Today: Trajectory Optimization

Big Picture

- Dynamic Programming
  - mesh (low dimensions)
  - function approx
- LQR
  - scales very well.
  - linearization only valid over a limited domain
- Lyapunov via SDP / SOS
  - $\forall x_i \Rightarrow \forall x$
  - 10+ dimensions
  - Proving stability so far, not synthesis
  - Somehow limited to simple functions. / might limit control design.
- Won't visit all x

Other extreme fundamental attack on dimension is to consider a single $x_0 = x(0)$ instead of a global policy

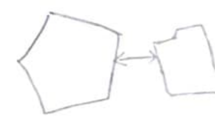
double integrator

Basic trajectory optimization formulation

$$\dot{x} = f(x, u)$$

$$\min_{u(\cdot)} \int_{t_0}^{t_f} g(x, u)\, dt$$

s.t. $\dot{x} = f(x, u)$



$x(0) = x_0$

\+ other constraints $\forall t \ |u(t)| \le 1$

$\forall t \ \mathrm{sdf}(x(t)) \ge 0$

no collisions.

Discrete-time case:

$$x[n+1] = Ax[n] + Bu[n]$$

Direct Transcription

$$\min_{u[\cdot]} \sum_{n=0}^{N} g(x[n], u[n]) \rightarrow \sum_{n=0}^{N} x^T[n] Q x[n] + u^T[n] R u[n]$$

$\Downarrow$

Quadratic Program

$Ax[n] + Bu[n]$

s.t. $x[n+1] = \cancel{A[\cdot}\ .\ \cancel{Ax[n] + B u[n]}$

$x[0] = x_0$

Transcribe to numerical implementation

idea #1: add $x[\cdot]$ as extra decision variables (~~stat~~ slack variables)

Can add linear constraints on u. x

$\forall n \; |u| \le 1 \qquad \forall n \; -2 \le x[n] \le 2$

idea # 2: solve $x[n]$ as func of $x_0, u[n], u[1] \ldots u[n-1]$.

Direct Shooting Method

$X[0] = X_0, \; X[1] = AX_0 + Bu[0],$

$X[2] = A(AX_0 + Bu[0]) + Bu[1]$.

Sparse constraints, dense constraints

add $x[\cdot]$ may speed up solvers

Quadratic cost is fine

linear is also fine

$$\min_{x[\cdot], u[\cdot]} \sum |x|_2 + |u|_2$$ another objective

min time $(x[n+1]$

find $x[\cdot]\; u[\cdot]$ s.t. $x[n+1] = Ax[n] + Bu[n]$

$X[0] = X_0$

$X[N] = X_G$

feasible problem, find minimum N that

a solution exists, larger than minimum time

a solution can be found

Back to continuous time

$$\dot{x} = Ax + Bu$$

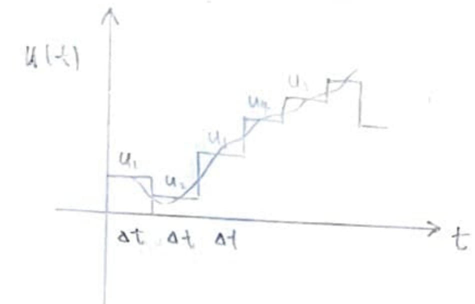
$$x[n+1] = x[n] + \Delta t\,[Ax[n] + Bu[n]]$$

Forward - Euler integration

close form exists $e^{A\Delta t}$

MATLAB ode45

$$x[n+1] = x[n] + dt\, f(x,u)$$ or Runge - Kutta, ...



model - predictive control

(solve traj opt at every timestep)

Non-linear system

$$\min_{u[\cdot]} \sum x^T Q x + u^T R u.$$

s.t. $x[n+1] = f(x[n], u[n])$

$x[0] = x_0$

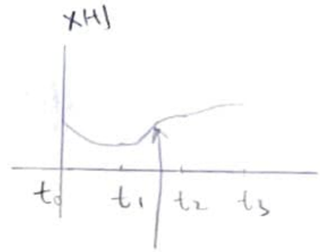
nonlinear system may have non-convex optimization problem

Continuous time ~~policy~~ control

$$\min_{x(\cdot),\, u(\cdot)} \quad \ldots$$

sweet spot # of params vs. numerical integration accuracy

Direct Collocation



$x(t)$ as a cubic spline

$u(t)$ is first-order spline

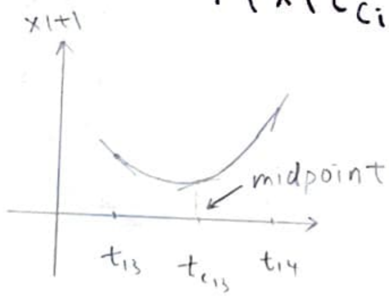
(FOH)

$x(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$

$\forall t_i \quad \dot{x}(t_i) = f(x(t_i), u(t_i))$

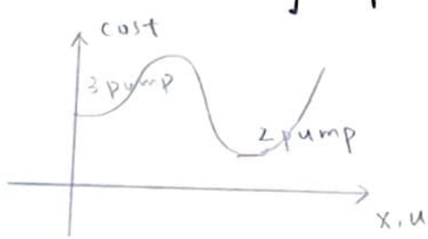
also at "collocation pts"

$$\dot{x}(t_{c_i}) = f(x(t_{c_i}), u(t_{c_i}))$$



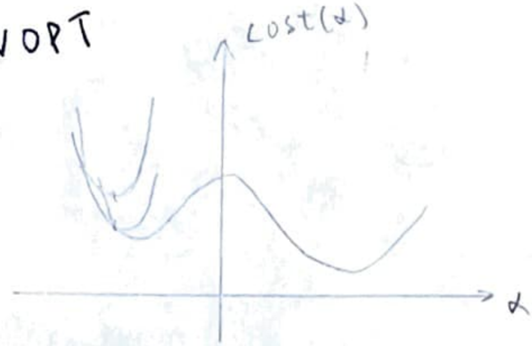
different from integration differential equation

pendulum swing up case



only have local minimum

SNOPT



Approximate using quadratic equation

Sequential

Quadratic Programming (QP)

Not guaranteed to work ?

not good in walking planning

~~Quadrator~~ Q

Quadrotor planning

non - convex optimization