Autonomous Control of a Commercial UAV

Project by:

Eric Adamany, Richard Dolan

Instructed by:

Dr. Sankalp Bhan

Overview

Tasks

- Autonomous flight control of Parrot Mambo Drone
- Vertical takeoff to an altitude of one meter
- Maintain position for five seconds
- Command a two-meter radius orbit, while maintaining altitude
- Stop after completion of one orbit
- Maintain position for five seconds
- Land drone at point of takeoff

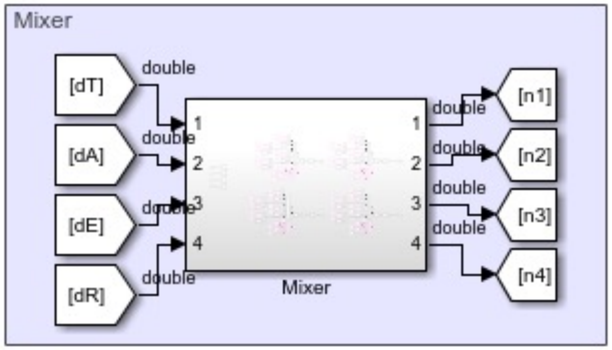
Sensors

- 2x Video cameras (optical flow → speed)
- 2x Ultrasonic Sensors (altitude)
- Pressure sensor (altitude)
- Gyroscopes (angular velocity)
- 3-axis accelerometer (acceleration)

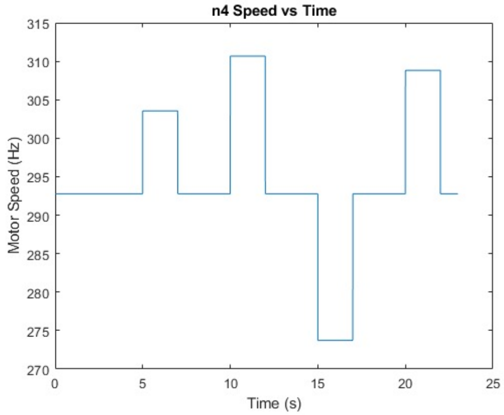
Implementation

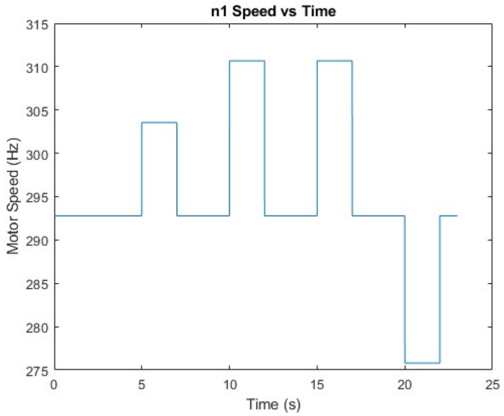
- Linearized control system (Simulation)
    - Plant consists of A, B, C, and D matrices derived from linearized equations of motion.
    - Closed-loop feedback control of all ten decoupled states
    - Commanded by throttle, aileron, elevator, and rudder
    - Used to test gains designed via pole placement and LQR approaches
- Non-linear control system (Simulation)
    - Closed-loop feedback control of nine decoupled states through block diagrams representing nonlinear equations of motion
- Flight test control system (Application)
    - Closed-loop feedback control of six of the system's states
    - State estimates derived from signals input from sensors on drone.
    - Drone interfaced with MATLAB & Simulink via software package & Bluetooth connection

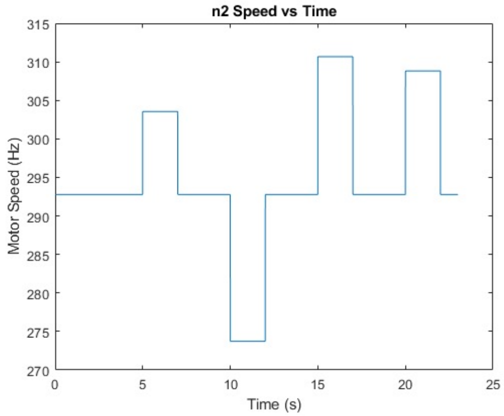
Motor Mixing

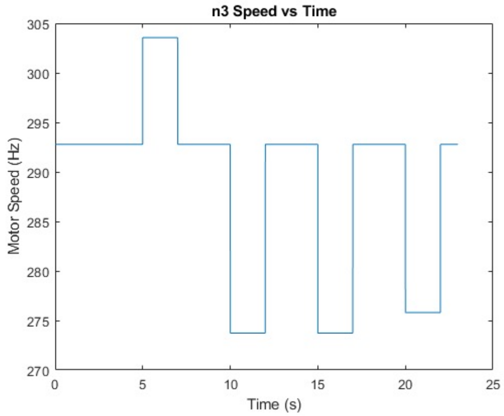


- @T = 5sec: Thrust step applied.
- @T = 10sec: Aileron step applied.
- @T = 15sec: Elevator step applied.
- @T = 20sec: Yaw step applied.





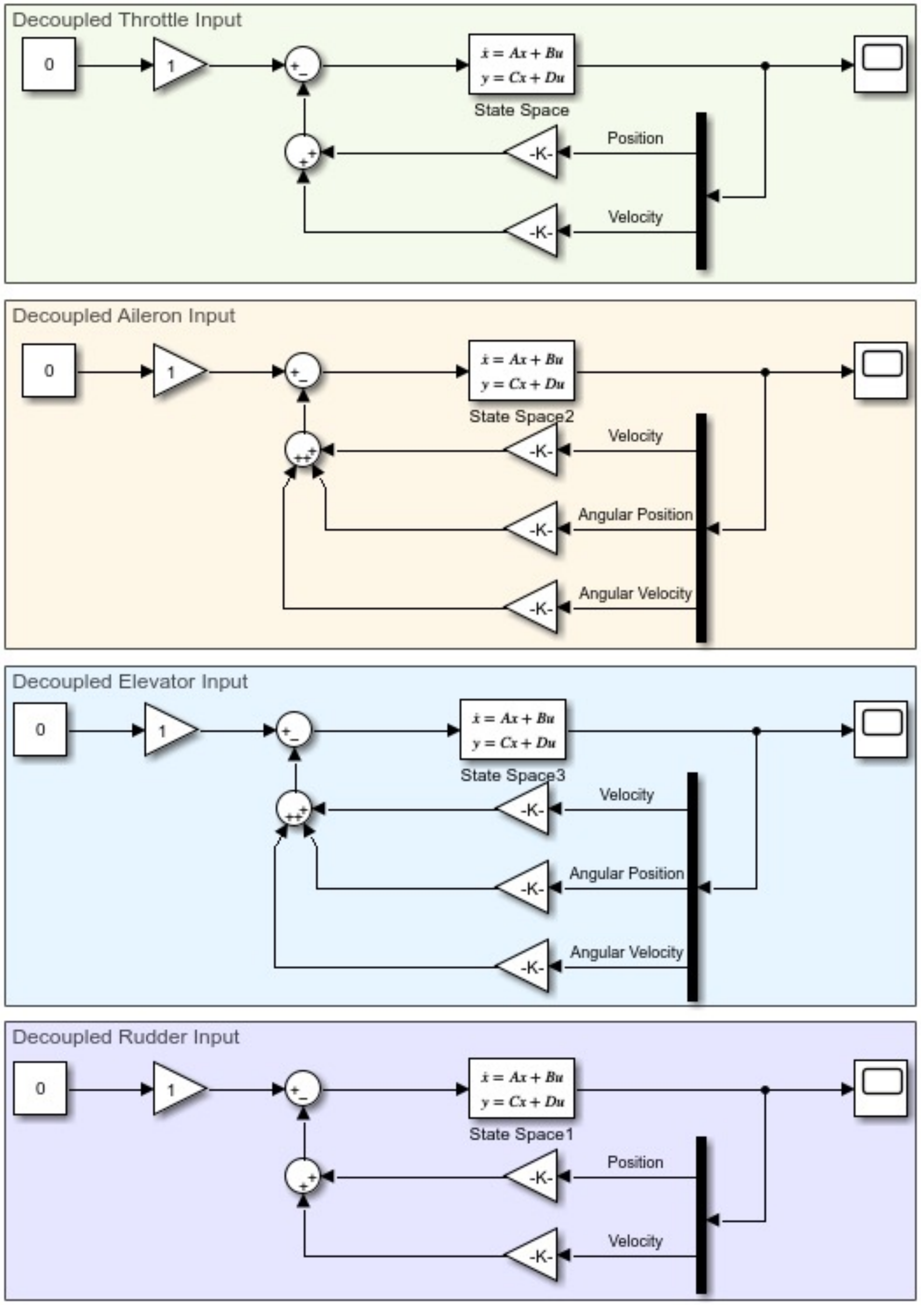




Linearized Control System

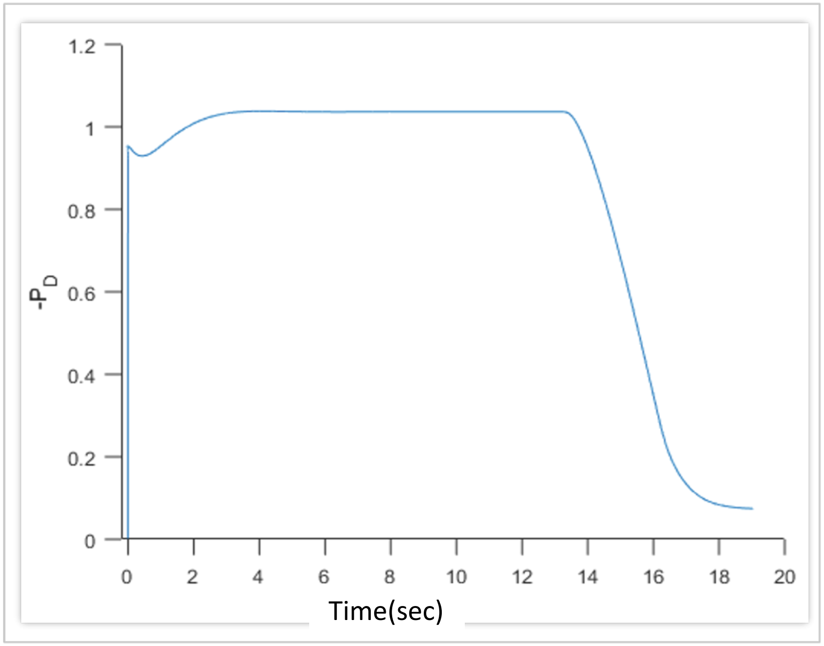
Assumptions

- 69F ambient temperature
- Forces along the X & Y axes not considered
- Air is at rest relative to the Earth
- Commanded motor speeds = ground truth motor speeds
- Propellers have constant moment of inertia
- Symmetric distribution of weight
- Geometry of a 2-D circle
- Earth's surface approximated as flat
- Gravity is uniform



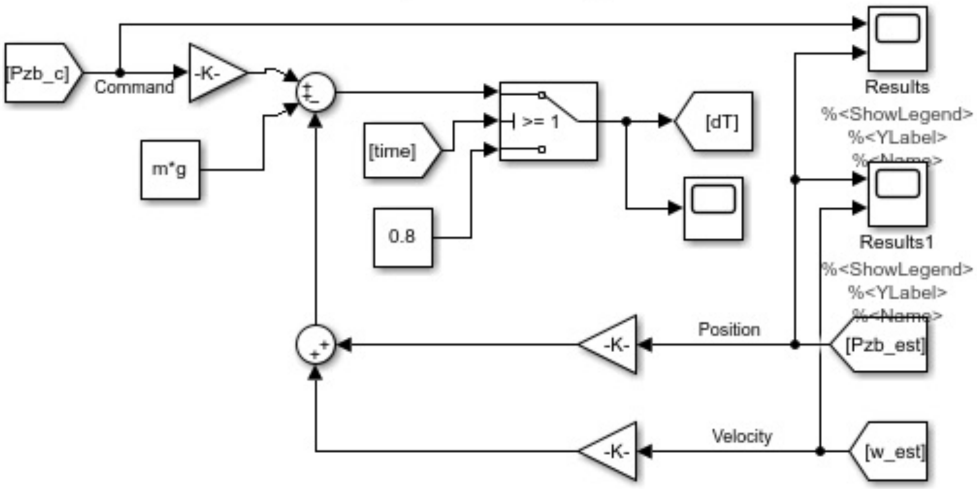
Take off, Hover, and Land via Simulation

- Using the nonlinear simulation with a P_{zb} Command of -1 at T = 0.
- Five seconds later, a command of 1 is given to u in order to create a 1m/s forward velocity.
- Starting at T=14sec, P_{zb} is slowly decreased to prepare for landing.

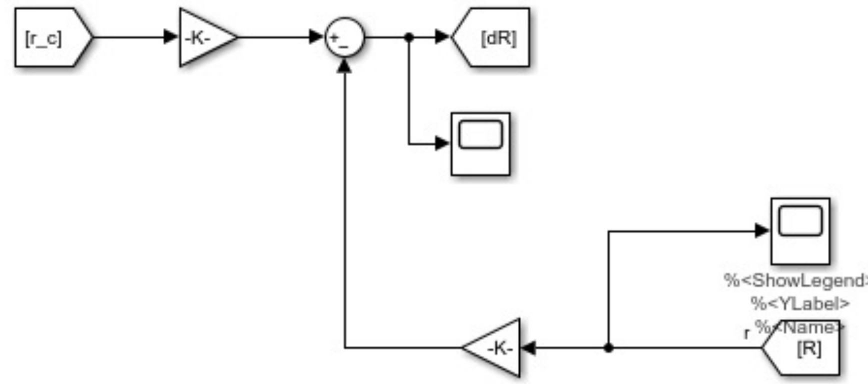


Nonlinear Control System

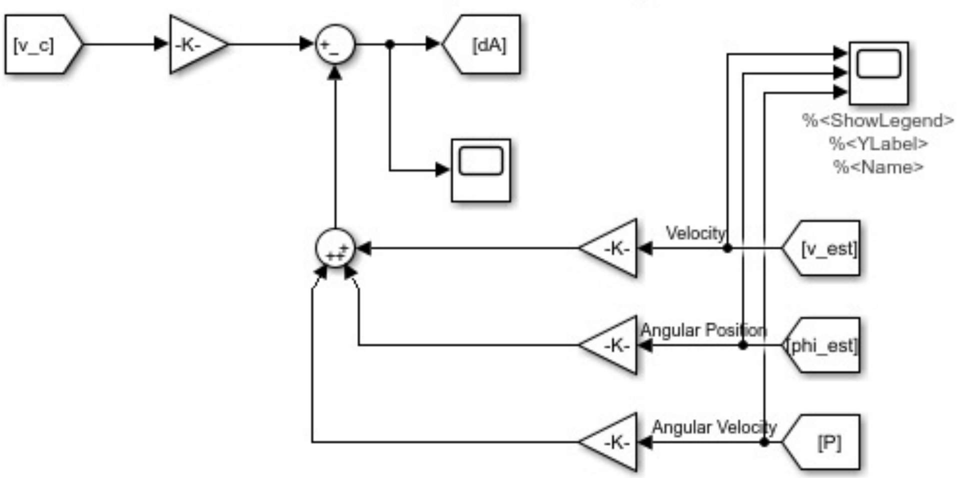
Control System - Decoupled Throttle Input



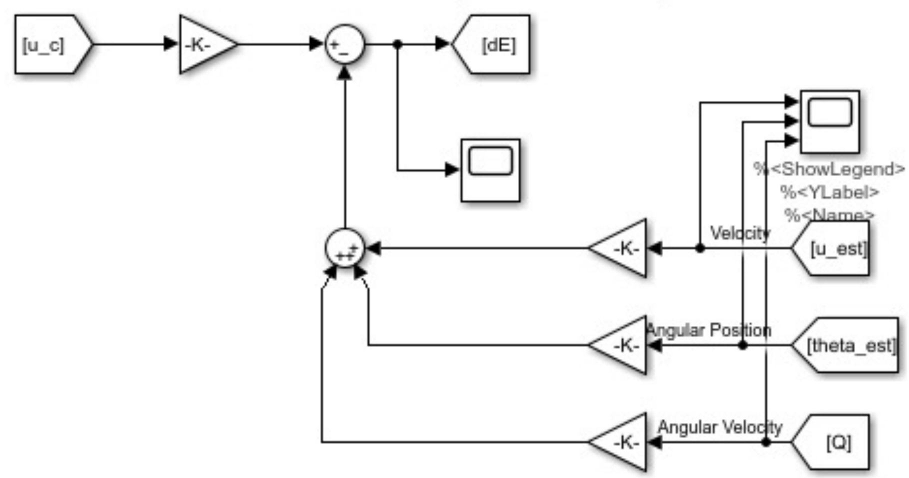
Control System - Decoupled Rudder Input



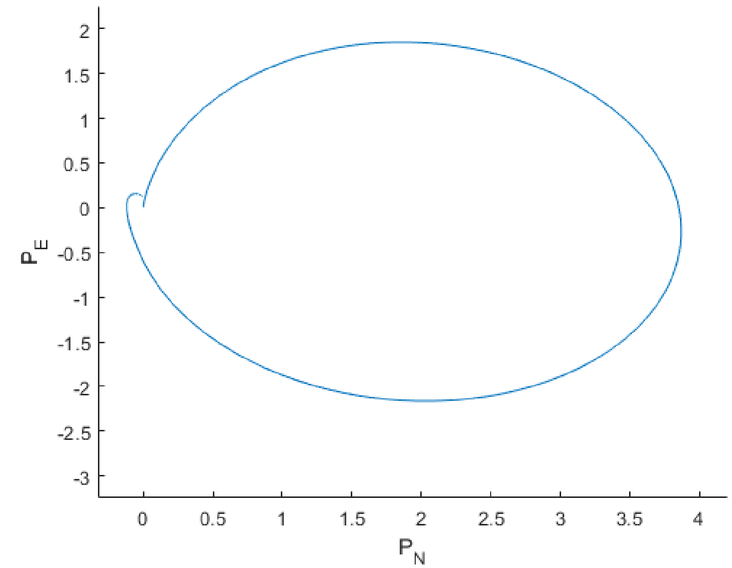
Control System - Decoupled Aileron Input



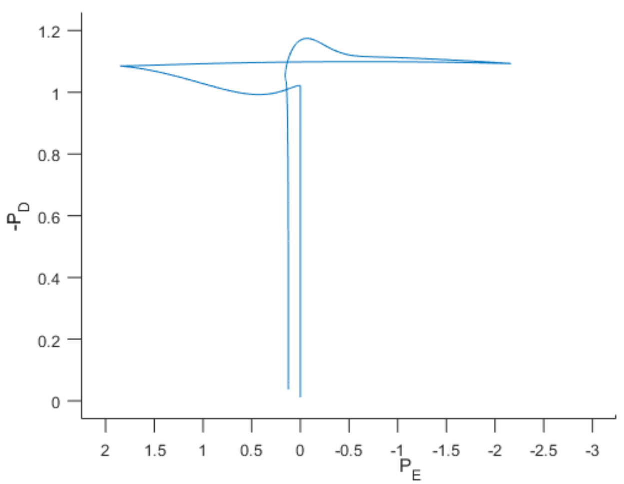
Control System - Decoupled Elevator Input

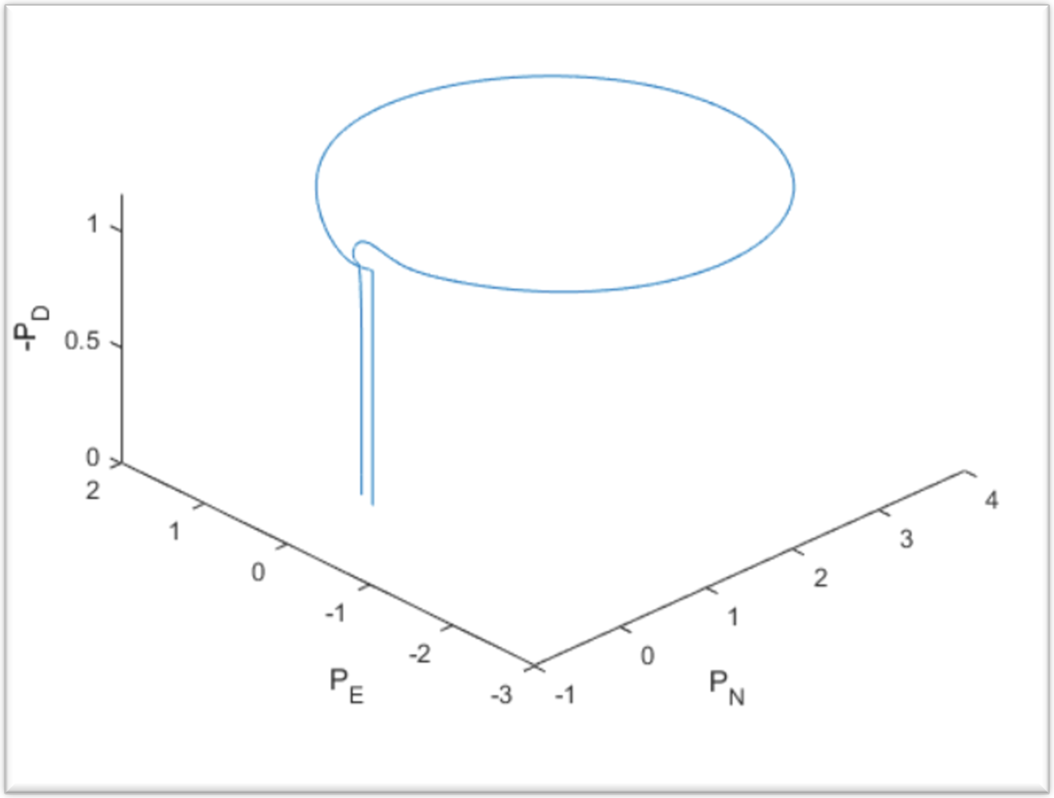


Orbit via Simulation



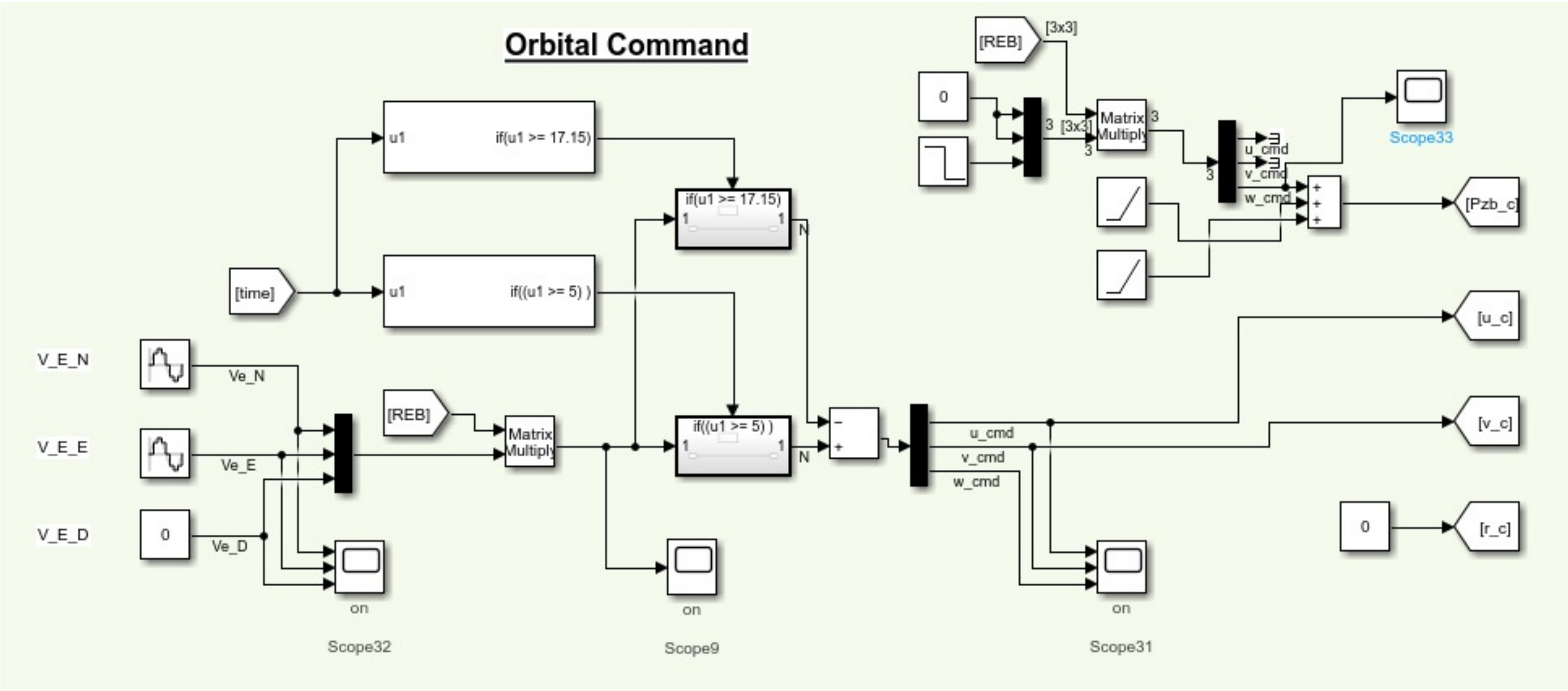
- Using the same nonlinear simulation, the command for P_{zb} was given and hover took place.
- Sine and cosine functions command states *u* and *v* in a circular, two-meter radius orbit





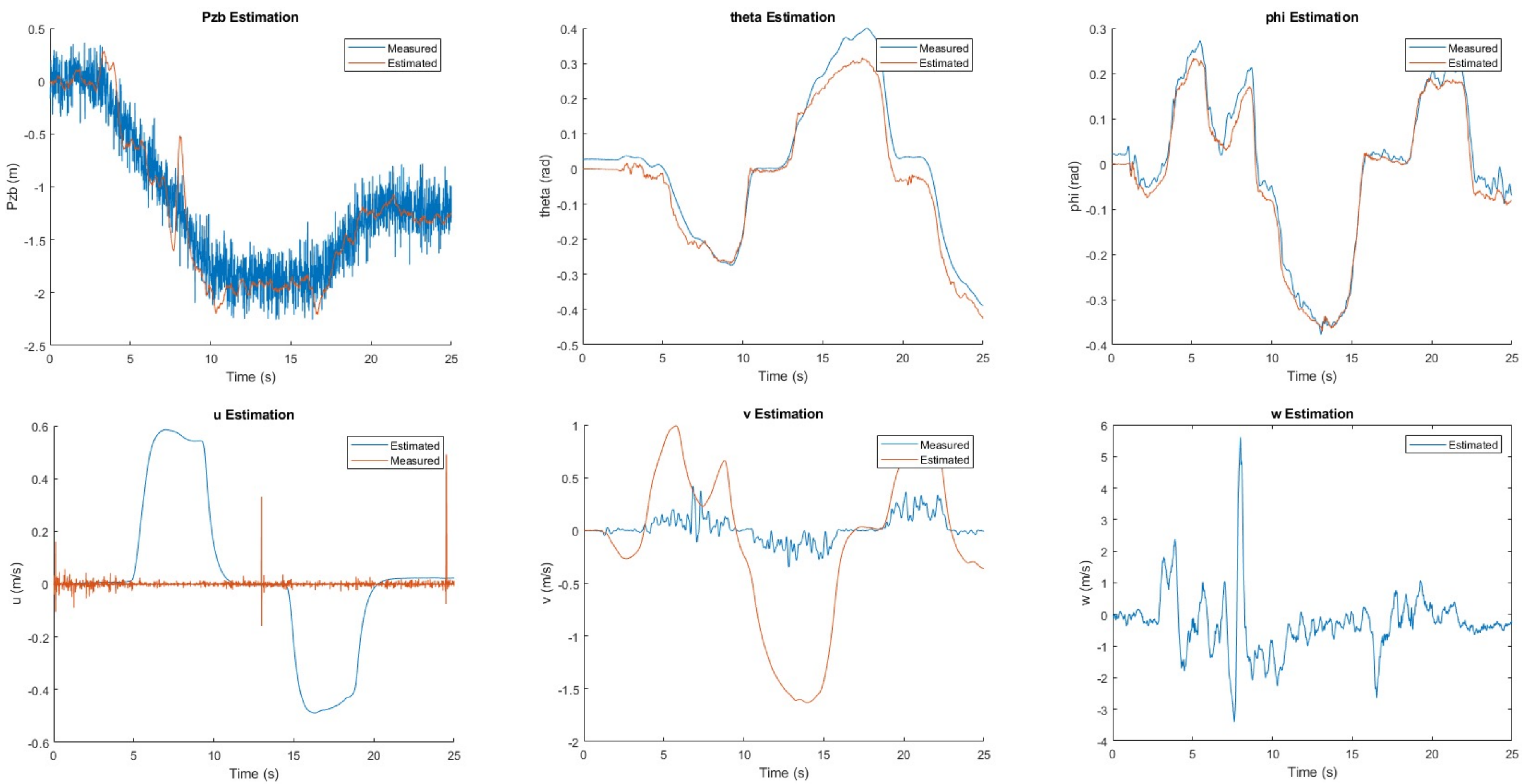
Orbit via Simulation

- Using the same nonlinear simulation, the command for P_{zb} was given and hover took place.
- Sine and cosine functions command states *u* and *v* in a circular, two-meter radius orbit



State Estimation

- Kalman filters process signals input from the drone's sensors to estimate six of the drone's actual states at a frequency of 200Hz.



State Estimation – Kalman Filters

Estimated States:

- P_{zb} – Position along the Z-Axis in the body frame
- u – Velocity along the X-Axis in the body frame
- v – Velocity along the Y-Axis in the body frame
- w – Velocity along the Z-Axis in the body frame
- Φ – Roll angle
- Θ – Pitch angle

States (measured only):

- p – body axis roll rate
- q – body axis pitch rate