16-745 Optimal Control Lecture 11

Reid Graves

February 20, 2025

1 Last Time

- Convex Optimization Overview
- Convex MPC

2 Today

- Nonlinear Trajectory Optimization
- Differential Dynamic Programming (DDP/Iterative LQR)

3 What About Nonlinear Dynamics?

- Linear stuff often works well, so use it if you can
- Nonlinear dynamics makes MPC problem non-convex
  $\Rightarrow$ No convergence (Optimality guarantees)
- Can work well in practice with effort

4 Nonlinear Trajectory Optimization Problem

$$\min_{x_{1:N}, u_{1:N-1}} \quad J = \sum_{k=1}^{N-1} l_k(x_k, u_k) + l_N(x_N)$$

$$l_k(x_k, u_k) + l_N(x_N) \quad \text{non-convex cost}$$

$$s.t. \quad x_k + 1 = f(x_k, u_k) \quad \leftarrow \text{ nonlinear}$$

$$x_k \in \mathcal{X}_k, \quad u_k \in \mathcal{U}_k \quad \text{non-convex constraints}$$

- Assume costs and constraints are C^2 (Continuous 2nd derivatives)

5 Differential Dynamic Programming (DDP)

- Nonlinear trajectory optimization method based on approximate DP

- Use a 2nd order Taylor expansion of cost to go in DP to compute Newton steps on non linear problem
- Very fast convergence is possible
- Can stop early in real-time application

5.1 Cost-to-go Expansion

$$V_k(x+\Delta x) \approx V_k(x) + p_k^T \Delta x + \frac{1}{2}\Delta x^T P_k \Delta x$$
$$p_N = \nabla_x l_N(x): \text{ Gradient}, \quad P_N = \nabla^2_{xx} l_N(x): \text{ Hessian}$$

5.2 Action-Value Function Expansion

$$S_k(x+\Delta x, u+\Delta u) \approx S_k(x,u) + \begin{bmatrix} g_x \\ g_u \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix}^T \begin{bmatrix} G_{xx} & G_{xu} \\ G_{ux} & G_{u,u} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix}$$
$$(G_{ux} = G_{xu}^T)$$
$$g = \text{ gradient terms} \quad G = \text{Hessian Terms}$$

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

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

$$\Rightarrow \Delta u_{k-1} = -G_{uu}^{-1}g_u - G_{uu}^{-1}G_{ux}\Delta x \quad (G_{uu}^{-1}g_u = d_{k-1}, \text{ Feed-forward term}), \quad (G_{uu}^{-1}G_{ux} = K_{k-1}, \text{ Feedback term})$$
$$= -d_{k-1} - K_{k-1}\Delta x$$

- Plug back into S_{k-1} to get $V_{k-1}(x+\Delta x)$:

$$\begin{aligned}\Rightarrow V_{k+1}(x+\Delta x) \approx V_{k-1}(x) + g_x^T\Delta x + g_u^T(-d_{k-1} - K_{k-1}\Delta x) + \frac{1}{2}\Delta x^T G_{xx}\Delta x \\ + \frac{1}{2}(d_{k-1} + K_{k-1}\Delta x)^T G_{uu}(d_{k-1} + K_{k-1}\Delta x) \\ - \frac{1}{2}\Delta x^T G_{xu}(d_{k-1} + K_{k-1}\Delta x) - \frac{1}{2}(d_{k-1} + K_{k-1}\Delta x)^T G_{ux}\Delta x\end{aligned}$$

So now we have:

$$\boxed{P_{k-1} = G_{xx} + K_{k-1}^T G_{uu} K_{k-1} - G_{xu}K_{k-1} - K_{k-1}^T G_{ux}}$$
$$\boxed{p_{k-1} = g_x - K_{k-1}g_u + K_{k-1}^T G_{uu} d_{k-1} - G_{xu}d_{k-1}}$$

5.3 Matrix Calculus

- Given $f(x): \mathbb{R}^n \to \mathbb{R}^m$, look at 2nd order Taylor expansion:
- If $m = 1$

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

- If $m > 1$

$$f(x+\Delta x) \approx f(x) + \frac{\partial f}{\partial x}\Delta x + \frac{1}{2}\left(\frac{\partial}{\partial x}\left[\frac{\partial f}{\partial x}\Delta x\right]\right)\Delta x$$
$$\frac{\partial f}{\partial x}\Delta x \in \mathbb{R}^{m\times n}$$

- for $m > 1$ $\frac{\partial^2 f}{\partial x^2}$ is a 3rd rank tensor. Think of "3D matrix" We need some tricks to deal with these.

- Kronnecker Product:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & \\ a_{21}B & a_{22}B & \dots & \\ \vdots & & \ddots & \end{bmatrix}$$
$$A \in \mathbb{R}^{l\times m} \quad B \in \mathbb{R}^{n\times p} \quad A\otimes B \in \mathbb{R}^{ln\times mp}$$

- Vectorization Operator

$$A = \begin{bmatrix} A_1 & A_2 & A_3 & \dots & A_m \end{bmatrix}$$
$$vec(A) = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_m \end{bmatrix} \in \mathbb{R}^{lm\times 1}$$

- The "vec trick"

$$vec(ABC) = (C^T \otimes A)vec(B)$$
$$\Rightarrow vec(AB) = (B^T \otimes I)vec(A) = (I \otimes A)vec(B)$$

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

$$\frac{\partial A(x)}{\partial x} = \frac{\partial vec(A)}{\partial x} \in \mathbb{R}^{lm\times n} \quad \text{implied whenever we diff a matrix}$$

- Taylor Expansion of $f(x)$:

$$f(x+\Delta x) \approx f(x) + \frac{\partial f}{\partial x}\Delta x + \frac{1}{2}(\Delta x^T \otimes I)\frac{\partial^2 f}{\partial x^2}\Delta x$$
$$\frac{\partial f}{\partial x} = A = \frac{\partial}{\partial x}[vec(IA(x)\Delta x)]$$
$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial vec(A)}{\partial x}$$

- Sometimes we need to diff through a transpose:

$$\frac{\partial}{\partial x}(A(x)^T B) = (B^T \otimes I)T\frac{\partial A}{\partial x}$$
$$Tvec(A) = vec(A^T) \quad T: \text{ Commutater Matrix}$$

5.4 Action-Value Function Derivatives

$$
\begin{aligned}
S_k(x,u) &= l_k(x,u) + V_{k+1}(f(x,u)) \\
\Rightarrow \frac{\partial S}{\partial x} &= \frac{\partial l}{\partial x} + \frac{\partial V}{\partial f}\frac{\partial f}{\partial x} \Rightarrow \boxed{g_x = \nabla_x l_k + A_k^T p_{k+1}} \quad \frac{\partial f}{\partial x} = A \\
\frac{\partial S}{\partial u} &= \frac{\partial l}{\partial u} + \frac{\partial V}{\partial f}\frac{\partial f}{\partial u} \Rightarrow \boxed{g_u = \nabla_u l_k + B_k^T p_{k+1}} \quad \frac{\partial f}{\partial u} = B \\
G_{xx} &= \frac{\partial g_x}{\partial x} = \nabla^2_{xx} l(x,u) + A_k^T P_{k+1} A_k + (P_{k+1} \otimes I)^T T \frac{\partial A_k}{\partial x} \\
G_{uu} &= \frac{\partial g_u}{\partial u} = \nabla^2_{uu} l(x,u) + B_k^T P_{k+1} B_k + (P_{k+1} \otimes I)^T T \frac{\partial B_k}{\partial u} \\
G_{xu} &= \frac{\partial g_x}{\partial u} = \nabla^2_{xu} l(x,u) + A_k^T P_{k+1} B_k + (p_{k+1} \otimes I)^T T \frac{\partial A_k}{\partial x}
\end{aligned}
$$

Call these the tensor terms:

$$
\begin{gathered}
(P_{k+1} \otimes I)^T T \frac{\partial A_k}{\partial x} \\
(P_{k+1} \otimes I)^T T \frac{\partial B_k}{\partial u} \\
(p_{k+1} \otimes I)^T T \frac{\partial A_k}{\partial x}
\end{gathered}
$$