Lecture 15 : Trajectory Optimization, Sequential Quadratic Programming, Convex - Concave Procedure

Goals :
- Intro to trajectory optimization (i.e. open-loop optimal control)
- Discuss its tractability
- Discuss two broad approximation techniques for non-convex problems based on convex optimization
    - Sequential Quadratic Programming (SQP)
    - Convex - Concave Procedure (CCP)

Trajectory Optimization (continuous time)

Consider the problem

$$\begin{array}{ll} \text{minimize} & \int_0^T g(x(t), u(t), t)\, dt \\ \text{subject to} & \dot{x}(t) = f(x(t), u(t), t) \\ & x(0) = x_0 , \quad x(T) \in \mathcal{X}_T \\ & (x(t), u(t)) \in Z \quad \forall t \in [0, T] \end{array}$$

with variables $u : \mathbb{R}_+ \rightarrow \mathbb{R}^m$, $x : \mathbb{R}_+ \rightarrow \mathbb{R}^n$

problem data $g, f, x_0, \mathcal{X}_T, Z$

- open-loop optimal control (no feedback)
- infinite-dimensional (variables are <u>functions</u> of time)
- can handle directly via calculus of variations, Pontryagin's Minimum Principle (a fascinating subject, though mathematically demanding & of limited practical use)
- Some special classes of systems to compute continuous time traj. algebraically via convex opt.
  - Differentially flat systems
- In practice, approximate via time-discretization
  - Simplest: Euler integration

$$x(t) = x_0 + \int_0^t f(x(t), u(t), t)\, dt$$

$$x(t+1) \approx \underbrace{x(t) + f(x(t), u(t), t)\, \Delta t}_{\bar{f}(x(t), u(t), t)}$$

with $u(t)$ piecewise constant

  - Many more complicated & accurate methods using more evaluations or higher derivatives of f
    - e.g. Runge Kutta methods

Trajectory Optimization (discrete time)

Consider

$$\begin{aligned} \text{minimize} \quad & \sum_{t=0}^{T-1} g_t(x_t, u_t) + g_T(x_T) \\ \text{subject to} \quad & x_{t+1} = f_t(x_t, u_t) \quad t = 0, \dots, T-1 \\ & x_0 = x_0, \quad x_T \in \mathcal{X}_T \\ & (x_t, u_t) \in Z \quad \forall t \end{aligned}$$

with variables $u_0, \dots, u_{T-1}, x_1, \dots, x_T$

problem data $g_t, f_t, x_0, \mathcal{X}_T, Z$

- now finite dimensional
- convexity ?
  - need convexity of g_t, $\mathcal{X}_T$, Z
  - need dynamics f to be <u>affine/linear</u>!
  - unfortunately trajectory opt. for nonlinear dynamical systems is computationally difficult in general

"Solving" Trajectory Optimization for Nonlinear Systems

- must resort to heuristics that (hopefully) find locally optimal solutions fast

- Broad approaches:

  1. "Solve" w/ general nonlinear solver, cross fingers, + hope for the best
     - only local, may not even converge to a feasible solution

  2. Iterative convex approximation
     - Sequential Quadratic Programming (SQP)
     - Convex - Concave Procedure (CCP)

  3. Global optimization hacks
     - e.g. genetic algorithms, simulated annealing, etc.
     - no guarantees, may be extremely slow
     - use with care; only recommended as a last resort

Sequential Quadratic Programming

- general method to approx. solve non-convex problems, often used for traj. opt. of nonlinear systems
- Basic idea: iteratively approximate by a QP by linearizing/quadratizing objective + constraints
- related to iterative algorithms for solving nonlinear convex problems (interior point)
  - natural extension of Newton's method to constrained problems

Consider the (generally non-convex) problem

$$
\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & h(x) = 0 && \text{(dynamics in traj. opt.)} \\
& g(x) \leq 0 && \text{(possibly non-convex constraints on state/input traj. in traj. opt.)}
\end{aligned}
$$

- assume f, g, h sufficiently differentiable
- suppose we have some guess/approximation x_k of the solution to start with

Quadratic Approximation

$$\begin{aligned} \underset{d_k}{\text{minimize}} \quad & q_k^T d_k + d_k^T P_k d_k \\ \text{subject to} \quad & h(x_k) + \nabla h(x_k)^T d_k = 0 \\ & g(x_k) + \nabla g(x_k)^T d_k \leq 0 \end{aligned} \qquad \text{(QP)}$$

where $d_k = x - x_k$ is the variable

- simply linearize constraints around current guess/iterate x_k
- typically objective not just quadratic approx. of f (w/ $q_k = \nabla f(x_k)$ and $P_k = \nabla^2 f(x_k)$) but instead uses a quadratic approx. of Lagrangian function

$$L(x, \nu, \lambda) = f(x) + \nu^T h(x) + \lambda^T g(x)$$

$$\longrightarrow q_k = \nabla_x L(x_k, \nu_k, \lambda_k), \quad P_k = \nabla_x^2 L(x_k, \nu_k, \lambda_k)$$

based on some guess/approximation of multipliers ν_k, λ_k

- takes constraints into account in objective approx. (can justify from KKT conditions)
- sometimes must "regularize" hessian, which can fail to be positive definite, making QP non-convex
  - Levenberg-Marquardt

- may also maintain a "trust region" within which approximations believed to be sufficiently accurate, and include as constraint in QP ($x \in T_k$)

- next generate new iterate from QP solution: let $(x_k^*, \nu_k^*, \lambda_k^*)$ denote optimal primal + dual of QP

$d_k^* + x_k$ (let $d_\nu^* = \nu_k^* - \nu_k$, $d_\lambda^* = \lambda_k^* - \lambda_k$)

now set

$$(*) \quad \begin{aligned} x_{k+1} &= x_k + \alpha d_k^* \\ \nu_{k+1} &= \nu_k + \alpha d_\nu^* \\ \lambda_{k+1} &= \lambda_k + \alpha d_\lambda^* \end{aligned}$$

step size $\alpha > 0$

and repeat w/ new iterates

Basic SQP Algorithm

Input: Initial guess (x_0, ν_0, λ_0)
Set $k=0$

while (not converged)

① Form + solve QP to obtain $(x_k^*, \nu_k^*, \lambda_k^*)$

② Choose α so that $\phi(x_k + \alpha d_k^*) < \phi(x_k)$

- "merit function" ϕ measures combo of objective value and constraint violation

③ Update iterates as in $(*)$

④ Set $k = k+1$ and go to ①

Comments:

- heuristic method, MAY give good local solution (even near optimal), but may fail to find a feasible solution (or even fail to converge)

- often depends heavily on quality of initial guess, but converges rapidly near a solution

- widely used in practice, especially for nonlinear trajectory optimization

Ex Double pendulum state transfer from Stanford EE364b

Convex - Concave Procedure (CCP)

- more sophisticated way to approx. certain non-convex problems, similar in spirit to SQP

Consider the problem

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) \\ \text{subject to} & f_i(x) - g_i(x) \leq 0 \qquad i=1,\dots,m \end{array}$$

where f_i, g_i are convex functions

- i.e. objective + constraints are differences of convex functions

- Seems quite specific, but actually broad
  - e.g. any C^2 function can be written in this form

Ex Boolean LP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \le b \\ & x_i \in \{0,1\} \end{array} \iff \begin{array}{ll} \text{min.} & c^T x \\ \text{subject to} & x_i^2 - x_i \le 0 \quad \forall i \\ & x_i - x_i^2 \le 0 \\ & Ax \le b \end{array}$$

$x_i(1-x_i) = 0$

$x_i - x_i^2 = 0 \iff \begin{array}{l} x_i - x_i^2 \ge 0 \\ x_i - x_i^2 \le 0 \end{array}$

Basic CCP Algorithm

Input: initial feasible point x_0
set $k = 0$

while (not converged)

① Convexify: $\hat{g}_i(x, x_k) = g_i(x_k) + \nabla g_i(x_k)^T (x - x_k)$

- linearize only concave terms

② Solve convex problem

$$\begin{array}{ll} \text{minimize} & f_0(x) - \hat{g}_0(x) \\ \text{subject to} & f_i(x) - \hat{g}_i(x) \le 0 \qquad i = 1, \dots, m \end{array}$$

③ Set $k = k+1$, go to ①

Comments:

- compared to SQP, more info retained at each iteration (convex terms retained exactly)

- line search unnecessary, but can include to encourage faster progress

- enabled by recent advances in general convex optimization software (beyond QP)

Nice recent paper: Lipp + Boyd, Optimization + Engineering 2016