1} + K_{N-1}\delta x)$$
$$- \tfrac{1}{2}\delta x^T G_{xu}(d_{N-1} + K_{N-1}\delta x) - \tfrac{1}{2}(d_{N-1} + K_{N-1}\delta x)^T G_{ux}\delta x$$

$$\Downarrow$$

$$\boxed{\begin{aligned} P_{N-1} &= G_{xx} + K_{N-1}^T G_{uu} K_{N-1} - G_{xu} K_{N-1} - K_{N-1}^T G_{ux} \\ p_{N-1} &= g_x - K_{N-1}^T g_u + K_{N-1}^T G_{uu} d_{N-1} - G_{xu} d_{N-1} \end{aligned}}$$

- Need some more math to calculate g and G

---

* Matrix Calculus:

  - given $f(x): \mathbb{R}^n \to \mathbb{R}^m$, look at 2nd order Taylor expansion:

  - if $m = 1$:

$$f(x+\delta x) \approx f(x) + \underbrace{\frac{\partial f}{\partial x}}_{\mathbb{R}^{1\times n}}\delta x + \frac{1}{2}\delta x^T \underbrace{\frac{\partial^2 f}{\partial x^2}}_{\mathbb{R}^{n\times n}}\delta x$$

  - if $m > 1$:

$$f(x+\delta x) \approx f(x) + \underbrace{\frac{\partial f}{\partial x}}_{\mathbb{R}^{m\times n}}\delta x + \underbrace{\frac{1}{2}\left(\frac{\partial}{\partial x}\left[\frac{\partial f}{\partial x}\delta x\right]\right)\Delta x}_{\text{ugly!}}$$

- for $m > 1$, $\frac{\partial^2 f}{\partial x^2}$ is a 3^{rd}-rank tensor. Think of this as a "3D matrix". We need some tricks to keep track of which dimensions we're multiplying along.

- Kronnecker Product:

$$\underbrace{A}_{l \times m} \otimes \underbrace{B}_{n \times p} = \underbrace{\begin{bmatrix} a_{11} B & a_{12} B & \cdots \\ a_{21} B & a_{22} & \cdots \\ \vdots & & \ddots \end{bmatrix}}_{ln \times mp}$$

- Vectorization Operator

$$\underbrace{A}_{l \times m} = [A_1 \quad A_2 \quad A_3 \cdots A_m]$$

column vectors

$$\text{vec}(A) = \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix}}_{lm \times 1}$$

- The "vec trick"

$$\text{vec}(ABC) = (C^T \otimes A)\, \text{vec}(B)$$

$$\Rightarrow \text{vec}(AB) = (B^T \otimes I)\, \text{vec}(A) = (I \otimes A)\, \text{vec}(B)$$

- If we want to diff a matrix w.r.t. a vector, vectorize the matrix:

$$\frac{\partial A(x)}{\partial x} = \underbrace{\frac{\partial\, vec(A)}{\partial x}}_{lm \times n}$$

(implied whenever we diff a matrix)

- Back to Taylor expansion of $f(x)$:

$$f(x+\Delta x) \approx f(x) + \underbrace{\frac{\partial f}{\partial x}}_{A} \Delta x + \frac{1}{2}(\Delta x^T \otimes I)\underbrace{\frac{\partial^2 f}{\partial x^2}}_{\frac{\partial vec(A)}{\partial x}} \Delta x$$

$$\frac{\partial}{\partial x}\left[vec(I A(x) \Delta x)\right] =$$

- Sometimes we need to diff through a transpose:

$$\frac{\partial}{\partial x}\left(A(x)^T B\right) = (B^T \otimes I)\underbrace{T}_{\text{"commutator matrix"}} \frac{\partial A}{\partial x}$$

$$T\, vec(A) = vec(A^T)$$

- Action-Value Function Derivatives:

$$S_n(x,u) = l_n(x,u) + V_{n+1}(f(x,u))$$

$$\Rightarrow \frac{\partial S}{\partial x} = \frac{\partial l}{\partial x} + \frac{\partial V}{\partial f}\underbrace{\frac{\partial f}{\partial x}}_{A} \Rightarrow \boxed{g_x = \nabla_x l + A_n^T p_{n+1}}$$

$$\frac{\partial S}{\partial u} = \frac{\partial l}{\partial u} + \frac{\partial V}{\partial f}\underbrace{\frac{\partial f}{\partial u}}_{B} \Rightarrow \boxed{g_u = \nabla_u l + B_n^T p_{n+1}}$$

$$\boxed{\begin{aligned}
G_{xx} &= \frac{\partial g_x}{\partial x} = \nabla^2_{xx} l(x,u) + A_n^T \nabla^2 V_{n+1} A_n + (p_{n+1}^T \otimes I)T \frac{\partial A_n}{\partial x} \\
G_{uu} &= \frac{\partial g_u}{\partial u} = \nabla^2_{uu} l(x,u) + B_n^T \nabla^2 V_{n+1} B_n + (p_{n+1}^T \otimes I)T \frac{\partial B_n}{\partial u} \\
G_{xu} &= \frac{\partial g_x}{\partial u} = \nabla^2_{xu} l(x,u) + A_n^T \nabla^2 V_{n+1} B_n + (p_{n+1}^T \otimes I)T \frac{\partial A_n}{\partial u}
\end{aligned}}$$

"Tensor Terms"