MEAM 620 Project 2 Phase 3

Due: Friday, April 26, 2013 *(Tentative)*

1 Overview

It is time to put together everything that you learned in this course! For this assignment, you will work as a team to implement on a vision-based control and trajectory following methodology using the KMel Nano+ quadrotor platform and software developed by KMel Robotics [1]. The ultimate goal of this project is to reproduce everything that you accomplished in Prject 1 Phase 4, but without the state feedback from Vicon.

There are two parts in this project, in the first part, you will use the *body frame* linear and angular velocities from the Vicon, and the detected AptilTags from camera images, to achieve vision-based autonomous flight. This part is *required* for all students. In the second part, you will use *only* sensor data (IMU and the detected AptilTags from camera images) to complete the same task, and the use of Vicon data is *not* allowed. The second part of this project is a bonus part. The completion of this part is *optional*, but it will of course lead to better grade.

You will need to sign up for times to fly online though the email that was sent. Due to limited space in the lab, one group at a time will be allowed to use the equipment for testing under the supervision of a TA. The hardware and software setup is almost the same as Project 1 Phase 4, with a few exceptions that will be pointed out in this document. Please refer to the Project 1 Phase 4 handout for detailed instruction of operating the quadrotor.

2 Sensor Data

The onboard microcontroller of the robot collects synchronized camera and IMU data and sends them to the desktop computer. The sensor data is decoded into standard MATLAB format. Note that since the sensor data is transmitted via wireless network, there may or may not be a sensor packet available during a specific iteration of the control loop. A sensor packet is a *struct* that contains following fields:

```
sensor.isReady      % True if a sensor packet is available, false othereise
sensor.t            % Time stamp for the sensor packet, different from the Vicon time
sensor.omegaImu     % Body frame angular velocity from the gyroscope
sensor.accImu       % Body frame linear acceleration from the accelerometer
sensor.img          % Undistorted image.
sensor.A            % Calibration matrix of the undistorted image
sensor.id           % IDs of all AprilTags detected, empty if no tag is detected in the image
sensor.p1           % Corner of the detect AprilTags in the image,
sensor.p2           % the ordering of the corners, and the distribution of the tags,
sensor.p3           % are the same as Project 2 Phase 1
sensor.p4
```

This struct is accessible in the controller (`student_control_hover`, etc). The sensor time stamp is different than the control time stamp, however, this should not matter since you only need the incremental time for your EKF. Also note that although the image is provided, it is possible to complete the project without image processing.

3 Vicon Data

In this project, you are not allowed to use the absolute pose feedback from the Vicon, which means that you are *not* allowed to use the following fields in the `qd` struct in your controller.

```
qd{qn}.pos          % Position, NOT allowed
qd{qn}.vel          % Linear velocity in the world frame, NOT allowed
qd{qn}.euler        % Orientation, NOT allowed
```

Instead, for the first part of the project, you will be given the linear and angular velocities in the *body frame* (IMU frame) of the quadrotor. You may use the following fields in your controller:

```
qd{qn}.vel_body     % Linear velocity in the body frame
qd{qn}.omega        % Angular velocity in the body frame
```

4 Extended Kalman Filter

In this project, you will need to use the Extended Kalman Filter (EKF) to estimated the pose (parts 1 and 2) and velocity (part 2 only) of the vehicle. An example MATLAB code of the 2D car simulation that was shown in the class is included for your reference. We suggest that you use the `simplify` function to shorten the symbolic representation of the state update equation, and then use the `jacobian` function to calculated the required Jacobian matrices.

To speedup the EKF development process, a sample dataset with sensor and Vicon data is included. You should use the dataset to develop your EKF offline before come to the lab for online testing.

You are also welcome to use other variants of the Kalman filter, such as the Unscented Kalman filter (UKF).

5 Special Instructions

5.1 Takeoff and Landing

You should takeoff the quadrotor by calling the `send_seqTakeoffToTags` sequence. This will lead the robot to the center of the lab space, where it can definitely see some tags. To land the robot from any place, simply call the `send_seqmsgLandOnCube` and the robot will go back to the cube. If you try to use your EKF when no AprilTags are visible, it is likely that you will crash and damage the robot.

5.2 EKF/Controller Initialization

Note that when a sequence is received, and your controller is called, there may or may not be a sensor packet available for initializing the EKF and/or the controller. In this case, you should defer the initialization step until the next sensor packet becomes available.

5.3 Coordinate Frames

The AptilTag-based pose estimator will give you the pose with respect to the origin of the mat with tags printed. The origin of the mat is the lower right most corner. As such, this coordinate frame is different from the Vicon coordinate system All your control and trajectory design should be done in the mat coordinate system.

5.4 Controller Gains

Due to the image transmission delay and the increased processing power requirement (your controller does not run at 100Hz anymore due to the time spent on vision processing) , you have to significantly reduce the gains of your controller. A set of suggested gains for the PD controller is shown as follows, You should use these gains as a starting point and tune your controller gains thereafter.

```
kpXY = 3;   % Position gain in X and Y directions
kdXY = 1.5; % Velocity gain in X and Y directions
kpZ  = 2;   % Position gain in Z direction
kdZ  = 1;   % Velocity gain in Z direction
```

5.5 Runtime Requirement

In this project, you will need to run your vision-based pose estimator and the EKF in real-time. It is very likely that this will require significantly more computation power than Project 1 Phase 4. As such, you should carefully monitor the computation time of all your algorithms (tag-based pose estimator, EKF, controller, trajectory generator). A 30ms total runtime bound will be considered as safe.

6 Development Steps

Please follow the suggested development flow to minimize the frustration. You must have a TA to verify the results of each step.

Part 1. :

(a) Develop the EKF offline using the dataset.

(b) Optimize the vision-based pose estimator and the EKF offline, such that they run in real-time.

(c) Autonomously control the quadrotor using Vicon position feedback, but run your estimator and EKF alongside, check the output of the EKF.

(d) Autonomously hover the quadrotor using the output of the EKF, finish the implementation in `student_control_hover_vision` and `send_seqmsgHoverVision`.

(e) Autonomously fly to a single waypoint, finish the implementation in `student_control_waypt_vision` and `send_seqmsgGoToWaypointVision`.

(f) Autonomous tracking of multiple waypoints, finish the implementation in `student_control_multi_waypt_vision` and `send_seqmsgFollowWaypointsVision`. For plotting purposes you should use `visionwaypts.mat` which will be provided and in the mat coordinate frame.

Part 2. : Follow the same development steps as part 1.

7 Submission

There are three deliverables for this assignment:

1. **Group Report**: Each group should submit a report approximately five pages in length (including plots - you don't need five pages of pure text). Please indicates clearly whether you complete the second part of the project or not. Your report should include but not be limited to: plots of tracking error versus time, position versus time, and 3D plots of paths. You may use the pose from the Vicon for comparison purpose. It should also include any information needed to replicate your EKF, such as values for Q and R and how you chose them.

   The report should be a pdf and should be sent to meam620@gmail.com.

2. **Individual “Report"**: Each person should send a one paragraph email to meam620@gmail.com with their name and group number and a short description of what they personally did for the group, and a short description of the contributions of their teammates.
3. **Code** : A .zip file of all code that you used for this phase. Send this with the report.

References

[1] KMel Robotics, http://kmelrobotics.com/