MiniProject Proposal: DJI Tello Control

Due: Fri. Nov. 13, 6 pm on Gradescope (1 per group).

1 Objectives

The goal of the Tello mini-project is to build on Lab5a and Lab5b to develop your own control application using the Tello MiniDrone. Requirements for the miniproject:

- The controller will be implemented in Python and use the Tello `rc x y z yaw` interface.
- The proposed application should include quantifiable performance metrics, such as time to target, precision to target, or tracking error for moving target. The position can be estimated using `aruco` tags and either the on-board Tello camera or an external camera.
- The proposed control algorithm should be prototyped using a simulated model such as from Lab5a.
- The mini-project report (due Tues 12/8 at 6 pm) should include a written portion documenting control strategies and parameters used, as well as a short video uploaded to `Youtube` showing the Tello in action.

Figure 1: DJI Tello quadrotor. Source: Amazon.com

2 Equipment and Tools

2.1 Project Equipment

This miniproject will use two main pieces of hardware: your laptop and your Tello. Your laptop will communicate with your Tello over wifi, so when running experiments you will not be able to access the internet unless you have a second wifi/network card. The python script will be running three threads. The first thread will be listening to the Tello on UDP port 8890 for state data from the Tello's sensors. The second thread will be running your controller and sending commands to your Tello on UDP port 8889. A third thread will be running visual tracking, either using the Tello camera or a fixed camera (e.g. your laptop webcam).

The Tello camera is at UDP port 11111. To enable the onboard camera, two commands must be sent first over port 8889. First send `command` and then `streamon` to enable camera streaming. An example Python3 code for reading video from Tello and displaying (tested under Ubuntu) is here:

`https://gist.github.com/f41ardu/b75da46dae383e5c835295600deef604`

(Make sure to use another window to send `command` and then `streamon` to enable video streaming on this port.) More example code for using the camera will be provided soon.

`https://github.com/ronf-ucb/TelloControl/blob/master/TelloKeyboardCommands.py`

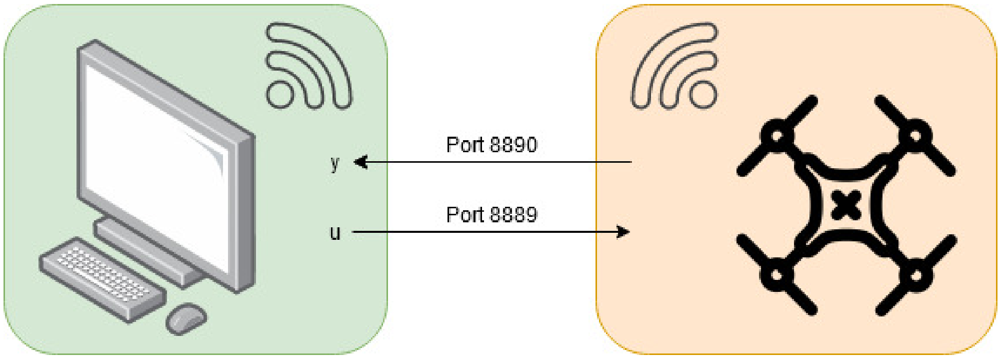

Figure 2: A system diagram how your laptop and your Tello communicate.

2.2 Visual Tags

Visual tags can be printed out and provide x-y coordinates in a field of view. `aruco` tags work with OpenCV which is available in Python. The references below give an easy introduction to `aruco` tags.

`https://medium.com/lifeandtech/custom-ar-tag-detection-tracking-manipulation-d7543b8569ea`

`https://mecaruco2.readthedocs.io/en/latest/notebooks_rst/Aruco/aruco_basics_video.html`

The tags work pretty well with a direct normal view. There are likely packages available to combine several tags to get orientation information. Multiple tags can appear in the same scene, so it is possible to have one tag on the Tello and other tags on navigation features or tracking targets.

3 Mini Project Proposal

For the miniProject, think of an *interesting* control problem to do with the Tello. *Interesting* would be an entertaining example video to show in future MEC134/EEC128 classes. Some illustrative examples to get you started thinking:

- Follow a swinging pendulum (or moving object)
- Fly Tello through a moving hoop
- Catch a slow paper airplane
- Catch a falling balloon
- Juggle a balloon (very hard)
- Fly through slalom poles

- Fly through an open box
- Mass estimation with a weight attached
- Nonlinear or novel techniques for tracking

For each of these projects (and likely other proposed projects), you can assume that items of interest have an *aruco* tag. For example, the pendulum (a weight on a string) would have a tag for tracking. If a laptop camera is used for tracking, then Tello would have a tag as well. Given the 10 Hz update rate, we don't expect the performance seen here:

`https://www.youtube.com/watch?v=3CR5y8qZf0Y`

Note: for safety reasons, all proposed miniProjects must be done indoors. The final report will include a video of your Tello control.

The purpose of the project proposal is to give you early feedback on your design and project ideas, to help to steer you away from things which we think are too hard. Since we haven't done anything complicated yet with Tello, there is uncertainty in actual degree of difficulty.

3.1 Outline for miniProject proposal

(Suggested length 2 pages or less, 10 pt minimum, 1 inch margins.)

1. (1 paragraph) Project concept- what will Tello do and how will it do it?
2. (1 paragraph) Basic project goals- what components and subsystems will need to be working? What performance is needed/expected, e.g. closed loop bandwidth?
3. (1 paragraph) Intermediate project goals- What pieces do you need to show working together? What could you do at slow speed if dynamic control is not possible due to slow visual tracking?
4. (1 paragraph) Final project goals- Assuming the basic and intermediate goals are met, and the pieces are all working well together, what will you try to achieve?
5. (1 paragraph) Performance metrics. The proposed application should include quantifiable performance metrics, such as time to target, precision to target, or tracking error for moving target.
6. (1 figure) Provide a controller block diagram labelling reference, sensed, and control signals.
7. (1 paragraph) Uncertainties. List all parts of the project you are uncertain about, e.g. using camera data with tags, operational space, lighting, communications, OpenCV (open computer vision) library, etc.

4 Installing Stuff

(This section is to get a head start; it does not need to be installed for the miniProject proposal.)

4.1 Installing OpenCV

The usual `conda` python package manager does not understand Opencv. To get opencv and aruco tag recognition

```
pip install opencv-python
pip install opencv-contrib-python
```

If install error do:

```
pip install opencv-contrib-python --user
```

4.2 Safety Rules

Please read and follow the rules of this lab:

- All lab and project experiments are to be done indoors.
- Wear safety glasses when flying drone.
- Do not try to grab the Tello drone in flight with bare hands.
- Keep children, pets, and others away from the experiment.
- Assume controllers can become unstable, leading to unexpected or dangerous behavior.
- The Tello Drone is not a toy - personal injury can result from contact with spinning blades. Keep prop guards on. Do not touch propellers when powered on.
- Choose an indoor area without ceiling obstacles, and away from fragile objects (china etc.).
- Follow safety guidelines as listed on the Ryze Robotics Website: in particular sections Propulsion System and Flight Batteries: `https://dl-cdn.ryzerobotics.com/downloads/Tello/20180211/Tello+Disclaimer+and+Safety+Guidelines+(EN)+v1.0.pdf`
- Make sure operation area is well-lit (flight control relies on visual feedback).
- Power down drone when not in use (it can overheat in less than 1 minute).
- Use only approved LiPo battery charger. Do not charge unattended.
- If you live in University property, please be aware of UCB campus drone policy: `https://campuspol.berkeley.edu/policies/drone(campus).pdf`