Evaluation of IMU and MOCAP Sensor Fusion on Quadrotor Controller
AE483 FINAL LAB

University of Illinois at Urbana-Champaign, Champaign, IL, 61820

In order to control a dynamic system in some desired way through a control action, it is necessary to have information about the system's states. Ideally, a controller would use state information taken directly from perfect sensors. In practice, it is often not possible to directly use sensor measurements for all states. Measurements from sensors can be corrupted from noise and bias, and in some cases, measurements are not available for all states. In these cases, it is necessary to estimate some state values. Specifically, the states can be predicted at the rate of the IMU measurements and corrected from MOCAP data. The IMU outputs acceleration measurements, which are integrated to produce estimates of position. The position estimates are corrected when new MOCAP data is available. State estimation is incredibly useful in designing a robust controller. Using a robust controller is particularly important when large and sudden disturbances are present. It is predicted that estimating the states at a faster rate will improve the performance of the quadrotor in maintaining its desired flight. The effectiveness of implementing state estimation is tested in two ways: using a flight trajectory with specified points and disturbing a hovering quadrotor in multiple axes. These tests are performed on the quadrotor when it is running the old controller and then when it is running the new controller that integrates IMU data. The results suggest that there is not a clear improvement in the quadrotor's performance in tracking a desired trajectory under normal conditions, but there is a clear improvement in the quadrotor's ability to stabilize itself after being disturbed.

I. Nomenclature

f	=	Forces applied to the system
K_f	=	Motor force constant
K_m	=	Moment constant
τ	=	Aerodynamic torque
σ	=	Spin rate
θ	=	Pitch angle
ω	=	Angular velocity
$\dot{\omega}$	=	Angular rate
J	=	Moment of inertia matrix
N	=	Matrix to convert between angular velocity and angular rate
T	=	Period
I	=	Inertia about Pitch Axis
m	=	Mass (kg)
g	=	Gravity (m/s^2)
K	=	Gain Matrix
ox	=	Drone X-position
oy	=	Drone Y-position
oz	=	Drone X-position
$ox_{desired}$	=	Desired X-position
$oy_{desired}$	=	Desired Y-position
$oz_{desired}$	=	Desired Z-position

II. Introduction

When designing control systems, knowledge of the current state of the system is not always available from direct sensor measurements. In these cases, state observers allow for estimation of state values. Knowledge of the states, whether from direct sensor measurements or from estimation, is necessary to apply state feedback. The sensors used in the current system, MOCAP and IMU, provide measurements for all states. However, it is possible to estimate the state values with an accuracy greater than that of the sensor measurements themselves. Both the MOCAP and the IMU measurements are corrupted with noise, and they are updated at different rates. MOCAP provides updated measurements at a rate of 50 Hz, while the IMU updates at a rate of 1000 Hz. Within the current framework, the rate at which the state values are updated is limited by the the speed of the slower system, which is MOCAP. The IMU data that updates in between the 50 HZ MOCAP measurements is essentially discarded. The Kalman Filter allows for the states to be estimated at the IMU rate of 1000 Hz. Implementation of the filter to estimate 3 states: x, y, and z-position, is detailed in the report "Hummingbird Onboard State Estimation" by Alex Faustino, Sean Finion, Shie-Jene Shan, and Brandon Wood. As outlined in this report, the states are propagated using the following state-space model:

$$x_{k+1} = \Phi_k x_k + \Gamma_k u_k + \omega_k \tag{1}$$

$$y_{k+1} = H_k x_k + v_k \tag{2}$$

where,

x represents the state vector, u represents the input vector, Φ represents the discretized state transition matrix, Γ represents the discretized input matrix, y represents the sensor measurements, and H represents the output matrix. ω and v represent the modeled process and measurement noise, respectively. These noise parameters are assumed to be random Gaussian variables with mean 0.

In general, the filter can be broken down into two steps: the prediction step and the correction step. During the prediction step, the system state is forward predicted using the model and the process noise covariance matrix. This step is defined by:

$$\widehat{x_k^-} = \Phi x_k + \Gamma_k u_k + \omega_k \tag{3}$$

$$\widehat{P_k^-} = \Phi P_{k-1} \Phi^T + Q \tag{4}$$

$\widehat{x_k^-}$ represents the forward predicted state based on the model, and $\widehat{P_k^-}$ represents the forward predicted error covariance matrix.

The correction step corrects the predicted state by using the sensor measurements. This correction step prevents the predictions from drifting too far from reality. This step is defined by:

$$L_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{5}$$

$$\widehat{x_k} = x_k^- + L_k (y_k - H x_k^-) \tag{6}$$

$$\widehat{P_k} = P_k^- - L_k (H P_k^- H^T + R) L_k^T \tag{7}$$

L_k represents the optimal Kalman gain matrix. The purpose of this parameter within the filter is analogous to the purpose K matrix in LQR feedback control. $\widehat{x_k}$ represents the corrected state estimate that is evaluated when sensor measurements are collected. $\widehat{P_k}$ represents the updated error covariance matrix. Note that the correction step does not necessarily occur after each prediction step. Within the system of this lab, the prediction steps should occur at a rate of 1000 Hz, while the correction step occurs at a rate of 50 Hz. The focus of this report differs from that of "Hummingbird Onboard State Estimation." In this implementation of state estimation, the position and velocity states are calculated from directly integrating IMU data. These estimates are then corrected using new MOCAP data when it is available. The implementation of this method is done on the onboard quadrotor code. The performance of this new controller is then compared to that of the old controller used in previous labs. The test done to evaluate this performance is a standard trajectory tracking test and a test where the quadrotor is disturbed.

III. Controller Design and Implementation

The controller used in this project is based on the controller used in Lab 4 but with an added integrator. This onboard integrator uses the IMU data to compute a quad position estimate between the MOCAPs 50 hz measurements. With this approach it is possible for the quad to estimate its position at a rate of 1000hz and then correct this estimate to

MOCAP measurement when current MOCAP data is. The equations of motion that define the system are represented as a nonlinear state space model that takes the form.

$$\dot{x} = f(x,u) \tag{8}$$

$\dot{x}$ represents the time derivative of the system states, x represents the states, and u represents the inputs.

In order to use the state equations for controller design, they are linearized about an equilibrium point using the first-order approximation. This takes the form:

$$f(x,u) \approx f(x_0,u_0) + \left.\frac{\partial f}{\partial x}\right|_{(x_0,u_0)} \Delta x + \left.\frac{\partial f}{\partial u}\right|_{(x_0,u_0)} \Delta u \tag{9}$$

The states of the current system and their time derivatives are defined as follows:

$$x = [o_x \; o_y \; o_z \; \theta_{yaw} \; \theta_{pitch} \; \theta_{roll} \; \dot{o}_x \; \dot{o}_y \; \dot{o}_z \; \dot{\theta}_{roll} \; \dot{\theta}_{pitch} \; \dot{\theta}_{yaw}]^T \tag{10}$$

$$\dot{x} = [\dot{o}_x \; \dot{o}_y \; \dot{o}_z \; \dot{\theta}_{yaw} \; \dot{\theta}_{pitch} \; \dot{\theta}_{roll} \; \ddot{o}_x \; \ddot{o}_y \; \ddot{o}_z \; \ddot{\theta}_{roll} \; \ddot{\theta}_{pitch} \; \ddot{\theta}_{yaw}]^T \tag{11}$$

The equilibrium states about which the linearization occurs are:

$$x_e = [0,0,0,0,0,0,0,0,0,0,0,0]^T \tag{12}$$

The resulting linearized state equation takes the form:

$$\dot{x} = Ax + Bu \tag{13}$$

Where,

$$A = \left.\frac{\partial f}{\partial x}\right|_{(x_0,u_0)} \qquad\qquad B = \left.\frac{\partial f}{\partial u}\right|_{(x_0,u_0)} \tag{14}$$

A feedback controller is implemented to control the states of the system towards their desired values. The particular feedback controller implemented is a linear-quadratic regulator (LQR), which provides a solution to a cost function that results in "optimal control". LQR takes matrices A, B, Q, and R as inputs and outputs a gain matrix, K. This process is iterated by changing the weights within Q and R to produce different K matrices. The Q matrix corresponds to penalty in state errors, while the R matrix corresponds to the energy of the control response in terms of the applied inputs.

The system inputs are then calculated with the following relationship:

$$u = -K(x - x_{desired}) \tag{15}$$

To add the IMU estimate of the x,y,z,$x_d ot$,$y_d ot$,$z_d ot$ states an integrator was coded on the on-board controller. The IMU is capable of measuring specific force in 3 axis of the drone frame. The specific force is a linear acceleration taken with respect to gravity. The AscTec Hummingbird user manual was found on the internet and used to interpret the calibrated raw data taken from the IMU. A table is included that describes how the IMU data is recorded.

Variable	Values	Description
short acc_x	-10000= -1g and 10000= 1g	specific force in x direction
short acc_y	-10000= -1g and 10000= 1g	specific force in y direction
short acc_z	-10000= -1g and 10000= 1g	specific force in z direction

Because this IMU data is recorded with respect to the quad body frame it must be rotated to the room frame to compare to the MOCAP x,y,z measurements. A rotation matrix is needed to convert the acceleration vector to the correct frame. The following matrix is the one used on the drone.

$$\begin{bmatrix} -sin(\theta_{pitch}) & 0 & 1 \\ cos(\theta_{pitch})sin(\theta_{roll}) & cos(\theta_{roll}) & 0 \\ cos(\theta_{pitch}cos(\theta_{roll}) & -sin(\theta_{roll}) & 0 \end{bmatrix}$$

To confirm that the acceleration data was being recorded as expected the ASCI tool was used to verify the IMUs measurements. It was found that the IMU detects the acceleration of gravity when the drone is not moving. An offset in the Z direction has to be included for the drone to provide accurate estimates of its position when integrating with respect to gravity. The equation used to offset the IMU data is included below. The IMU data can also be easily converted to units of m/s^2 by dividing by the raw data by 1000. The equations used to convert the IMU data on the drone are below.

$$\ddot{o}_{x,IMU} = \frac{RO\_ALL\_Data.acc\_x}{1000}$$

$$\ddot{o}_{y,IMU} = \frac{RO\_ALL\_Data.acc\_y}{1000}$$

$$\ddot{o}_{z,IMU} = -(\frac{RO\_ALL\_Data.acc\_z}{1000} + 9.81)$$

Once the acceleration vector is of the correct magnitude and in the correct reference frame the acceleration vector is integrated to get a velocity and integrated again to get an estimate for x,y, and z positions. A basic Euler integration scheme was implemented on the drone. The time steps are equal to the drone processor refresh rate of main.c which is equal to 1000hz. The implementation of the integrator is shown below.

$$\dot{o}_{x,IMU} = \frac{(acc\_x\_old + acc\_x)}{2} * 0.001 + \dot{o}_{x,IMU}$$

$$\dot{o}_{y,IMU} = \frac{(acc\_y\_old + acc\_y)}{2} * 0.001 + \dot{o}_{y,IMU}$$

$$\dot{o}_{z,IMU} = \frac{(acc\_z\_old + acc\_z)}{2} * 0.001 + \dot{o}_{z,IMU}$$

$$o_{x,IMU} = \frac{(vel\_x\_old + vel\_x)}{2} * 0.001 + o_{x,IMU}$$

$$o_{y,IMU} = \frac{(vel\_y\_old + vel_y)}{2} * 0.001 + o_{x,IMU}$$

$$o_{z,IMU} = \frac{(vel\_z\_old + vel\_z)}{2} * 0.001 + o_{x,IMU}$$

This integrated IMU data is used as the state estimate x in the equation $y = -k(x - x_{desired})$ as the quads states when no current MOCAP data is present. When current MOCAP data is present the IMU position is corrected to the MOCAP estimate. This is done because the MOCAP measurement has less error than the IMU position measurement as a drift error will build up quadratically from integration over time of IMU data. To improve this controller sensor weights can be assigned to the IMU and MOCAP measurements. These weights would allow the operator to tune the performance of how the position estimate is computed. To optimally determine these weights a Q and R matrix can be designed and an observer gain could be implemented.

IV. Experimental Results

A. Methodology

In order to assess the performance of the controller after implementing the integration method, two types of tests are conducted. First, the quadrotor is flown through a set of standard points. This test is run twice: once with the controller from previous labs and once with the new controller. This test allows for a comparison between the two controllers' abilities to track a desired sequence of points. Evaluation of the results will determine whether or not using integrated IMU data in the new controller improves the ability of the quadrotor to track its desired state.

The second type of test involves introducing disturbances to the quadrotor while it is hovering at a height of 1 meter. This test is also run once with each of the controllers. This test allows for evaluation of the quadrotor's response to disturbance. Comparison of its behavior between each of the tests, specifically how quickly it returns to the equilibrium state after being disturbed, is evaluated. The combination of point-tracking tests and disturbance response tests is used to ultimately asses the performance of the new controller in relation to the old controller.

B. Results

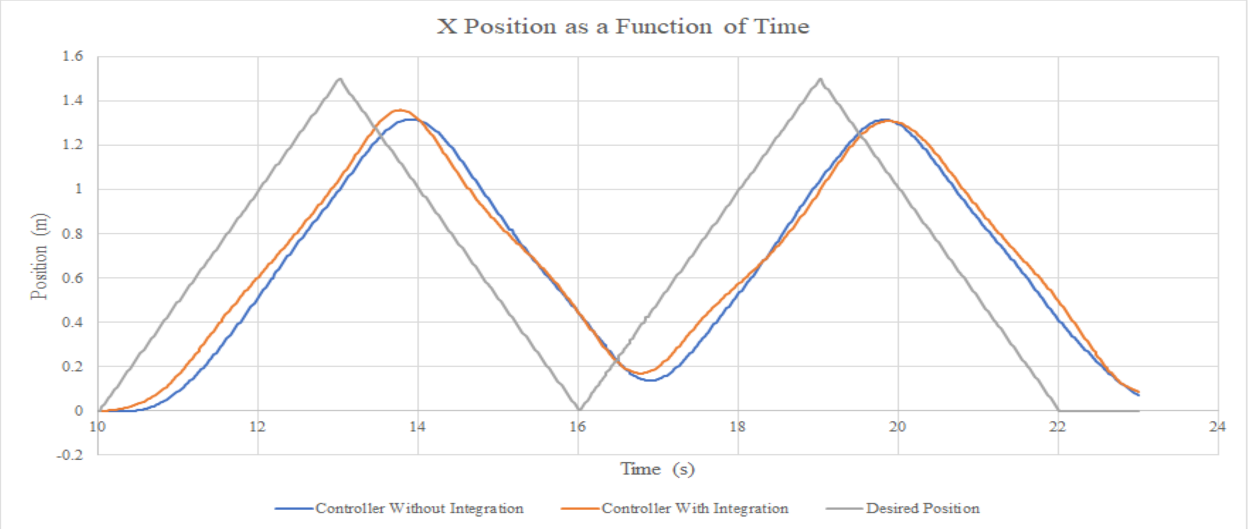

Fig. 1 This figure displays the position of the quadrotor on the x axis as a function of time during the point-tracking flight test.

This figure displays the results of the point-tracking flight test that was conducted. The desired position curve indicates that the quadrotor attempted to track the following sequence of positions on the x-axis: 0m, 1.5m, 0m, 1.5m. The positions between these specified points follow a linear path. The other two curves on the graph display the actual quadrotor trajectories from the two controllers. Initial comparison between these curves shows that there does not seem to be significant difference between the controller with integration and the controller without integration. Each of the curves follows the same trend, and differences in the trajectories are minor. Further analysis of the error between the actual trajectories and the desired trajectory quantifies the controller differences.

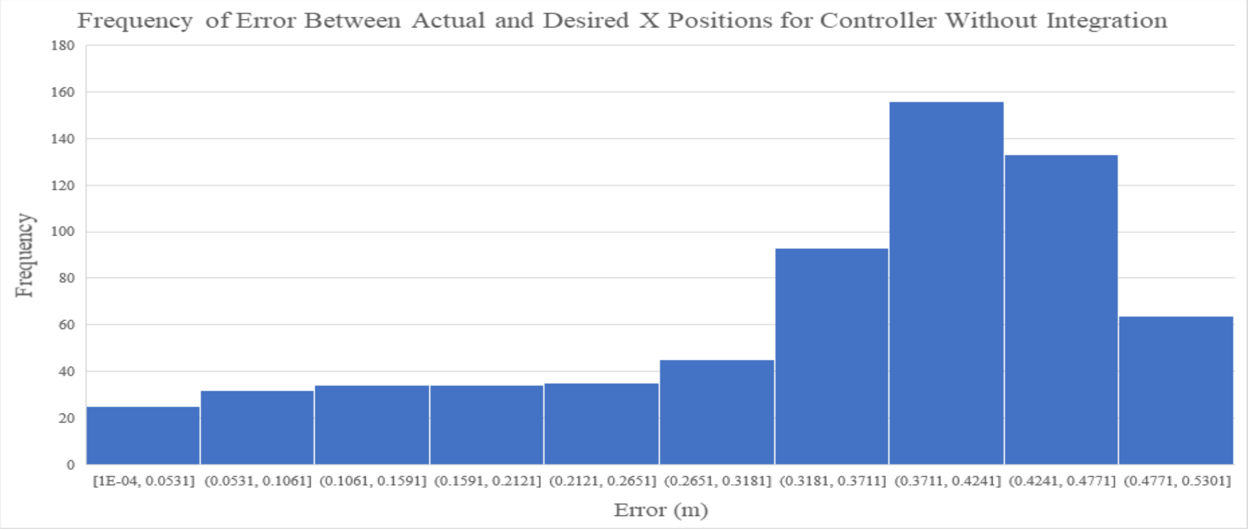

Fig. 2 This figure displays the distribution of the magnitude of error between the actual x-position and the desired x-position for the controller without integration

From the distribution of error magnitude, the median and mean errors are 0.3838 and 0.3398 meters, respectively. The sum-squared total error between the actual and desired position is 524.446.

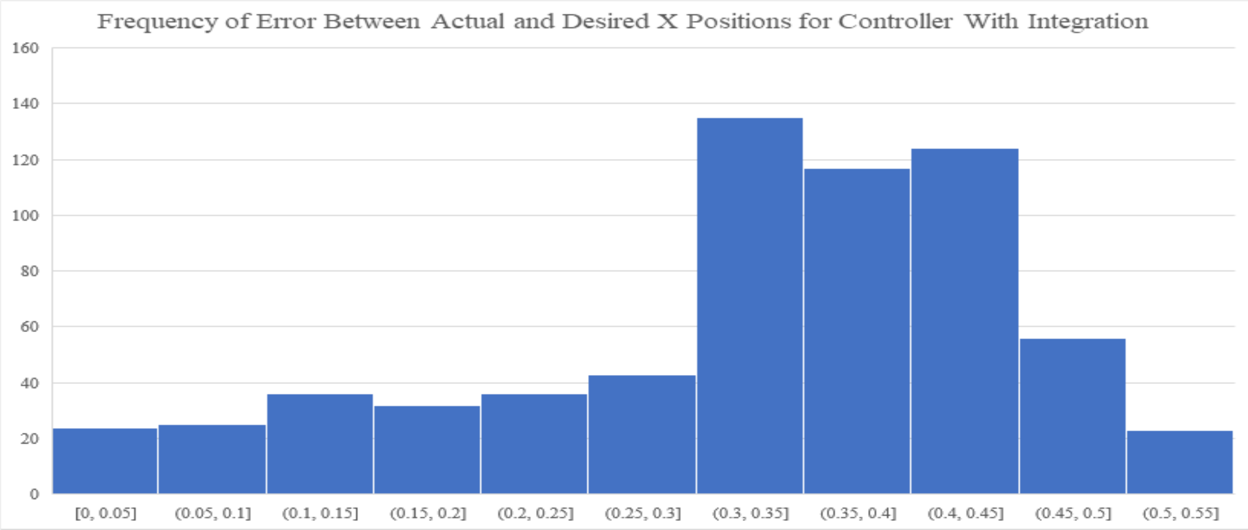

Fig. 3 This figure displays the distribution of the magnitude of error between the actual x-position and the desired x-position for the controller with integration

From this distribution, it is seen that the median and mean errors are 0.3482 meters and 0.3248 meters, respectively. The median error between actual and desired position for the controller with integration is 9.1% smaller than that of the old controller. The mean error for the controller with integration is 4.4% smaller than that of the old controller. These results indicate that while the controller with integration has less error in tracking the desired position, the difference is minor. The initial qualitative analysis of the trajectory curve is confirmed through analysis of the error distributions. An error difference this small may be attributed to different quadrotors used for testing or other random influences. In order to truly confirm that the controller with integration performed better in this aspect, more trials are required to eliminate bias.

The next section of results covers the disturbance testing. The quadrotor was disturbed in the following ways: translationally on the y-axis, translationally on the z-axis, and rotationally with yaw angle.

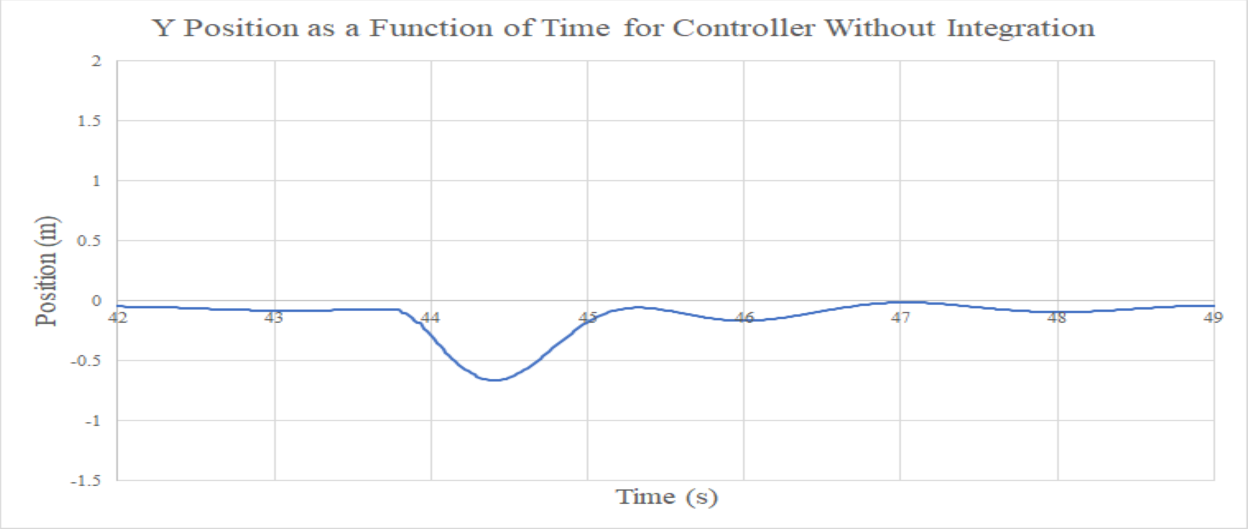

Fig. 4 This figure displays the quadrotor y-position for the controller without integration. The disturbance is in the y-axis.

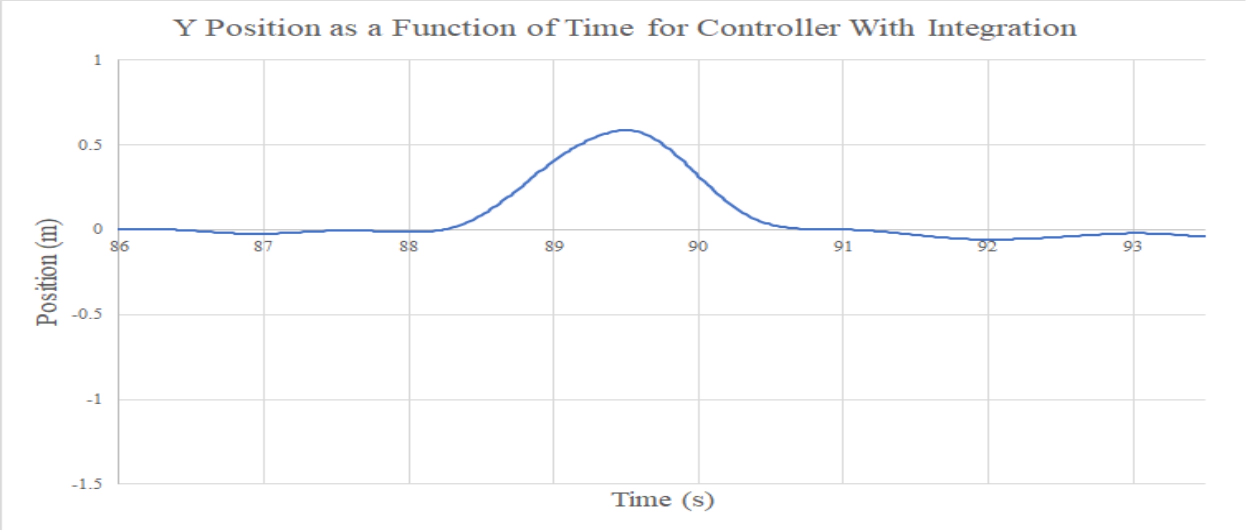

Fig. 5 This figure displays the quadrotor y-position for the controller with integration. The disturbance is in the y-axis.

Figures 4 and 5 above can be compared to evaluate the quadrotor's ability to respond to disturbances. While the disturbances between the tests of the old and new controllers are not the same, visual analysis of the curves are still informative. From the two graphs, it can be seen that the disturbances introduced have a magnitude of slightly more than 0.5 meters in both cases. Figure 5 indicates that once the quadrotor reaches its disturbed position, it returns back to its equilibrium position more quickly than in the test represented in Figure 4. The curve in Figure 4 indicates that the quadrotor running the controller without integration had oscillatory motion upon returning from the disturbance, while the quadrotor running the controller with integration had no oscillatory motion. As a preliminary assessment of the different controllers' performance in terms of responding to disturbance, the results of y-axis disturbance indicates that the controller with integration performs better.

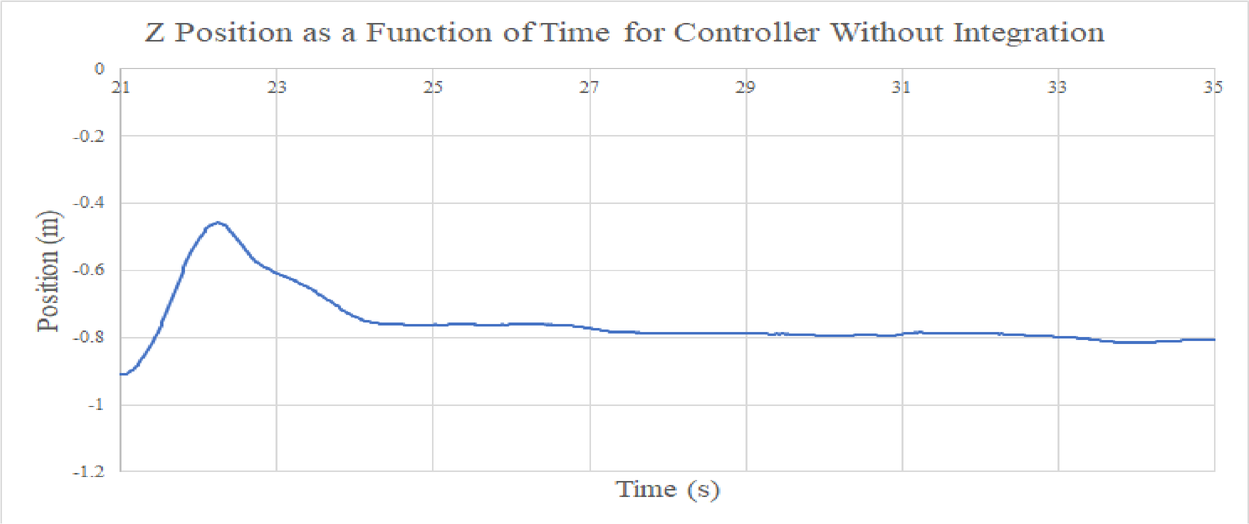

Fig. 6 This figure displays the quadrotor z-position for the controller without integration. The disturbance is in the z-axis.

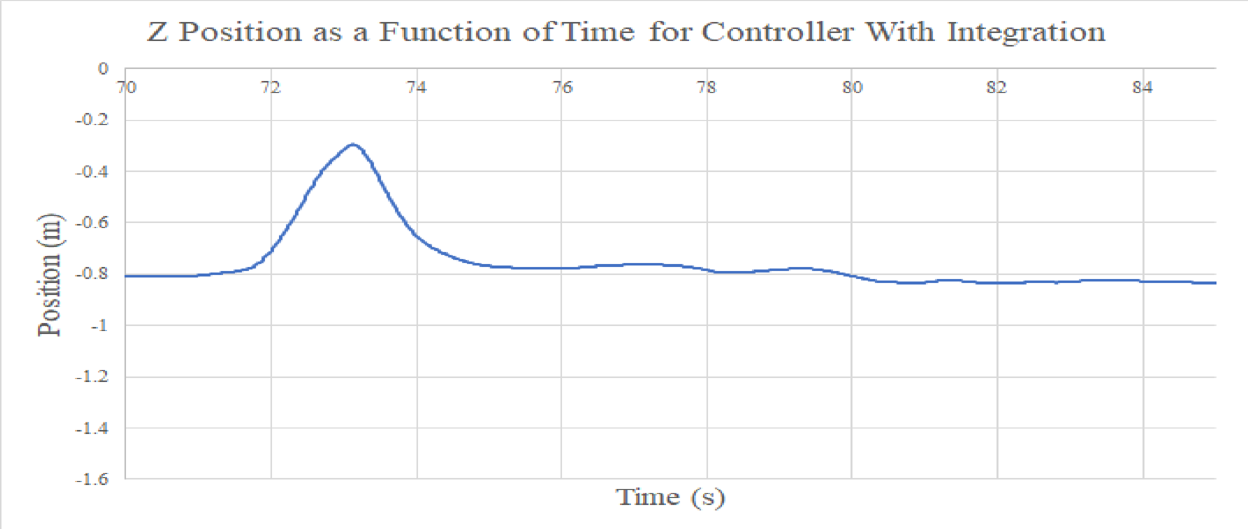

Fig. 7 This figure displays the quadrotor z-position for the controller with integration. The disturbance is in the z-axis.

The analysis of disturbance response is then continued with a disturbance to the quadrotor in the z-axis. Figure 6 indicates that the quadrotor running the controller without integration was disturbed to a height of slightly lower than 0.5 meters. Figure 7 indicates that the quadrotor running the controller with integration was disturbed to a height slightly lower than 0.3 meters. From a comparison of the two curves, it can be seen that the quadrotor running the new controller was able to return to its equilibrium position more quickly than the quadrotor with the old controller. The quicker response indicates that the new controller performs better in terms of disturbance response. Additional testing can be used to further analyze difference in performance as it relates to disturbance and quadrotor stability.

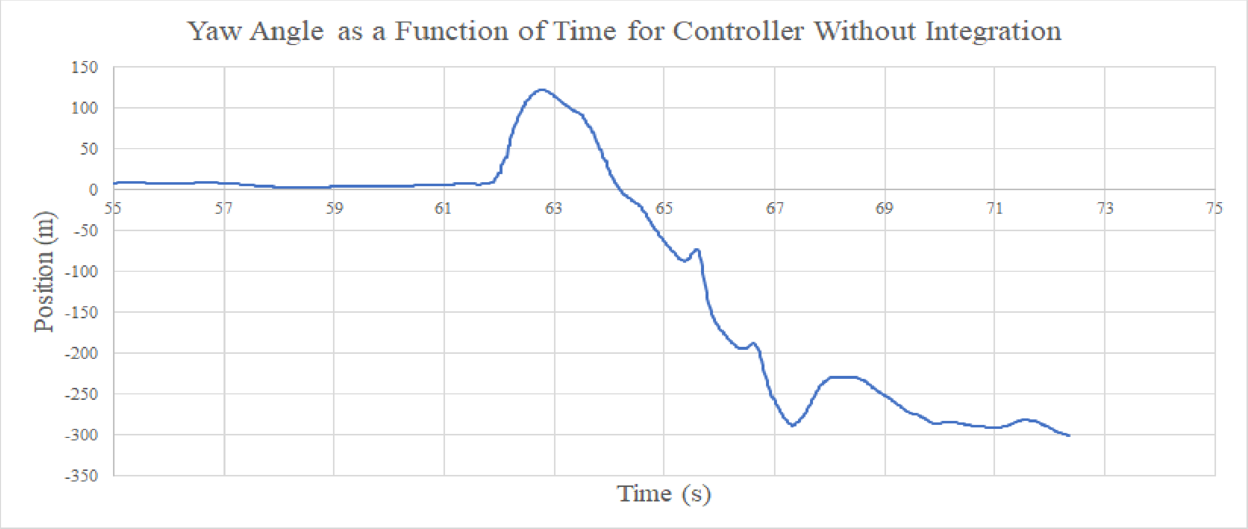

Fig. 8 This figure displays the quadrotor yaw angle for the controller without integration. The disturbance is in the yaw angle of the quadrotor.

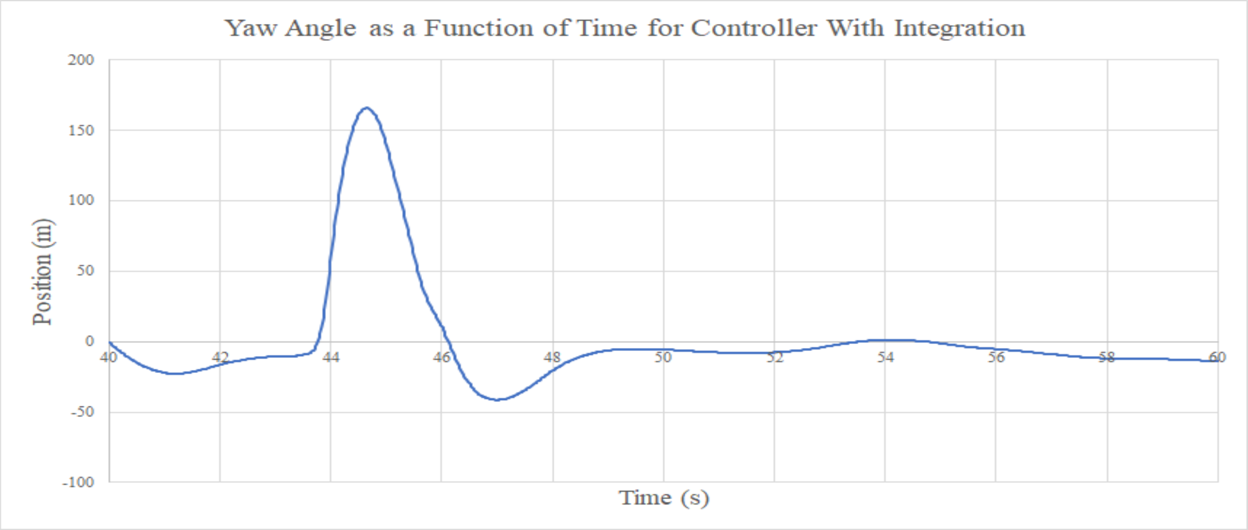

Fig. 9 This figure displays the quadrotor yaw angle for the controller with integration. The disturbance is in the yaw angle of the quadrotor.

Figures 8 and 9 display the results from introducing yaw disturbances to the quadrotor running the two different controllers. Figure 8 indicates that the quadrotor running the controller without integration was disturbed to an angle of approximately 120 degrees. The figure indicates that the quadrotor was then unable to recover from this disturbance. The yaw angle moves back towards zero but then grows more negative until eventually the quadrotor is turned off. The old controller is clearly unable to handle yaw disturbances of a large magnitude. Figure 9 shows that the quadrotor running the controller with integration was disturbed to a yaw angle of about 160 degrees. In contrast, this figure indicates that the new controller was certainly able to handle large yaw disturbances. The yaw angle returns back towards zero and settles after slightly overshooting its equilibrium position. The results from yaw disturbance analysis suggests that the controller using integration is far more successful in stabilizing the quadrotor with respect to disturbance. While the difference was not as apparent in the translational disturbances, the difference is certainly clear for yaw stability.

V. Conclusion

The controller used in the previous labs used strictly MOCAP data to update the x, y, and z positions and linear velocities. This data was updated at a frequency of 50 Hz. In this project, this controller was modified so that the position and linear velocity states are updated by integrating IMU data at a rate of 1000 Hz. These states are then corrected every 50 Hz from the MOCAP data. The results from the standard test designed do not necessarily show a definitive difference between the old controller and new controller's ability to track a desired state. The minor improved performance from the new controller may be attributed to differences among drones and random bias. More trials would account for this bias and give a more definitive result. From this test, it is concluded that under normal conditions, updating state information more frequently than the MOCAP rate does not improve performance. The difference in performance, however, is more clear in the analysis of disturbance response. When the quadrotor was disturbed translationally in the y and z axes, it returned to its equilibrium position more quickly when running the new controller. When the yaw angle was disturbed, the old controller was unable to recover, while the new controller recovered very well. This difference is key in establishing superior performance of the controller with integration as it relates to quadrotor stability.

VI. Acknowledgements

We woud like to thank the TA's of this class who supported us through late nights working on the labs. We would also like to thank Professor Dan Block and Professor Bretl for helping us expand our knowldege of control systems through this course and for making the lab space available to us.

VII. References