Control Allocation in Single-Chamber Solid-Propellant DACS via Differentiable Programming-Based Constrained LM Optimization

Jaehyeok Cha[1†], Gyubin Park[1†], and Jong-Han Kim[1*]

***Abstract*— We propose a differentiable programming-based control allocation framework for a single-chamber solid-propellant Divert and Attitude Control System. The method explicitly captures the nonlinear coupling between chamber pressure, nozzle throat areas, and thrust using physics-based differentiable models. Desired force and torque commands are mapped to actuator inputs via a constrained Levenberg–Marquardt optimization, where physical bounds are enforced through proximal projection, and gradients are computed using automatic differentiation. Closed-loop six-degree-of-freedom simulations indicate that the proposed framework is computationally efficient and tractable for onboard implementation and achieves precise attitude stabilization and minimal miss distance even under strong nonlinear coupling.**

I. Introduction

The rapid advancement of long-range ballistic missiles has heightened the strategic importance of missile defense systems [1]. Missile defense engagements are typically categorized into boost, midcourse, and terminal phases, with exo-atmospheric midcourse interception serving as a cornerstone of multi-layered defense architectures. In this near-space environment, aerodynamic control is negligible, necessitating an autonomous kill vehicle (KV) that operates independently after separation from a booster. The KV utilizes a hit-to-kill concept, relying on direct kinetic impact to ensure target neutralization. This approach has become the industry standard due to its superior lethality against submunitions and higher mass efficiency compared to fragmentation warheads, which require complex fuzing mechanisms and impose significant weight penalties [2].

To achieve high-precision interception without aerodynamic control authority, the KV relies on a Divert and Attitude Control System (DACS) during the terminal homing phase [3]. The DACS executes two primary functions: (i) divert control to track lateral acceleration commands derived from a guidance law (*e.g.*, proportional navigation), steering the KV toward the target [4]; and (ii) attitude control to generate angular acceleration for maintaining seeker line-of-sight (LOS) alignment and tracking accuracy [5].

This work was supported in part by the Theater Defense Research Center funded by Defense Acquisition Program Administration (DAPA) under Grant UD240002SD, and in part by the Korea Research Institute for Defense Technology Planning and Advancement (KRIT) under Grant KRIT-CT-22-030 through the Reusable Unmanned Space Vehicle Research Center, funded by the Korea government (DAPA). Equal support was received from both funding sources. *(Corresponding author: J.-H. Kim)*

[1]The authors are with the Department of Aerospace Engineering and the Program in Aerospace Systems Convergence at Inha University, Incheon, Republic of Korea. J.-H. Kim is also with the Aerospace Systems Research Institute at Inha University. {jaehyeokc,gyubin}@inha.edu, jonghank@inha.ac.kr

[†]These authors contributed equally to this work.

Conventional DACS implementations have predominantly employed liquid-propellant thrusters due to their fine thrust modulation capability. However, concerns regarding structural complexity, limited storability, and safety have driven the adoption of solid-propellant gas generators, offering inherent simplicity, reliability, and rapid response. A primary limitation of solid propulsion is the difficulty of thrust regulation, which has been addressed through pintle-actuated systems that modulate the nozzle throat area to control chamber pressure and thrust [6], [7], [8]. In multi-nozzle configurations sharing a single chamber, however, pintle actuation introduces strong cross-coupling, as the movement of one pintle alters the global chamber pressure and consequently affects the thrust of all other nozzles. Coupled with nonlinear thermofluid dynamics, this interaction yields a complex nonlinear control problem [9], motivating the development of an integrated, differentiable optimization-based framework.

Several studies have addressed the complexity of continuous-thrust pintle systems, primarily through decoupled or hierarchical control architectures. Early efforts pioneered adaptive PI-based pressure regulation integrated with sequential torque allocation logic [10], or bifurcated the control task into inner-loop pressure tracking and outer-loop thrust distribution [11]. To enhance robustness, more integrated schemes have emerged, such as adaptive sliding-mode control paired with a nonlinear disturbance observer (NDO) to concurrently regulate pressure and thrust despite actuator nonlinearities [12]. Complementing these real-time strategies, other research has pursued offline optimization to generate pressure command profiles that minimize propellant consumption by accounting for time-varying vehicle mass [13].

A fundamental limitation of existing real-time strategies [10], [11], [12] is their reliance on problem decomposition at the expense of holistic optimization. These methods typically employ sequential schemes—such as pressure regulation followed by torque allocation [10]—or relegate strong nonlinear cross-channel interactions to external disturbances to be rejected via observers [12]. Because these approaches do not explicitly solve the fully coupled allocation problem, their achievable performance depends on the quality of the decomposition and associated controller tuning. This motivates the development of a unified framework capable of navigating the full nonlinear landscape of the coupled DACS dynamics.

This study develops a unified differentiable modeling and optimization framework for a single-chamber solid-propellant DACS. The nonlinear combustion-chamber dynamics induced by variable nozzle throat areas are formulated via differen-

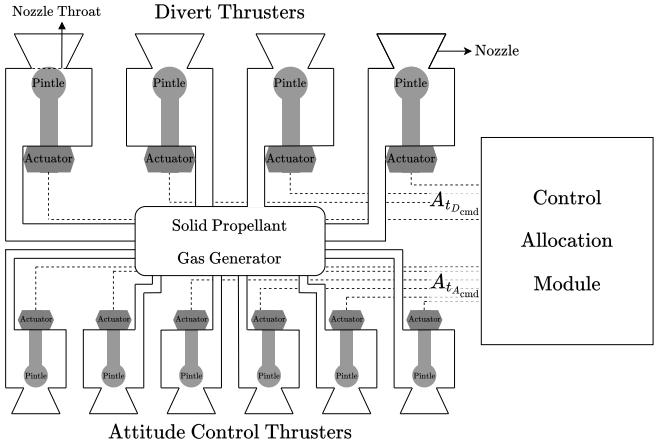

Fig. 1. Single-chamber solid-propellant DACS architecture considered in this study. All divert and attitude control thrusters share a common combustion chamber.

tiable programming [14], [15], enabling high-fidelity gradient computation of pressure and thrust with respect to actuator states. Leveraging this differentiable model, the control allocation (CA) task is posed as a nonlinear constrained optimization problem to determine the optimal throat areas that satisfy lateral force and attitude torque commands. This problem is solved using a constrained Levenberg–Marquardt (LM) algorithm [16], [17], which integrates the coupled system physics directly into the solver rather than treating them as external disturbances. The effectiveness of the proposed framework is demonstrated through six-degree-of-freedom (6-DoF) closed-loop simulations, with Monte-Carlo analyses confirming its robustness against initial condition variations and thrust uncertainties.

II. System Modeling

The DACS mounted on the KV comprises a solid-propellant gas generator, divert and attitude control thrusters, and a CA module (see Fig. 1). This study considers a single-chamber configuration in which all thrusters share a common combustion chamber. The gas generator delivers high-pressure gas to the thrusters, while the CA module regulates individual nozzle throat areas to generate the lateral forces and attitude torques required for guidance and control.

The mass conservation within the combustion chamber is expressed as:

$$\dot{m}_p = \frac{d(\rho_c V_c)}{dt} + \dot{m}_e, \tag{1}$$

where $\dot{m}_p$ denotes the mass generation rate from propellant combustion. The variables ρ_c and V_c represent the chamber gas density and free volume, respectively, such that the term $d(\rho_c V_c)/dt$ accounts for the rate of gas accumulation within the chamber. Finally, $\dot{m}_e$ is the total exhaust mass flow rate.

When the flow is choked at the nozzle throat, the mass flow rate through the nozzle is given by:

$$\dot{m}_e = A_t P_c \sqrt{\frac{\gamma}{RT_c}\Big(\frac{2}{\gamma+1}\Big)^{\frac{\gamma+1}{\gamma-1}}}, \tag{2}$$

where A_t is the nozzle throat area, P_c the chamber pressure, and R, γ, and T_c are the gas constant, specific heat ratio, and chamber temperature, respectively.

For a solid-propellant gas generator, the mass generation rate can be written as:

$$\dot{m}_p = A_b \dot{r}_b \rho_p, \tag{3}$$

where A_b, $\dot{r}_b$ and ρ_p are burning surface area, burning rate, and the propellant density, respectively.

The burning rate is modeled using Saint Robert's law [18], which relates the burning rate to the chamber pressure:

$$\dot{r}_b = a_b P_c^n, \tag{4}$$

where a_b is an empirical coefficient depending on the ambient grain temperature, and n is the pressure exponent [19]. Combining (1)-(4) with the ideal gas law yields the following chamber pressure dynamics:

$$\frac{dP_c}{dt} = \frac{RT_c A_b a_b \rho_p P_c^n}{V_c} - \frac{P_c}{V_c}\frac{dV_c}{dt} - \frac{RT_c P_c A_{t_{\text{total}}}}{V_c}\sqrt{\frac{\gamma}{RT_c}\Big(\frac{2}{\gamma+1}\Big)^{\frac{\gamma+1}{\gamma-1}}}, \tag{5}$$

where $A_{t_{\text{total}}}$ is the sum of all nozzle throat areas, and the dynamics of the chamber free volume can be expressed as:

$$V_c = V_{c_0} + A_b \int \dot{r}_b dt, \tag{6}$$

which follows from the geometric expansion of the burning grain via $dV_c/dt = A_b \dot{r}_b$. Equation (5) indicates that the chamber pressure dynamics depend strongly on the total nozzle throat area, thus coupling the dynamics of all thrusters that share the common chamber.

Each thruster $i \in \{D_1, \ldots, D_4, A_1, \ldots, A_6\}$ produces thrust F_i proportional to the chamber pressure and its respective throat area A_{t_i} [11]:

$$F_i = C_{F_i} A_{t_i} P_c, \tag{7}$$

where C_{F_i} is the thrust coefficient defined as:

$$C_{F_i} = \sqrt{\frac{2\gamma^2}{\gamma-1}\Big(\frac{2}{\gamma+1}\Big)^{\frac{\gamma+1}{\gamma-1}}\Big\{1-\Big(\frac{P_{e_i}}{P_c}\Big)^{\frac{\gamma-1}{\gamma}}\Big\}} + \Big(\frac{P_{e_i}-P_a}{P_c}\Big)\frac{A_{e_i}}{A_{t_i}}, \tag{8}$$

where P_{e_i} and A_{e_i} denote the exit pressure and exit area of nozzle i, and P_a is the ambient pressure. The first term represents the momentum-thrust contribution due to isentropic expansion, whereas the second term represents the pressure-thrust correction caused by the mismatch between the exit and ambient pressures.

The configuration of divert and attitude control thrusters on the KV is illustrated in Fig. 2, and the resulting lateral forces and torques in the body frame are expressed as [5]:

$$\begin{aligned} F_y &= -F_{D_2} + F_{D_4} - F_{A_2} - F_{A_3} + F_{A_5} + F_{A_6}, \\ F_z &= F_{D_1} - F_{D_3} + F_{A_1} - F_{A_4}, \\ \tau_r &= -bF_{A_2} + bF_{A_3} - bF_{A_5} + bF_{A_6}, \\ \tau_p &= aF_{A_1} - aF_{A_4}, \\ \tau_y &= aF_{A_2} + aF_{A_3} - aF_{A_5} - aF_{A_6}, \end{aligned} \tag{9}$$

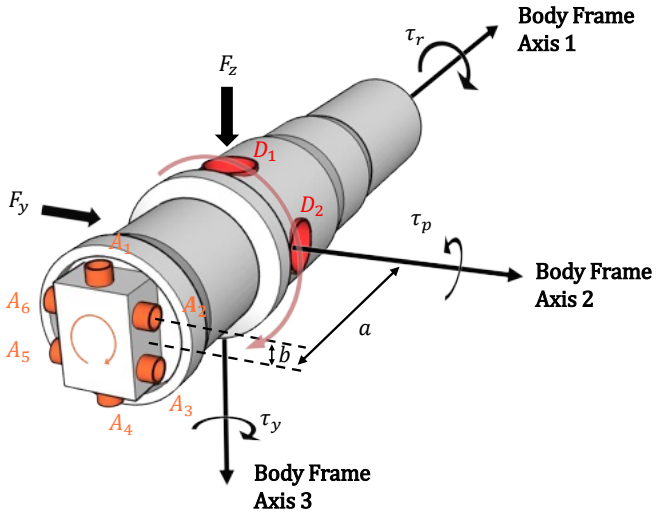

Fig. 2. KV thruster layout and body-axis definitions.

where (F_y, F_z) represent the lateral forces in the body coordinate frame and (τ_r, τ_p, τ_y) denote the corresponding roll, pitch, and yaw torques. The parameters a, b are the lever arms of the attitude control thrusters along the body axes.

III. Problem Formulation

A. Force and Torque Commands

During the terminal homing phase, the KV's guidance and control loops generate lateral acceleration and angular acceleration commands via proportional navigation (PN) and a proportional-derivative (PD) attitude controller, respectively. The attitude controller computes angular acceleration commands, maintaining the strapdown seeker's LOS alignment with the target. As the controller aligns the vehicle's body x-axis with the LOS vector, a lateral control acceleration perpendicular to the LOS is produced, implementing a True Proportional Navigation (TPN) law.

Assuming a North-East-Down (NED) inertial frame, the three-dimensional TPN guidance is expressed as [20], [21]:

$$u_{\mathrm{NED}}^{\mathrm{TPN}} = N V_{TM} \times \Omega_{TM}, \tag{10}$$

where N is the navigation constant and Ω_{TM} is the rotation vector of the LOS, defined as

$$\Omega_{TM} = \frac{R_{TM} \times V_{TM}}{\|R_{TM}\|^2}. \tag{11}$$

Here, $R_{TM} = R_T - R_M$ and $V_{TM} = V_T - V_M$ denote the relative position and velocity vectors between the target and the interceptor, respectively.

The body-frame acceleration command, u_{cmd}, which incorporates both the TPN guidance law and gravity compensation, is computed by first calculating the total required acceleration in the NED frame. This is achieved by subtracting the gravitational acceleration vector g from the TPN command $u_{\mathrm{NED}}^{\mathrm{TPN}}$. This resultant vector is then transformed into the body frame using the direction cosine matrix $C_{\mathrm{N}}^{\mathrm{B}}$:

$$u_{\mathrm{cmd}} = C_{\mathrm{N}}^{\mathrm{B}}(u_{\mathrm{NED}}^{\mathrm{TPN}} - g), \tag{12}$$

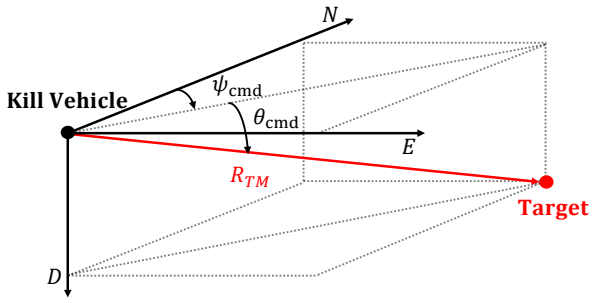

Fig. 3. Reference Euler-angle commands for seeker line-of-sight alignment.

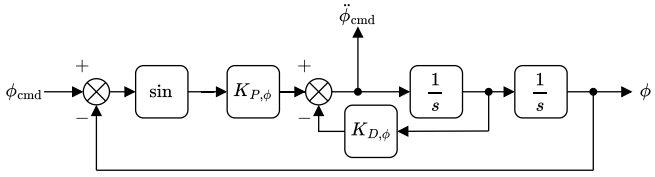

Fig. 4. Second-order PD attitude-control structure used to generate angular-acceleration commands.

where $C_{\mathrm{N}}^{\mathrm{B}}$ is the direction cosine matrix from NED to body frame, and g is the gravitational acceleration vector in NED coordinates.

Because the KV cannot generate thrust along its longitudinal (body-x) axis, only the lateral components of this command vector are utilized. Therefore, the y and z elements of u_{cmd} (denoted $u_{\mathrm{cmd},y}$ and $u_{\mathrm{cmd},z}$) correspond to the final commanded lateral accelerations that serve as inputs to the CA module.

The Euler angle commands that keep the seeker aligned with the target are computed by decomposing the LOS vector, as shown in Fig. 3. The Euler angle commands are defined as [22]:

$$\begin{aligned} \phi_{\mathrm{cmd}} &= 0, \\ \theta_{\mathrm{cmd}} &= \tan^{-1}\left(-R_{TM,D}/\sqrt{R_{TM,N}^2 + R_{TM,E}^2}\right), \\ \psi_{\mathrm{cmd}} &= \tan^{-1}\left(R_{TM,E}/R_{TM,N}\right), \end{aligned} \tag{13}$$

where the roll command angle ϕ_{cmd} is fixed at zero for seeker stabilization.

The PD attitude controller based on a second-order reference system (illustrated in Fig. 4) tracks the reference attitude and computes the commanded angular accelerations as follows:

$$\begin{aligned} \ddot{\phi}_{\mathrm{cmd}} &= K_P^{\phi} \sin(\phi_{\mathrm{cmd}} - \phi) - K_D^{\phi}\dot{\phi}, \\ \ddot{\theta}_{\mathrm{cmd}} &= K_P^{\theta} \sin(\theta_{\mathrm{cmd}} - \theta) - K_D^{\theta}\dot{\theta}, \\ \ddot{\psi}_{\mathrm{cmd}} &= K_P^{\psi} \sin(\psi_{\mathrm{cmd}} - \psi) - K_D^{\psi}\dot{\psi}. \end{aligned} \tag{14}$$

The final commanded lateral forces and torques provided to the CA module are as follows:

$$y_{\mathrm{cmd}} \triangleq \begin{bmatrix} m u_{y,\mathrm{cmd}} \\ m u_{z,\mathrm{cmd}} \\ I_{xx}\ddot{\phi}_{\mathrm{cmd}} \\ I_{yy}\ddot{\theta}_{\mathrm{cmd}} \\ I_{zz}\ddot{\psi}_{\mathrm{cmd}} \end{bmatrix} = \begin{bmatrix} F_{y,\mathrm{cmd}} \\ F_{z,\mathrm{cmd}} \\ \tau_{r,\mathrm{cmd}} \\ \tau_{p,\mathrm{cmd}} \\ \tau_{y,\mathrm{cmd}} \end{bmatrix}, \tag{15}$$

where m is the vehicle mass, and I_{xx}, I_{yy}, I_{zz} are the principal moments of inertia.

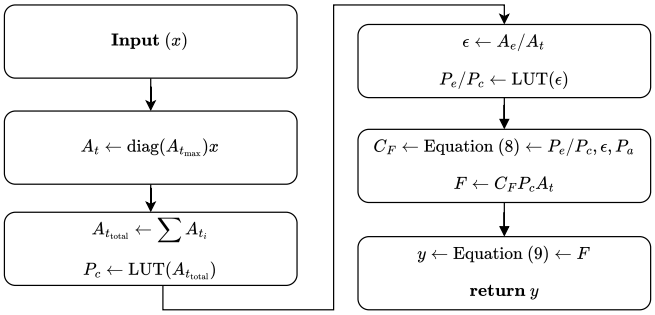

Fig. 5. Computation flow of quasi-steady-state dynamics used for total force and torque vector computation.

B. Quasi-Steady-State Combustion Chamber Model

To ensure real-time feasibility, a quasi-steady-state model is derived by assuming $\dot{P}_c \approx 0$ in (5). Under the assumptions of choked flow and isentropic expansion, the nozzle expansion ratio $\epsilon_i = A_{e_i}/A_{t_i}$ is related to the pressure ratio P_{e_i}/P_c through the following transcendental equation:

$$\frac{1}{\epsilon_i} = \left(\frac{\gamma+1}{2}\right)^{\frac{1}{\gamma-1}} \left(\frac{P_{e_i}}{P_c}\right)^{\frac{1}{\gamma}} \sqrt{\frac{\gamma+1}{\gamma-1}\left\{1-\left(\frac{P_{e_i}}{P_c}\right)^{\frac{\gamma-1}{\gamma}}\right\}}. \tag{16}$$

Notably, both the expansion ratio relation in (16) and the resulting quasi-steady-state pressure equation are transcendental, lacking closed-form analytical solutions for P_{e_i}/P_c and P_c. Therefore, we numerically invert these relations to construct pre-computed lookup tables (LUTs). Leveraging these LUTs, the thrust F_i for a given normalized throat area vector x can be efficiently evaluated via (7) and (8). This process defines a nonlinear mapping $y = f(x)$ that relates actuator inputs to the total force and torque vector, as illustrated in Fig. 5. This mapping serves as the fundamental model for the optimization-based allocation framework presented in the next section.

IV. Control Allocation via Differentiable Constrained LM Method

To solve the CA problem, this study employs a constrained LM algorithm, which regularizes the update step within a trust region and is well matched to the residual least-squares structure of the allocation problem. Relative to a generic SQP scheme, the projected LM update provides a simpler per-iteration procedure under box constraints, which is attractive for repeated online solution [16]. The optimization problem is formulated as:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|W^{1/2}\hat{g}(x;x^{(k)})\|^2 + \lambda^{(k)}\|x - x^{(k)}\|^2 \\ \text{subject to} \quad & 0 \le x \le 1, \end{aligned} \tag{17}$$

where $\hat{g}(x;x^{(k)})$ is the local linearization of the normalized control-error $g(x) \triangleq W_g(f(x) - y_{\mathrm{cmd}})$. Here, the weighting matrix W_g performs row scaling to compensate for the large difference in orders of magnitude between force and torque commands, while x is adjusted to minimize the tracking error. Meanwhile, $\lambda^{(k)}$ regulates the update step size to ensure the validity of the linear approximation.

In (17), the first term minimizes the weighted control-error to ensure that the generated forces and torques precisely track y_{cmd} across all control channels. To satisfy the physical limits of the throat areas, the box constraints $0 \le x \le 1$ are enforced via proximal projection, which clips the solution to the feasible region at each iteration.

Since the mapping $f(x)$ involves complex nonlinear coupling and lacks a closed-form expression, the control-error $g(x)$ behaves as a nonlinear implicit function, making manual derivation of its Jacobian $Dg(x)$ impractical. To address this, we adopt a differentiable programming approach leveraging the automatic differentiation capabilities of JAX [15]. Unlike numerical finite-difference methods, JAX enables autodiff-consistent Jacobian evaluation of the interpolated quasi-steady model, improving numerical consistency and supporting efficient online computation.

The overall optimization procedure is summarized in Algorithm 1. At each discrete time step, the constrained LM algorithm iteratively solves the subproblem to a maximum of K iterations, with early termination if both optimality (ϵ_{opt}) and step-size (ϵ_{step}) convergence criteria are met. To accelerate convergence, a warm-start strategy is employed where the initial guess $(x_t^{(0)}, \lambda_t^{(0)})$ is set to the previous step's optimal solution (x_{t-1}, λ_{t-1}). This leverages the temporal continuity of the flight trajectory and pressure dynamics, and empirically leads to minimal iteration counts in the tested closed-loop simulations. The stationarity of the constrained problem is verified by the projected gradient g_{proj}, defined as [23]:

$$g_{\mathrm{proj},i} = \begin{cases} 0 & \text{if } x_i = 0 \text{ and } (Dg^T g)_i > 0 \\ 0 & \text{if } x_i = 1 \text{ and } (Dg^T g)_i < 0 \\ (Dg^T g)_i & \text{otherwise.} \end{cases} \tag{18}$$

For smooth residual mappings with convex box constraints, projected LM methods are known to possess strong local convergence properties under standard regularity assumptions [17]; in the present interpolated and nonconvex setting, this serves as a local justification rather than a global convergence guarantee.

V. Numerical Experiments

A. Monte-Carlo Simulation

To evaluate the robustness of the proposed CA algorithm, a 500-run Monte-Carlo simulation was conducted within a closed-loop engagement scenario. The KV and target are modeled as a 6-DoF rigid body and a 3-DoF point mass, respectively. In each trial, the nozzle throat areas x obtained from Algorithm 1 are applied as actuator commands to (7)-(9) to produce the realized control forces. The simulation parameters are summarized in Table I, Table II, and Table III. Initial attitude errors, body rate biases, and geometric misalignments follow a uniform distribution, while the thrust uncertainty ΔF_i follows a Gaussian distribution. The guidance law utilizes a navigation constant typical for hypersonic interceptions. For real-time applicability, the maximum number of LM iterations, K, is strictly enforced to guarantee a deterministic execution time.

Algorithm 1: Constrained Levenberg-Marquardt Method for Control Allocation.

Input:

y_{cmd} // Force, torque commands
x_{t-1} // Previous optimal solution
λ_{t-1} // Previous damping parameter
K // Maximum iteration number
$\epsilon_{\text{opt}}, \epsilon_{\text{step}}$ // Convergence tolerance

Output:

x_t // Updated optimal solution
λ_t // Updated damping parameter

Initialization:

$\lambda^{(k)} \leftarrow \lambda_{t-1}, \quad x^{(k)} \leftarrow x_{t-1}, \quad g_k \leftarrow g(x^{(k)})$

for $k \leftarrow 0$ **to** $K-1$ **do**

// Optimality evaluation:
$g_{\text{opt}} \leftarrow Dg_k^T g_k$

// Column Scaling:
$W \leftarrow \left[\text{diag}(Dg_k^T Dg_k)\right]^{-1}$

// LM update step:
$\Delta x^{(k)} \leftarrow (Dg_k W)^T (Dg_k W Dg_k^T + \lambda^{(k)} I)^{-1} g_k$

// Projection:
$x_{\text{clipped}} \leftarrow \text{clip}(x^{(k)} - \Delta x^{(k)}, 0, 1)$
$\Delta x_{\text{proj}}^{(k)} \leftarrow x^{(k)} - x_{\text{clipped}}$

// Convergence check:
$g_{\text{proj}} \leftarrow \text{ProjectedGradient}(x^{(k)}, g_{\text{opt}})$
if $\|g_{\text{proj}}\|_\infty < \epsilon_{\text{opt}}$ **and** $\|\Delta x_{\text{proj}}^{(k)}\| < \epsilon_{\text{step}}$ **then**
break

$g_{\text{new}} \leftarrow g(x_{\text{clipped}})$

// Update solution and damping:
if $\|g_{\text{new}}\|^2 < \|g_k\|^2$ **then**
$x^{(k+1)} \leftarrow x_{\text{clipped}}$
$g_{k+1} \leftarrow g_{\text{new}}$
$\lambda^{(k+1)} \leftarrow 0.8\lambda^{(k)}$
else
$\lambda^{(k+1)} \leftarrow 2.0\lambda^{(k)}$

return $x_t \leftarrow x^{(k)}, \quad \lambda_t \leftarrow \lambda^{(k)}$

The proposed framework demonstrates high-bandwidth potential, with execution times summarized in Table IV. These results validate the computational tractability of the approach for real-time applications. Although Python's inherent overhead caused occasional peak latencies, the overall statistical profile—including the TPN-PD loop—confirms the feasibility of a fully embedded implementation.

The solver's robustness is evidenced by the iteration history in Fig. 6. Leveraging temporal continuity, the warm-start strategy maintains a minimal iteration count (1 or 2) after the initial transient. Conversely, cold-start scenarios exhibit higher latency as the solver exhausts its iteration quota without reaching convergence, highlighting the critical role of warm-start initialization. The proposed algorithm stabilizes attitude and boresight errors within the seeker's field-of-view (FOV) in less than 0.5 s (Figs. 7 and 8), ensuring a continuous target lock. Notably, the miss distance—analytically determined via the closest point of approach to capture the geometric minimum—is maintained below 0.013 cm (Fig. 9). These results demonstrate that the proposed CA achieves the sub-millimeter precision required for reliable hit-to-kill kinetic impacts under stochastic uncertainties.

TABLE I
INITIAL CONDITIONS, UNCERTAINTY RANGES, AND SOLVER SETTINGS

Parameter	Value	Unit	Description
$\|R_{TM}\|$	40	km	Relative distance
$\|V_{TM}\|$	4420	m/s	Closing speed
$\Delta\phi_0, \Delta\theta_0, \Delta\psi_0$	$\pm(10, 20, 30)$	deg	Missile initial attitude error
$\Delta p_0, \Delta q_0, \Delta r_0$	$\pm(0.5, 0.5, 0.5)$	deg/s	Missile initial body rate error
$\Delta\alpha$	± 1.5	deg	Target position within seeker FOV
$\Delta\alpha_v$	± 1.0	deg	Closing velocity alignment error
ΔF_i	0.05 (3σ)	–	Each thruster force uncertainty
x_0	$(1, \ldots, 1)$	–	Initial nozzle throat area (normalized)
dt	1×10^{-3}	s	Simulation timestep
ϵ_{opt}	1×10^{-5}	–	LM optimality condition tolerance
ϵ_{step}	1×10^{-2}	–	LM step size tolerance
λ_0	1	–	Initial LM damping parameter
K	5	–	Maximum LM iteration count

TABLE II
MISSILE MASS PROPERTIES, LEVER ARMS, AND GUIDANCE/CONTROLLER PARAMETERS

Parameter	Value	Unit	Description
m_0	80	kg	Missile initial mass
I_{xx}, I_{yy}, I_{zz}	(1.6, 10.4, 10.4)	kg m^2	Moments of inertia about body axes
a, b	(0.748, 0.084)	m	Torque lever arms
$K_P^\phi, K_P^\theta, K_P^\psi$	(400, 400, 400)	–	PD proportional gain
$K_D^\phi, K_D^\theta, K_D^\psi$	(28.3, 28.3, 28.3)	–	PD derivative gain
N	4	–	Navigation constant (TPN law)

TABLE III
DACS GEOMETRY AND COMBUSTION PARAMETERS

Parameter	Value	Unit	Description
$A_{t_{D,\max}}$	1.895×10^{-4}	m^2	Maximum divert nozzle throat area
$A_{t_{A,\max}}$	3.159×10^{-6}	m^2	Maximum attitude nozzle throat area
R	352	J/(kg K)	Specific gas constant
T_c	2500	K	Combustion chamber temperature
P_{c_0}	1	MPa	Initial chamber pressure
V_{c_0}	0.01	m^3	Chamber free volume
A_b	0.047	m^2	Propellant burning surface area
ρ_p	1680	kg/m^3	Propellant density
γ	1.25	–	Specific heat ratio
a_b	2.912×10^{-6}	–	Burn rate coefficient
n	0.6	–	Pressure exponent of burn rate

TABLE IV
ONLINE COMPUTATION TIME

Module	Computation Time [ms]		
	Mean	Std. dev.	Max
Control Allocation (CA)	1.078	0.323	7.476
Guidance and Control (TPN-PD)	0.070	0.002	0.911

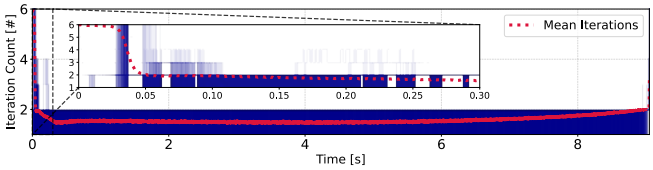

Fig. 6. Mean LM iteration count in the Monte-Carlo study.

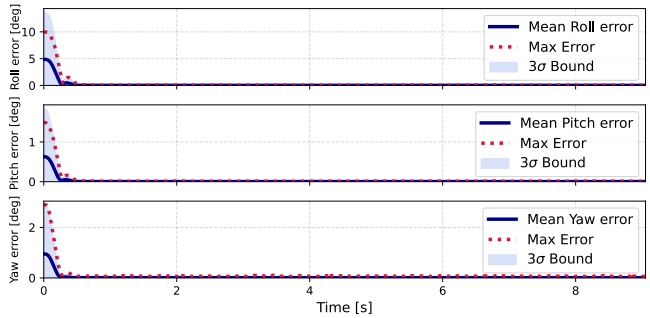

Fig. 7. Time histories of the roll, pitch, and yaw attitude errors in the Monte-Carlo study.

VI. Concluding Remarks

This paper presented a differentiable programming-based constrained LM optimization framework for CA in a single-chamber solid-propellant DACS. By leveraging Jacobians via automatic differentiation, the algorithm efficiently computes optimal nozzle configurations that satisfy physical throat constraints while minimizing tracking errors. Closed-loop 6-DoF simulations, including a 500-run Monte-Carlo analysis, verified that the proposed framework exhibits stable empirical convergence and robust hit-to-kill performance under stochastic thrust variations, attitude and body-rate perturbations, and geometric misalignment uncertainties. These results demonstrate the algorithm's high-precision control capability and support the computational tractability of the proposed framework for onboard implementation.

While this study focuses on a quasi-steady-state approach, future efforts will integrate full chamber pressure dynamics directly into the optimization loop and include quantitative comparisons against representative decoupled strategies [10], [11], [12] and generic constrained nonlinear solvers such as SQP. Additional studies on sensitivity to combustion-parameter uncertainties, sensor noise, and actuator latencies will further clarify the achievable real-time performance of this differentiable optimization framework for next-generation interceptor systems.

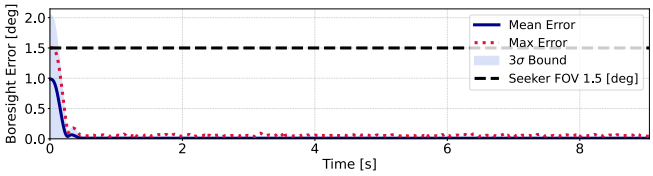

Fig. 8. Boresight-error history in the Monte-Carlo study.

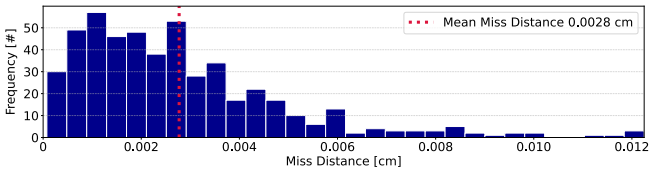

Fig. 9. Miss distance histogram in the Monte-Carlo study.

References

[1] K. M. Slayer, "Hypersonic weapons: Background and issues for congress," Congressional Research Service, Tech. Rep., 2022.

[2] A. Sleiman and J. Harris, "Requirements comparison hit-to-kill vs warhead for TMD," THAAD Project Office, Tech. Rep., 1999.

[3] G. Sullins, "Exo-atmospheric intercepts- bringing new challenges to standard missile," *Johns Hopkins APL Technical Digest*, vol. 22, no. 3, pp. 260–274, 2001.

[4] G. Park, J. Choi, D. H. Jeong, and J.-H. Kim, "Optimal impact angle guidance via first-order optimization under nonconvex constraints," in *2024 American Control Conference (ACC)*. IEEE, 2024, pp. 778–784.

[5] G. Park, S.-Y. Lee, and J.-H. Kim, "ADMM-based control allocation for kill vehicle guidance," in *Symposium of the Korean Institute of Communications and Information Sciences*, 2021, pp. 571–572.

[6] Y.-B. Wang, M. Ji, and H. Chang, "Modeling and dynamic characteristics analysis on solid attitude control motor using pintle thrusters," *Aerospace Science and Technology*, vol. 106, p. 106130, 2020.

[7] V. Saravanan, J. Ko, S. Lee, N. Murugan, and V. S. Kumar, "Conceptual aerodynamic design of pintle nozzle for variable-thrust propulsion," *International Journal of Aeronautical and Space Sciences*, vol. 21, no. 1, pp. 1–14, 2020.

[8] D. Yan, Z. Zhao, A. Song, F. Li, L. Ye, G. Zhao, and S. Ma, "Transient characteristics of fluidic pintle nozzle in a solid rocket motor," *Aerospace*, vol. 11, no. 3, p. 243, 2024.

[9] J. Napior and V. Garmy, "Controllable solid propulsion for launch vehicle and spacecraft application," in *57th International Astronautical Congress*, 2006, pp. C4–2.

[10] S. Joner and I. Quinquis, "Control of an exoatmospheric kill vehicle with a solid propulsion attitude control system," in *AIAA guidance, navigation, and control conference and exhibit*, 2006, p. 6572.

[11] W. Lee, Y. Eun, H. Bang, and H. Lee, "Efficient thrust distribution with adaptive pressure control for multinozzle solid propulsion system," *Journal of Propulsion and Power*, vol. 29, no. 6, pp. 1410–1419, 2013.

[12] H. Lee and H. Bang, "Efficient thrust management algorithm for variable thrust solid propulsion system with multi-nozzles," *Journal of Spacecraft and Rockets*, vol. 57, no. 2, pp. 328–345, 2020.

[13] H. Lee, I.-S. Jeon, and J. Suk, "Efficient generation of combustion chamber pressure command for multi-nozzle solid propulsion system," *International Journal of Aeronautical and Space Sciences*, vol. 26, pp. 2750–2767, 2025.

[14] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[15] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, Y. Katariya, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: Composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/jax-ml/jax

[16] A. Ranganathan, "The Levenberg-Marquardt algorithm," *Tutorial on LM algorithm*, vol. 11, no. 1, pp. 101–110, 2004.

[17] C. Kanzow, N. Yamashita, and M. Fukushima, "Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints," *Journal of Computational and Applied Mathematics*, vol. 172, no. 2, pp. 375–397, 2004.

[18] J. Bergmans and R. Di Salvo, "Solid rocket closed-loop pressure control," in *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2003, p. 4968.

[19] G. P. Sutton and O. Biblarz, *Rocket Propulsion Elements*, 8th ed. John Wiley & Sons, 2010.

[20] P. Zarchan, *Tactical and Strategic Missile Guidance*, 7th ed. American Institute of Aeronautics and Astronautics, Inc., 2019.

[21] H.-M. Jeon, T. Y. Kang, J. Park, and C.-K. Ryoo, "High-altitude terminal control of anti-air missiles with a terminal booster and front lateral impulse thrusters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 2, pp. 2071–2081, 2024.

[22] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*, 3rd ed. AIAA Education Series, 2014.

[23] C. T. Kelley, *Iterative methods for optimization*. SIAM, 1999.