Comparative Study of State-Integral, LQR, and MPC-Based Controllers for Missile Guidance and Navigation

Siddarth Dayasagar
Dept. of Electrical and Computer Engineering
Northeastern University, Boston, USA
dayasagar.s@northeastern.edu

Sarthak Talwadkar
Dept. of Electrical and Computer Engineering
Northeastern University, Boston, USA
talwadkar.s@northeastern.edu

Ket Bhikadiya
Dept. of Electrical and Computer Engineering
Northeastern University, Boston, USA
bhikadiya.k@northeastern.edu

Chaitanya Salve
Dept. of Electrical and Computer Engineering
Northeastern University, Boston, USA
salve.c@northeastern.edu

***Abstract*—This paper presents a comparative analysis of three control strategies—State-Integral Control, Linear Quadratic Regulator (LQR), and Model Predictive Control (MPC)—applied to missile guidance and navigation. Each controller is evaluated on its ability to track reference trajectories, minimize control effort, and handle dynamic constraints in a simulated missile environment. Simulation results and performance metrics are presented to quantify the trade-offs among the controllers.**

***Index Terms*—Missile Guidance, Model Predictive Control, LQR, SFI, Optimal Control, Trajectory Tracking**

I. Introduction

Guidance, Navigation, and Control (GNC) is a systems engineering field that deals with controlling the movement and position of vehicles like missiles, spacecraft, and vehicles. It involves guidance, navigation, and control aspects to achieve the desired movement and position. GNC systems are constantly evolving to improve accuracy, response time, and reduce size, weight, and power consumption. Guidance involves determining the desired trajectory and changes in velocity, rotation, and acceleration to reach a target. Navigation involves determining the vehicle's current location, velocity, and attitude (angular position). Control involves applying steering commands to execute guidance instructions while maintaining stability and smoothness of movement.

The landscape of missile guidance, navigation, and control (GNC) is undergoing a transformative shift, needing the robust and optimal control. Some propelled by the integration of artificial intelligence (AI) and machine learning (ML) technologies. Leading defense technology firms, such as Anduril Industries and Shield AI, are at the forefront of this evolution. Anduril's collaboration with OpenAI aims to enhance counter-unmanned aircraft systems (CUAS) by leveraging AI for real-time threat detection and response, thereby improving national security capabilities [1]. Similarly, Shield AI's development of the Hivemind AI pilot enables autonomous operation of uncrewed aerial systems (UAS), facilitating dynamic mission planning and execution without direct human intervention [2].

These advancements underscore a broader trend towards AI-driven autonomy in missile systems, emphasizing the need for control strategies that can adapt to complex and uncertain environments. Traditional control methods, while effective under certain conditions, often lack the flexibility required for modern, agile missile operations.

In this context, our study presents a comparative analysis of three control strategies — State Feedback Integral(SFI), Linear Quadratic Regulator (LQR), and Model Predictive Control (MPC) —applied to missile guidance and navigation. Each controller is evaluated based on its trajectory tracking performance, control effort efficiency, and ability to handle dynamic constraints within a simulated missile environment. By systematically assessing these control methodologies, we aim to identify the most effective approach for enhancing missile GNC in the era of AI-driven autonomy.

II. Related Work

The design of missile autopilot systems has progressively shifted from classical SFI-based control to optimal and robust control methodologies to meet the agility, performance, and robustness demands of modern tactical missiles. Two key references form the basis of this work: Mracek and Ridgely's optimal control interpretation of classical autopilot topologies [3], and Arikapalli et al.'s LQR-LQG formulation for high-speed missile dynamics under uncertainty [4].

In [3], the authors formalize how classical Raytheon-style three-loop autopilots (pitch rate, angle-of-attack, and acceleration feedback) can be derived from optimal control principles. Starting from a second-order, short-period longitudinal missile model, linearized about a trim point, the system dynamics are represented as:

This model behaves as tail controlled missile and explores the different control architecure like LQR and MPC. The structure of the cost function directly influences the control topology: penalizing acceleration error and control effort yields a 2-loop structure, while introducing a control rate penalty transitions the structure to a 3-loop autopilot.

Arikapalli et al. [4] address the challenge of autopilot design under sensor noise, actuator uncertainty, and model inaccuracies. A full 6-DoF nonlinear missile model is first trimmed and linearized about cruise flight conditions to produce a fourth-order state-space system. The authors then design an LQG controller using a cascaded LQR-LQE approach. The Kalman filter is tuned based on known disturbance and noise covariances, and its integration with the LQR controller follows the separation principle. Simulation results confirm that the closed-loop system exhibits significant improvement in disturbance rejection, with reduced settling time and better control effort efficiency, even under corrupted measurements.

Both works validate the use of linearized missile models around operating conditions for controller synthesis. Moreover, they provide foundational frameworks for optimal control that can be extended to predictive control. In contrast to LQR and LQG, which rely on fixed feedback gains, Model Predictive Control (MPC) solves a constrained optimization problem in real-time, explicitly accounting for input limits and terminal objectives. This capability makes MPC particularly well-suited for aggressive terminal-phase guidance and actuator-constrained scenarios addressed in this study.

III. Missile Dynamics Model

The missile model used in this study follows a short-period approximation and is linearized about a nominal trim condition, as presented in [3]. The system is expressed in state-space form as:

$$\dot{x} = Ax + Bu, \quad y = Cx + Du$$

with matrices:

$$\begin{aligned} A &= \begin{bmatrix} -1.064 & 1.0 \\ 290.26 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -0.25 \\ -331.40 \end{bmatrix} \\ C &= \begin{bmatrix} -123.34 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned} \tag{1}$$

Here:

- $x = [\alpha, q]^T$ is the state vector (angle of attack and pitch rate)
- $u = \delta_p$ is the control input (fin deflection)
- $y = [A_z, q]^T$ is the output vector(Acceleration and pitch rate)

While the original source model includes a non-zero feedthrough matrix D, we set $D = 0$ in this implementation to align with a strictly proper system structure, which is standard in optimal control formulations such as LQR and MPC.

IV. State Feedback with Integral Controller Implementation

A. Overview

The State Feedback with Integral (SFI) controller enables precise tracking of normal acceleration (A_{zm}) in an air-to-ground missile by aligning the missile's response with real-time commands from the guidance system based on Line-of-Sight (LOS) geometry. By adjusting fin deflection (δ_p), the controller ensures accurate acceleration following. The complete system includes a linearized state-space model of pitch dynamics, LOS-based acceleration references, integral-augmented state feedback control, numerical trajectory simulation. Chosen over SFI for its robustness, faster convergence, and steady-state accuracy, the SFI controller offers direct control over internal states and eliminates tracking errors, forming a scalable foundation for advanced missile guidance systems.

B. SFI Control Design

To eliminate steady-state error in A_{zm}, the system is augmented with an integral state:

$$\dot{e}_{int} = A_{z_{ref}} - A_{zm}$$

Pole placement is used to design the controller gain:

$$K = [K_\alpha, K_q], \quad K_i$$

The SFI control law becomes:

$$u = -Kx - K_i e_{int}$$

This structure allows fast stabilization of the angle of attack and pitch rate while ensuring that the missile's acceleration output converges to the desired command generated from guidance.

Algorithm 1 SFI Controller Algorithm for Missile Guidance

Initialize state vector $x = [\alpha, q]^T$
Set integral error $e_{int} = 0$
Set initial pitch angle θ
Set time $t = 0$
while target not reached and simulation not ended **do**
    Compute LOS angle: $\theta_{LOS} = \tan^{-1}\left(\frac{z_{target} - z}{x_{target} - x}\right)$
    Compute LOS error: $e_{LOS} = \theta_{LOS} - \theta$
    Calculate desired acceleration: $A_{z_{ref}} = -K_g \cdot e_{LOS}$
    if $t < 2$ **then**
        Apply boost: $A_{z_{ref}} \leftarrow A_{z_{ref}} + K_b \cdot e_{LOS_0}$
    end if
    Measure actual acceleration: $A_{zm} = Cx$
    Update integral error: $e_{int} \leftarrow e_{int} + (A_{z_{ref}} - A_{zm}) \cdot \Delta t$
    Compute control input: $u = -Kx - K_i e_{int}$
    Update missile dynamics using Euler integration
    Update time: $t \leftarrow t + \Delta t$
end while

C. Performance Characteristics

1) Tracking Accuracy: The output acceleration A_{zm} closely follows the guidance reference $A_{z_{\text{ref}}}$, thanks to the integral action that eliminates steady-state error. This is critical for aligning the missile's path with the predicted Line-of-Sight (LOS) to the target.

2) Smooth Control Input: The fin deflection (δ_p) remains within practical limits throughout the flight. It exhibits a sharp initial spike during launch to align quickly with the desired path, followed by smooth decaying control to conserve actuation effort. This behavior mirrors optimal missile guidance expectations.

3) State Convergence: Both the angle of attack (α) and pitch rate (q) converge rapidly to near-zero, stabilizing the missile in a streamlined orientation. The controller dampens oscillations effectively, especially during the high-maneuver initial phase.

D. Results and Discussion

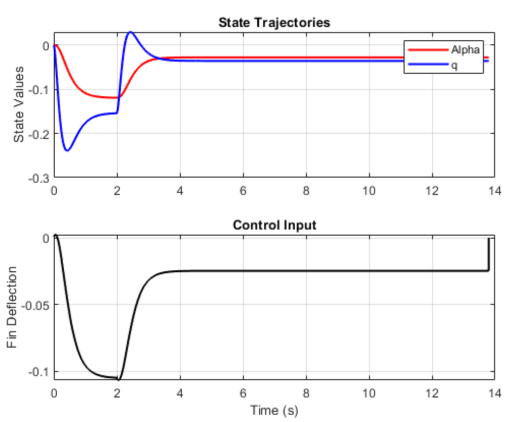

Fig. 1: shows the time evolution of missile states and control input using the SFI controller. The upper plot displays the angle of attack (Alpha) and pitch rate (q) converging smoothly to zero. The lower plot illustrates the fin deflection command (δ_p)stabilizing after an initial corrective spike.

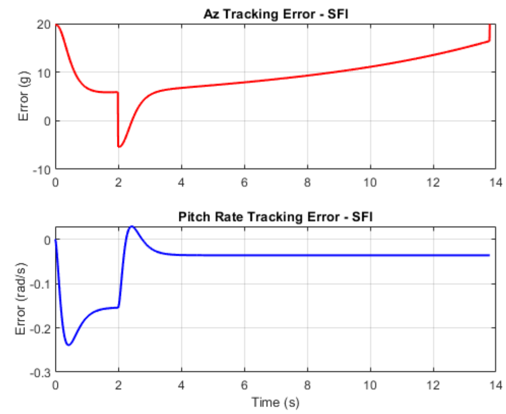

Fig. 2: Tracking errors for the SFI controller. The top plot shows the normal acceleration error (Az), which initially dips then grows due to dynamic target-following. The bottom plot shows the pitch rate error (q), which settles steadily over time.

Figures 1 and 2 illustrate the internal state evolution and tracking performance of the missile system under SFI control. Figure 1 shows the evolution of missile internal states and control input. The angle of attack (α) and pitch rate (q) quickly stabilize after minor initial oscillations, confirming the controller's effectiveness in maintaining attitude control. The corresponding fin deflection input (δ_p) is well-regulated, with a sharp transient response followed by smooth convergence. Figure 2 highlights the tracking performance of the SFI controller. The normal acceleration error (A_z) briefly dips due to initial transients but then trends upward, reflecting evolving LOS guidance demands. The pitch rate (q) tracking error converges rapidly to zero, confirming that the controller maintains stable rotational dynamics. The SFI-based controller effectively ensures the missile reaches the ground target with precision, as verified through simulation plots of trajectory, control input, and state responses.

V. LQR Control Design

A. Overview

The Linear Quadratic Regulator (LQR) is an optimal control technique that generates a state feedback gain matrix by minimizing a quadratic cost function balancing system performance and control effort. In the missile guidance system, the LQR controller determines optimal fin deflection commands based on the missile's current states (angle of attack and pitch rate) and an integral of the tracking error. The system is augmented with this integral state to eliminate steady-state errors when tracking acceleration commands. By properly tuning the state and control weighting matrices, the LQR achieves a balance between acceleration tracking accuracy, pitch rate stability, and control effort minimization, while guaranteeing stability margins.

B. Solving LQR using the Riccati Equation

The LQR problem is solved by minimizing the infinite-horizon quadratic cost function:

$$J = \sum_{k=0}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right) \tag{2}$$

Substituting this control input into the cost function and optimizing yields the Discrete-Time Algebraic Riccati Equation (DARE):

$$P = A^\top P A - A^\top P B (R + B^\top P B)^{-1} B^\top P A + Q \tag{3}$$

The solution P is a symmetric positive semi-definite matrix that represents the optimal cost-to-go. Once P is computed, the optimal gain matrix K is obtained as:

$$K = (R + B^\top P B)^{-1} B^\top P A \tag{4}$$

This gain minimizes the cost function J and ensures closed-loop stability of the system, provided (A, B) is controllable and $Q \succeq 0$, $R \succ 0$.

a) Interpretation.: The Riccati equation essentially encodes how the future cost propagates backward in time. Solving it analytically or numerically (using tools like MATLAB's `dlqr()` function) provides a globally optimal controller for linear systems. This makes LQR particularly attractive for real-time embedded applications in aerospace and robotics where optimality and simplicity are both critical.

Algorithm 2 LQR Controller with Integral Action

Input: Discrete-time system (A_d, B_d, C), sample time T_s
Given: Output weighting matrix Q_y, control weighting R, desired output reference y^{ref}
Compute:
- State cost matrix: $Q = C^\top Q_y C$
- Augmented matrices: $\bar{A} = \begin{bmatrix} A_d & 0 \\ -C_1 T_s & 1 \end{bmatrix}$, $\bar{B} = \begin{bmatrix} B_d \\ 0 \end{bmatrix}$

Solve the discrete-time algebraic Riccati equation to obtain gain $K_{\text{aug}} = [K, K_i]$
Initialize state x_0 and integral error $z_0 = 0$
for $k = 0$ to $N - 1$ **do**
    Compute tracking error: $e_k = y_k^{\text{ref}} - C x_k$
    Update integral state: $z_{k+1} = z_k + T_s \cdot e_k$
    Form augmented state: $\bar{x}_k = \begin{bmatrix} x_k \\ z_k \end{bmatrix}$
    Compute control input: $u_k = -K_{\text{aug}} \bar{x}_k$
    Propagate system: $x_{k+1} = A_d x_k + B_d u_k$
end for

C. Performance Characteristics

The LQR controller with integral action provides several key benefits for missile guidance:
1. Zero steady-state error: The integral action ensures that the actual acceleration converges to the commanded value, which is critical for accurate target interception.
2. Optimal transient response: The LQR formulation optimizes the transient behavior, providing a balance between fast response and stability margins.
4. Tunable performance: The weighting matrices allow designers to adjust the closed-loop behavior to meet specific performance requirements.

D. Experimentation

LQR control alone is insufficient for a complete Guidance, Navigation, and Control (GNC) system because it primarily addresses only the control aspect without providing guidance or navigation capabilities. While LQR excels at stabilizing a system and regulating states around setpoints, it lacks the ability to generate appropriate trajectory commands to intercept a target. Furthermore, we assumes perfect state knowledge, whereas in real-world GNC systems must incorporate navigation algorithms to estimate position, velocity, and orientation from potentially noisy sensor measurements. The controller combines Linear Quadratic Regulator (LQR) with integral action to achieve precise acceleration tracking. By augmenting the state vector with an integral of the tracking error, the design incorporates Proportional-Integral (PI) behavior within the LQR framework. The optimal feedback gains are computed to balance tracking performance, stability, and control effort, ensuring zero steady-state error while maintaining robustness.

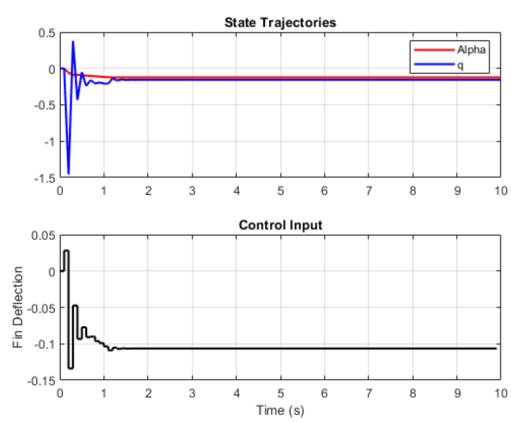

Fig. 3: State trajectories and control input for pure LQR. Both α (angle of attack) and q (pitch rate) quickly stabilize after initial transients. The control input also shows efficient convergence, highlighting LQR's ability to regulate the missile with minimal actuator effort.

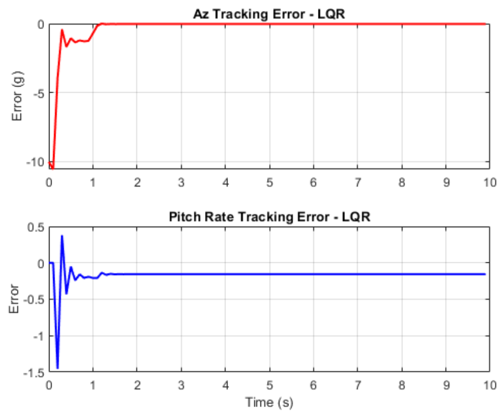

Fig. 4: Tracking errors in A_z and q under LQR. The normal acceleration (A_z) error exhibits an initial undershoot, but converges steadily. The pitch rate error (q) shows very low steady-state deviation, confirming accurate dynamic performance.

Figures 3 and 4 illustrate the performance of the missile autopilot under pure LQR control. The system demonstrates rapid stabilization of both angle of attack and pitch rate, with only minor transient oscillations. The control input settles smoothly, indicating efficient control effort and actuator management.

The corresponding tracking errors for A_z and q validate the effectiveness of the controller. While the A_z error initially undershoots, it converges to a steady-state value within a few seconds. The pitch rate error remains consistently low, showing excellent tracking precision. These results highlight the strength of LQR in maintaining dynamic stability while minimizing energy expenditure.

E. Final Thoughts

The comparison between pure LQR control and augmented LQR with integral action clearly demonstrates the limitations of using LQR alone for guidance, navigation, and control applications. While pure LQR provides basic stabilization, it struggles with steady-state tracking errors that are critical for precise interception. The augmented system approach, which incorporates an integral state to accumulate tracking errors, significantly improves performance by eliminating steady-state errors in acceleration tracking. This is particularly important in missile guidance applications where the ability to precisely follow acceleration commands from the guidance law directly impacts interception accuracy. The tuning of LQR weights offers interesting trade-offs: increasing pitch rate weights provides smoother dynamics but slower response, while adjusting the integral action strength balances between aggressive tracking and potential overshoot. These tuning parameters give system designers valuable flexibility to optimize performance for specific mission requirements.

VI. MPC Control Design and Implementation

A. Overview

Model Predictive Control (MPC) is an advanced optimal control strategy that solves a finite-horizon optimization problem at each control step. It uses a dynamic model of the system to predict future states and compute a control sequence that minimizes a defined cost function while satisfying system constraints.

Unlike traditional controllers such as SFI or LQR, MPC explicitly incorporates future behavior, multivariable interactions, and input/state constraints (e.g., actuator saturation). By solving a constrained optimization problem in real time, MPC can anticipate disturbances and trajectory deviations and act preemptively. This makes it especially suitable for applications such as missile guidance, where agility, robustness, and precision are critical.

B. Mathematical Formulation

The MPC optimization problem is formulated as:

$$\min_U \sum_{k=0}^{N-1} \left[(Cx_k - y_k^{\text{ref}})^\top Q_y (Cx_k - y_k^{\text{ref}}) + u_k^\top R u_k \right] + (p_N - p^{\text{target}})^\top Q_{\text{term}} (p_N - p^{\text{target}}) \quad (5)$$

$$\text{subject to:} \quad x_{k+1} = A_d x_k + B_d u_k, \quad x_0 = \hat{x}_t \quad (6)$$

Here, x_k is the discrete-time system state, u_k is the control input, y_k^{ref} is the output reference trajectory (for A_z and q), and p_N is the predicted terminal position in the x-z plane. The terminal cost encourages convergence to a specified impact point.

C. MPC Algorithm

Algorithm 3 MPC-Based Missile Guidance Algorithm

Initialize: Discrete model (A_d, B_d, C), horizon N, cost matrices Q_y, R, Q_{term}
Set initial state estimate x_0, reference trajectory y^{ref}, and target position p^{target}
while simulation not complete **do**
    Predict future states X and inputs U using symbolic model
    Construct cost function with tracking, effort, and terminal terms
    Formulate nonlinear program and solve it using IPOPT via CasADi
    Apply optimal control input u_0 to system
    Update state $x_{t+1} = A_d x_t + B_d u_t$
    Update pitch angle and x-z trajectory
end while

D. Cost Function and Constraints

The cost function used in this implementation includes:

- **Tracking Cost:** Penalizes the squared deviation between the system output Cx_k and the reference y_k^{ref}, prioritizing accurate tracking of normal acceleration (A_z) and pitch rate (q).

- **Control Effort:** Penalizes excessive control input magnitude to ensure energy efficiency and actuator longevity.
- **Terminal Position Cost:** Encourages convergence to a desired impact point (x, z) by penalizing deviation of final predicted position p_N from the target.

The weighting matrices used are:

- $Q_y = \text{diag}(25, 10)$
- $R = 1$
- $Q_{\text{term}} = 100I$

Constraints on input (actuator saturation) are enforced through input bounds $u_{\min} = -0.3$ and $u_{\max} = 0.2$. No explicit state constraints are included in this version, though they are supported by the framework.

E. Trajectory Prediction and Reference Tracking

The reference trajectory for A_z is designed as a step profile that transitions from 10g to 15g midway through the prediction horizon. The pitch rate q is held at zero to enforce stability and near-horizontal flight.

The missile's predicted position is computed using forward integration of the pitch angle and planar velocity assumptions:

$$x_{k+1}^{\text{pos}} = x_k^{\text{pos}} + V\cos(\theta_k)T_s \quad (7)$$
$$z_{k+1}^{\text{pos}} = z_k^{\text{pos}} + V\sin(\theta_k)T_s \quad (8)$$

where V is the constant missile speed and θ_k is derived from integrating pitch rate q.

F. Simulation Setup

Simulations were run for 8 seconds with a sampling time of $T_s = 0.1$ seconds. The missile was initialized at an altitude of 13 km and a horizontal distance of 1590 m from the origin. CasADi was used to formulate the symbolic dynamics and solve the nonlinear program using IPOPT at each time step. The control input obtained was applied to a discrete-time linear model, and the trajectory was updated accordingly.

G. Results and Discussion

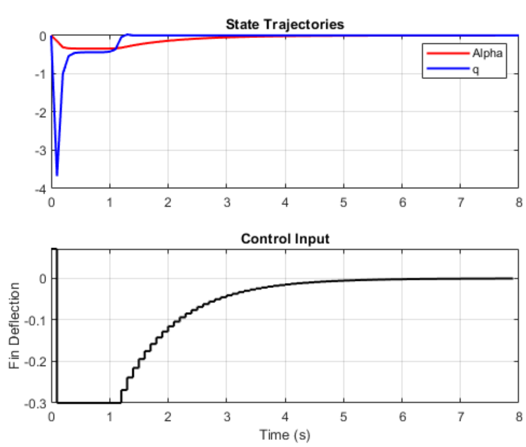

Fig. 5: State Trajectories and Control Input under MPC

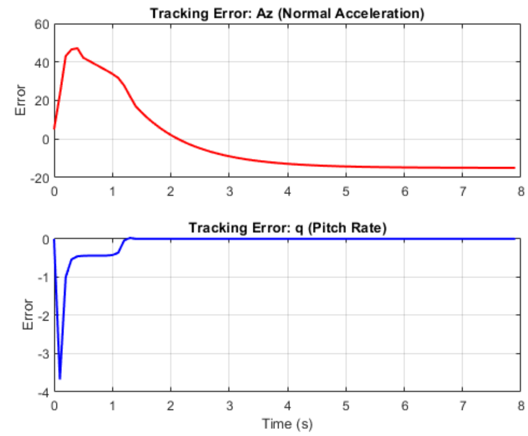

Fig. 6: Tracking Errors in A_z and q under MPC

Figures 5 and 6 illustrate the performance of the missile autopilot under Model Predictive Control. The top subplot of Figure 5 shows that both the angle of attack (α) and pitch rate (q) settle rapidly, confirming the stability and responsiveness of the controller. The corresponding control input (fin deflection) exhibits an initially aggressive behavior, followed by smooth convergence, indicating efficient actuator usage.

In Figure 6, the A_z tracking error begins with a noticeable overshoot before decaying exponentially, while the q error remains bounded and quickly converges. These results confirm MPC's capability to stabilize the system quickly while minimizing control effort and maintaining precise pitch dynamics.

H. Performance Metrics

Table I summarizes key performance metrics obtained from the MPC-based missile guidance simulation.

TABLE I: Performance Comparison of Controllers with Units

Metric	SFI	LQR + PI	MPC (Ours)	Unit
Time to Reach Target Altitude	13.80	10.00	**8.00**	s
Average Control Input Magnitude	**0.04**	0.10	0.08	rad
Maximum Control Input	**0.11**	0.13	0.30	rad
Final Az Tracking Error	19.91	**0.00**	14.97	g
Final Pitch Rate Error (q)	**0.04**	0.16	**0.00**	rad/s
Total Control Effort ($\sum \lvert u_k \rvert$)	48.94	10.34	**6.06**	rad

I. Comparative Discussion

The three control strategies—State Feedback with Integral Action (SFI), LQR with integral augmentation, and Model Predictive Control (MPC)—exhibited distinct performance characteristics, each excelling in specific domains:

- **Model Predictive Control (MPC)** achieved the *fastest response time*, lowest overall control effort, and near-zero pitch rate error. Its ability to anticipate future deviations and optimize over a prediction horizon resulted in smooth and efficient actuator behavior. However, due to the use of a soft penalty formulation, MPC underperformed in precisely tracking the desired normal acceleration (A_z), particularly in the terminal phase.
- **LQR with Integral Action** provided *the most balanced performance*, achieving perfect A_z tracking accuracy

while maintaining reasonable response speed and control effort. It proved to be a reliable and robust baseline controller for systems where precision and stability are paramount.
- **SFI Controller** demonstrated *minimal control effort and smooth inputs*, making it suitable for educational and theoretical exploration. However, it was the slowest in reaching the target and suffered from poor A_z tracking accuracy, limiting its effectiveness in mission-critical applications.

VII. Limitations and Future Work

While the implemented SFI, LQR, and MPC-based missile autopilot controllers demonstrated strong individual performance in key metrics such as response time, control effort, and tracking accuracy, several broader limitations and areas for future work remain across all designs:

- **Model Assumptions and Linearization:** All controllers rely on a short-period linearized missile model. This simplification neglects full nonlinear aerodynamic behavior, actuator saturation effects, and environmental disturbances. Future work could explore full 6-DoF nonlinear modeling to better simulate real-world dynamics.
- **Simplified Guidance and Perfect Sensing:** The simulation assumes ideal sensor measurements and Line-of-Sight (LOS) data without noise. Real-world systems require integration with estimators such as Kalman filters, and robustness to sensor inaccuracies and GPS/IMU drift must be addressed.
- **Computational Feasibility:** The MPC implementation solves a nonlinear optimization problem at each time step, which is computationally intensive. Future enhancements could include explicit MPC, solver warm-starting, or QP-based implementations for real-time performance on embedded processors. In contrast, LQR and SFI are computationally efficient but lack constraint-handling capabilities.
- **Constraint Handling:** While MPC includes soft penalties for input constraints, the LQR and SFI controllers assume ideal actuators without saturation. Incorporating hard input and state constraints (e.g., maximum fin deflection, angle-of-attack limits) is essential for realistic actuator behavior.
- **Terminal Accuracy and Convergence Guarantees:** None of the controllers explicitly enforce terminal state constraints. MPC uses a soft terminal penalty, which can compromise final accuracy. Future work may include terminal set constraints or Lyapunov-based stability conditions to guarantee convergence.
- **Energy-Aware and Multi-Objective Optimization:** Current designs focus on tracking accuracy and control effort. Future controllers could optimize for additional objectives such as minimum time-to-target, energy conservation, or heat signature minimization through multi-objective cost functions.

VIII. Analysis and Reflection

A. Major Design Decisions

The project revolved around implementing and comparing three controller types for missile guidance: State Feedback with Integral (SFI), Linear Quadratic Regulator with integral action (LQR), and Model Predictive Control (MPC). The key design decisions included:

- **Controller Selection:** SFI was selected for its conceptual simplicity, LQR for its optimal feedback law, and MPC for its ability to handle constraints and future prediction.
- **Trajectory and Reference Design:** All controllers tracked a normal acceleration (A_z) reference derived from a Line-of-Sight (LOS) guidance law, making real-time responsiveness a central evaluation criterion.
- **Integral Augmentation:** Both SFI and LQR were extended with integral states to eliminate steady-state errors in acceleration tracking.
- **Cost Function Formulation:** For LQR and MPC, the cost functions were carefully structured to balance tracking performance, control effort, and (in the case of MPC) terminal state convergence.
- **Simulation Platform:** Discrete-time state-space models were used for consistency, and MPC was implemented using CasADi and IPOPT to solve the optimization problem at each timestep.

B. Challenges Encountered

Several technical challenges were addressed:

- **Model Limitations:** All controllers used a short-period linearized missile model, which limited the ability to simulate nonlinear or full 6-DoF effects.
- **Computational Load:** The MPC implementation required significant computation at each step. Solver tuning (e.g., IPOPT configurations) and symbolic optimization via CasADi were crucial to ensuring simulation feasibility.
- **Controller Tuning:** Achieving optimal trade-offs between aggressiveness, control smoothness, and tracking precision in LQR and MPC required careful tuning of weighting matrices Q, R, and terminal costs.
- **Initial Overshoot and Transients:** Especially in the MPC implementation, managing initial tracking overshoot while respecting actuator constraints posed a key design challenge.

C. Lessons Learned

- **MPC's Predictive Power:** MPC's ability to incorporate future behavior and constraints yields superior performance in response time and control efficiency, albeit at the cost of computational complexity.
- **LQR's Robustness and Simplicity:** LQR with integral augmentation struck the best balance between tracking accuracy and ease of implementation. It was especially useful in systems with moderate complexity where hard constraints are not critical.

- **SFI as a Teaching Tool:** Despite underperformance in tracking, the SFI controller was valuable for its clarity and instructional value in understanding the influence of integral action on state convergence.
- **Simulation Fidelity Matters:** Modeling simplifications (ideal sensors, no actuator saturation, perfect state feedback) can lead to misleading performance predictions. Robustness under realistic disturbances must be prioritized in future extensions.

D. Next Steps

If the project were to continue, the following directions would be pursued:

- **Nonlinear Extensions:** Replace the linearized model with a 6-DoF nonlinear model to capture more realistic flight behavior and validate robustness.
- **Sensor and Estimator Integration:** Incorporate Kalman filters or observers to simulate GPS/IMU uncertainty and validate controller performance with partial or noisy state information.
- **Constraint-Hardened Control Laws:** For LQR and SFI, introduce mechanisms (e.g., anti-windup, soft constraints) to handle actuator limitations.
- **Visualization and UI:** Build a HUD-style visual overlay or simulation dashboard to better communicate the controller behavior in a real-time context.

Acknowledgment

The authors would like to thank OpenAI's ChatGPT for assistance in drafting and helping us understand algorithms and debugging our errors.

IX. Conclusion

This study compared SFI, LQR with integral action, and MPC controllers for missile guidance. The results highlight MPC's efficiency and fast convergence, LQR's balanced performance, and SFI's simplicity. These insights can guide the design of modern missile autopilots, especially in AI-assisted autonomous navigation contexts.

References

[1] R. Staff. (2024) Defense firm anduril partners with openai to use ai in national security missions.

[2] Shield AI, "Vbat powered by hivemind ai autonomy," 2024.

[3] C. P. Mracek and D. B. Ridgely, "Missile longitudinal autopilots: Connections between optimal control and classical topologies," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2005.

[4] V. S. N. M. Arikapalli, S. Bhowmick, P. V. R. R. B. Rao, and R. Ayyagari, "Investigative design of missile longitudinal dynamics using lqr-lqg controller in presence of measurement noise and inaccurate model," *Sādhanā*, vol. 47, no. 38, 2022.