UDACITY

< Return to Classroom

DISCUSS ON STUDENT HUB

Building a Controller

REVIEW

CODE REVIEW 4

HISTORY

Meets Specifications

Dear Future Flying Car Engineer,
Congratulations!

You have successfully completed the third project in the nanodegree. Your drones pass all scenarios with flying colors. Your coding style is consistent and highly readable, and on top of it all, you did a fantastic job on the extra challenges!

- In the next section of the course, you will use this project as the starter code for the Building an Estimator project; as a quick breakdown, you will learn how to use quaternions to help improve your controller's overall performance, as well as learn how to implement GPS and other update methods.

I hope you're enjoying the course so far, and hope you stay safe and keep up the good work! Good luck with the rest of the nanodegree projects and beyond!

Writeup

The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.

Awesome job on your writeup! Unfortunately, the images/videos in the results folder were somehow lost in the submission project, as seen below:

Fortunately, everything passed the scenario tests, and came out great!

Implemented Controller

The controller should be a proportional controller on body rates to commanded moments. The controller should take into account the moments of inertia of the drone when calculating the commanded moments.

Good job here! Your controller is proportional on body rates to commanded moments and it takes into account the moments of inertia of the drone when calculating the commanded moments. 👏🏻

The controller should use the acceleration and thrust commands, in addition to the vehicle attitude to output a body rate command. The controller should account for the non-linear transformation from local accelerations to body rates. Note that the drone's mass should be accounted for when calculating the target angles.

Well done! Your controller meets every specification mentioned in this rubric point.

The controller should use both the down position and the down velocity to command thrust. Ensure that the output value is indeed thrust (the drone's mass needs to be accounted for) and that the thrust includes the

Output value is indeed thrust (the drone's mass needs to be accounted for) and that the thrust includes the non-linear effects from non-zero roll/pitch angles.

Additionally, the C++ altitude controller should contain an integrator to handle the weight non-idealities presented in scenario 4.

Output value is indeed thrust and it also includes the non-linear effects from non-zero pitch angles. Nice work! 👍

The controller should use the local NE position and velocity to generate a commanded local acceleration.

Local NE position and velocity has been used to generate commanded local acceleration. Very neat implementation!

The controller can be a linear/proportional heading controller to yaw rate commands (non-linear transformation not required).

Signal of the yaw angle has been verified and a proportional controller was implemented! Good job!

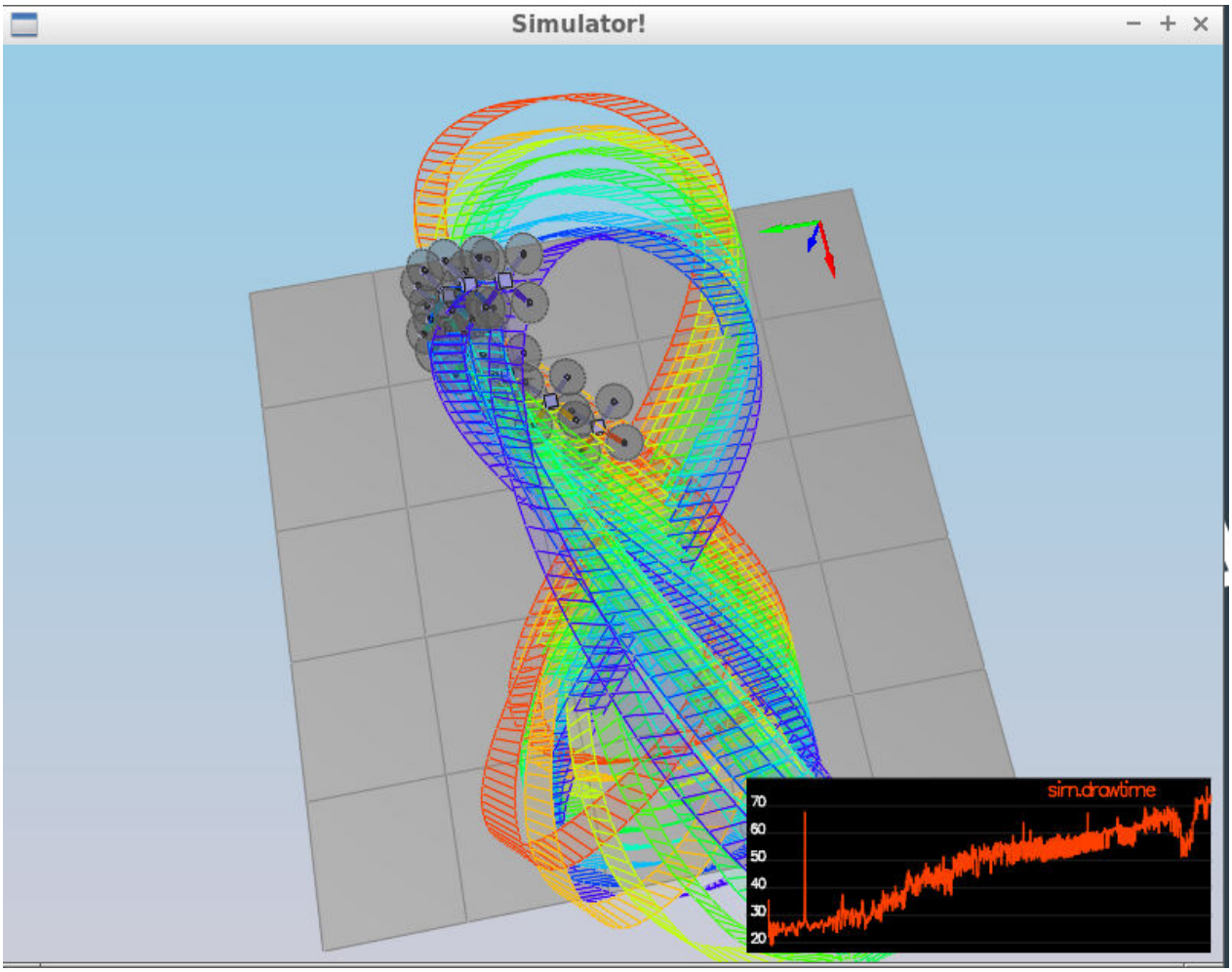
The thrust and moments should be converted to the appropriate 4 different desired thrust forces for the moments. Ensure that the dimensions of the drone are properly accounted for when calculating thrust from moments.

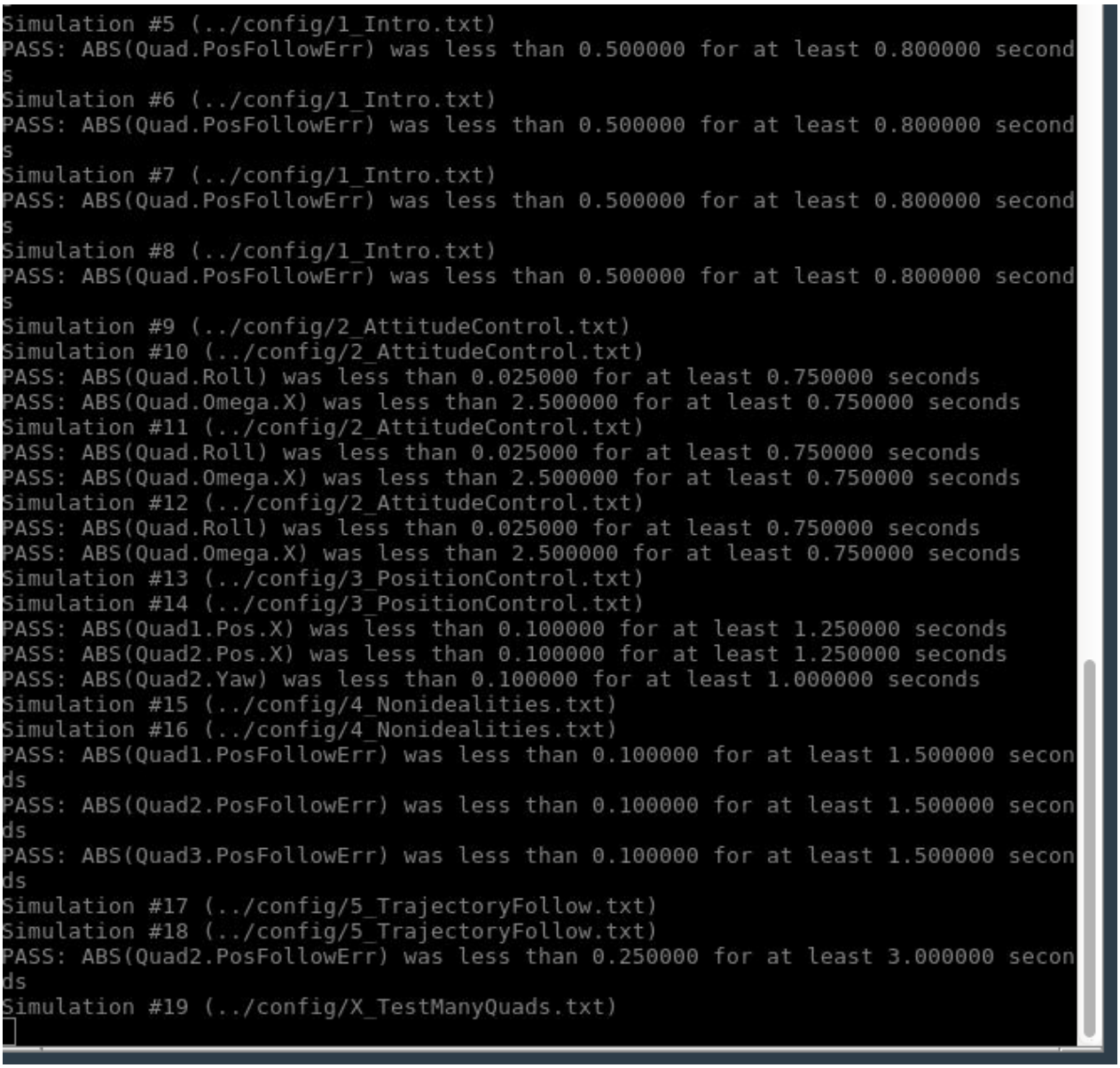
This specification was satisfied as well. All dimensions of the drone are properly accounted for. Very good job!

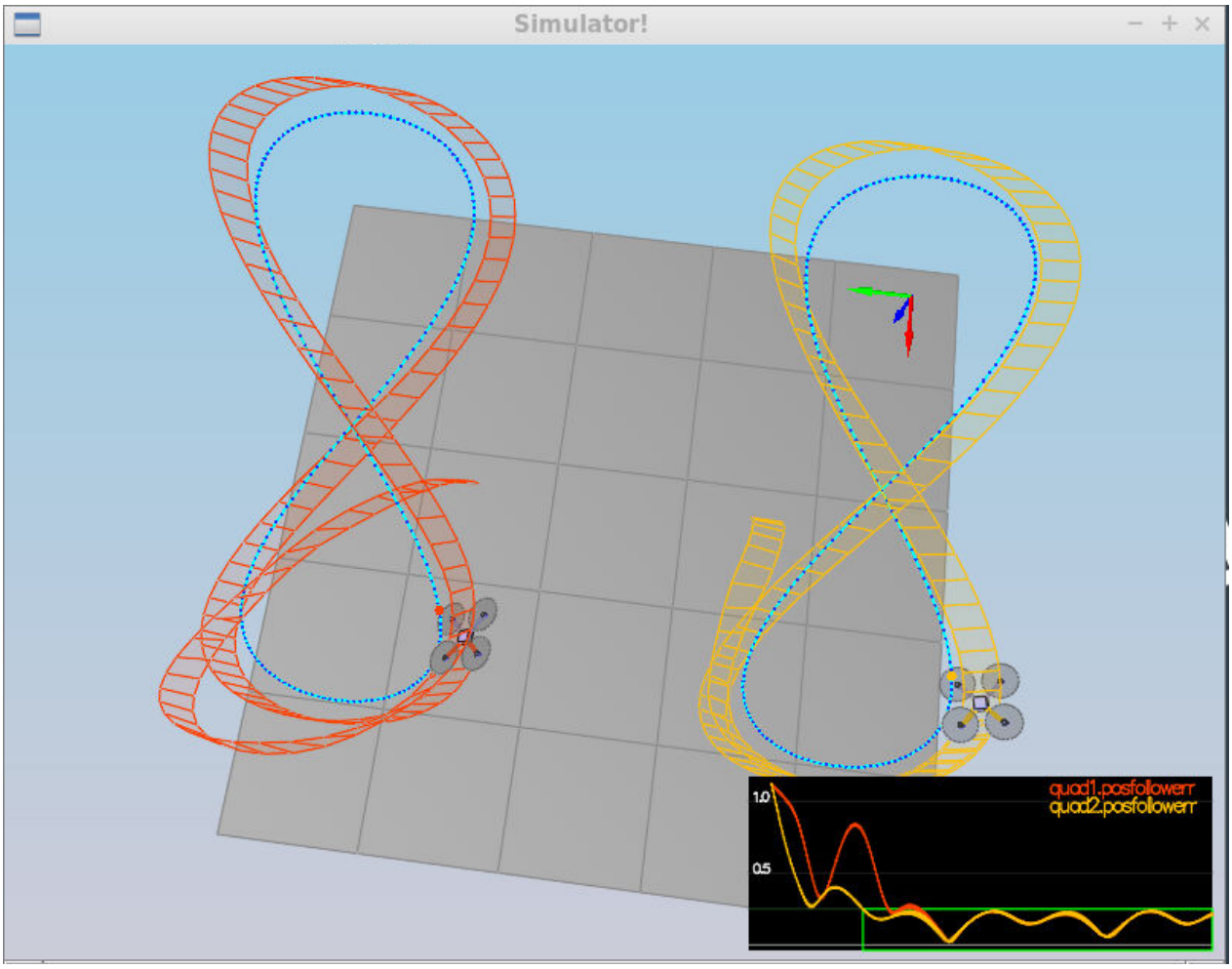
Flight Evaluation

Ensure that in each scenario the drone looks stable and performs the required task. Specifically check that the student's controller is able to handle the non-linearities of scenario 4 (all three drones in the scenario should be able to perform the required task with the same control gains used).

Beautiful work on getting your controller to pass each scenario flawlessly! You even did an incredible job on the extra credit as seen below:


Simulator!
sim.drawtime
70
60
50
40
30
20


Simulation #5 (../config/1_Intro.txt)
PASS: ABS(Quad.PosFollowErr) was less than 0.500000 for at least 0.800000 second
s
Simulation #6 (../config/1_Intro.txt)
PASS: ABS(Quad.PosFollowErr) was less than 0.500000 for at least 0.800000 second
s
Simulation #7 (../config/1_Intro.txt)
PASS: ABS(Quad.PosFollowErr) was less than 0.500000 for at least 0.800000 second
s
Simulation #8 (../config/1_Intro.txt)
PASS: ABS(Quad.PosFollowErr) was less than 0.500000 for at least 0.800000 second
s
Simulation #9 (../config/2_AttitudeControl.txt)
Simulation #10 (../config/2_AttitudeControl.txt)
PASS: ABS(Quad.Roll) was less than 0.025000 for at least 0.750000 seconds
PASS: ABS(Quad.Omega.X) was less than 2.500000 for at least 0.750000 seconds
Simulation #11 (../config/2_AttitudeControl.txt)
PASS: ABS(Quad.Roll) was less than 0.025000 for at least 0.750000 seconds
PASS: ABS(Quad.Omega.X) was less than 2.500000 for at least 0.750000 seconds
Simulation #12 (../config/2_AttitudeControl.txt)
PASS: ABS(Quad.Roll) was less than 0.025000 for at least 0.750000 seconds
PASS: ABS(Quad.Omega.X) was less than 2.500000 for at least 0.750000 seconds
Simulation #13 (../config/3_PositionControl.txt)
Simulation #14 (../config/3_PositionControl.txt)
PASS: ABS(Quad1.Pos.X) was less than 0.100000 for at least 1.250000 seconds
PASS: ABS(Quad2.Pos.X) was less than 0.100000 for at least 1.250000 seconds
PASS: ABS(Quad2.Yaw) was less than 0.100000 for at least 1.000000 seconds
Simulation #15 (../config/4_Nonidealities.txt)
Simulation #16 (../config/4_Nonidealities.txt)
PASS: ABS(Quad1.PosFollowErr) was less than 0.100000 for at least 1.500000 secon
ds
PASS: ABS(Quad2.PosFollowErr) was less than 0.100000 for at least 1.500000 secon
ds
PASS: ABS(Quad3.PosFollowErr) was less than 0.100000 for at least 1.500000 secon
ds
Simulation #17 (../config/5_TrajectoryFollow.txt)
Simulation #18 (../config/5_TrajectoryFollow.txt)
PASS: ABS(Quad2.PosFollowErr) was less than 0.250000 for at least 3.000000 secon
ds
Simulation #19 (../config/X_TestManyQuads.txt)



DOWNLOAD PROJECT

4 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review

START