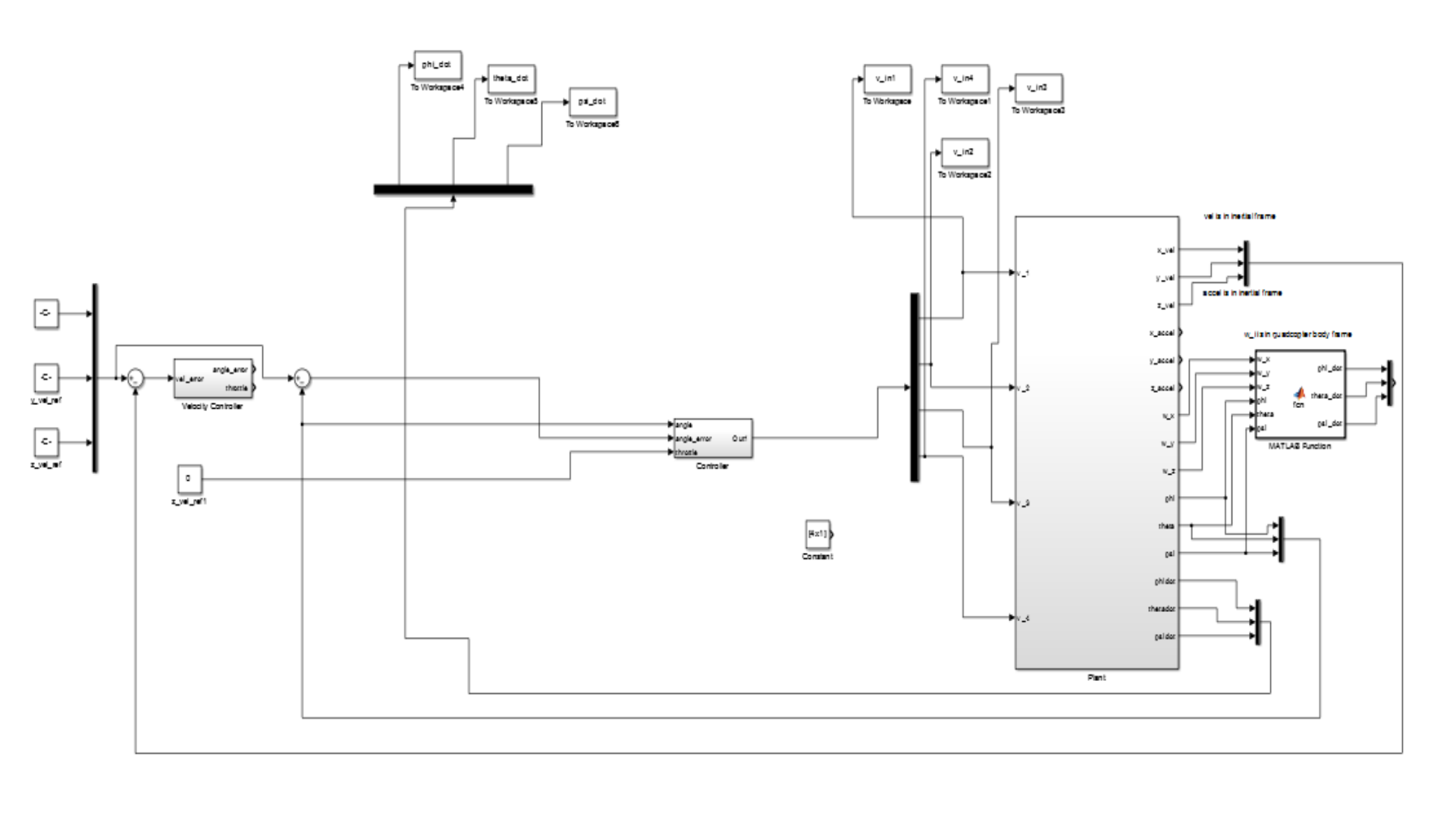
Quadqwopter

Ken Katagiri, Jeffrey Lu, Lawrence Ng

Model, Physical Dynamics Simulation

- What values should we output to the motors given a desired torque

Inspiration from Andrew Gibiansky, Harvey Mudd student


phi_dot
To Workspace4
theta_dot
To Workspace5
psi_dot
To Workspace6
v_in1
To Workspace
v_in4
To Workspace1
v_in3
To Workspace3
v_in2
To Workspace2
vel is in inertial frame
accel is in inertial frame
w_i is in quadcopter body frame
MATLAB Function
fcn
Plant
Constant
Velocity Controller
Controller
x_vel_ref
y_vel_ref
z_vel_ref
x_vel_ref1

Simulation: Quadcopter Dynamics

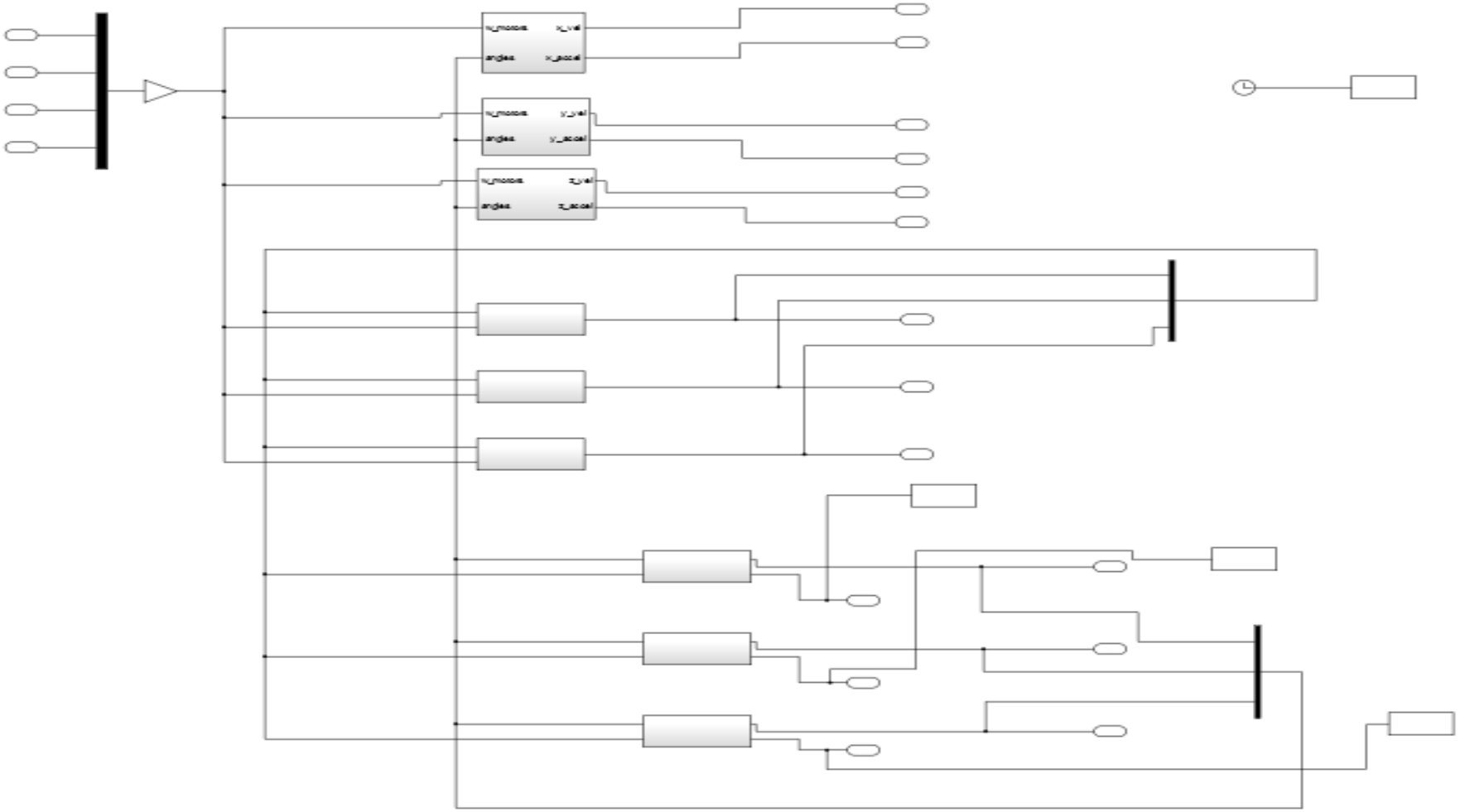
$$\dot{\omega} = \begin{bmatrix} \tau_\phi {I_{xx}}^{-1} \\ \tau_\theta {I_{yy}}^{-1} \\ \tau_\psi {I_{zz}}^{-1} \end{bmatrix} - \begin{bmatrix} \frac{I_{yy} - I_{zz}}{I_{xx}} \omega_y \omega_z \\ \frac{I_{zz} - I_{xx}}{I_{yy}} \omega_x \omega_z \\ \frac{I_{xx} - I_{yy}}{I_{zz}} \omega_x \omega_y \end{bmatrix}$$

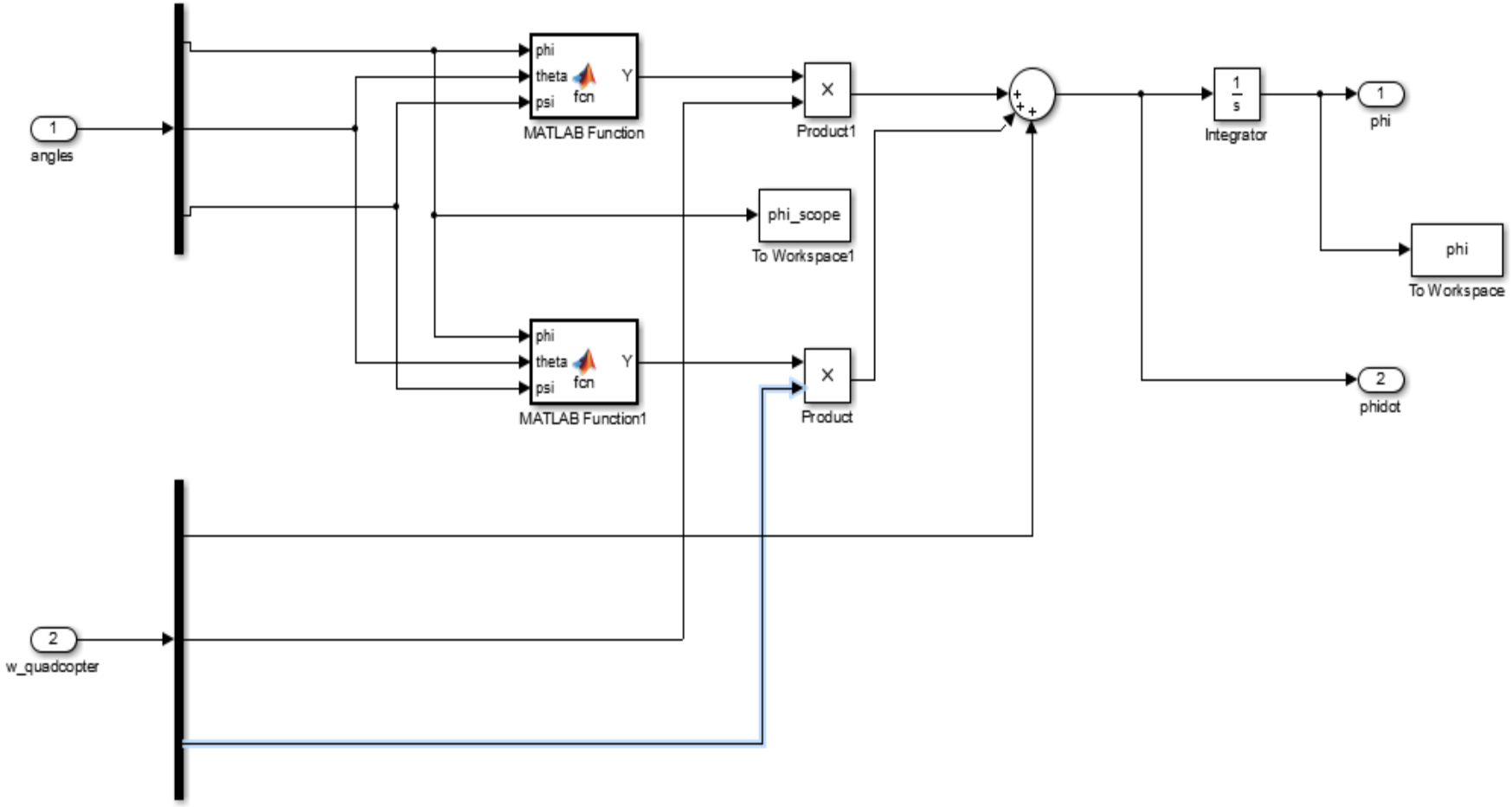
$$m\ddot{x} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RT_B + F_D$$

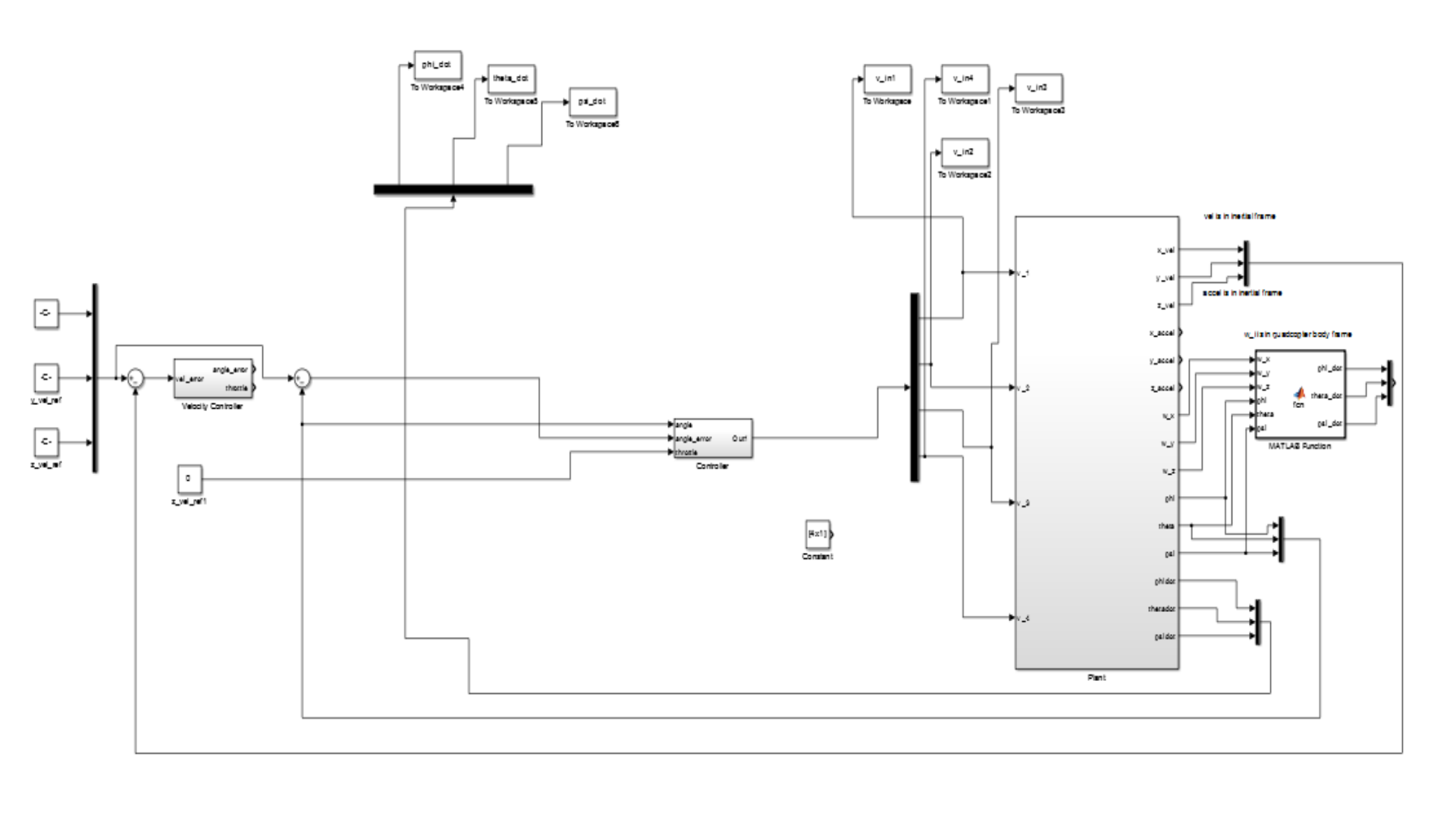
$$\omega = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \dot{\theta}$$

source: <http://andrew.gibiansky.com/downloads/pdf/Quadcopter%20Dynamics,%20Simulation,%20and%20Control.pdf> Feb. 21, 2013, Gibiansky, Andrew

Andrew Gibiansky: A current student at Harvey Mudd College studying mathematics and 'computational everything'. Extensive background in computer science. Former Intern at Google and Counsyl.




1
angles
phi
theta
psi
fcn
Y
MATLAB Function
Product1
phi_scope
To Workspace1
MATLAB Function1
Product
1
s
Integrator
1
phi
phi
To Workspace
2
phidot
2
w_quadcopter

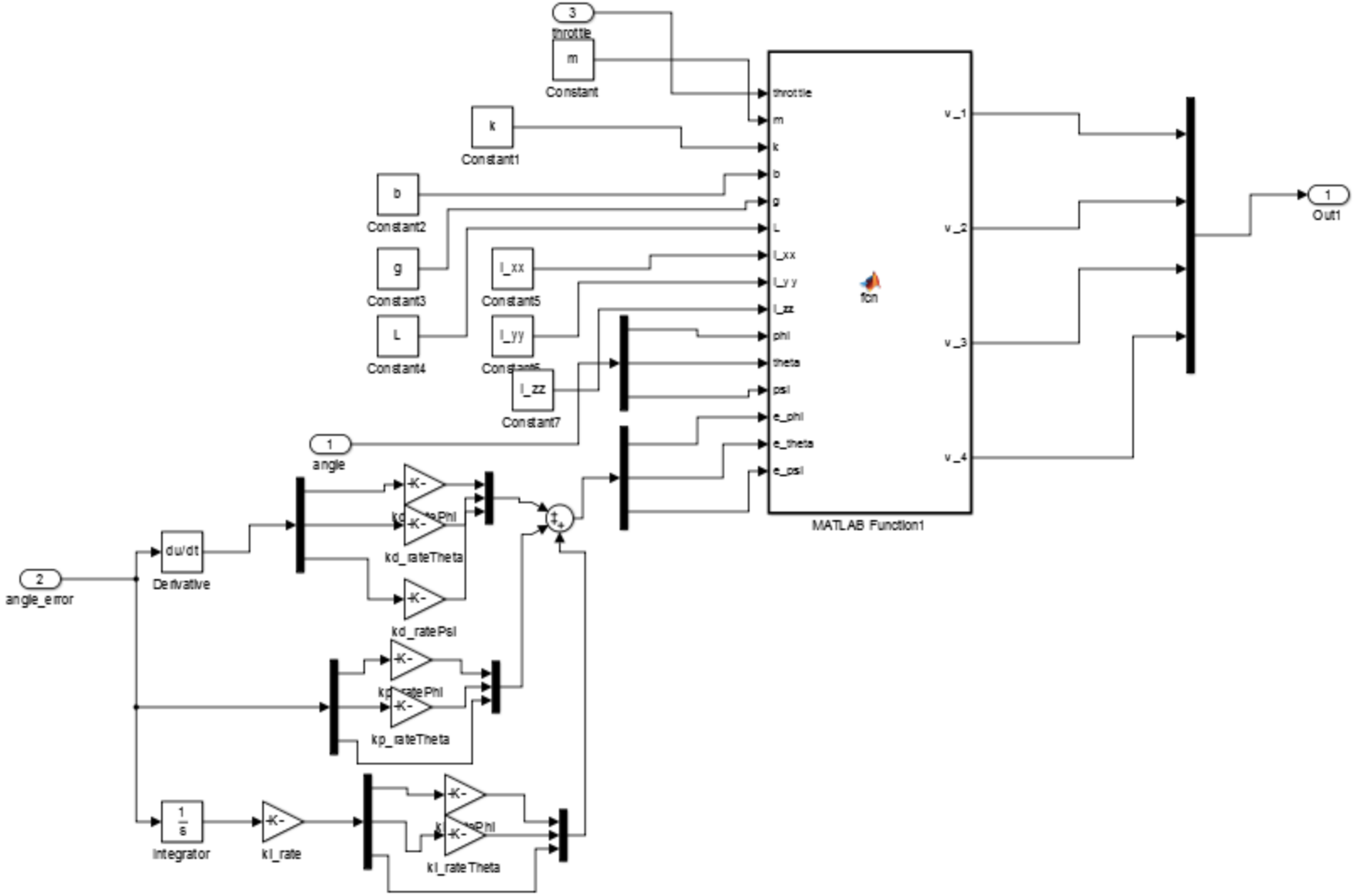

phi_dot
To Workspace4
theta_dot
To Workspace5
psi_dot
To Workspace6
v_in1
To Workspace
v_in4
To Workspace1
v_in3
To Workspace3
v_in2
To Workspace2
vel is in inertial frame
accel is in inertial frame
w_i is in quadcopter body frame
MATLAB Function
fcn
Plant
Constant
Velocity Controller
Controller
x_vel_ref
y_vel_ref
z_vel_ref
x_vel_ref1

$$\gamma_1 = \frac{mg}{4k\cos\theta\cos\phi} - \frac{2be_\phi I_{xx} + e_\psi I_{zz} kL}{4bkL}$$

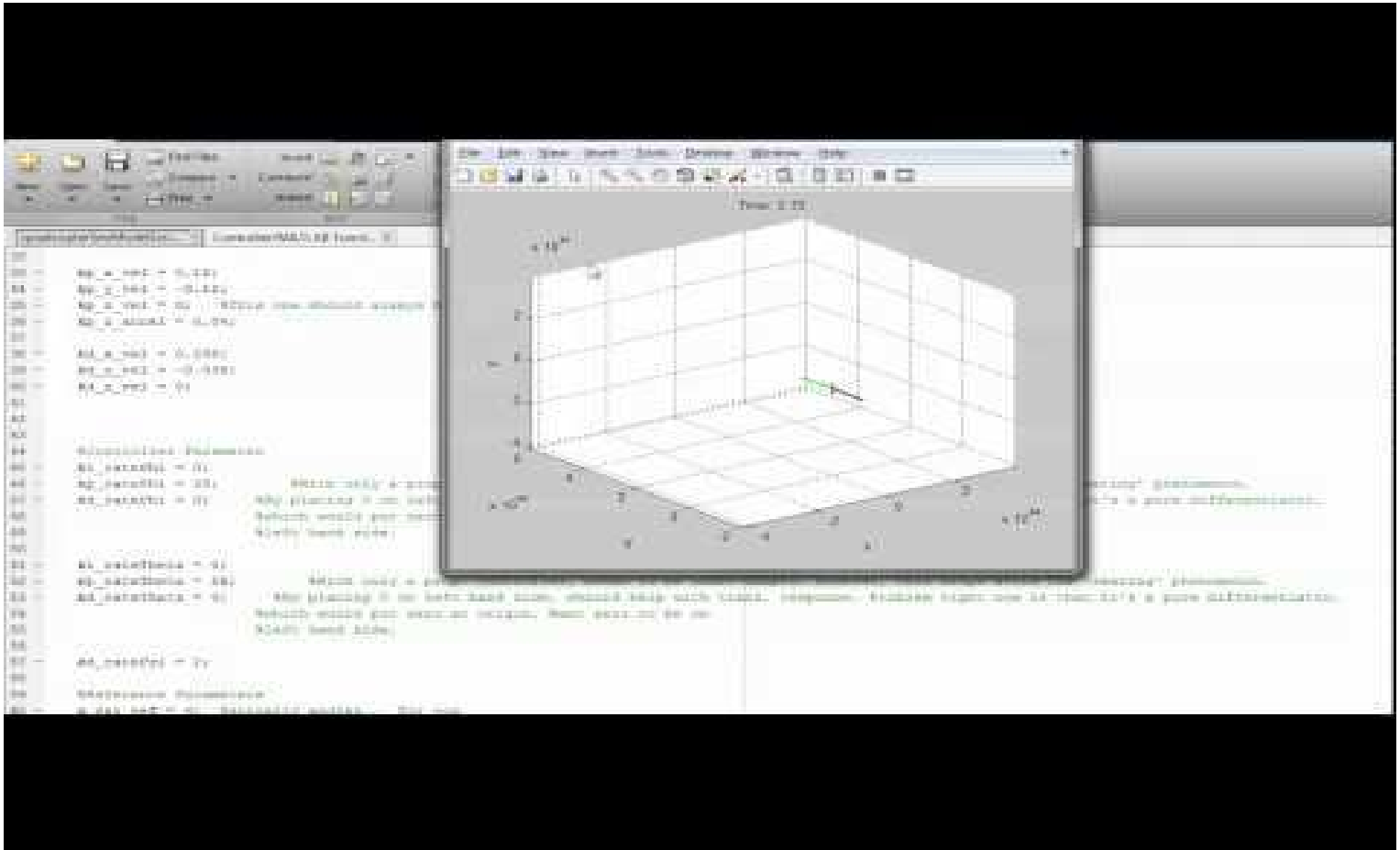
$$\gamma_2 = \frac{mg}{4k\cos\theta\cos\phi} + \frac{e_\psi I_{zz}}{4b} - \frac{e_\theta I_{yy}}{2kL}$$

$$\gamma_3 = \frac{mg}{4k\cos\theta\cos\phi} - \frac{-2be_\phi I_{xx} + e_\psi I_{zz} kL}{4bkL}$$

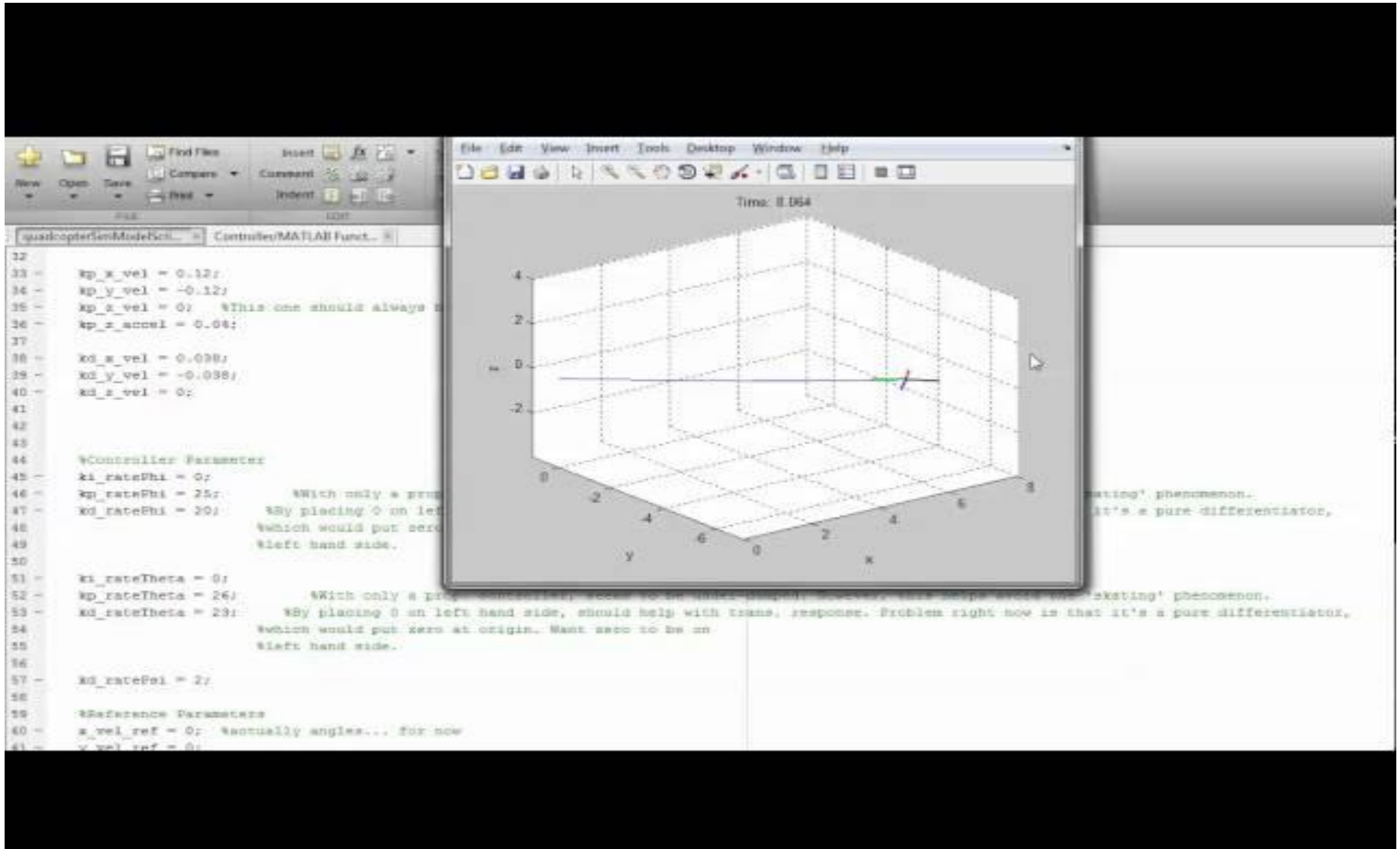
$$\gamma_4 = \frac{mg}{4k\cos\theta\cos\phi} + \frac{e_\psi I_{zz}}{4b} + \frac{e_\theta I_{yy}}{2kL}$$


3
throttle
m
Constant
k
Constant1
b
Constant2
g
Constant3
L
Constant4
I_xx
Constant5
I_yy
Constant6
I_zz
Constant7
throttle
m
k
b
g
L
I_xx
I_y y
I_zz
phi
theta
psi
e_phi
e_theta
e_psi
fcn
v_1
v_2
v_3
v_4
MATLAB Function1
1
Out1
1
angle
2
angle_error
du/dt
Derivative
-K-
kd_rateTheta
kd_ratePsi
kp_rateTheta
1
s
Integrator
ki_rate
ki_rateTheta

Simulation: P Controller



Simulation: PD Controller



Implementation

- Our onboard processor use the Timer1 Arduino library for a single timed interrupt to poll sensors, receive user input, and calculate and actuate motor outputs.
- The ServoTimer2 Arduino library was used to time the PWM signals to each motor.

Implementation

- Sensors were calibrated using an affine model.
- An exponential smoothing filter was used to filter the accelerometer readings.
- For the angular velocity readings, and the extrapolated angle values from the angular velocity and acceleration, we used a Kalman filter, implemented by Kristian Lauszus of TKJ Electronics, to help reduce noise.

Source: https://github.com/TKJElectronics/KalmanFilter/

Implementation

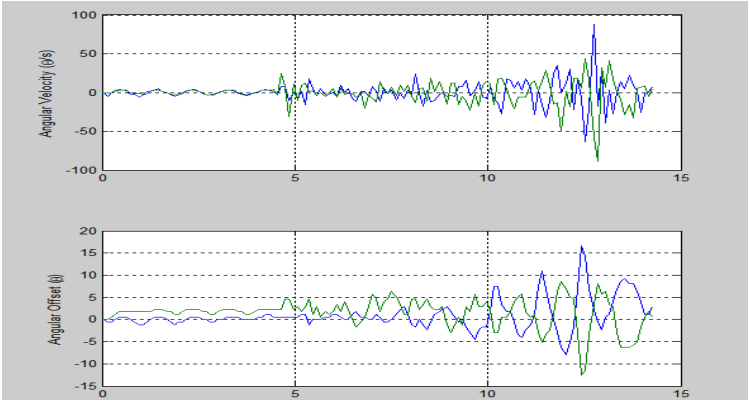
- A simple communication protocol was used to send signals from the user, to the onboard processor.
- There were 4 bytes of information that we had to send, but since the processor reads a byte at a time from a sequence, we needed to be able to show where every four bytes began and ended.
- We abstracted the information into packets, with a simple byte header marking where the packet began, and a checksum at the end to ensure data integrity.

Implementation

- We wanted to have an insight of the internal state
- We had onboard XBee send state information in a CSV format when not servicing interrupts, and used MATLAB to plot the values.

Verification

- Plotting state variables with Matlab



- Harness to minimize risk of crashing while collecting data from hardware.

Post Mortem

- Large source of noise is due to unbalanced propellers.
- We were afraid to try our project for a long time without restraints.

Next Steps

- Buy a tool to properly balance propellers to improve performance
- Test with outside disturbances.
- Add finer grain control with a joystick.
- Add some type of collision avoidance