3/17: Constraints & MPC

Announcements
- midterm survey due today (3/17)
- project milestone due Thurs. (3/19)
- HW3 due 3/26

Last time:
- iLQR

Today:
- constrained traj. opt (SCP)
- model-predictive control (MPC)
- HW2 winners!

Recall: iLQR

optimal control: $\min_{x_{0:T}, u_{0:T-1}} \; c_T(x_T) + \sum_{t=0}^{T-1} c_t(x_t, u_t)$

s.t. $x_{t+1} = f(x_t, u_t)$

$x_0 = x(0)$

× $x_t \in \mathcal{X} = \mathbb{R}^n$

× $u_t \in \mathcal{U} = \mathbb{R}^m$

Key idea: leveraging LQR for nonlinear systems
if we have no constraints, we can approximate

$A_t = \frac{\partial f}{\partial x}\Big|_{\bar{x}_t, \bar{u}_t}$ $\quad B_t = \frac{\partial f}{\partial u}$ $\quad c_t \approx \begin{bmatrix} q_t \\ r_t \end{bmatrix}^T \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \frac{1}{2}\begin{bmatrix} x_t \\ u_t \end{bmatrix}^T \begin{bmatrix} Q_t & S_t \\ S_t & R_t \end{bmatrix}\begin{bmatrix} x_t \\ u_t \end{bmatrix}$

linear dyn. — quadratic cost

Issues w/ iLQR:
1. Linearizations can be ... bad
2. constraints are hard

Ways to resolve constraints:

1. use penalties – penalty methods ρ const, $\rho \to \infty$
   AL (ALTRO)
2. sequential convex programming (SCP)
   Just like iLQR we're going appx. the problem about a current guess $\bar{x}_{0:T}, \bar{u}_{0:T-1}$

$$c_t(x_t, u_t) \approx c_t(\bar{x}_t, \bar{u}_t) + \begin{bmatrix} q_t \\ r_t \end{bmatrix}^T \begin{bmatrix} x_t - \bar{x}_t \\ u_t - \bar{u}_t \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \; \end{bmatrix} \begin{bmatrix} Q_t & S_t \\ S_t^T & R_t \end{bmatrix}\begin{bmatrix} \; \end{bmatrix}$$

$q_t = \nabla_x c_t \quad Q_t = \nabla^2_{xx} c_t \quad S_t = \nabla^2_{xu} c_t$

$r_t = \nabla_u c_t \quad R_t = \nabla^2_{uu} c_t$

$$x_{t+1} \hat{=} A_t(x_t - \bar{x}_t) + B_t(u_t - \bar{u}_t) + f(\bar{x}_t, \bar{u}_t)$$

$$g(x_t) \le 0 \Rightarrow g(\bar{x}_t) + \frac{\partial g}{\partial x}(x_t - \bar{x}_t) \le 0$$

$$\Rightarrow G_t x_t + h_t \le 0$$

$z = [x_0, x_1, \dots, u_0, u_1, \dots]$

$\min_z \; \frac{1}{2} z^T P z + q^T z$

s.t. $Az = b$

$Gz \le h$

Key idea: solve a sequence of these problems

$(\bar{x}, \bar{u})^k \to \dots x^*, u^*$

Alg1 (sequential convex programming)
input initial guess $\bar{x}_{0:T}, \bar{u}_{0:T-1}$
repeat until convergence:
 linearize dynamics, all non-convex constraints
 "quadraticize" cost, all about $\bar{x}, \bar{u}$
 $\bar{x}, \bar{u} \leftarrow$ solve-subproblem ( )

---

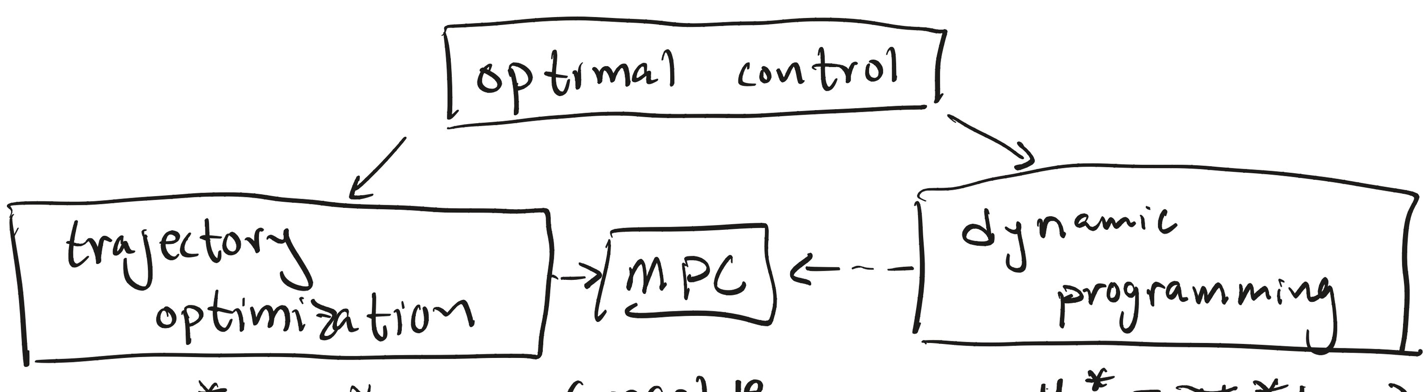
iLQR	SCP
+ faster (Riccati)	– slower (usually)
+ gives feedback $u_t = \bar{u}_t - K_t(x_t - \bar{x}_t) - k_t$	– no feedback (can appx. w/ iLQR pass)
– very sensitive to initial guess ($\bar{x}_t = f(\bar{x}_{t-1}, u_{t-1})$) uses rollouts	+ less sensitive to init., can init. w/ $\bar{x}$ not satisfying the dynamics
	++ can incorporate constraints
+ dynamically feasible	– not dynamically feasible

SNOPT

$\tilde{x}_t = 0.1 \cdot x_t$

$x_t \le 1$ ✓

$x_t \le 0.5$ ✓

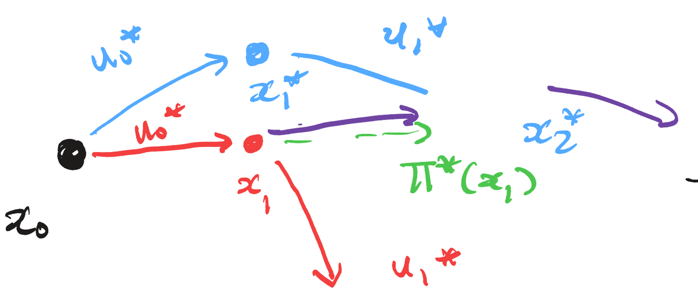


$x^*_{0:T}, u^*_{0:T-1}$ (open-loop) — cheap, but brittle

(resolve traj. opt online)

$u^*_t = \pi^*(x_t)$ (closed-loop) — expensive, but robust

Recall: compounding error

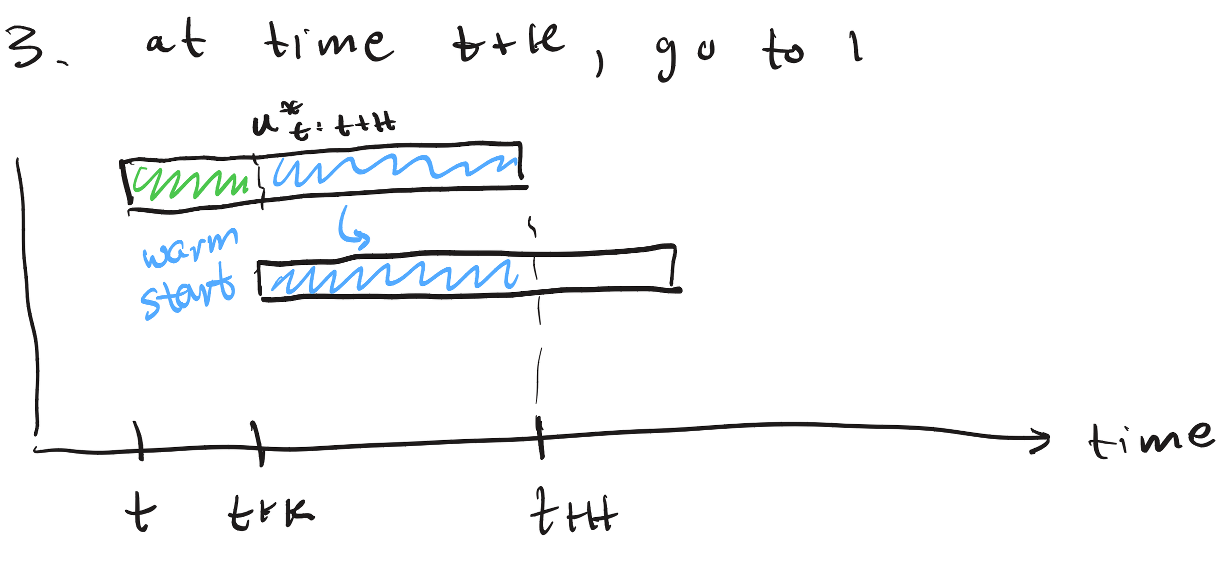


MPC: recompute x^*, u^* @ each timestep

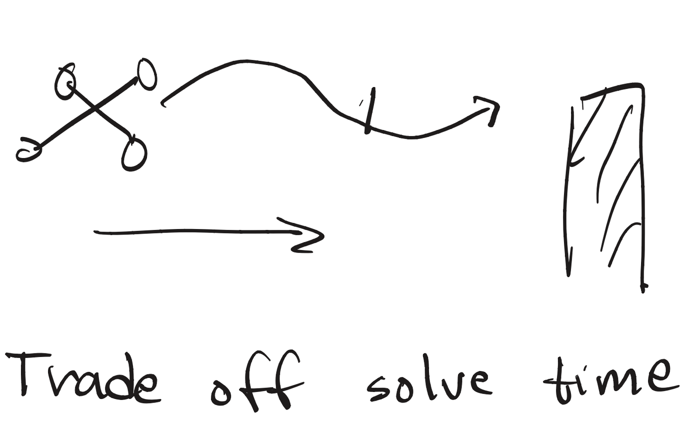
Two approximations:
i) truncate our horizon
ii) known/deterministic dynamics

Fixing these w/ fast replanning

Key steps:
1. @ time t, solve finite-horizon OC problem for $t: t+H$, from current state x_t, w/ initial guess shift(u^*_{prev}) "warm start"
2. apply solution $u^*_{t:t+H}$ for K timesteps
3. at time $t+k$, go to 1



Key design decisions:
1. Terminal cost/constraints



Think carefully about "recursive feasibility"

2. Trade off solve time/solution quality