CS8803: ACRL, Spring 2019: Homework 1

This assignment must be completed **individually**.

Due: Wednesday, Feb 13th, 3pm

1 The Problem

This problem is designed to help you better understand LQR by designing a controller for an unstable robot.[1] While the system is relatively mildly non-linear about hover, helicopter control and (especially) VSTOL aircraft control really drove the development of modern robust and optimal control techniques that handle *model uncertainty*, as these systems often are difficult to model well, have delays, and exhibit non-minimum phase behavior.

Figure 1: An autonomous helicopter at CMU running a robust controller.

[1] We thank Pieter Abbeel for the helicopter simulator, code, and problem suggestions.

2 System Description

You will work with a helicopter model that simulates an XCell Tempest helicopter in the flight regime close to hover. It uses the simplest dynamics that can be expressed in the frame of the vehicle. We use (n, e, d) to refer to North, East, Down (this is a common flight frame). The state vector is represented by position (n, e, d), orientation (quaternion q), velocity $(\dot{n}, \dot{e}, \dot{d})$, and angular rate ($(p, q, r)$, in the helicopter's coordinate frame). The standard controls for a helicopter are as follows:

- u1 and u2: The latitudinal (left-right) and longitudinal (front-back) cyclic pitch controls. They are also often referred to by elevator and aileron. They change the pitch angle of the main rotor throughout each cycle and can thereby cause the helicopter to pitch forward/backwards or to roll sideways. By pitching and rolling the helicopter, the pilot can affect the direction of the main thrust, and hence the direction in which the helicopter moves.

- u3: The tail rotor collective pitch control. This control is also often referred to by "rudder" and it affects the tail rotor thrust. It can be used to yaw (turn) the helicopter.

- u4: The main rotor collective pitch control, similarly to the cyclic controls, causes the main rotor blades to rotate along an axis that turns along the length of the rotor blades, and thereby affects the angle at which the main rotor's blades are tilted relative to the plane of rotation. As the main rotor blades sweep through the air, they generate upward thrust that increases with this angle.

By varying the collective pitch angle, we can affect the main rotor's thrust. Let Fx, Fy, Fz denote the forces working on the helicopter along the helicopters forward (x) axis, its sideways to the right (y) axis, and its downward (z) axis. Similarly, we let Tx, Ty, Tz denote the torques working on the helicopter along each of these axes. We let (u,v,w) denote the velocity of the helicopter along (x,y,z). Helicopter dynamics can be determined then by computing the forces and torques acting on the vehicle.

$$
\begin{aligned}
Fx &= Gx + C_x[u] && (1)\\
Fy &= Gy + C_y[1; v] && (2)\\
Fz &= Gz + C_z[1; w; u4] && (3)\\
Tx &= C_p[1; p; u1] && (4)\\
Ty &= C_q[1; q; u2] && (5)\\
Tz &= C_r[1; r; u3] && (6)
\end{aligned}
$$

Here (Gx,Gy,Gz) denotes gravity in the frame of the helicopter; C are vectors that parameterize the helicopter model. These parameters are instantiated in the code and have been determined by fitting them to flight data. From forces and torques we find linear and angular accelerations, which in turn we can integrate over time to obtain the helicopter's state.

2.1 Robustness

For control purposes, state is augmented with the past control inputs, and the past change in control inputs. Due to under-modeling of higher order dynamics, it is crucial not to produce higher frequency commands than necessary. In fact, without this, LQR techniques– as well as μ-synthesis and more advanced linear control strategies– have been tried many times leading to unstable or very low performance control [1, 3].

Interestingly, simple policy search methods and even hand-designed controllers using classical techniques perform more than adequately for hover. [2]. We will investigate these issues further in the assignment. The files:*heli.m* and *init_setup.m* contain the implementation and relevant comments.

3 Questions

Each question to be addressed is in the file *ACRL_HW1.m*. The initial questions are straightforward implementations of an open loop control and a closed-loop stabilizing controller. We then investigate the controller's performance in the presence of delay. The final questions ask you to design a controller for hovering this model by policy search on the non-linear model.

4 What to Turn in

Turn in a zip file containing your code (the modified *ACRL_HW1.m* file) and a report answering each question.

References

[1] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *In Advances in Neural Information Processing Systems 19*, page 2007. MIT Press, 2007.

[2] J. Andrew Bagnell and Jeff Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings of the International Conference on Robotics and Automation 2001*. IEEE, May 2001.

[3] William Messner Marco La Civita, G. Papageorgiou and Takeo Kanade. Design and flight testing of a gain-scheduled h-infinity loop shaping controller for wide-envelope flight of a robotic helicopter. In *Proceedings of the 2003 American Control Conference*, pages 4195 – 4200, June 2003.