HiSST: 4th International Conference on High-Speed Vehicle Science & Technology

22–26 September 2025, Tours, France

CEAS
Council of European Aerospace Societies

KL-Divergence Model Predictive Control for On–Off Thrusters in Spaceplane

You-Jeong Lee[1] *Dain Yoon*[1] *Se-Hwan An*[1] *Chang-Hun Lee*[1] *Hyoung-Sik Choi*[2] *Seungchan Shin*[2]

Abstract

This paper presents a sampling-based discrete model predictive control (KL-MPC) framework for spaceplane equipped with on/off thrusters. In contrast to conventional Model Predictive Path Integral (MPPI) control, which perturbs continuous inputs with Gaussian noise, the proposed approach represents actuator commands using Bernoulli distributions and updates their probabilities through cost-weighted rollouts. This discrete formulation inherently accommodates binary firing logic without heuristic thresholding, while actuator dynamics are captured through a first-order lag model. The cost function is evaluated in moment-tracking coordinates to simultaneously account for reference tracking accuracy, fuel efficiency, and switching reduction. Simulation studies on a reentry spaceplane demonstrate that the proposed method mitigates overshoot and inefficiency associated with pseudo-inverse thresholding, while achieving performance comparable to an optimization-based MPC–MILP controller at sub-millisecond runtimes. The results demonstrate that KL-MPC achieves high-performance attitude control while remaining computationally efficient and explicitly respecting discrete actuator constraints.

Keywords: *Attitude control, Spaceplane, On/Off thrusters, Discrete Model Predictive Control (KL-MPC), KL-divergence minimization*

Nomenclature

Latin

t – Time index
i – Actuator index
K – Number of rollouts
T – Prediction horizon
Δt – Simulation/control time step
x_t – System state at time t
$U = \{u_t\}$ – Control sequence over the horizon
$u_t \in \{0,1\}^n$ – Binary actuator command vector
$u_{t,i}^{\mathrm{cmd}}$ – Commanded input
$u_{t,i}^{\mathrm{act}}$ – Actual actuator state
$\theta = \{\pi_{t,i}\}$ – Bernoulli parameters
M_t – Control moment vector
M_t^{ref} – Reference moment
B – Thruster effectiveness (allocation) matrix
$\ell(x_t, u_t)$ – Stage cost
$S(U)$, S_k – Trajectory cost , cost of rollout k
$p^*(U)$, $q_\theta(U)$ – Target and proposal distributions
$D_{\mathrm{KL}}(\cdot\|\cdot)$ – KL divergence
w_t, w_f, w_{sw} – Weights for tracking, fuel, switching

Greek

λ – Temperature parameter
η – Threshold for binary execution
τ_{act} – Actuator time constant (lag)

Superscripts

cmd – Commanded (pre-lag)
act – Actual (post-lag)

Subscripts

t – Time index
i – Actuator index

1. Introduction

On/off thrusters, such as reaction control systems (RCS), are widely employed in spaceplanes and reentry vehicles, where actuators deliver discrete impulses rather than continuous forces [1]. Control

[1]*Department of Aerospace Engineering, KAIST, Daejeon, Republic of Korea, lyj801@kaist.ac.kr*
[1]*Department of Aerospace Engineering, KAIST, Daejeon, Republic of Korea, dainyoon@kaist.ac.kr*
[1]*Department of Aerospace Engineering, KAIST, Daejeon, Republic of Korea, shan99@kaist.ac.kr*
[1]*Department of Aerospace Engineering, KAIST, Daejeon, Republic of Korea, lckdgns@kaist.ac.kr*
[2]*Korea Aerospace Research Institute, Daejeon, Republic of Korea, chs@kari.re.kr*
[2]*Korea Aerospace Research Institute, Daejeon, Republic of Korea, seungchan@kari.re.kr*

strategies for such systems have traditionally followed two main tracks. Early approaches relied on heuristic methods such as bang–bang control and pulse-width modulation (PWM) [2], which are computationally efficient but prone to chattering, inefficiency, and limited capability to enforce constraints. More recently, optimization-based techniques, including mixed-integer linear programming (MILP) [3] and model predictive control (MPC), have been introduced to explicitly capture actuator discreteness and fuel usage [4], with some works extending MPC formulations to PWM-based actuation [5]. While these methods can deliver near-optimal performance, their computational complexity generally precludes deployment in high-frequency flight control.

A clear trade-off therefore arises depending on the control frequency. For high-frequency applications such as tactical missiles or reentry vehicles operating at millisecond-level cycles, heuristic methods remain prevalent for their real-time feasibility but offer limited optimality [1]. Conversely, for slower tasks such as satellite attitude control or vehicle guidance, optimization-based methods become feasible, though they often require solve times on the order of seconds. The open challenge is to design controllers that capture discrete actuator characteristics and constraints while remaining computationally tractable at high update rates [4].

Sampling-based optimal control has emerged as a promising approach to bridge this gap. Among them, Model Predictive Path Integral (MPPI) control is particularly attractive, combining near-optimal performance with tractable computation by exploiting massively parallel rollouts [6, 7]. However, the standard MPPI formulation relies on Gaussian perturbations around continuous control inputs, limiting its applicability to binary or discrete actuators. More generally, MPPI can be interpreted as a Kullback–Leibler (KL) divergence minimization problem, where the optimal control distribution is approximated within a parameterized family [8]. This interpretation provides a principled foundation for sampling-based MPC, but existing formulations remain confined to Gaussian exploration.

An influential study by Williams et al. [7] established this information-theoretic view of MPPI, deriving the update law as a KL divergence minimization under Gaussian sampling. While this perspective justified the classical MPPI update, it did not address systems with discrete or binary actuators. In contrast, the present work develops an extension of KL-based sampling MPC, grounded in a KL perspective that naturally yields maximum-likelihood updates from rollout data. Building on this foundation, we introduce *KL-MPC*, a framework that parameterizes actuator commands as Bernoulli random variables and updates their activation probabilities through cost-weighted rollouts. This discrete specialization directly handles binary firing commands without heuristic thresholding, incorporates actuator lag through a first-order model, and remains compatible with other discrete distributions. By embedding these elements in a sampling-based MPC setting, KL-MPC extends the path-integral family of methods beyond Gaussian exploration to a broader class of systems governed by discrete actuation.

To validate the proposed approach, we apply KL-MPC to the attitude control of a reentry spaceplane inspired by the X-37B configuration. The baseline autopilot follows a nonlinear three-loop structure [9], which generates pseudo-control inputs subsequently allocated to on/off RCS thrusters. Conventional approaches either quantize continuous commands into discrete firings, offering simplicity, or solve MILPs that capture actuator discreteness more accurately at the expense of computational cost. In contrast, KL-MPC directly models binary actuator distributions, combining efficiency with accuracy to deliver near-oracle tracking performance at sub-millisecond runtimes. The main contributions of this paper are:

- Proposing a sampling-based discrete MPC framework that models on/off thruster commands as Bernoulli random variables, enabling direct handling of binary actuation without heuristic thresholding.
- Designing the control law across both inner-loop rate control and allocation, using a predictive reference trajectory that accounts for actuator lag and discrete firing dynamics.
- Validating the method on reentry vehicle attitude control, showing that KL-MPC achieves oracle-level tracking accuracy and efficiency.

The remainder of the paper is organized as follows. Section 2 reviews background and notation. Section 3 details the proposed KL-MPC formulation. Section 4 presents simulation studies with comparisons against baseline controllers. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Preliminaries

2.1. Discrete Actuator Systems

Many aerospace vehicles rely on actuators that operate in an inherently discrete manner rather than producing continuous-valued forces. A representative case is the Reaction Control System (RCS) of spaceplanes or reentry vehicles, where each thruster is either fully on or fully off [10]. The resulting control input is therefore constrained to a binary domain,

$$u_t \in \{0,1\}^n, \tag{1}$$

with n actuators. This discreteness introduces a fundamental challenge: feasible commands must be chosen from a combinatorial set whose size grows exponentially with n and the horizon length, making classical convex optimization unsuitable for real-time control.

2.2. Sampling-Based MPC

Model Predictive Control (MPC) is a receding-horizon method in which future control actions $U = \{u_t\}_{t=0}^{T-1}$ are optimized while explicitly accounting for system dynamics and operational constraints. The optimization is formulated as

$$U^* = \arg\min_U \sum_{\tau=0}^{T-1} \ell(x_\tau, u_\tau) + \ell_T(x_T), \tag{2}$$

subject to

$$x_{\tau+1} = f(x_\tau, u_\tau), \tag{3}$$

with $\ell(\cdot)$ and $\ell_T(\cdot)$ denoting stage and terminal costs. From the optimized sequence, only the initial control u_0^* is applied, after which the prediction horizon is shifted forward. While this structure provides strong guarantees on constraint satisfaction and long-term performance, solving Eq.(2) exactly is computationally prohibitive for nonlinear dynamics or discrete inputs.

Sampling-based approaches approximate the MPC solution through Monte Carlo rollouts, replacing deterministic optimization with stochastic search in control space. The algorithm iteratively samples input sequences, evaluates their performance through rollout simulations, and updates the proposal distribution to emphasize low-cost trajectories. Two popular representatives are the Cross-Entropy Method (CEM) and Model Predictive Path Integral (MPPI) control [6, 7]. Both methods exploit massively parallel computation, which substantially improves scalability and makes them well suited for time-critical applications in robotics and aerospace.

2.3. KL-Divergence Formulation

A unifying perspective is to interpret these sampling-based controllers as minimizing the Kullback–Leibler (KL) divergence between a target distribution and a parameterized proposal. The *optimal control distribution* is defined as a probability distribution weighted by trajectory costs,

$$p^*(U) \propto p(U)\,\exp\big(-S(U)/\lambda\big), \tag{4}$$

where $S(U)$ is the trajectory cost and λ is a temperature parameter. The proposal distribution $q_\theta(U)$, parameterized by θ, is updated by

$$\theta^* = \arg\min_\theta D_{\mathrm{KL}}(p^*(U) \,\|\, q_\theta(U))\,. \tag{5}$$

This information-theoretic formulation, originally derived in the context of path integral control [8], shows that the exponential weighting used in MPPI and CEM can be interpreted as a maximum-likelihood update biased toward low-cost rollouts [7].

This information-theoretic viewpoint reveals that methods such as MPPI and CEM can be interpreted as maximum-likelihood updates guided by trajectory costs. In this sense, the KL divergence provides a principled bridge between stochastic search in control space and optimal control objectives, enabling sampling-based MPC to be understood within a unified statistical framework.

3. Methodology

3.1. KL-MPC with Bernoulli Parameterization

We introduce a discrete sampling-based MPC framework designed to on/off thrusters. Control inputs are parameterized as Bernoulli random variables, such that

$$q_\theta(U) = \prod_{t=0}^{T-1} \prod_{i=1}^{n} \mathrm{Bern}(u_{t,i}; \pi_{t,i}), \qquad \pi_{t,i} \in [0,1], \tag{6}$$

with $\pi_{t,i}$ denoting the activation probability of actuator i at time t. This formulation naturally captures the binary nature of thrusters while avoiding heuristic thresholding of continuous inputs.

3.2. Weighted MLE Update

At each iteration, K candidate sequences are sampled from q_θ, simulated through the system dynamics with actuator lag, and scored via the cost S_k. With this choice of sampling distribution, the importance ratio simplifies, and the normalized weights reduce to

$$v_k = \frac{1}{z}\,\exp(-S_k/\lambda), \qquad \tilde{v}_k = \frac{v_k}{\sum_{k=1}^{K} v_k} \tag{7}$$

the update for the Bernoulli parameter is

$$\pi_{t,i}^{\text{new}} = \sum_{k=1}^{K} \tilde{v}_k \, u_{t,i}^{(k)}. \tag{8}$$

This corresponds to a weighted maximum-likelihood estimate (MLE), shifting the proposal distribution closer to the optimal distribution q^*.

3.3. Receding Horizon and Cost Modeling

The proposed controller integrates actuator dynamics and allocation objectives within a receding horizon formulation. The commanded angular-rate input ω^{cmd} is shaped through a first-order error dynamics model, derived via nonlinear dynamic inversion (NDI), to produce a moment reference trajectory M^{ref}. This construction ensures that the reference moment is consistent with actuator delay and feasible for tracking with on/off thrusters.

At each stage, thruster commands are evaluated with the cost function

$$\ell(x_t, u_t) = w_t \big\| M_t^{\text{ref}} - \mathbf{B} u_t^{\text{act}} \big\|^2 + w_f \|u_t\|_1 + w_{\text{sw}} \|u_t - u_{t-1}\|_1, \tag{9}$$

where $\mathbf{B}$ is the control effectiveness matrix and w_t, w_f, w_{sw} weight the relative importance of tracking, fuel efficiency, and switching activity, respectively. For each Monte Carlo rollout k, the stage costs are accumulated to form the trajectory cost

$$S_k = \sum_{t=0}^{T-1} \ell(x_t^{(k)}, u_t^{(k)}). \tag{10}$$

The rollout dynamics explicitly incorporate rigid-body rotation and actuator lag. The rigid-body rotational dynamics are propagated as

$$\dot{\omega} = J^{-1}\big(M - \omega \times (J\omega)\big), \tag{11}$$

where J is the inertia matrix, ω is the body angular velocity, and $M = \mathbf{B}u$ denotes the applied control moment. The actuator dynamics are modeled by a first-order lag for each thruster,

$$\dot{u}_{t,i} = \tfrac{1}{\tau_{\text{act}}}\big(u_{t,i}^{\text{cmd}} - u_{t,i}\big), \tag{12}$$

with time constant τ_{act}.

Finally, warm-start initialization with the pseudo-inverse allocation $\mathbf{B}^{\dagger} M^{\text{ref}}$ accelerates convergence of the sampling process while retaining the stochastic nature of the updates.

3.4. Algorithmic Summary

The proposed KL-MPC procedure can be summarized as follows:

Algorithm 1 KL-MPC with Bernoulli Sampling

Require: K rollouts, horizon T, temperature λ, initial Bernoulli parameters $\theta = \{\pi_{t,i}\}$
Ensure: first-step command $u^{\text{cmd}} \in \{0,1\}^n$
1: **while** task not completed **do**
2: **for** $k = 1$ to K **do**
3: Sample trajectory $U^{(k)}$ with $u_{t,i}^{(k)} \sim \text{Bern}(\pi_{t,i})$
4: Propagate dynamics with $U^{(k)}$ and compute total cost S_k
5: **end for**
6: Compute normalized weights $v_k \propto \exp(-(S_k - S_{\min})/\lambda)$
7: Update parameters: $\pi_{t,i}^{\text{new}} \leftarrow \sum_k v_k u_{t,i}^{(k)}$
8: Apply first-step command $u_{0,i}^{\text{cmd}} = \mathbf{1}[\pi_{0,i}^{\text{new}} \geq \eta]$
9: Shift horizon and update state
10: **end while**

4. Simulation Study

4.1. Model

The simulation environment emulates the X-37B reentry spaceplane under nominal reentry conditions at Mach 30 and an altitude of $100\,\text{km}$. A 5-DOF rigid-body model is used, including translational and rotational dynamics, while guidance laws are omitted to focus solely on the attitude control performance. The vehicle is equipped with twelve on/off RCS thrusters, each modeled with binary actuation and first-order valve dynamics. All simulations are performed on a CPU-based platform.

The translational equations of motion are

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} f_{B,x}^a \\ f_{B,y}^a \\ f_{B,z}^a \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_{B,x}^t \\ f_{B,y}^t \\ f_{B,z}^t \end{bmatrix} + g \begin{bmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{13}$$

where m is the mass, (u, v, w) are body-axis velocities, (p, q, r) are angular rates, and f_B^a, f_B^t denote aerodynamic and thruster forces in body coordinates.

Rotational motion follows Euler's rigid-body dynamics:

$$I_{xx}\dot{p} - I_{xz}\dot{r} - I_{xz}pq + (I_{zz} - I_{yy})qr = L_a + L_t, \tag{14}$$

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})pr + I_{xz}(p^2 - r^2) = M_a + M_t, \tag{15}$$

$$I_{zz}\dot{r} - I_{xz}\dot{p} + (I_{yy} - I_{xx})pq + I_{xz}qr = N_a + N_t, \tag{16}$$

with inertia components $I_{xx}, I_{yy}, I_{zz}, I_{xz}$ and aerodynamic/propulsive moments (L_a, M_a, N_a) and (L_t, M_t, N_t).

Each RCS thruster is commanded by a binary input $u_{t,i}^{\text{cmd}} \in \{0, 1\}$, while the actual valve state $u_{t,i}$ follows a first-order lag:

$$\dot{u}_{t,i} = \frac{1}{\tau_{\text{act}}}(u_{t,i}^{\text{cmd}} - u_{t,i}), \tag{17}$$

with time constant τ_{act}. This formulation captures both the discrete actuation and the finite response delay of the thruster valves.

4.2. Baseline Controllers

Three representative controllers are considered in this study. The underlying information-theoretic update (importance weights and KL-view), our contribution in experiments is the discrete Bernoulli parameterization at the actuator level.

(a) **3-Loop Controller:** A representative continuous-control baseline that does not explicitly account for actuator discreteness. Pseudo-control moments are computed through the inner-loop structure and then mapped to actuator commands via the pseudoinverse of the effectiveness matrix. Since the resulting u is continuous, a simple thresholding scheme (threshold 0.5) is applied to obtain binary on/off commands. This approach is widely used for its simplicity and very low computational cost.

(b) **MPC-MILP Controller (Oracle):** A model predictive control formulation expressed as a mixed-integer linear program. This controller jointly optimizes inner-loop angular-rate tracking and actuator allocation while enforcing binary thruster constraints. It provides near-optimal tracking performance with smooth transients and is employed as an oracle reference, although its computational burden precludes real-time implementation.

(c) **KL-MPC (Proposed):** A sampling-based MPC implementation that uses Bernoulli sampling for actuator commands and cost-weighted probability updates. The scheme simultaneously addresses inner-loop tracking and discrete actuator allocation, directly handling binary thruster commands and actuator lag without heuristic thresholding.

4.3. Simulation Setup

All controllers are tuned to achieve comparable closed-loop bandwidth under identical vehicle dynamics, actuator lag, and cost structure. The 3-Loop baseline implements angular-rate feedback in the inner loop using second-order error dynamics characterized by (ζ, ω_n), whereas both MPC-MILP and KL-MPC employ first-order error dynamics parameterized by time constants τ. For the MPC-based controllers (MPC-MILP and KL-MPC), the simulation employs a timestep of $\Delta t = 0.01$ s with a prediction horizon of $T = 5$. For KL-MPC, $K = 512$ rollouts are generated with temperature parameter $\lambda = 1.0$. Performance metrics are averaged over $N_{\text{seeds}} = 20$ randomized Monte Carlo trials. The principal tuning parameters are summarized in Table 1.

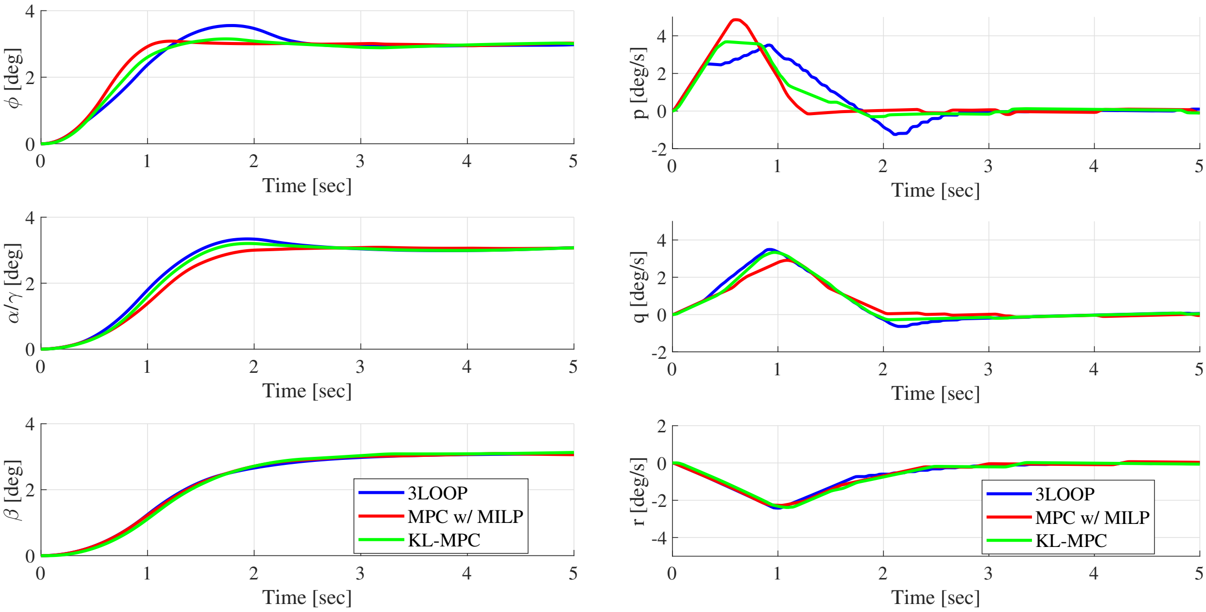
Table 1. Controller tuning parameters used in simulation.

Parameter	3-Loop	MPC-MILP	KL-MPC
Inner loop (Roll)	$\zeta_R = 0.8,\ \omega_{n,R} = 5$ Hz	$\tau_R = 1/12$ s	$\tau_R = 1/12$ s
Inner loop (Pitch)	$\zeta_P = 0.7,\ \omega_{n,P} = 3$ Hz	$\tau_P = 1/6$ s	$\tau_P = 1/6$ s
Inner loop (Yaw)	$\zeta_Y = 0.7,\ \omega_{n,Y} = 3$ Hz	$\tau_Y = 1/5$ s	$\tau_Y = 1/5$ s
Prediction horizon T	–	5	5
Weight w_t (Roll, Pitch, Yaw)	–	$15,\ 3,\ 3$	$0.5,\ 0.02,\ 0.02$
Weight w_f	–	100	1200
Weight w_{sw}	–	2800	1800
Sampling rollouts K	–	–	512
Temperature λ	–	–	1

4.4. Results and Analysis

4.4.1. Step Tracking Responses

Figure 1 shows step tracking responses for Euler angles and angular rates. The 3-Loop baseline exhibits noticeable overshoot and slower convergence. MPC-MILP (oracle) eliminates overshoot and yields smooth transients. KL-MPC closely follows the oracle while avoiding the inefficiency of naive thresholding.

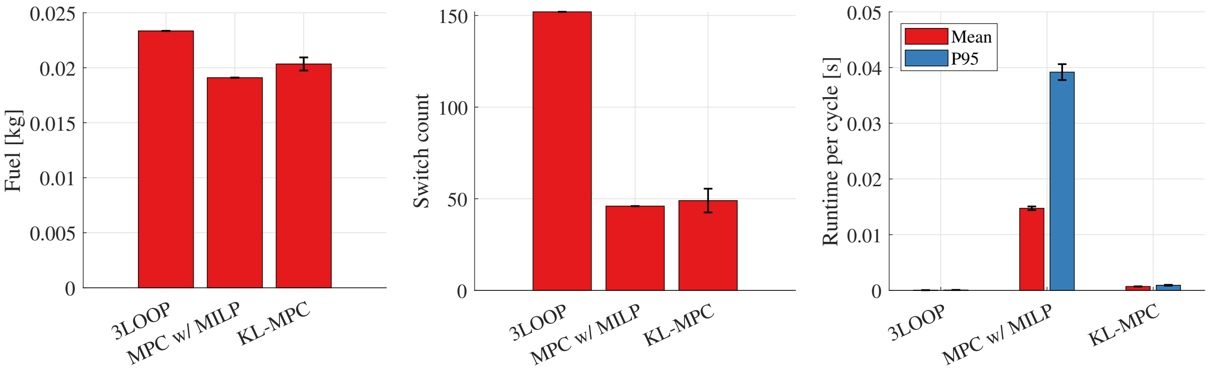


(a) Euler angles ϕ, α, β. **(b)** Angular rates p, q, r.

Fig 1. Step tracking responses for the three compared methods.

4.4.2. Aggregate Metrics

Figure 2 and Table 2 summarize tracking accuracy, fuel consumption, switching activity, and runtime. The 3-Loop baseline shows the largest fuel usage (23.35 g) and switching count (152). MPC-MILP achieves the lowest fuel and switching (19.10 g, 46) but requires substantially higher runtimes (mean 14.7 ms, 95th percentile 39 ms). KL-MPC attains fuel 20.35 g and switching 49, approaching the oracle while maintaining sub-millisecond runtimes (mean 0.70 ms, 95th percentile 0.92 ms).



(a) Fuel consumption (total). **(b)** Switching count. **(c)** Runtime per cycle (mean and 95th percentile).

Fig 2. Efficiency metrics across three methods (3-Loop, MPC-MILP, KL-MPC).

4.5. Effect of Predictive and Constant References on Performance

While the previous simulations confirmed that sampling-based MPC is capable of accommodating binary actuators, an important question remains as to why the framework should be embedded not only at the allocation stage but also within the inner-loop control architecture. The rationale lies in the observation that achieving high-performance control with on/off thrusters requires anticipation of the future evolution

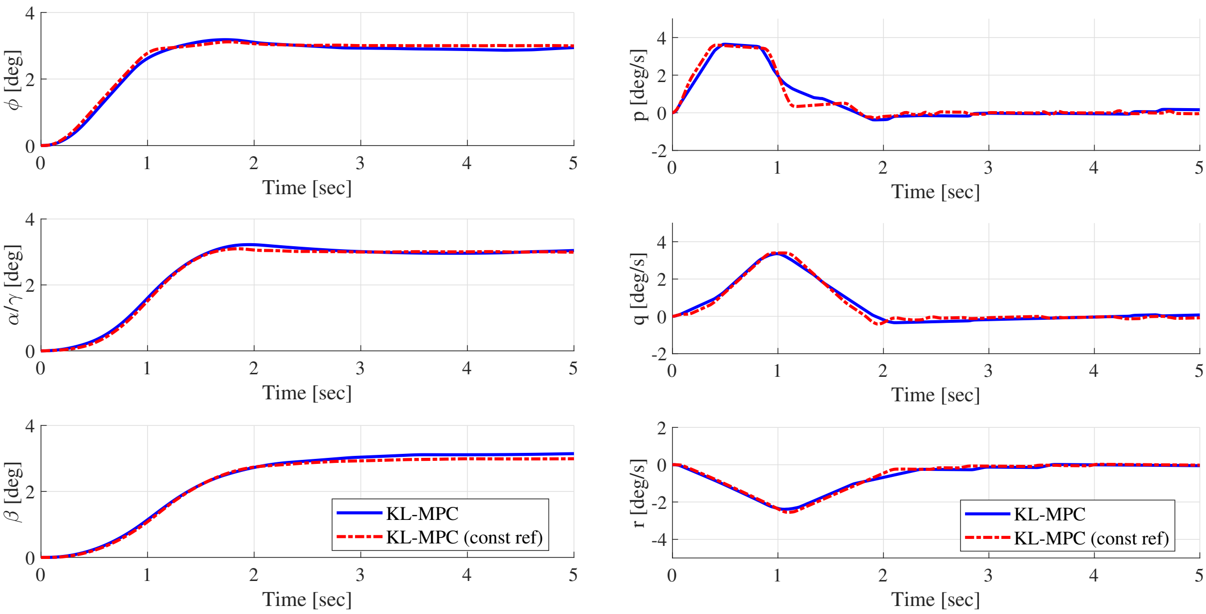
Table 2. Aggregate performance on step tracking (mean $\pm$ std over N_{seeds} runs).

Metric	3-Loop PI	MPC-MILP	KL-MPC (Proposed)
Angle RMSE [deg]	1.9629 ± 0.0000	**1.9594** ± 0.0000	1.9849 ± 0.0046
Rate RMSE [deg/s]	2.1041 ± 0.0000	2.0606 ± 0.0000	**2.0383** ± 0.56
Fuel [g]	23.35 ± 0.00	**19.10** ± 0.00	20.35 ± 0.59
Switches	152.0 ± 0.0	**46.0** ± 0.0	49.0 ± 6.5
Time [ms]	**0.0428** ± 0.0040	14.74 ± 0.33	0.700 ± 0.013
Time95 [ms]	**0.0647** ± 0.0091	39.18 ± 1.43	0.923 ± 0.071

of the commanded moment, rather than merely matching the instantaneous demand. To examine this hypothesis, we conduct a comparative study between two variants of the proposed controller that differ only in the construction of the reference moment trajectory: (i) **KL-MPC**, which employs a first-order error-dynamics model to generate a smooth reference trajectory $M^{\text{ref}}(t)$ over the horizon; and (ii) **KL-MPC (constant reference)**, which maintains a fixed reference across the horizon.

4.5.1. Step Tracking Responses and Aggregate Metrics

Figure 3 compares the closed-loop responses under step commands. While the angle trajectories of the two variants appear broadly similar, the constant-reference case exhibits less smooth convergence. This difference is more clearly visible in the angular rate responses, where the constant-reference method produces chattering-like oscillations reminiscent of the 3-Loop+PI allocation. Nevertheless, unlike the heuristic thresholding case, the constant-reference variant does not generate overshoot in the angle response, indicating a modest improvement over purely heuristic control.



(a) Euler angles ϕ, α, β. **(b)** Angular rates p, q, r.

Fig 3. Step tracking responses: KL-MPC and KL-MPC (const ref).

We further evaluate both controllers over 20 randomized runs, with aggregate metrics reported in Table 3. The predictive-reference variant of KL-MPC achieves lower fuel consumption (0.0204 ± 0.0006 g) and fewer switchings (49 ± 6.5) compared with its constant-reference counterpart (0.0274 ± 0.0012 g, 124.5 ± 16.9). Angle- and rate-tracking errors remain comparable between the two, yet the constant reference variant exhibits degraded efficiency, effectively reverting toward the performance profile of the heuristic 3-Loop+PI controller. These results highlight that, in discrete actuator systems, accurate tracking alone is insufficient: exploiting predictive reference dynamics is essential to simultaneously suppress chattering, reduce fuel expenditure, and minimize switching activity.

Table 3. Aggregate performance of KL-MPC and KL-MPC (const ref) (mean $\pm$ std over N_{seeds} runs).

Metric	KL-MPC	KL-MPC (const ref)
Angle RMSE [deg]	1.9849 ± 0.0046	$\mathbf{1.9825} \pm 0.0038$
Rate RMSE [deg/s]	$\mathbf{2.0383} \pm 0.0097$	2.0710 ± 0.0085
Fuel [g]	$\mathbf{20.35} \pm 0.59$	27.37 ± 1.22
Switches [count]	$\mathbf{49.0} \pm 6.49$	124.5 ± 16.95
Time [ms]	$\mathbf{0.700} \pm 0.013$	0.718 ± 0.021
Time95 [ms]	$\mathbf{0.923} \pm 0.071$	0.981 ± 0.045

4.6. Discussion

The 3-Loop+PI baseline remains attractive for its computational efficiency and straightforward implementation, but its reliance on heuristic allocation leads to overshoot, oscillatory responses, and inefficient actuator usage. In contrast, the MPC-MILP benchmark explicitly enforces actuator discreteness and switching constraints, achieving near-oracle tracking accuracy and fuel efficiency. However, the associated computational burden renders it impractical for high-frequency control cycles, particularly in reentry or agile maneuvering scenarios.

The proposed KL-MPC closes this gap by combining the strengths of sampling-based MPC with a Bernoulli parameterization tailored for on/off thrusters. Across all baseline comparisons, it consistently achieves accurate tracking, reduced fuel consumption, and minimized switching, while maintaining runtimes on the order of microseconds, thus supporting real-time feasibility.

The ablation study further clarifies the role of predictive reference modeling. While both KL-MPC variants maintain comparable tracking accuracy, the constant-reference formulation incurs substantially higher fuel usage and switching counts, effectively reverting to the behavior of heuristic allocation methods. This indicates that predictive reference dynamics are essential not merely for smoother trajectories but also for realizing the efficiency benefits of discrete optimal control. Overall, the results indicate that achieving high-performance control with binary actuators necessitates discrete-conscious optimization together with predictive modeling, thereby establishing KL-MPC as a feasible and theoretically consistent approach for attitude regulation under on/off actuation limits.

5. Conclusion

This paper presented a sampling-based discrete predictive control framework, termed KL-MPC, for attitude control with on/off thrusters. Building upon the information-theoretic formulation of sampling-based MPC [8], the method introduces a Bernoulli parameterization and probability-domain updates, thereby enabling direct treatment of binary actuator constraints and actuator lag without resorting to heuristic thresholding. A moment-tracking cost function was employed to balance reference tracking accuracy, fuel consumption, and switching activity.

Comparative simulations were conducted in a representative spaceplane reentry scenario, using a unified 3-Loop autopilot structure with three allocation strategies: PI allocation, MPC-MILP allocation, and the proposed KL-MPC allocation. The PI allocator achieved sub-millisecond runtimes but exhibited overshoot, chattering, and inefficient actuator utilization. The MPC-MILP allocator achieved near-oracle tracking accuracy and fuel efficiency, yet incurred runtimes on the order of tens of milliseconds per cycle, which limits real-time applicability. The proposed KL-MPC bridged these extremes, delivering oracle-level accuracy and efficiency while maintaining runtimes below one millisecond, thus striking a practical balance between real-time feasibility and constraint-aware optimality.

An additional ablation study highlighted the importance of predictive reference modeling. While both predictive and constant-reference variants of KL-MPC preserved accurate angle tracking, the constant-reference formulation suffered from excessive fuel usage and switching activity, yielding performance comparable to heuristic PI allocation. This finding underscores that discrete actuator systems require not only constraint-aware optimization but also predictive reference dynamics to fully exploit the benefits of sampling-based MPC.

Future work will extend the framework to six-degree-of-freedom vehicle dynamics, integrate mixed allocation with aerodynamic control surfaces, and pursue hardware-in-the-loop validation. Beyond aerospace applications, KL-MPC holds promise for a broad class of discrete decision-making problems, including on/off actuator networks, logistics routing, and mission-level scheduling, underscoring its potential as a versatile and high-performance control architecture for systems governed by binary constraints. Moreover, while the present study focuses on Bernoulli parameterization, the framework readily generalizes to other discrete distributions, further broadening its applicability to systems with more complex actuation or decision structures.

Acknowledgment

This work was partly supported by the Korea Institute for Advancement of Technology funded by the government (Ministry of Trade, Industry and Energy) in 2025 (Project No. RS-2022-KI002569, Specialized Workforce Development Program for an expert-to-be building up in the Urban Air Mobility), and by the Korea Aerospace Research Institute (KARI)'s own research project titled "Development of Control Performance Analysis Program for Spaceplanes".

References

[1] Johansen, T.A., Fossen, T.I.: Control allocation—a survey. Automatica **49**(5), 1087–1103 (2013)

[2] Bernelli-Zazzera, F., Mantegazza, P., Nurzia, V.: Multi-pulse-width modulated control of linear systems. Journal of guidance, control, and dynamics **21**(1), 64–70 (1998)

[3] Doman, D.B., Gamble, B.J., Ngo, A.D.: Quantized control allocation of reaction control jets and aerodynamic control surfaces. Journal of Guidance, Control, and Dynamics **32**(1), 13–24 (2009)

[4] Leomanni, M., Garulli, A., Giannitrapani, A., Scortecci, F.: An mpc-based attitude control system for all-electric spacecraft with on/off actuators. In: 52nd IEEE Conference on Decision and Control, pp. 4853–4858 (2013). IEEE

[5] Vazquez, R., Gavilan, F., Camacho, E.F.: Pulse-width predictive control for ltv systems with application to spacecraft rendezvous. Control Engineering Practice **60**, 199–210 (2017)

[6] Williams, G., Aldrich, A., Theodorou, E.A.: Model predictive path integral control: From theory to parallel computation. Journal of Guidance, Control, and Dynamics **40**(2), 344–357 (2017)

[7] Williams, G., Wagener, N., Goldfain, B., Drews, P., Rehg, J.M., Boots, B., Theodorou, E.A.: Information theoretic mpc for model-based reinforcement learning. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1714–1721 (2017). IEEE

[8] Theodorou, E.A., Todorov, E.: Relative entropy and free energy dualities: Connections to path integral and kl control. In: 2012 Ieee 51st Ieee Conference on Decision and Control (cdc), pp. 1466–1473 (2012). IEEE

[9] Yoon, D., Lee, C.-H.: Nonlinear three-loop autopilot design for spaceplanes. In: 2021 21st International Conference on Control, Automation and Systems (ICCAS), pp. 1502–1508 (2021). IEEE

[10] Hall, R.A., Hough, S., Orphee, C., Clements, K.: Design and stability of an on-orbit attitude control system using reaction control thrusters. In: AIAA Guidance, Navigation, and Control Conference, p. 0087 (2016)