Design, Simulation and Testing of a Linear Quadratic Regulator for Attitude Control of a Fin-Tab Rocket

David P. Reynolds*, Karsten Caillet†, Matthew Foley‡, Aditya Swami §, Andrés Chiva Ruiz ¶
Georgia Institute of Technology, Atlanta, GA, 30332

This paper details an actively controlled fin-tab rocket developed by the Georgia Tech Guidance Navigation and Controls (GNC) team, a project team of the Ramblin' Rocket Club (RRC), designed to achieve precise attitude control and trajectory correction. The rocket has a 3-meter fuselage and four fins, each driven by high-torque servo actuators. An aerodynamic model was developed to characterize the relationship between fin deflection angles and the resulting control moments. The servo actuators were characterized through a first order model to approximate the servo dynamics. Based on the aerodynamic model, a gain-scheduled Linear Quadratic Regulator (LQR) controller was developed to correct attitude errors in flight. An Extended Kalman Filter was implemented for state estimation, fusing data from an onboard Inertial Measurement Unit (IMU) and GPS, providing real-time estimates of the rocket's states. Finally, a Monte Carlo simulation was run with varying thrust, mass, actuator dynamics, aerodynamics, and wind conditions to validate the controller's robustness.

I. Introduction

The Guidance, Navigation, and Control (GNC) team at the Georgia Institute of Technology aims to develop, simulate and validate a fin-tab-controlled rocket for active attitude stabilization. The fin-tab rocket has a 3-meter fuselage with four fins located near the base, each equipped with independently actuated surfaces to deflect the wind stream to aerodynamics moments. The development of an active attitude control system for a rocket presents a non-trivial engineering challenge due to the rapidly changing dynamics of the rocket. As the motor burns, the moment of inertia and center of mass shift at a nonlinear rate, while the rocket simultaneously traverses a varying atmosphere at a continuously changing speed. These factors necessitate a control architecture that is capable of maintaining stability and performance throughout the entire flight profile. To address this, a gain scheduled Linear Quadratic Regulator (LQR) is used with 47 breakpoints defined along a nominal trajectory. These breakpoints are used to simplify the nonlinear six-degree-of-freedom dynamics into a Linear Time-Invariant (LTI) system from which linear control can be implemented. The development of an accurate state estimator is also essential, as the controller relies on real-time knowledge of the vehicle's states to compute the appropriate control input. To this end, a pre-flight attitude estimator, as well as an in-flight Extended Kalman Filter were developed. The testing and validation of both are hinged upon developing a high-fidelity simulation of the rocket. Controller performance is ensured through specific mission requirements. The simulated flight yaw angle must not exceed 3.5° deviation during the yaw maneuver angle of 20 degrees and must remain within the same margin once the rocket resumes straight flight. The simulation framework developed to this end is described in a companion paper [1]. In the following sections, we detail the high-fidelity modeling of the rocket's dynamics, the synthesis of the gain-scheduled LQR controller, the development and validation state estimation algorithms, and the performance metrics that demonstrate active attitude stabilization.

II. Dynamical Modeling

A. States and Frames

The rocket is simulated on GNCSIM [1], the team's custom rocket simulation framework. It makes use of 14 state variables: mass (m), position in the inertial frame ($\vec{p}_I$), velocity in the body frame ($\vec{v}_B$), a quaternion representing the

*GNC Controls Co-Lead, Graduate, Daniel Guggenheim School of Aerospace Engineering, AIAA Student Member 1789019
†GNC Controls Co-Lead, Undergraduate, Daniel Guggenheim School of Aerospace Engineering, AIAA Student Member 1537725
‡Controls Team Member, Undergraduate, Daniel Guggenheim School of Aerospace Engineering, AIAA Student Member 1931532
§Controls Team Member, Undergraduate, Daniel Guggenheim School of Aerospace Engineering, AIAA Student Member 1930856
¶Controls Team Member, Undergraduate, Daniel Guggenheim School of Aerospace Engineering, AIAA Student Member 1930708

orientation from the inertial frame to the body frame ($\vec{q}_{IB}$) and the angular rate in the body frame ($\vec{\omega}_B$). The inertial frame follows the East (x), North (y), Up (z) (ENU) convention, and the body frame has x pointed out of the nose, y aligned with North at launch, and z completing the right hand rule [1].

The 14 state variables follow a set of ordinary differential equations (ODE) which assume the inertial frame is fixed to the ground, gravity is constant, no Coriolis acceleration due to the Earth's rotation, and rigid body dynamics, except mass which is not constant, as it varies with thrust [1].

B. Acting Forces and Moments

The dynamics of the vehicle are governed by four primary sources which generate forces and moments on the body. The first is gravitational force, defined in the inertial frame as $\mathbf{F}_I^g = [0, 0, -m(t)g]^T$, where $m(t)$ is the mass at the current time and g is the standard acceleration due to gravity (9.81 m/s^2). The second force is thrust from the AeroTech M650W motor, which is modeled in the body frame as $\mathbf{F}_B^T = [T(t), 0, 0]^T$, where $T(t)$ is obtained from the thrust curve of the motor, shown in Fig. 1. Both thrust and gravity do not generate a moment on the vehicle as they are assumed to act through the rocket's center of mass. Additionally, forces and moments on all three axes arise from the passive aerodynamics of the vehicle (F_B^a, M_B^a), which are obtained from the rocket geometry and fin geometry through the use of Digital DATCOM ([1]). Finally, deflections of each of the four fin-tabs generate active forces and moments on all three axes of the rocket (F_B^d, M_B^d) and enable the control of the rocket's attitude during flight. With $\mathbf{C}_{BI}(\vec{q}_{IB})$ representing the rotation from inertial to body frame, we obtain the total forces and moments applied at the center of mass, and expressed in the body frame:

$$F_B = \mathbf{C}_{BI}(\vec{q}_{IB})F_I^g + F_B^a + F_B^T + F_B^d \tag{1}$$

$$M_B = M_B^a + M_B^d \tag{2}$$

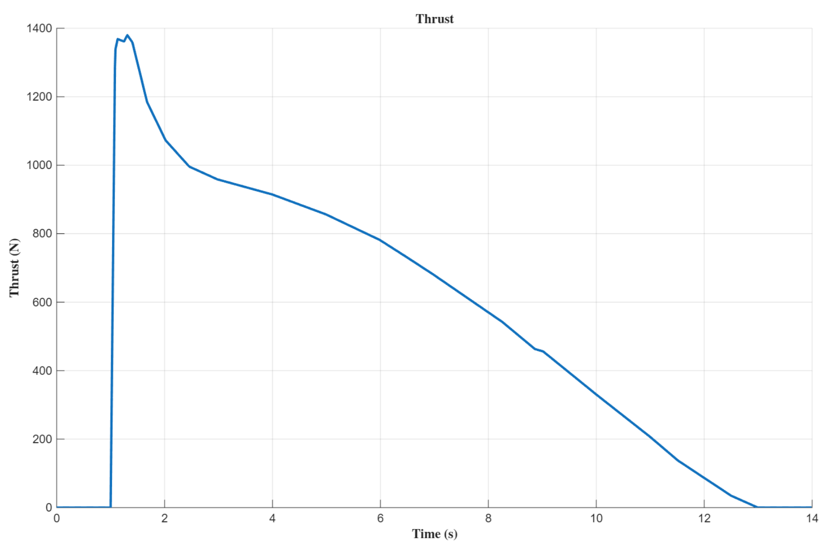

Fig. 1 AeroTech M650W Thrust Curve.

C. Control Allocation

To translate the control efforts for roll, pitch, and yaw into fin deflections, a control allocation matrix is employed. This matrix is used to map the control inputs of pitch, yaw, and roll into fin deflections as shown by the formula below.

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} u_{roll} \\ u_{pitch} \\ u_{yaw} \end{bmatrix} \tag{3}$$

In this configuration, the roll command deflects two actuators, generating a roll moment to act on the rocket. The pitch and yaw commands deflect pairs of fins differentially to generate the required pitching and yawing moments. This allocation scheme leverages the geometric symmetry of the fin layout to decouple the roll, pitch and yaw axes. The fin deflection commands are then used to move the actuators, causing an aerodynamic moment to act on the vehicle.

D. Actuator Model

In order to accurately model the servo actuator on the rocket, a first-order model approximation is used with maximum deflection and maximum deflection rate. The time constant (τ), maximum rate ($\dot{\delta}_{\text{max}}$) and maximum deflection (δ_{max}) were gathered from the datasheet of the servos. Further system identification will be necessary to confirm these parameters. The actuators are shown as part of the fin-tab assembly in Fig 2. The following equations show the actuator dynamics given a commanded deflection (δ_{cmd}):

$$\dot{\delta}(t) = \frac{1}{\tau}(\delta_{\text{cmd}} - \delta(t)) \tag{4}$$

$$\dot{\delta}(t) \leq \dot{\delta}_{\text{max}} \tag{5}$$

$$-\delta_{\text{max}} \leq \delta(t) \leq \delta_{\text{max}} \tag{6}$$

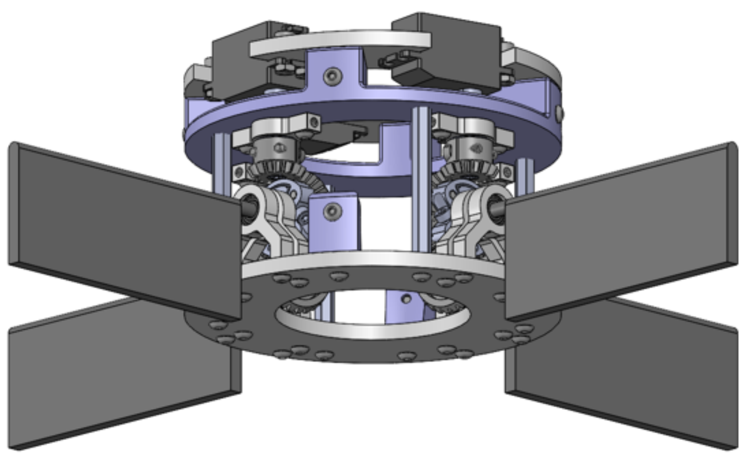

Fig. 2 fin-tab Assembly.

III. Controller Design

In order to design a controller, a dynamical system of the rocket must be represented in the form $\dot{\vec{x}} = f(\vec{x}, \vec{u}, t)$ where f represents a set of 14-state, nonlinear ordinary differential equations governing the vehicle's motion, and $\vec{u}$ is the vector of control commands for pitch, yaw, and roll. To develop a Linear Time-Invariant (LTI) controller, the nonlinear dynamics must be linearized using breakpoints. These points are derived from a nominal trajectory, which is obtained through Hermite-Simpson collocation [1], [2]. For the flight of the fin-tab rocket, a total of 47 breakpoints are used, which equates to approximately two breakpoints per second during ascent. This density of breakpoints was chosen for good observed attitude tracking performance in simulation, while minimizing the computational overhead associated with excessive breakpoints. A linear controller is developed for every breakpoint indexed by fractional accumulated velocity. The gains are linearly interpolated during the flight to prevent high-frequency chatter when switching between controllers. This ensures that the linear controller remains valid throughout the flight profile.

A. Reduced-Order Model

To develop the longitudinal and lateral control laws, the 14-state nonlinear dynamics are simplified into a reduced-order, 3-state Linear Time-Invariant (LTI) model. This reduced order model uses angle of attack (α), pitch rate (q), and pitch (θ).

$$\begin{aligned} \alpha &= \text{atan2}(v_z, v_x) \\ q &= \omega_y \\ \theta &= \sin^{-1}(2(q_3 q_1 - q_0 q_2)) \end{aligned} \tag{7}$$

To achieve this, a function $g(\vec{x}_{14})$ was defined to take in the full 14-dimensional state vector ($\vec{x}_{14}$) and convert it to the 3-dimensional state vector of the longitudinal dynamics ($\vec{x}_3$). Because the rocket is axially symmetric, the longitudinal and lateral dynamics are identical, allowing for the same reduced order model to be applied for both dynamics.

$$\vec{x}_3 = g(\vec{x}_{14}) \tag{8}$$

The full-order system is then linearized around a breakpoint using finite difference to obtain Jacobians of the dynamics and control matrix [3].

$$A_{14} = \frac{\partial f}{\partial \vec{x}_{14}} \qquad B_{14} = \frac{\partial f}{\partial \vec{u}} \tag{9}$$

The transformation J_g is defined as a mapping from the 14-dimensional state space to the reduced 3-dimensional space of longitudinal dynamics. The transformation T is defined as the Moore-Penrose pseudoinverse of J_g to lift the reduced state back into the full-order space [4].

$$J_g = \frac{\partial g}{\partial \vec{x}_{14}} \qquad T = J_g^{\dagger} \tag{10}$$

The final system matrix of the reduced order system (A_3) and the control matrix (B_3) are found and the LTI system can be developed.

$$A_3 = J_g A_{14} T \qquad B_3 = J_g B_{14} \tag{11}$$

$$\dot{\vec{x}}_3 = A_3 \vec{x}_3 + B_3 \vec{u} \tag{12}$$

The development of the reduced-order model for the roll channel follows an identical procedure, with the state vector including only roll rate (p) and roll angle (ϕ).

B. Controller

For every breakpoint, a Linear-Quadratic Regulator (LQR) is used to regulate the rocket along a nominal trajectory. The LQR controller was chosen for its simple implementation in MATLAB, ability to handle multi-input multi-output (MIMO) systems, and ease of tuning. The tuning is done by setting the weights of a positive-semidefinite symmetric matrix Q and a positive-definite symmetric matrix R [5]. The Q matrix defines the scaling cost for the state, and the R matrix defines the scaling cost for control. For a continuous-time infinite-horizon LQR the cost function is defined as follows.

$$J = \int_0^{\infty} (\vec{x}^{\intercal} Q \vec{x} + \vec{u}^{\intercal} R \vec{u})\, dt \tag{13}$$

The resulting control law is defined as

$$\vec{u} = -K(\vec{x} - \vec{x}_{ref}) \tag{14}$$

Where K is a matrix found by solving the Algebraic Riccati Equation (ARE), $\vec{x}$ is the reduced state vector and $\vec{x}_{ref}$ is the nominal reduced state vector that the controller attempts to regulate to. Solving the Riccati equation is done by using the lqr command in Matlab. Using the reduced order models, three independent LQR controllers can be developed to control for pitch, yaw and roll in a decoupled manner. The weighting matrices Q and R are defined for the reduced order model. For the 3-state longitudinal and lateral controllers the Q and R matrices are defined as follows:

$$Q_3 = \text{diag}(q_\alpha, q_q, q_\theta) = \text{diag}(1,1,1) \qquad R_3 = r_\delta = 2 \tag{15}$$

Similarly, for the 2-state roll channel, the matrices are defined as:

$$Q_2 = \text{diag}(q_p, q_\phi) = \text{diag}(1,1) \qquad R_{roll} = r_\delta = 20 \tag{16}$$

By using the reduced order models (A_3, B_3, A_2, B_2), the weighting matrices (Q_3, R_3, Q_2, R_2), the gain matrix K_3 and K_2 are computed for each respective channel. This decoupling allows for independent tuning of the performance for pitch, yaw, and roll.

At every time step, the full state is mapped into the reduced states to define the error for each controller. The error is used to get the control using the K matrix. Then each control command is passed through the allocation matrix H to obtain fin command deflections. Passing these fin deflection commands through the actuator model deflects the airstream to generate the necessary control moments.

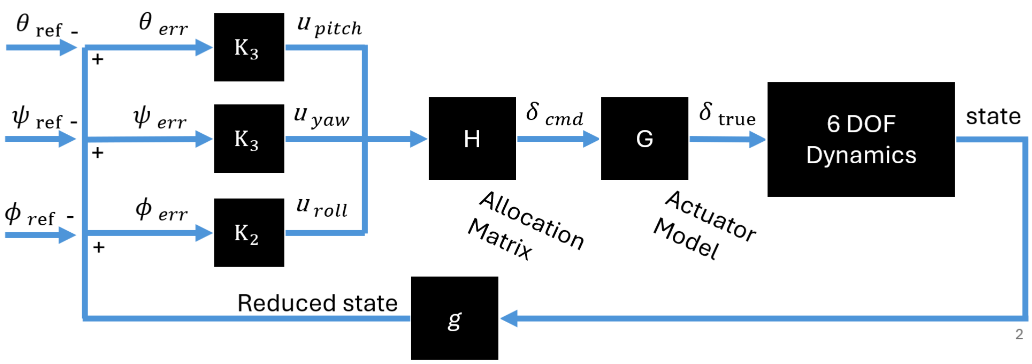

Fig. 3 Control architecture for Fin-Tab Rocket.

IV. State Estimation

Closed-loop control of the fin-tab assembly relies on precise vehicle attitude, position, and velocity data. These states are estimated by fusing measurements from an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS). The 6-axis IMU provides measurements of specific force and angular velocity at a high rate, allowing for continuous state propagation; the GPS supplies a lower-frequency position measurement to reduce inertial drift. The state estimation architecture separates the preflight calibration from this in-flight filtering. First, sensor biases and scale factors are identified on the ground, and the initial attitude is established using magnetometer and accelerometer data. Then, during flight, an extended Kalman filter (EKF) propagates the translational states using the accelerometer measurements with corrections to the position from the GPS. Attitude estimation is achieved through gyroscope dead reckoning, where the quaternion is integrated forward using the angular rate measurements, with no attitude correction from an external sensor. An initial attitude estimate is obtained on the launch pad using measurements of gravity and the Earth's magnetic field. These vectors are measured in the body frame and compared with their known representations in the inertial frame. The resulting attitude estimation problem is formulated as Wahba's problem and solved using Davenport's Q-method to obtain the quaternion that rotates the inertial frame into the body frame.

A. Bias Estimation

Low-cost commercial-off-the-shelf (COTS) sensors exhibit substantial bias that are the dominant factor in error growth for short-term navigation. Due to the short flight time, the startup biases are treated as constant throughout the flight and can be estimated immediately before flight while the vehicle is on the ground. With the vehicle fixed on the ground, IMU data is collected where the true angular rate is zero and the only specific force acting on the IMU is gravity. A linear Kalman Filter is used to estimate the constant bias state defined as the following [6].

$$\vec{x_b} = [\beta_a^T \beta_\omega^T] \tag{17}$$

In this state $\beta_a, \beta_\omega \in R^3$ are the turn-on biases of the accelerometer and the gyroscope, respectively. Because the system is static, the discrete-time process model represents a constant bias system.

$$\vec{x}_{b,k+1} = F\vec{x}_{b,k}, \quad F = I_6 \tag{18}$$

The measurement model directly relates the raw IMU outputs to the bias state with gyroscope measurements modeled as bias alone, while accelerometer measurements are modeled as bias plus the gravity vector expressed in the body frame. The final bias estimate is stored to be subtracted from all IMU measurements during flight.

The magnetometer is calibrated separately to account for hard iron and soft iron distortions prior to flight. The magnetometer measurement model is given by [7].

$$\vec{\tilde{m}} = S_I^{-1}\left(\vec{m}_E + \vec{m}_{e(t)}\right) + \vec{b}_{HI} + \vec{m}_{i(t)} \tag{19}$$

where $\vec{\tilde{m}}$ is the measured magnetic field vector in the body frame, $\vec{m}_E$ is the true Earth magnetic field vector, $\vec{m}_{e(t)}$ represents external time-varying magnetic disturbances, $\vec{b}_{HI}$ is the hard iron bias vector, $S_I \in \mathbb{R}^{3\times 3}$ is the soft iron distortion matrix, and $\vec{m}_{i(t)}$ represents measurement noise.

A compensation model is then applied to remove these effects:

$$\vec{m}_c = S_I\left(\vec{\tilde{m}} - \vec{b}_{HI}\right) \tag{20}$$

Rather than developing a custom magnetometer calibration procedure, open-source calibration tools such as Magneto can be used to estimate the soft iron scaling matrix and hard iron offsets from collected calibration data prior to flight.

B. Scale Factor Determination

In addition to additive bias, inertial sensors often exhibit scale-factor errors that impact measurements. These errors are treated as fixed and measured offline prior to flight. Accelerometer scale factors are determined by aligning each sensor axis both parallel and anti-parallel to the gravity vector, and comparing the corrected measurement magnitudes to the known local gravity. Gyroscope scale factors are identified using known-rate rotations along each axis of the IMU. The bias is first subtracted from the raw measurement, after which the scale factor correction is applied. This ordering ensures that the scale factor is applied only to the true signal and not to the additive offset. The resulting corrected signals are then used by the EKF.

C. Initial Attitude Estimation

An accurate initial attitude is required to correctly rotate body-frame accelerations into the flat-Earth frame. After bias calibration, the vehicle remains stationary while an initial orientation is computed using Davenport's Q-method. The accelerometer provides a measurement of the local gravity direction, while the magnetometer provides a measurement of the Earth's magnetic field vector. These measured body-frame vectors are paired with their corresponding known inertial-frame reference vectors to form Wahba's problem.

Wahba's problem determines the rotation $R \in SO(3)$ that minimizes the weighted least-squares cost [8].

$$J(R) = \frac{1}{2}\sum_{i=1}^{N} w_i \left\|\vec{b}_i - R\vec{r}_i\right\|^2, \tag{21}$$

where $\vec{b}_i$ are measured body-frame vectors, $\vec{r}_i$ are known inertial-frame reference vectors, and w_i are scalar weights. This minimization is equivalent to maximizing

$$\max_R \ \mathrm{tr}(RB^T), \tag{22}$$

where

$$B = \sum_{i=1}^{N} w_i \vec{b}_i \vec{r}_i^T. \tag{23}$$

Davenport's Q-method solves this problem by forming the matrix

$$S = B + B^T, \qquad \sigma = \mathrm{tr}(B) \tag{24}$$

$$\vec{Z} = \begin{bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{bmatrix}, \tag{25}$$

and constructing the Davenport K-matrix

$$K = \begin{bmatrix} S - \sigma I_3 & \vec{Z} \\ \vec{Z}^T & \sigma \end{bmatrix}. \tag{26}$$

The optimal quaternion $\vec{q}_{opt}$ is the unit eigenvector corresponding to the maximum eigenvalue $\lambda_{\max}$ of

$$K\vec{q} = \lambda\vec{q}, \tag{27}$$

with normalization

$$\|\vec{q}_{opt}\| = 1. \tag{28}$$

This formulation allows explicit weighting of each measurement vector, enabling greater confidence to be placed on higher-quality measurements while reducing sensitivity to noisier measurements such as the magnetometer. The resulting quaternion is normalized and used to initialize the in-flight attitude estimate for propagation within the Extended Kalman Filter.

D. In-Flight State Estimation

1. State Vector

The Extended Kalman Filter (EKF) estimates on a six-element state vector comprising position and velocity expressed in the ENU inertial frame, $\vec{x} = \begin{bmatrix} \vec{p_I}^T & \vec{v_I}^T \end{bmatrix}^T$. Attitude is tracked separately through a quaternion propagated by the gyroscope integration described below and is thus not included in the EKF state vector.

2. In Flight Attitude Estimation

Attitude is propagated using the strap-down quaternion integration algorithm. At each IMU measurement, the corrected gyroscope output is converted into an instantaneous rotation quaternion. This is calculated as a rotation by angle $\theta = \|\omega\|\Delta t$ about the unit axis $\hat{a} = \omega/\|\omega\|$, giving the incremental rotation quaternion

$$\delta q = \begin{bmatrix} \cos(\theta/2) \\ \hat{a}_x \sin(\theta/2) \\ \hat{a}_y \sin(\theta/2) \\ \hat{a}_z \sin(\theta/2) \end{bmatrix}. \tag{29}$$

The past timestep quaternion is then updated by quaternion multiplication $q_{k+1} = q_k \otimes \delta q$ and renormalized. Because the attitude propagation relies on integrating gyroscope measurements, bias will cause drift over time. The current implementation corrects for known biases using a pre-flight estimated bias vector, but a future enhancement will implement a multiplicative EKF for quaternions.

3. Extended Kalman Filter

The EKF prediction step propagates the state estimate forward using a kinematic model driven by the accelerometer measurement [9]:

$$\vec{p}_{k+1} = \vec{p}_k + \vec{v}_k\Delta t + \vec{a}_I\frac{\Delta t^2}{2} \tag{30}$$

$$\vec{v}_{k+1} = \vec{v}_k + \vec{a}_I\Delta t \tag{31}$$

where the inertial frame acceleration is obtained by first correcting the accelerometer output for the IMU-to-CM lever arm, then rotating from the body frame into the inertial frame, and finally subtracting gravity as shown below.

$$\vec{F}_{s,\mathrm{CM}} = \vec{F}_{s,\mathrm{IMU}} + \vec{\omega}_B \times \vec{v}_{CG} + \vec{\omega}_B \times (\vec{\omega}_B \times \vec{r}_{\mathrm{IMU}}) \tag{32}$$

$$\vec{a}_I = C_{IB}\vec{F}_{s,\text{CM}} + \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} \tag{33}$$

The error covariance is propagated using the discrete-time state-transition matrix derived from the kinematic model,

$$P_{k+1|k} = F\, P_{k|k}\, F^T + Q \tag{34}$$

where Q is the process noise covariance tuned to account for accelerometer noise, σ_a, and unmodeled dynamics.

$$Q = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & 0 & \frac{\Delta t^3}{2} & 0 & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & 0 & \frac{\Delta t^4}{4} & 0 & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & 0 & \Delta t^2 & 0 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & 0 & \Delta t^2 & 0 \\ 0 & 0 & \frac{\Delta t^3}{2} & 0 & 0 & \Delta t^2 \end{bmatrix}$$

The state transition matrix F is obtained from the Jacobian of the process model f with respect to the state x,

$$F = \left.\frac{\partial f}{\partial x}\right|_{x_{k|k}} \tag{35}$$

Since the kinematic model is linear, this Jacobian is exact and evaluates to the closed-form discrete-time matrix,

$$F = \begin{bmatrix} I_3 & \Delta t\, I_3 \\ 0_3 & I_3 \end{bmatrix}$$

where I_3 is the 3×3 identity matrix, 0_3 is the 3×3 zero matrix, and Δt is the discretization timestep.

When a GPS measurement is available, the update step is performed. The observation model selects the position states,

$$H = \begin{bmatrix} I_3 & 0_3 \end{bmatrix}$$

and the Kalman gain is computed as

$$K = P_{k+1|k} H^T \left(H P_{k+1|k} H^T + R \right)^{-1} \tag{36}$$

The state and covariance are then corrected through the measurement step:

$$\vec{x}_{k+1|k+1} = \vec{x}_{k+1|k} + K\left(\vec{z} - H\vec{x}_{k+1|k}\right) \tag{37}$$

$$P_{k+1|k+1} = (I_6 - KH) P_{k+1|k} \tag{38}$$

where z is the GPS measurement and R is the GPS noise covariance constructed using the GPS horizontal and vertical accuracy parameters. Since the GPS outputs at a much slower rate than the IMU, the measurement step is only performed if a GPS measurement is available. Otherwise, the filter propagates the states using only IMU measurements. The state and covariance are recorded at each time step, providing a record of estimation uncertainty useful for algorithm and sensor analysis. Fig. 4 shows in-flight state estimation performance across a 100 runs given the same truth data.

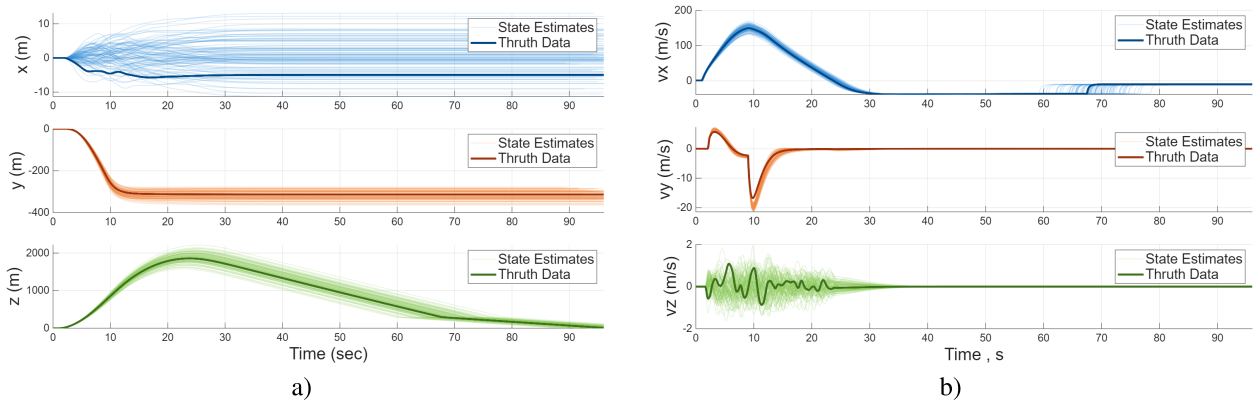

Fig. 4 EKF State Estimation Performance: a) Position, b) Velocity

V. Controller Validation

A. Linearization Validation

In order to validate the linear reduced order models, a comparison was done between the full nonlinear dynamics and the simplified linear model using an impulse response. For a fair comparison, the rocket was initialized with a velocity of 20 m/s along the x body axis, with aerodynamics as the only forces acting on the vehicle. As shown in Figure 5, an impulse was applied to the pitch channel of the rocket to observe the resulting pitch and pitch rate response. The blue line represents the linear response from the reduced order model and the green represents the full nonlinear dynamics. From both responses, there is a near perfect overlap in the time domain suggesting that the reduced order model successfully captures the vehicle's lateral and longitudinal dynamics. Furthermore, a pole zero map confirms the stability of the reduced-order system with two complex conjugate poles on the left-half plane and a single integrator.

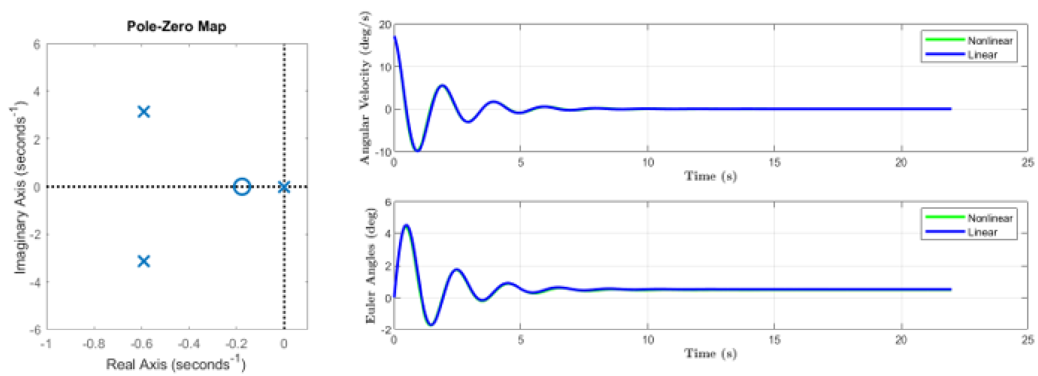

Fig. 5 Impulse Response of linear and nonlinear models for fin-tab Rocket.

B. Monte Carlo

Next, Monte Carlo simulations are run with the LQR controller to verify that the yaw mission requirements are met, ensuring that the vehicle follows the desired attitude profile. This analysis demonstrates how parametric variations in the rocket's physical characteristics can still result in successful mission completion. A Monte Carlo simulation with 1000 runs is conducted using Latin Hypercube Sampling (LHS), which ensures a good sampling spread in parameter space. The simulation varies 46 parameters with ±10% 3σ uncertainty bounds, [10] including aerodynamic scale factors (C_a, C_m, C_n), mass properties (wet and dry mass, longitudinal and rotational moments of inertia, center of gravity locations), motor characteristics (total impulse, burn time, ignition time), geometric parameters (nose length, body length, rocket diameter), actuator dynamics (maximum deflection, rate limits, time constant), parachute drag coefficients, and launch rail parameters (length and orientation), reference wind speed, base wind direction, turbulence intensity and sensor parameters (accelerometer noise σ_a, navigation covariance scale, GPS vertical and horizontal accuracy, and GPS output probability). Critical inter-dependencies between parameters are preserved. For example, when total impulse or burn time are varied, the thrust curve is scaled appropriately in both magnitude and time while maintaining constant specific impulse (I_{sp}). Similarly, dry mass was recalculated based on the perturbed wet mass and total impulse to ensure physically consistent propellant mass across all simulations.

Figure 6 shows the mean and 3-sigma deviation of the yaw angle results alongside the mission requirements. The fin-tab rocket yaw angle successfully fits within requirements for all the samples, which validates the LQR controller performance in simulations, confirming readiness for a real flight.

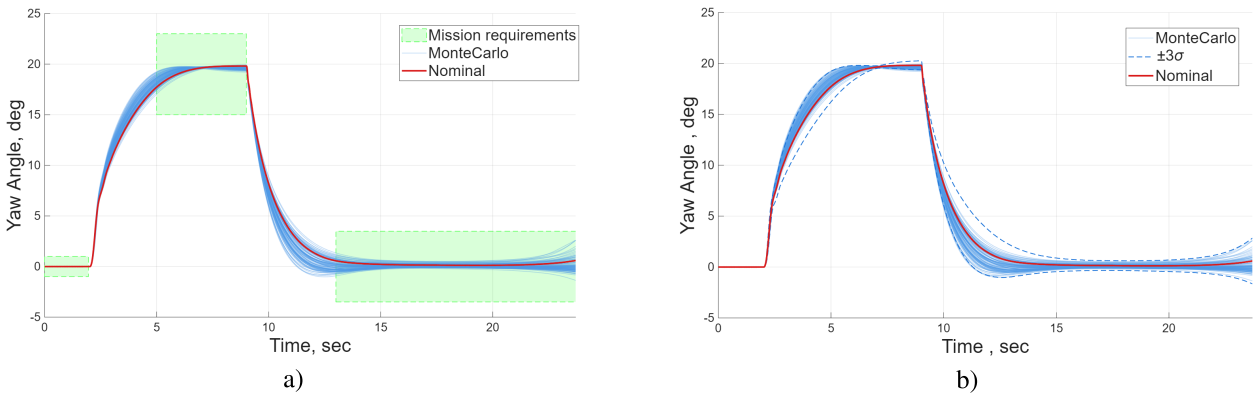

Fig. 6 Yaw angle mean and 3σ distribution with mission requirements for 1000-sample Monte Carlo simulation.

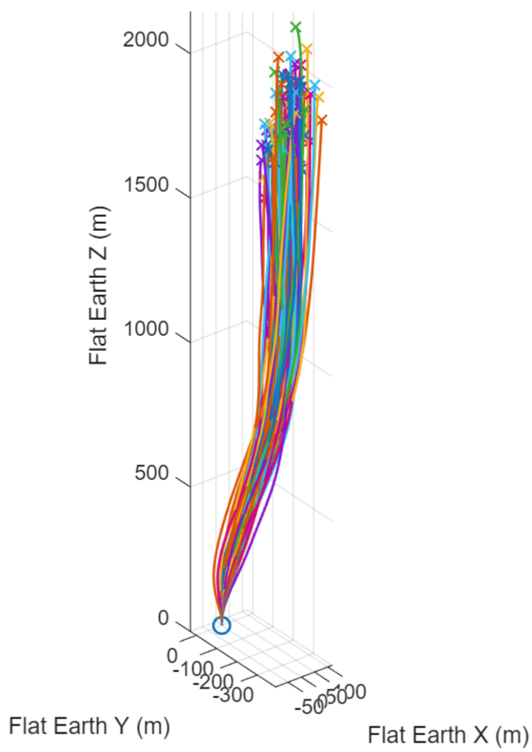

Fig. 7 Closed-loop 3-dimensional trajectories for a 1000-sample Monte Carlo simulation.

VI. Conclusion

This paper presented the design and high-fidelity simulation of an active attitude control system for a fin-tab rocket. By using gain scheduling and a decoupled LQR architecture, the rocket attitude dynamics could successfully be controlled. Specifically, the controller demonstrated the ability to perform a 20 degree yaw maneuver then return back to vertical flight. Furthermore, the integration of an initial attitude estimation and the Extended Kalman Filter provided a robust state estimate, fusing IMU and GPS data. The performance of the GNC architecture was tested by using 1000 Monte Carlo runs that vary 46 physical and environmental parameters, including aerodynamic uncertainties and actuator dynamics. These results confirm that the controller maintains robust stability and successfully bounds attitude errors, proving resilient against ±10% 3σ uncertainty bounds applied across a wide range of aerodynamic, vehicle parameters, and environmental conditions. Future work will focus on Hardware-in-the-Loop (HWIL) testing to further verify that the rocket will meet mission requirements before flight.

References

[1] Caillet, K., Reynolds, D. P., Vadia, K., Foley, C., Swami, A., and Chiva Ruiz, A., "GNCSIM: A High-Fidelity 6-DOF Simulation Framework for Closed-Loop Rocket Navigation and Control," *AIAA Region II Student Conference*, 2026. Submitted.

[2] Kelly, M., "OptimTraj - Trajectory Optimization for Matlab," , 2017. URL `https://github.com/MatthewPeterKelly/OptimTraj`.

[3] Kisabo, A. B., and Adebimpe, A. F., "State-Space Modeling of a Rocket for Optimal Control Design," *InTech Open*, 5 June 2019.

[4] Gu, C., "Model Order Reduction of Nonlinear Dynamical Systems," Technical Report UCB/EECS-2012-217, University of California at Berkeley, Berkeley, CA, USA, December 2012. URL `http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-217.html`.

[5] Ogata, K., *Modern Control Engineering, Fifth Edition*, Pearson, 25 August 2009.

[6] Groves, P. D., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed., Artech House, Norwood, MA, USA, 2013.

[7] VectorNav Technologies LLC, "Magnetometer Hard & Soft Iron Calibration (HSI Calibration)," Inertial Navigation Primer, VectorNav, 2026. Available online: `https://www.vectornav.com/resources/inertial-navigation-primer/specifications--and--error-budgets/specs-hsicalibration`.

[8] Zanetti, R., Ainscough, T., Christian, J., and Spanos, P. D., "Q-Method Extended Kalman Filter," *Proceedings of the AAS/AIAA Space Flight Mechanics Winter Meeting*, Springfield, VA, USA, 2013. NASA Technical Report No. 20120017927, Report No. JSC-CN-27491; available at `https://ntrs.nasa.gov/api/citations/20120017927/downloads/20120017927.pdf`.

[9] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, Springer New York, New York, NY, USA, 2014.

[10] Hanson, J., and Beard, B., "Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Analysis," *NASA Technical Report*, 1 September 2010.