Eric Reyes

22 November 2025

Active Rocket Controls

Applying Control Theory to Our Flight Computer

ME 476C - Section 001

Introduction

The Active Rocket Control (ARC) capstone aims to design and build a rocket with an actively controlled flight system. This incorporates a flight computer that utilizes an onboard inertial measurement unit (IMU) sensor to regulate the rocket's orientation during flight. Active control allows for the rocket to stay within the ±2° range of roll for z-axis optimization and improves overall flight stability that passive stabilization cannot achieve.

As demonstrated in *Multi-Mode Electric Actuator Dynamic Modelling for Missile Fin Control*, closed-loop fin control using gyroscopic feedback is used in guided flight systems (like our flight computer that interfaces with an IMU sensor), where actuators are regulated through proportional-derivative control loops to maintain stability [1].

Being mindful of the standard of closed-loop fin control through gyro capabilities, this report is an analytical study of the flight computer that explores how a closed-loop roll-rate control system can be designed and implemented for the ARC rocket. Using control-theory methods, the analysis evaluates whether such a system is feasible given the available avionics, IMU sensing capabilities, and fin-actuation hardware.

Assumptions

IMU Assumptions

The flight computer uses the *Adafruit LSM6DSO32 6 DoF Accelerometer and Gyroscope* sensor [2]. This provides tri-axial angular velocity and acceleration measurements. For this analysis, the following assumptions can be made based on the manufacturer's datasheet:

Angular rate measurement range: ±2000 °/s

Acceleration measurement range: ±32 g

Sampling frequency: $f \geq 200$ Hz

Latency: negligible because the actuator and airframe response will be much higher

IMU orientation: the gyro's z-axis will be aligned with rocket roll axis

Small-angle assumption: the motion about the roll axis is small enough for dynamics to be linearized

These assumptions consider the IMU to be an ideal angular-rate sensor in the feedback loop like standard gyroscopic sensor implementations noted in *Multi-Mode Electric Actuator Dynamic Modelling for Missile Fin Control* [1].

Physical Assumptions

With assumptions of the IMU sensor in mind, physical modeling assumptions also must be made for this analysis. These assumptions define how the rocket's roll dynamics, actuator behavior, and control inputs should be represented mathematically, which will then allow for the closed-loop system to be analyzed using control-theory standard methods:

Rigid-Body: the rocket is modeled as a rigid body rotating about its longitudinal axis. Structural flexes are neglected.

Small-Angle Linearization: like with the IMU, roll motion is assumed to be within the ±2° requirement, allowing for linear roll dynamics to be represented using first and second order transfer functions from control theory.

Moment of Inertia: The rocket's moment of inertia about the roll axis, I, is assumed to be constant over the duration of flight where roll control is active.

Fin-generated Roll Torque: this is assumed to be linearly proportional to the fin deflection angle

Fin actuator: the servo is represented as a first-order system, which means it responds smoothly with finite speed

External disturbances: wind and thrust misalignment are neglected to be able to simplify closed-loop behavior

These assumptions allow for roll dynamics to be written in the standard rigid-body form:

$$I\dot{\omega} = \tau_{fin} \qquad (1)$$

which allows us to use classical control modeling for the closed-loop roll-rate controller [1], [3].

Modeling

Based on the assumptions that were made previously, the flight computer and fin-actuated roll control system can be represented as a linear closed-loop control system. The following models describe the rocket roll dynamics, the controller of the flight computer, and the actuator that produces the fin-generated torque.

Roll Dynamics

From rigid body rotational dynamics, roll-rate about the rocket's longitudinal axis can be defined with

$$I\dot{\omega}(t) = \tau_{fin}(t) \qquad (2)$$

Where I is the moment of inertia about the roll axis, $\dot{\omega}(t)$ is the roll rate, and $\tau_{fin}(t)$ is the torque generated from the active fins. Assuming fin behavior is linear, fin torque is proportional to fin deflection angle $\delta(t)$,

$$\tau_{fin}(t) = K_f\delta(t) \qquad (3)$$

K_f is the effective fin torque coefficient. Taking the Laplace transform of (2)-(3) and solving for the transfer function from fin deflection to roll rate gets

$$P(s) = \frac{\Omega(s)}{\Delta(s)} = \frac{K_f}{Is} \quad (4)$$

This is the standard single-integrator roll-rate model used in classical control systems [3].

Controller Model (PD Control)

A proportional-derivative (PD) controller is chosen because it is commonly used in fin-actuated aerospace systems [1]. Its implementation is also very straightforward on flight computers. The controller computes fin deflection based on roll-rate error as shown:

$$e(t) = \omega_{\text{cmd}}(t) - \omega(t) \quad (5)$$

The PD control law can be written as

$$\delta(t) = K_p e(t) + K_d \dot{e}(t) \quad (6)$$

Where K_p is the proportional gain and K_d is the derivative gain. In the Laplace domain, the controller transfer function becomes

$$C(s) = \frac{\varDelta(s)}{E(s)} = K_p + K_d s \quad (7)$$

For roll stabilization, the commanded roll rate is $\omega_{\text{cmd}}(t) = 0$, so the controller is aiming to drive the measured roll rate toward zero for stability.

Actuator Model

The fin actuator or servo is modeled as a first-order system with time constant τ_a, which represents that the fin cannot move instantaneously to the commanded angle. The actuator transfer function is written as

$$A(s) = \frac{\varDelta_{\text{fin}}(s)}{\varDelta_{\text{cmd}}(s)} = \frac{1}{\tau_a s+1} \quad (8)$$

$\varDelta_{\text{cmd}}(s)$ is the commanded fin angle from the controller and $\varDelta_{\text{fin}}(s)$ is the actual fin angle applied to the rocket. This form aligns with simplified actuator models used in PD-controlled fin systems [1]. The actuator dynamics are represented through τ_a. If this value is small, or in other words, the actuator response is fast, the fin is assumed to track the commanded angle closely.

Closed-Loop Roll-Rate Control System

Combining previous models of the controller C(s), actuator A(s), and roll dynamics P(s) in a unity-feedback configuration results in the closed-loop transfer function from commanded roll rate $\Omega_{cmd}(s)$ to the actual roll rate $\Omega(s)$,

$$T(s) = \frac{\Omega(s)}{\Omega_{\mathrm{cmd}}(s)} = \frac{C(s)A(s)P(s)}{1+C(s)A(s)P(s)} \qquad (9)$$

And so, substituting (4), (7), and (8) into (9) gives

$$T(s) = \frac{\left(K_p+K_d s\right)\left(\frac{1}{\tau_a s+1}\right)\left(\frac{K_f}{Is}\right)}{1+\left(K_p+K_d s\right)\left(\frac{1}{\tau_a s+1}\right)\left(\frac{K_f}{Is}\right)} \qquad (10)$$

This final expression represents the simplified closed-loop roll-rate dynamics under the assumption that were previously mentioned. This is consistent with standard feedback formulations used in control theory [3] and with the gyro-stabilized fin-control model presented in *Multi-Mode Electric Actuator Dynamic Modelling for Missile Fin Control* [1].

Results

Closed-Loop Transfer Function

The closed-loop transfer function as derived in equation (10) shows that the system can be stabilized by ensuring the proportional and derivative gains are chosen nicely. A PD controller is well-suited for this because it can reduce overshoot while responding quickly to changes in roll rate. As long as the gains are not set too high for the servo's response speed, the rocket will resist unwanted rolling, and the system will stay stable. This confirms that the closed-loop controller is feasible through control-theory.

Rocket Body Relation to Dynamics

The RockSim model of our rocket confirms that the body is symmetric and rigid, which supports the assumption made earlier on the rocket's roll dynamics (rigid body, small-angle, linear torque) matches the physical nature of our rocket's design. This figure demonstrates that the mathematical modelling done reflects the actual rocket geometry.

Figure 1: RockSim Model Based on *HI-TECH Precision Rocket*

Flight Computer Hardware

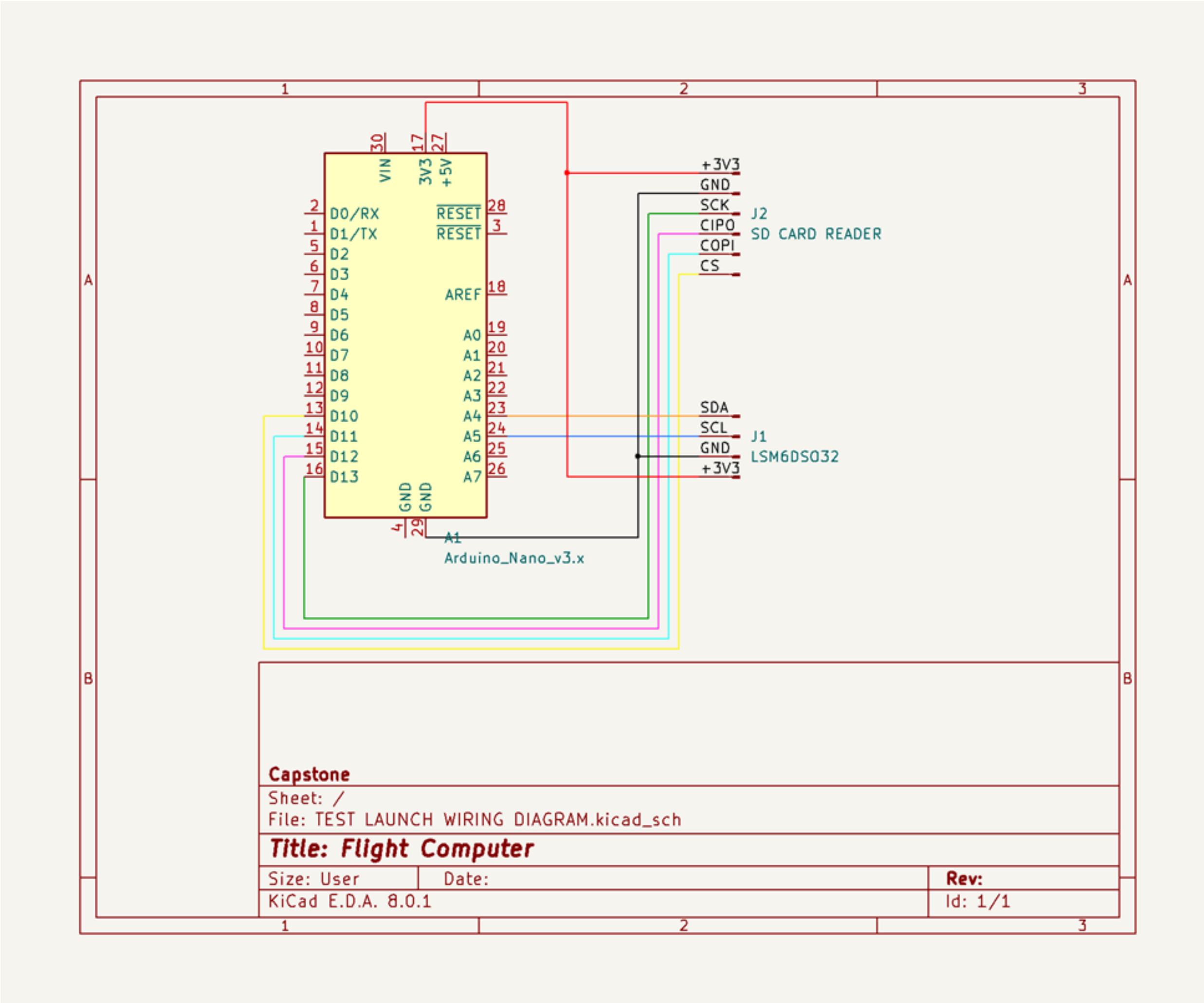

Figure 2: Schematic of ARC Flight-Computer from EE Team

The EE schematic shows how the IMU, Arduino Nano, and SD card reader are wired together. With these parts implemented into the PCB, this hardware allows the flight computer to read IMU roll-rate data, perform calculations, and command the active fin actuators during flight. This hardware list verifies that avionics can support a closed-loop control loop in real time because this implementation of parts allows for the function to happen.

Flight Computer Control Flow

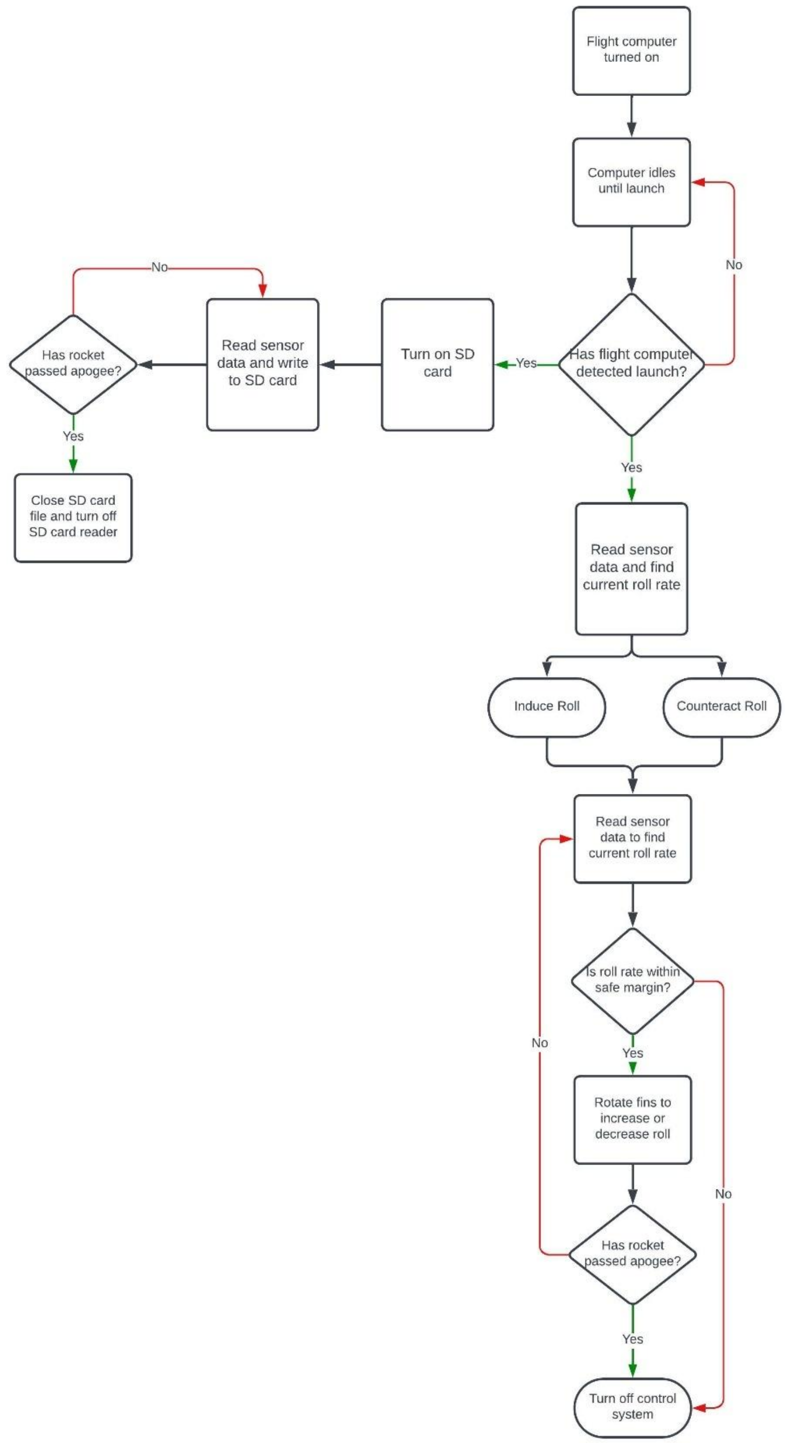

Figure 3: EE division's flowchart depicting flight-computer logic

The EE Team's flowchart shows how the flight computer will actually run the control system. Once launch is detected, the computer will run performing these actions:

1. Read IMU sensor data
2. Compute roll rate and either induce or counteract roll
3. Check if the roll rate is outside the allowed margin
4. Rotate fins as needed
5. Continues until apogee

These processes match the closed-loop behavior predicted in equation (9) and show that the control method can be implemented in software.

Conclusion

The modeling, diagrams, and hardware layout all show that the ARC rocket can support a closed loop roll controller. The IMU provides close to ideal (accurate) measurements, the Arduino can compute PD control quickly, and the fins can respond fast enough to correct roll. Overall, the system is feasible with the available avionics and mechanical design. Moving forward, the EE division of our capstone team can use this analysis as a framework to guide the development of the flight computer source code and implement the closed-loop roll-rate control logic.

References

[1] A. S. Gurav and R. N. Yadav, "Multi-Mode Electric Actuator Dynamic Modelling for Missile Fin Control," *Aerospace*, vol. 4, no. 2, pp. 1–19, 2017. Accessed: Nov. 20, 2025.

[2] Adafruit Industries, "Adafruit LSM6DSO32 6-DoF Accelerometer and Gyroscope – STEMMA QT / Qwiic," Datasheet. Accessed: Nov. 20, 2025.

[3] S. A. Frank, *Control Theory Tutorial: Basic Concepts Illustrated by Software Examples*, 2021. Accessed: Nov. 20, 2025.