CS 287, Fall 2012
Problem Set #2 : Nonlinear Optimization for Control, Time-Varying LQR for Nonlinear Systems, Feasible Newton Step Interior Point Methods

---

Deliverable: pdf write-up by Monday October 29th, 23:59pm to be emailed to pabbeel@cs.berkeley.edu, with subject: PS2. Some of the starter code will generate plots. Don't include all of them, include whatever is sufficient to show that you correctly solved the problem. Whenever including a figure, make sure it is accompanied by a discussion of that figure. Ball-park figure for number of pages: 6.

Refer to the class webpage for the homework policy.

Various starter files are provided on the course website.

You are free to use any convex programming solver of your choice for this assignment. If you don't have in-depth experience with a particular solver already, I recommend you try out CVX. CVX can solve a broad class of convex optimization problems, including linear programs, but also quadratic programs (which you will need for future assignments!).

1. 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 2 seconds the helicopter is kept in place, for the next 2 seconds the helicopter is made to perform half of a forward flip, and for the last 2 seconds 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:

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

where $\bar{x}$ is the tentative trajectory provided to you. To solve this problem, use sequential convex optimization by iteratively linearizing the dynamics model (f), and repeatedly solving problems of the type:

$$
\begin{array}{ll}
\min_{x,u,s} & \sum_{t=1}^{T}(x_t - \bar{x}_t)^\top Q(x_t - \bar{x}_t) + \sum_{t=1}^{T}(u_t - \bar{u}_t)^\top R(u_t - \bar{u}_t) + C\sum_{t=1}^{T} s_t^\top s_t \\
\text{s.t.} & \forall t \in 1,\ldots,T: \quad x_{t+1} = A_t(x_t - \hat{x}_t) + B_t(u_t - \hat{u}_t) + c_t + s_t \\
& \forall t \in 1,\ldots,T: \quad \|x_t - \hat{x}_t\|_2 \leq \varepsilon_x \\
& \forall t \in 1,\ldots,T: \quad \|u_t - \hat{u}_t\|_2 \leq \varepsilon_u \\
& \forall t \in 1,\ldots,t: \quad u_t \in \mathcal{U}_t
\end{array}
$$

where $\bar{x}, \bar{u}$ are the tentative target trajectory you are provided with, $\hat{x}, \hat{u}$ are the current points of linearization, which are taken from the solution of the previous iteration (i.e., use collocation). A couple of noteworthy additional variables that were introduced above: (i) $\varepsilon_x, \varepsilon_u$ define the trust regions within which the optimization is constrained such as to stay within a region where the linearization is a reasonable approximation, (ii) s_t are slack variables, they allow for the dynamics constraint to be violated, which can be necessary to make these intermediate, linearized problems feasible, especially for underactuated systems like helicopters, (iii) C scales the penalty on the slack variables deviation from zero and is in initially set to a small value, but gets increased in the next iteration. In my own solution, I used eight outer iterations, and had C take on the values $10^{k/2}$, with i being the outer iteration index.

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.

2. Feasible Newton Step Interior Point Methods

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

(a) **Unconstrained.** Implement Newton's method with backtracking line search for unconstrained optimization.

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

(c) **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.