CS 287, Fall 2015 Problem Set #3
Convex Optimization, Sequential Convex Programming, Optimization-based Motion Planning and Control

---

Deliverable: pdf write-up by Wednesday October 7th, 11:59pm, submitted through Gradescope. **Your pdf write-up should be typeset as follows: answers to Question 1(a), 1(b), 1(c), and 1(d) an page 1; answer to Question 2 on pages 2 and 3; answer to Question 3 on page 4; answer to Question 4 on page 5. Make sure to follow the requested typesetting, and insert blank regions (or blank pages) as necessary. Thanks!**

Refer to the class webpage for the homework policy.

Various starter files are provided on the course website.

1. Feasible Newton Step Interior Point Methods

For these questions report the data requested in the starter file, and a paragraph or two about your implementation.

(a) **Gradients and Hessians.** Implement numerical computation of gradients and hessians.

(b) **Unconstrained.** Implement gradient descent and Newton's method with backtracking line search for unconstrained optimization.

(c) **Equality Constrained.** Implement the feasible Newton step method (method 2 from the slides) with backtracking line search for equality constrained optimization.

(d) **Equality and Inequality Constrained.** Implement the feasible Newton step method (method 2 from the slides) with backtracking line search for equality and inequality constrained optimization. Use the log-barrier method to handle the inequalities.

2. Sequential Quadratic Programming (SQP)

In this question you get to implement a generic SQP-based solver for non-convex optimization problems. See `penalty_sqp.m`, and look for `TODO` and `YOUR_CODE_HERE` for the parts you need to write. The script `test_penalty_sqp.m` contains a collection of test problems. You should check that your solver solves them to at least precision 10^{-3}. (Note that some of these problems have numerically challenging features.) As your solution write-up, just provide the solution path found by your algorithm, which is saved to a png file by the script.

3. Trajopt: Trajectory Optimization for Motion Planning

In this question you get to implement a trajectory optimization algorithm that uses signed distances to (try to) enforce that the trajectory is collision-free. In particular, it will solve the

following problem:

$$\min_{\boldsymbol{\theta}_{1:T}} \sum_t \frac{1}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2$$

subject to

$$\boldsymbol{\theta}_1 = start$$

$$\boldsymbol{\theta}_T = goal$$

$$|\boldsymbol{\theta}^j_{t+1} - \boldsymbol{\theta}^j_t| \leq .2 \quad \text{for } j \in \{1,2,3\}, t \in \{1,2,\ldots,T-1\}$$

$$\text{SignedDistance}(\text{Robot}, \text{Obstacle}) \geq d_{\text{safe}}$$

$$\forall \ \text{Obstacle} \in \text{Obstacles}$$

See `test_traj_opt.m` and look for `TODO` and `YOUR_CODE_HERE` for the parts you need to write. Note this questions relies on your solution to Question 2, and you'll need to make sure that your code for Question 2 is in the matlab path. Four sets of obstacles are provided: `obstacles{0,1,2,3,4}.mat`. The optimizer should successfully find solutions for 0, 1, and 2, and it will fail on 3 and 4 by getting stuck in a bad local optimum. As your solution write-up, just provide the paths found for each test case by saving the last figure from running `test_traj_opt.m`.

4. Nonlinear Optimization for Control and Time-Varying LQR for Nonlinear Systems

In this question you get to design a controller for helicopter aerobatics! The starter file provides a tentative trajectory for the helicopter to fly: for the first one second the helicopter is kept in place, for the next two seconds the helicopter is made to perform half of a forward flip, and for the last one second the helicopter is hovering upside down.

(a) **Nonlinear Optimization for Control.** The first step is to turn this crude guess of an aerobatic trajectory for the helicopter, which is physically not realizable for the helicopter, into a reasonable target trajectory to control around. Do so by solving the following optimization problem:

$$\begin{aligned}
\min_{x,u} \quad & \sum_{t=1}^{T} (x_t - \bar{x}_t)^\top Q (x_t - \bar{x}_t) + \sum_{t=1}^{T} (u_t - \bar{u}_t)^\top R (u_t - \bar{u}_t) \\
\text{s.t.} \quad & \forall t \in 1,\ldots,T: \quad x_{t+1} = f(x_t, u_t) \\
& x_0 = x_{\text{init}} \\
& \forall t \in 1,\ldots,t: \quad u_t \in \mathcal{U}_t
\end{aligned}$$

where $\bar{x}$ is the tentative trajectory provided to you. To solve this problem, use sequential convex programming. Start by looking in `q4_starter.m`. Report plots of the trajectory your SCP implementation found.

(b) **Time-Varying LQR for Nonlinear Systems.** The second step is to design the feedback controller to keep the helicopter on the target trajectory. First run the open loop sequence of controls you found from part (a). Report on whether it succeeds at executing the aerobatic trajectory accurately? Next, implement a finite-horizon, time-varying LQR controller that provides a sequence of linear feedback controllers that attempt to stay on the target trajectory. Report plots of the trajectories executed by this feedback controller.