Last Time:

- Convex MPC Examples
- Nonlinear Traj Opt
- DDP/iLQR

Today:

- DDP details + extensions
- Constraints
- Free/minimum time problems

---

* DDP Recap:

- Solve the unconstrained Traj opt problem:

$$\min_{\substack{x_{1:N} \\ u_{1:N-1}}} \quad J = \sum_{n=1}^{N-1} l(x_n, u_n) + l_N(x_N)$$

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

- Backward Pass:

$$V_n(x+\Delta x) \approx V(x) + p_n^T \Delta x + \frac{1}{2}\Delta x^T P_n \Delta x$$

$$P_N = \nabla^2 l_N(x) \quad , \quad p_N = \nabla l_N(x)$$

$$V_{N-1}(x+\Delta x) = \min_{\Delta u} S(x+\Delta x, u+\Delta u)$$

$$\Rightarrow \boxed{\begin{aligned} \Delta u_{n-1} &= -d_{n-1} - K_{n-1}\Delta x_{n-1} \\ P_{n-1} &= G_{xx} + K^T G_{uu} K - G_{xu} K - K^T G_{ux} \\ p_{n-1} &= g_x - K^T g_u + K^T G_{uu} d - G_{xu} d \end{aligned}}$$

- Forward Rollout

$\Delta J = 0$

$x_1' = x_1$

for $k = 1 : N-1$

$u_k' = u_k - \alpha d_k - K_k (x_k' - x_k)$

$x_{k+1}' = f(x_k', u_k'$

$\Delta J \leftarrow \Delta J + \alpha g_{u_k}^T d_k$

end

- Line Search :

$\alpha = 1$

do :

$x', u', \Delta J = \text{rollout}(x, u, d, K, \alpha)$

$\alpha \leftarrow c\alpha$

while $J(x', u') > J(x, u) - b\Delta J)$

$x, u \leftarrow x', u'$

- Repeat until $\|d\|_\infty < tol$

Armijo parameters : $c \sim 1/2$, $b \sim .0001 - .01$

* Example:

  - Cartpole + acrobot swing up

  - DDP can converge in fewer iterations but iterations are more expensive. iLQR often wins in wall-clock time

  - Problem is nonconvex $\Rightarrow$ can land in different local optima depending on initial guess.

* Regularization

  - Just like standard Newton, $V(x)$ and/or $S(x,u)$ Hessians can become indefinite in backward pass

  - Definitely necessary for DDP, often a good idea with iLQR as well.

  - Many options for regularizing:

    * Add a multiple of identity to $\nabla^2 S(x,u)$, just like standard Newton.

    * Regularize P_n or G_n as needed in the backward pass.

    * Regularize just $G_{uu} = \nabla^2_{uu} S(x,u)$ (this is the only matrix you have to invert):

$$d = G_{uu}^{-1} g_u \quad , \quad K = G_{uu}^{-1} G_{ux}$$

  - This last one is a good choice for iLQR but not DDP.

- Regularization should not be required for iLQR but can be necessary due to floating point error.

* DDP Notes:

+ Can be very fast (itterations + wall-clock)

+ One of the most efficient methods due to exploitation of DP structure

+ Always dynamically feasible due to forward rollout ⇒ can always execute on robot

+ Comes with TVLQR tracking controller for free.

  ⇒ can be very effective for online use

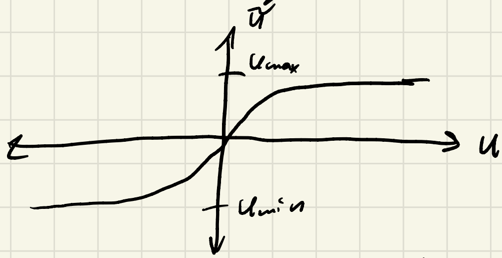
- Does not natively handle constraints

- Does not support infeasible initial guess for state trajectory due to forward rollout. Bad "maze" "bug trap" problems

- Can suffer from numerical ill-conditioning on long trajectories.

* Handling Constraints in DDP

- Many options depending on type of constraint
- Torque limits are often handled with a "squashing function" e.g. $\tanh(u)$:



$$\tilde{u} = u_{max} \tanh\left(u / u_{max} \right)$$

- Effective, but adds nonlinearity and may need more iterations.
- Better option: solve box-constrained QP in the backward pass:

$$\delta u = \underset{\delta u}{\operatorname{argmin}} \; S(x+\delta x, u+\delta u)$$

$$\text{s.t.} \quad u_{min} \le u + \delta u \le u_{max}$$

- State constraints are harder. Often penalties are added to cost function. Can cause ill-conditioning
- Better option: Wrap entire DDP algorithm in an Augmented Lagrangian method.
- AL method adds linear (multiplier) and quadratic (penalty) terms to the cost $\Rightarrow$ fits DDP nicely.