AER1217: DEVELOPMENT OF UAS

Centre for Aerial Robotics Research and Education

January 15, 2018

- **Coordinator**
Professor H. H-T. Liu
Institute for Aerospace Studies (UTIAS)
Room 113, Tel: (416) 667-7928, Fax: (416) 667-7799
Email: liu@utias.utoronto.ca
- **Teaching Assistants**
Mr. Yih Tang Yeo, yihtang.yeo@mail.utoronto.ca
Mr. Joshua Furtado, joshua.furtado@mail.utoronto.ca

Curriculum

This graduate course is offered as part of CARRE program in UAVs. This course is the second part of CARRE core courses, followed by AER1216: Fundamentals of UAVs, which covers the fundamental principles related to UAV design. AER1216 is the prerequisite of this course, unless approved by the instructor. In this course, the focus is placed on the development of unmanned aerial systems (UAS), with the theme of autonomy in navigation and control, as well as flight performance analysis and evaluation.

- Course Materials
  - Course Presentation Handouts
  - available through U of T portal (blackboard)
- Lectures
  - Mondays, 9h00-12h00 @ Lecture Hall (UTIAS)
- Marking
  - Problem Sets (20%)
  - Laboratory (40%)
  - Project (40%)

Grading

- Problem Sets (based on lectures) (20%)
- Laboratory (40%)
  - Lab 1 and Report (10%)
  - Lab 2 and Report (10%)
  - Lab 3 and Report (10%)
  - Lab 4 and Report (10%)
- Autonomous UVS Project (40%)
  - Project Review Presentation (5%)
  - Project Demonstration/Competition (15%)
  - Project Report (20%)

Tentative Agenda I

1. Introduction (Liu) Jan.15
   overview of the course, cases of previous year's labs and projects
2. Overview of AER1216: Design (Grant) Jan.22
3. Quadrotor dynamics and control (Schoellig) Jan.29
4. Lab 1: Instructions and Office Hours (TAs) Feb.05
5. Navigation for UAVs (Kelly) Feb.12
6. Lab 2 Feb.26
7. Path Planning for UAVs (Waslander) Mar.05
8. Computer vision for UAVs (Kelly) Mar.12
9. Lab 3 Mar.19
10. Instrumentation and sensor payloads for UAVs (Armenakis) Mar.26 or TBD

Tentative Agenda II

11. Lab 4 Apr.02
12. Guest Lecture Apr.09
13. Project Review Apr.16
14. Project Demonstration Apr.23
15. Final Report Due Apr.30

Lab 1: Quadrotor Simulation and Control Design I

This is the first lab in a series designed to complement the lecture material and help you with the course project. This lab requires the design of a quadrotor position controller, and implementation in ROS. It is set up to work with Gazebo, which simulates your interface with the Parrot AR.Drone UAV and the Vicon motion capture system for position and attitude measurements. To be able to complete this lab within your dedicated time slot, you should have preliminary knowledge of ROS, and Python programming. As a team you will be working on the Lenovo IdeaPad Y700. The course only has three of these so the members of your team are encouraged to bring a laptop of their own. This will allow each member to work on separate portions of the lab simultaneously.

1. Design and implementation of the position controller,

Lab 1: Quadrotor Simulation and Control Design II

2. Implementation of a ROS node that publishes control commands and subscribes to quadrotor actual and desired positions, and
3. Implementation of a ROS node that publishes a time based trajectory.

Lab 2: From Quadrotor Simulation to the Real World I

This is the second lab in a series designed to complement the lecture material and help you with the course project. This lab takes what you have done in the first lab and transfers it to the implementation on a Parrot AR.Drone 2.0 in the indoor robotics lab. You will be receiving position and attitude information of the quadrotor from the Vicon motion capture system. The `vicon_bridge` package will be used to receive the quadrotor pose, and the `ardrone_autonomy` package will be used to send commands to the AR.Drone.

1. Modification of `keyboard_controller.py` to use high-level keyboard commands,
2. Inclusion of the modified `keyboard_controller.py` into your Lab 1 ROS architecture, and
3. Modification of the ROS node from Lab1 that publishes a time-based trajectory to fly linear and circular trajectories.

Lab 3: Georeferencing Using UAV Payload Data I

This is the third lab in a series designed to complement the lecture material and help you with the course project. In this lab, you will be using the Parrot AR.Drone 2.0 bottom-facing camera (64° diagonal FOV, 640px ? 360px) to detect and locate targets of interest on the ground. The data from the Vicon motion capture system and the images from the quadrotor will be used to find each targets location on the ground within the inertial Vicon fixed reference frame. As in the previous lab, the Vicon motion capture system will also be used for the controller of the quadrotor.
This lab requires you to design an image processing algorithm that will assign georeferenced coordinates to pixels of an image based on quadrotor position and attitude. In preparation for the project and the fourth lab, it is highly encouraged for you to conduct image processing in real-time using OpenCV, but this is not necessary to complete this lab. To help you finish the lab within

Lab 3: Georeferencing Using UAV Payload Data II

your allocated time, there is a pre-lab component that will require you to have the trajectory ready before you start.

1. Design and implementation of the desired trajectory (pre-lab),
2. Modification of `DroneVideoDisplay` to start automatic image capture of the bottom-facing camera and save time-stamped images, and
3. Implementation of a georeferencing algorithm that analyzes images and AR.Drone pose data to deter- mine the coordinates of each target of interest.

Lab 4: Flying Through Hoops I

This is the fourth and final lab in a series designed to complement the lecture material and help you with the course project. In this lab, you will be using the Parrot AR.Drone 2.0 front-facing camera (92° diagonal FOV, 640px ? 360px) to detect one hoop and fly through it. This lab requires you to design an image processing algorithm that will detect a hoop, estimate its relative or absolute vertical position, and adjust the desired z-position in real-time. This z-position feeds into your quadrotor altitude controller so that the altitude error between the quadrotor and the hoop is driven to zero. Combined with the knowledge of the hoop plane and horizontal position, the quadrotor can fly through it without hitting the sides.

1. Hoop detection algorithm (pre-lab),
2. Hoop vertical position estimation algorithm, and

Lab 4: Flying Through Hoops II

3. Modification of DroneVideoDisplay to run the above algorithms and publish updated desired positions to the controller.

Project: Autonomous Drone Racing I

In the final project you will be attempting to set the fastest time along a track with a Parrot AR.Drone 2.0 while flying through a number of hoops and maintaining high tracking accuracy. The track is unknown to you beforehand and you will have one opportunity to survey the area with the quadrotor to determine the trajectory that will need to be followed during the timed event. No hoops will be present during this time. In the timed event, the hoop locations and their altitudes are also unknown but will be placed on the trajectory. You are required to detect the hoops and estimate their altitudes in real-time to fly through them during the timed event.
The project consists of a set of deliverables spread out over the last month of the course. In addition to the demonstration described above, the project also includes a short review presentation prior to the demonstration and a final report

Project: Autonomous Drone Racing II

documenting your approach and results to be submitted afterwards. This handout is primarily focused on the demonstration and how it will be scored/graded, and the breakdown of the final report. Unlike the labs, minimal guidance will be provided to you regarding what code to write or modify to achieve the project objectives. You are encouraged to write code in a modular way because the code will constitute a portion of the project report to be submitted after the demonstration.

Project: What's New?

- set up different colours of Hoops, under different colour, different actions are required;
- make the hoops smaller