

AI-Augmented Model Predictive Control for Safe and Adaptive Rendezvous and Proximity Operations

Luca Sportelli^{*}, Tyler Barr[†], Cagri Kilic[‡], and Di Wu[§]

Department of Aerospace Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114

Autonomous rendezvous and proximity operations (RPO) in adversarial orbital environments require guidance architectures balancing target pursuit, safety preservation, and real-time adaptability under dynamically evolving interaction conditions. Although learning-based approaches show promise, their application to safety-critical orbital robotics remains limited by concerns regarding interpretability, robustness, and constraint awareness. This work presents an adaptive Model Predictive Control (MPC) framework for autonomous spacecraft RPO in multi-agent adversarial scenarios. The proposed architecture combines a constrained receding-horizon MPC formulation with a data-driven supervisory tuning layer that adjusts controller parameters from offline closed-loop evaluation and online interaction geometry. Relative motion follows Clohessy–Wiltshire (CW) dynamics, enabling computationally efficient finite-horizon prediction and real-time quadratic optimization. The MPC formulation incorporates actuator limits, predictive keep-out-zone constraints, slack-variable feasibility handling, and optional Control Barrier Function (CBF) safety filtering. Rather than generating thrust commands directly, the adaptive layer modifies interpretable MPC parameters, including tracking weights, safety penalties, minimum-separation objectives, and keep-out-zone objectives. The framework was evaluated in the official Kerbal Space Program Differential Game (KSPDG) Capture-the-Satellite environment through Monte Carlo simulations. Results demonstrate improved closed-loop robustness, adaptive maneuvering behavior, and rendezvous performance compared with fixed-parameter MPC while preserving safety-aware operation and real-time feasibility, providing a modular, interpretable foundation for adaptive spacecraft RPO.

Nomenclature

A_c = continuous-time state matrix

^{*}Ph.D. Student, Department of Aerospace Engineering, Embry–Riddle Aeronautical University, Daytona Beach, FL 32114. Corresponding author: sportelli@my.erau.edu

[†]Undergraduate Student, Department of Aerospace Engineering, Embry–Riddle Aeronautical University, Daytona Beach, FL 32114.

[‡]Assistant Professor, Department of Aerospace Engineering, Embry–Riddle Aeronautical University, Daytona Beach, FL 32114.

[§]Assistant Professor, Department of Aerospace Engineering, Embry–Riddle Aeronautical University, Daytona Beach, FL 32114.

A_d	=	discrete-time state transition matrix
B_c	=	continuous-time control matrix
B_d	=	discrete-time control matrix
d_{BG}	=	Bandit–Guard separation distance, m
d_{BL}	=	Bandit–Lady separation distance, m
d_{KOZ}	=	keep-out-zone radius, m
$d_{\min}$	=	minimum-separation objective, m
$d_{\min, \text{eff}}$	=	effective minimum-separation objective, m
H_p	=	prediction horizon
H_u	=	control horizon
K_{KOZ}	=	active KOZ prediction horizon
h	=	Control Barrier Function safety function
J	=	MPC objective function
k_0, k_1	=	Control Barrier Function gains
n	=	orbital mean motion, rad/s
P_k	=	supervisory parameter vector
$P^\star$	=	offline-optimized supervisory parameter vector
Q	=	state-tracking weighting matrix
R	=	control-effort weighting matrix
r_0	=	reference-orbit radius, m
s_i	=	slack variable
T	=	control-horizon lifting matrix
$u_{\max}$	=	actuator acceleration limit
$\mathbf{u}$	=	control input vector
$\mathbf{U}_k$	=	stacked control decision sequence
w_U	=	control-effort weight
w_{KOZ}	=	KOZ safety weight
$w_{L, \text{pos}}$	=	Lady position-tracking weight
$w_{L, \text{vel}}$	=	Lady velocity-tracking weight
$\mathbf{x}$	=	spacecraft state vector
$\mathbf{X}_k$	=	stacked predicted state sequence
$\mathbf{z}$	=	QP decision vector

Φ	=	stacked state-transition prediction matrix
Γ	=	stacked control prediction matrix
γ_K	=	KOZ constraint decay coefficient
Δt	=	controller update timestep, s
λ_s	=	slack-variable penalty weight
μ	=	gravitational parameter, m^3/s^2
σ_{H_p}	=	terminal-cost multiplier

Subscripts

B	=	Bandit spacecraft
G	=	Guard spacecraft
L	=	Lady spacecraft
KOZ	=	keep-out zone

I. Introduction

Autonomous RPO remain a central challenge in orbital robotics, particularly in adversarial or uncooperative scenarios where safety, performance, and adaptability must be balanced in real time. These operations are becoming increasingly relevant for applications including on-orbit servicing, inspection, debris mitigation, and infrastructure protection in congested orbital environments.

MPC has been widely adopted in robotics and aerospace systems due to its ability to explicitly enforce constraints while optimizing future system behavior over a receding horizon [1, 2]. In orbital robotics, MPC formulations based on CW or Hill dynamics provide computationally efficient guidance strategies compatible with real-time operation and constrained maneuvering [3–5].

At the same time, learning-based approaches have demonstrated promising capabilities in adaptive guidance, trajectory optimization, and multi-agent interaction problems. Recent studies have highlighted the growing role of artificial intelligence techniques within spacecraft guidance, navigation, and control architectures, particularly for autonomous decision-making and adaptive mission operations [6]. However, the direct deployment of end-to-end learning architectures in safety-critical orbital environments remains challenging due to limited interpretability, robustness concerns, and the difficulty of guaranteeing constraint-aware behavior.

This work proposes an adaptive MPC framework for autonomous RPO in adversarial multi-agent scenarios. The proposed architecture preserves a constrained receding-horizon MPC structure throughout execution, while an outer-loop adaptive layer modifies selected optimization-level parameters governing the predictive controller behavior. The proposed architecture combines linearized orbital dynamics, constrained convex optimization, and adaptive parameter

tuning into a modular and interpretable control framework.

Figure 1 provides a visual overview of the proposed AI-augmented MPC framework, highlighting the interaction between the constrained receding-horizon MPC layer, predictive safety mechanisms, and the adaptive supervisory parameter-update process. State and geometry features extracted from the orbital interaction are processed by the adaptive supervisory layer, which updates interpretable MPC optimization parameters using safety and KOZ monitoring information. The constrained MPC/QP layer remains responsible for generating spacecraft thrust commands under orbital dynamics, actuator, and predictive safety constraints.

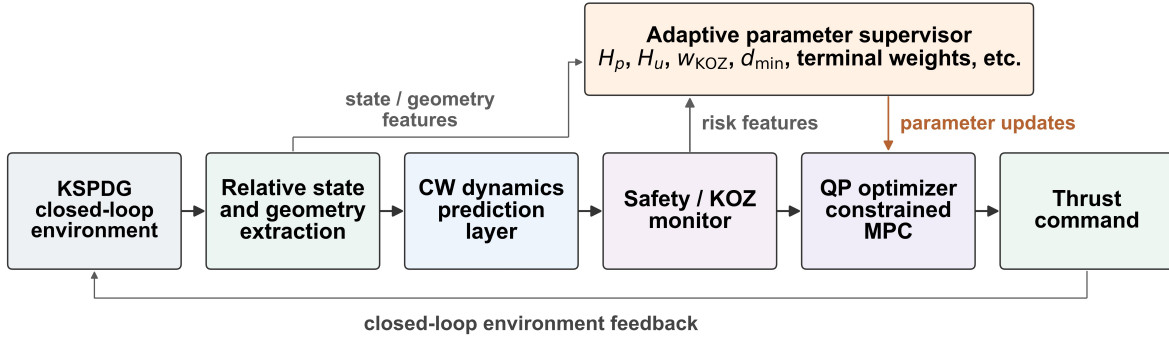


Fig. 1 Overview of the proposed AI-augmented MPC framework for adversarial RPO.

The framework is organized as two clearly separated loops. An *inner loop* solves a constrained receding-horizon quadratic program at every controller tick using the current measured state, fixed dynamics model, and active constraint set (Eq. 15). An *outer loop* updates the parameter vector P_k of the inner-loop optimization problem through offline parameter optimization and online supervisory adaptation based on the evolving orbital interaction geometry and safety conditions. The inner-loop dynamics model, actuator bounds, and predictive safety constraints are invariant to outer-loop decisions, so the proposed architecture remains a receding-horizon MPC in the classical sense [1, 2] with a supervisory tuning layer above it.

Importantly, the adaptive supervisory layer does not directly command thrust actions. Instead, it modifies high-level optimization parameters governing the MPC problem, including tracking weights, control penalties, prediction horizons, and safety-related objective terms. The constrained receding-horizon optimization structure itself remains unchanged throughout execution.

This separation between adaptive strategic tuning and low-level constrained control preserves the interpretability and constraint-awareness of the MPC formulation while enabling adaptive behavior in dynamically evolving orbital interactions.

The proposed framework was evaluated within the AIAA Capture the Satellite competition environment based on

the KSPDG framework and the associated SpaceGym environment [7]. The considered scenario involved a competitive multi-agent RPO interaction between a pursuing spacecraft (*Bandit*), a target vehicle (*Lady*), and an adversarial spacecraft (*Guard*), where the corresponding guidance and control strategy achieved first-place performance during the final competition phase. Experimental results demonstrated robust adaptive MPC behavior together with improved success-rate trends relative to representative fixed-parameter MPC baselines while maintaining real-time computational feasibility and safety-aware operation.

Although optimization parameters are updated adaptively during execution, the control inputs are always generated through the repeated online solution of a constrained finite-horizon MPC problem using the current system state, prediction model, and active constraint set. Consequently, the supervisory adaptation process operates at the tuning level rather than replacing the underlying MPC formulation.

The main contributions of this work are summarized as follows:

- An AI-augmented MPC framework for adversarial RPO that combines constrained receding-horizon optimization with adaptive supervisory parameter tuning while preserving the underlying MPC structure;
- A supervisory adaptation strategy that operates on physically interpretable MPC parameters rather than directly generating control commands, enabling adaptive behavior while maintaining transparency and constraint-aware operation;
- An interpretable autonomy framework that relates adaptive controller behavior to physically meaningful quantities, including safety penalties, minimum-separation objectives, and evolving orbital interaction geometry;
- Large-scale Monte Carlo validation within the official KSPDG Capture-the-Satellite environment, demonstrating improved mission performance and adaptive adversarial maneuvering behavior relative to fixed-parameter MPC baselines.

II. Related Work

MPC has been extensively studied for spacecraft RPO due to its ability to explicitly handle actuator limitations, collision-avoidance constraints, and trajectory optimization within a unified receding-horizon framework [4, 5]. Additional MPC-based proximity operations frameworks have investigated guaranteed feasibility and thruster-constrained maneuver planning for small-body and orbital rendezvous scenarios [8]. Linearized relative orbital dynamics based on CW formulations are particularly attractive for real-time guidance and optimization because they preserve computational tractability while providing sufficiently accurate local relative motion representations for many proximity operations scenarios [9]. Automated rendezvous and docking architectures have historically relied on constrained guidance and navigation formulations capable of preserving operational safety, fault tolerance, and real-time maneuvering feasibility during close-proximity operations [10].

Early applications of constrained optimization and MPC to spacecraft proximity operations demonstrated the ability

to explicitly handle actuator limitations, safety constraints, and fuel-efficiency objectives within a unified guidance framework. For example, Richards et al. [11] investigated constrained trajectory optimization for spacecraft formation flying under state and actuator limitations, while Breger and How [12] developed fuel-optimal rendezvous trajectories with passive collision-avoidance constraints for autonomous spacecraft proximity operations.

More recently, learning-based and data-driven methods have been explored for adaptive spacecraft guidance, autonomous rendezvous, and multi-agent maneuver planning in dynamically evolving orbital environments. Recent studies have investigated reinforcement-learning-enhanced MPC architectures and adaptive decision-making frameworks that combine optimization-based control with data-driven adaptation under uncertainty [13, 14]. He et al. [15] demonstrated that reinforcement learning combined with opponent-modeling strategies can improve adaptive decision-making in competitive multi-agent environments, highlighting the potential of learning-based approaches for dynamically evolving adversarial interactions. However, end-to-end learned control architectures often suffer from limited interpretability and difficulty enforcing hard safety constraints, which remains a major concern for safety-critical orbital robotics applications. This has motivated growing interest in safe reinforcement learning and safe learning control frameworks that explicitly incorporate safety constraints, robustness, and certifiable decision-making capabilities into learning-based systems [16, 17].

To address some of the limitations of purely learning-based control architectures, several recent works have investigated combinations of optimization and learning in order to improve adaptability while preserving safety, structure, and constraint-awareness within autonomous control systems [18, 19]. In particular, learning-enhanced control frameworks have been explored to improve prediction quality, online parameter adaptation, and strategic decision-making behavior while retaining optimization-based control structures [20]. Similarly, multi-agent trajectory optimization and differential game formulations have demonstrated the importance of explicitly modeling interactions among autonomous agents operating in dynamically coupled environments [21].

The framework proposed in this work differs from end-to-end learning approaches in that the adaptive layer does not directly generate control actions. Instead, adaptation is restricted to the modification of high-level optimization parameters governing the MPC problem. The constrained receding-horizon optimization process, orbital dynamics model, and safety constraints remain active throughout execution. This preserves the interpretability and constraint-aware structure of the controller while enabling adaptive strategic behavior in adversarial RPO scenarios.

III. Problem Formulation

The considered RPO scenario involves three spacecraft operating in close relative motion:

- a pursuing spacecraft, denoted as the *Bandit*;
- a target spacecraft, denoted as the *Lady*;
- an adversarial spacecraft, denoted as the *Guard*.

The objective of the *Bandit* is to approach the *Lady* while avoiding unsafe proximity with the *Guard*. The scenario therefore combines target tracking, constrained maneuvering, and adversarial interaction within a single constrained autonomous guidance problem. In the official competition environment, the *Lady* and *Guard* spacecraft behaviors were governed by the challenge simulation framework and could evolve dynamically throughout the interaction. The resulting interaction scenario required the *Bandit* spacecraft to approach the target while reacting to dynamically evolving adversarial behavior from the *Guard* vehicle together with periodic evasive maneuvering behavior from the *Lady* spacecraft.

The relative dynamics used by the controller are formulated using CW dynamics about a circular reference orbit [22]. The reference orbit is not intended to imply that all vehicles remain fixed on identical circular trajectories; rather, it provides the local orbital frame and mean motion used to construct a computationally efficient linearized prediction model. The spacecraft states are represented in a Local Vertical Local Horizontal (LVLH) frame associated with this reference orbit, while the MPC tracking problem is expressed using relative coordinates with respect to the *Lady* spacecraft.

At each control update, the *Bandit* relative state used by the controller is defined as in Eq. 1.

$$x(k) = [r_x \quad r_y \quad r_z \quad v_x \quad v_y \quad v_z]^T. \quad (1)$$

r_x , r_y , and r_z denote the *Bandit* position relative to the *Lady*, and v_x , v_y , and v_z denote the corresponding relative velocity components. In this local tracking frame, the *Lady* is treated as the instantaneous tracking reference, while the *Guard* state is represented relative to the same frame for predictive safety assessment.

The control input is defined as in Eq. 2.

$$u(k) = [u_x \quad u_y \quad u_z]^T. \quad (2)$$

u_x , u_y , and u_z correspond to commanded translational accelerations along the local frame axes.

The autonomous guidance objective is formulated as a constrained multi-objective control problem. The controller seeks to:

- minimize the relative distance between the *Bandit* and the *Lady*;
- maintain safe separation between the *Bandit* and the *Guard*;
- limit excessive control effort;
- satisfy actuator and safety constraints.

Safety constraints include thrust magnitude limits and predictive KOZ constraints around the *Guard* spacecraft. The KOZ constraints are evaluated over the prediction horizon (H_p) whenever the predicted *Guard* proximity falls within a predefined safety threshold. Slack variables are introduced to soften selected safety constraints and preserve numerical

feasibility in highly adversarial configurations [23]. These slack variables are penalized in the MPC cost function so that constraint relaxation is discouraged unless required to avoid infeasibility.

The proposed framework operates within a receding-horizon control architecture. At each timestep, updated relative state information is used to solve a constrained optimization problem, and only the first control action is applied. The supervisory layer adjusts selected optimization-level MPC parameters according to the active interaction geometry and mission state, but it does not directly generate thrust commands. The underlying constrained MPC structure, orbital dynamics model, actuator limits, and safety constraints remain active throughout execution.

IV. Methods

The overall control architecture introduced in Fig. 1 combines predictive orbital dynamics, constrained receding-horizon optimization, safety-aware constraint handling, and adaptive supervisory parameter tuning within a unified autonomous guidance framework.

A. Relative Orbital Dynamics

Relative motion prediction within the MPC framework is based on the CW, linearized orbital dynamics model [5, 22]. The CW formulation provides a computationally efficient linear time-invariant approximation of relative spacecraft motion about a circular reference orbit and is widely used in RPO due to its compatibility with real-time guidance and optimization frameworks.

The linearized translational relative dynamics in the LVLH frame are governed by the CW equations reported in Eq. 3. In the LVLH convention adopted in this work, the x axis denotes the radial direction, the y axis denotes the along-track direction, and the z axis denotes the orbit-normal direction.

$$\begin{aligned}\ddot{x} - 2n\dot{y} - 3n^2x &= u_x, \\ \ddot{y} + 2n\dot{x} &= u_y, \\ \ddot{z} + n^2z &= u_z.\end{aligned}\tag{3}$$

u_x , u_y , and u_z are the commanded translational accelerations, and $n = \sqrt{\mu/r_0^3}$ is the mean motion associated with a circular reference orbit of radius r_0 , with μ the gravitational parameter of the primary body.

The continuous-time dynamics are rewritten in state-space form as shown in Eq. 4.

$$\dot{\mathbf{x}}(t) = A_c \mathbf{x}(t) + B_c \mathbf{u}(t).\tag{4}$$

The continuous-time state and control vectors associated with the CW prediction model are defined in Eq. 5.

$$\mathbf{x}(t) = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T, \quad \mathbf{u}(t) = [u_x \ u_y \ u_z]^T. \quad (5)$$

For real-time MPC implementation, the dynamics are discretized using a fixed controller update timestep Δt . A zero-order hold on $\mathbf{u}$ is assumed between samples, and the discrete-time prediction model is obtained via the standard matrix-exponential identity in Eq. 6.

$$\begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix} \Delta t\right). \quad (6)$$

This yields the discrete-time prediction model in Eq. 7.

$$\mathbf{x}_{k+1} = A_d \mathbf{x}_k + B_d \mathbf{u}_k. \quad (7)$$

$A_d \in \mathbb{R}^{6 \times 6}$ and $B_d \in \mathbb{R}^{6 \times 3}$ are the discrete-time state transition and control matrices used by the predictive controller.

The CW model was selected because it preserves the convex structure required for efficient Quadratic Programming (QP) while remaining sufficiently accurate for the local relative motion conditions encountered during the considered RPO scenario. Although the operational environment in the competition framework may include nonlinear and dynamically evolving interactions among spacecraft, the linearized prediction model provided adequate real-time predictive capability while maintaining moderate computational cost.

B. Model Predictive Control Formulation

At each control update, the proposed controller solves a constrained finite-horizon optimization problem using the discrete-time prediction model defined in Eq. 7. The MPC framework operates in a receding-horizon fashion: after solving the optimization problem, only the first control action is applied before the optimization is repeated using updated state information.

The resulting constrained finite-horizon optimization problem is reported in Eq. 8, subject to the system dynamics, actuator limitations, and predictive safety constraints. The decision variables consist of the control sequence over the control horizon (H_u) and, when predictive KOZ constraints are active, the associated safety slack variables.

$$\min_{\mathbf{u}, \mathbf{s}} J \quad \text{s.t.} \quad \begin{cases} \mathbf{x}(k+i+1) = A_d \mathbf{x}(k+i) + B_d \mathbf{u}(k+i), \\ \|\mathbf{u}(k+i)\| \leq u_{\max}, \\ \|\mathbf{u}(k+i) - \mathbf{u}(k+i-1)\| \leq \Delta u_{\max}, \\ g_{\text{KOZ}}(\mathbf{x}(k+i)) + s_i \geq 0, \\ s_i \geq 0. \end{cases} \quad (8)$$

The MPC objective function is formulated as the finite-horizon quadratic cost reported in Eq. 9. The first term penalizes the predicted relative tracking error between the *Bandit* and *Lady* spacecraft over the prediction horizon. The matrices Q penalize the relative tracking errors, while R penalizes control effort over the control horizon. The slack variables s_i soften selected predictive KOZ constraints over the active safety horizon K_{KOZ} and are penalized by λ_s to discourage constraint relaxation unless required for feasibility.

$$J = \sum_{i=1}^{H_p-1} \|\mathbf{x}_B(k+i) - \mathbf{x}_L(k+i)\|_Q^2 + \sigma_{H_p} \|\mathbf{x}_B(k+H_p) - \mathbf{x}_L(k+H_p)\|_Q^2 \\ + \sum_{i=0}^{H_u-1} \|\mathbf{u}(k+i)\|_R^2 + \sum_{i=1}^{K_{\text{KOZ}}} \lambda_s s_i^2. \quad (9)$$

The receding-horizon prediction structure and the corresponding predictive safety constraint evaluation are illustrated in Fig. 2. The figure highlights how the optimizer evaluates future *Bandit* and *Guard* trajectories over the prediction horizon while enforcing safety-aware trajectory adjustments when KOZ interaction risk is present. At each control update, the optimizer propagates the predicted *Bandit* trajectory over the prediction horizon H_p using the discrete-time CW dynamics model. Predicted *Guard* positions are simultaneously evaluated to identify potential KOZ violations. When predicted safety violations are detected within the horizon, safety constraints and associated slack variables are activated in the optimization problem, allowing the controller to trade rendezvous performance against safety preservation while maintaining feasibility.

1. Stacked prediction

H_p denotes the prediction horizon and $H_u \leq H_p$ the control horizon. The stacked predicted state and decision input sequences are defined in Eq. 10.

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_{k+1} \\ \vdots \\ \mathbf{x}_{k+H_p} \end{bmatrix} \in \mathbb{R}^{6H_p}, \quad \mathbf{U}_k = \begin{bmatrix} \mathbf{u}_k \\ \vdots \\ \mathbf{u}_{k+H_u-1} \end{bmatrix} \in \mathbb{R}^{3H_u}. \quad (10)$$

MPC prediction horizon and safety constraints

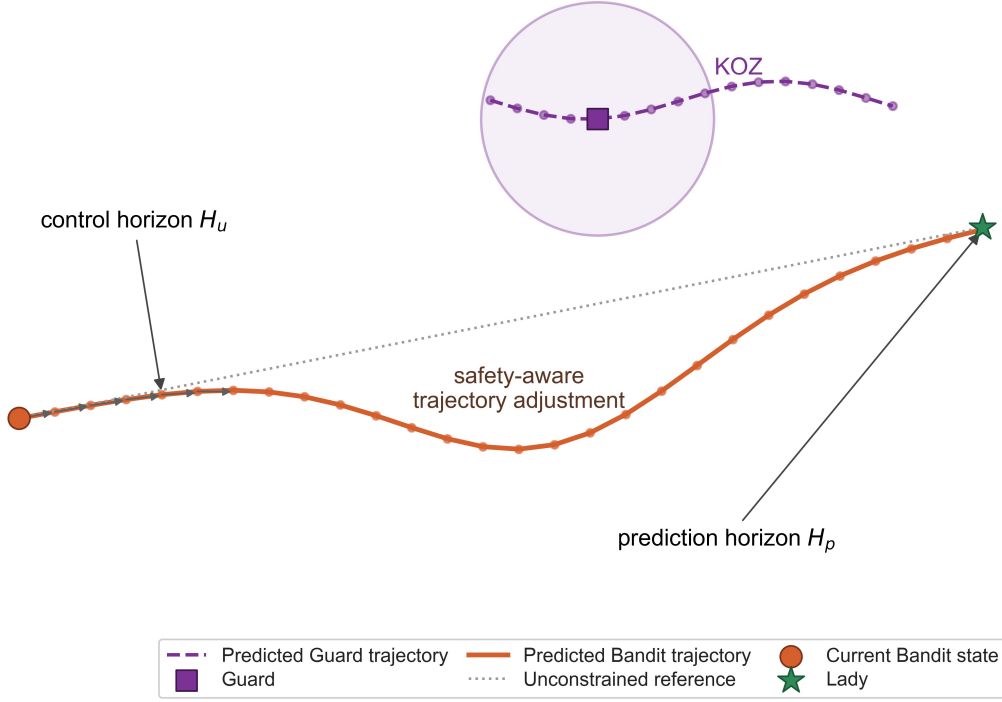


Fig. 2 Receding-horizon MPC prediction process and predictive safety-constraint evaluation.

Repeated application of Eq. 7 yields the stacked prediction equation shown in Eq. 11, where $\Phi \in \mathbb{R}^{6H_p \times 6}$ and $\Gamma \in \mathbb{R}^{6H_p \times 3H_p}$ are the standard block-Toeplitz prediction matrices defined in Eq. 12, and $T \in \mathbb{R}^{3H_p \times 3H_u}$ is a block lower-triangular lifting matrix mapping the H_u decision controls to the H_p applied controls, holding the input at the last decision value for $i \geq H_u$.

$$\mathbf{X}_k = \Phi \mathbf{x}_k + \Gamma T \mathbf{U}_k. \quad (11)$$

$$\Phi = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^{H_p} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} B_d & 0 & \cdots & 0 \\ A_d B_d & B_d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_d^{H_p-1} B_d & A_d^{H_p-2} B_d & \cdots & B_d \end{bmatrix}. \quad (12)$$

2. Cost-function notation

Using $\|v\|_M^2 \triangleq v^\top M v$, the weighting matrices in Eq. 9 are block-diagonal, as in Eq. 13, and a terminal-cost multiplier $\sigma_{H_p} \geq 1$ is applied to the last prediction block to encourage longer-horizon strategic maneuver planning. The terminal multiplier takes the value $\text{term}_{\text{final}}$ when the *Bandit* is inside the final-approach distance and $\text{term}_{\text{base}}$ otherwise.

$$Q = \text{blkdiag}(w_{L,\text{pos}} I_3, w_{L,\text{vel}} I_3), \quad R = w_U I_3. \quad (13)$$

3. Decision variable and parametric QP

The decision variable is defined as the concatenation of the control sequence and the active slack vector as shown in Eq. 14.

$$\mathbf{z} = \begin{bmatrix} \mathbf{U}_k \\ \mathbf{s} \end{bmatrix} \in \mathbb{R}^{3H_u + K_{\text{KOZ}}}. \quad (14)$$

Substituting the stacked prediction model of Eq. 11 into the MPC cost function of Eq. 9 yields the convex parametric quadratic program given in Eq. 15.

$$\mathbf{z}^*(\mathbf{x}_k, P_k) = \arg \min_{\mathbf{z}} \frac{1}{2} \mathbf{z}^\top H(P_k) \mathbf{z} + f(\mathbf{x}_k, P_k)^\top \mathbf{z} \quad \text{s.t.} \quad A(\mathbf{x}_k, P_k) \mathbf{z} \leq b(\mathbf{x}_k, P_k), \quad \mathbf{z}_{lb} \leq \mathbf{z} \leq \mathbf{z}_{ub}. \quad (15)$$

$P_k \in \mathcal{P}$ is the supervisory parameter vector and enters only the QP data (H, f, A, b) , never replacing the optimization itself. With the notation introduced above and $\bar{Q} = I_{H_p} \otimes Q$ (with the last block scaled by σ_{H_p}) and $\bar{R} = w_U I_{3H_u}$, the QP data take the closed forms reported in Eq. 16.

$$H = \begin{bmatrix} T^\top \Gamma^\top \bar{Q} \Gamma T + \bar{R} & 0 \\ 0 & \lambda_s I_{K_{\text{KOZ}}} \end{bmatrix}, \quad f = \begin{bmatrix} -T^\top \Gamma^\top \bar{Q} (\mathbf{X}_L^{\text{ref}} - \Phi \mathbf{x}_k) \\ \mathbf{0} \end{bmatrix}. \quad (16)$$

The linear inequality constraints $A\mathbf{z} \leq b$ encode the predictive safety constraints introduced in Section IV.C. Translational actuator limits are enforced through the per-axis box constraints given in Eq. 17, matching the per-axis thrust capability of the *Bandit*.

$$-u_{\max,j} \leq \mathbf{z}_j \leq +u_{\max,j}, \quad j = 1, \dots, 3H_u. \quad (17)$$

Slack variables are additionally constrained according to $0 \leq s_i \leq s_{\max}$.

4. Receding-horizon control law

The applied control input is obtained from the first control block of the optimizer solution, as in Eq. 18.

$$\mathbf{u}_k = \mathbf{z}_{1:3}^*(\mathbf{x}_k, P_k). \quad (18)$$

The optimization problem of Eq. 15 is subsequently re-solved at time step $k+1$ using the updated spacecraft state and supervisory parameter vector. This corresponds to the standard receding-horizon MPC construction [1, 2] with online parameter dependence on P_k rather than fixed controller tuning. When the optimization problem of Eq. 15 returns an infeasible status, a distance-scheduled PD control law saturated according to the actuator bounds of Eq. 17 is applied as a numerical fallback strategy. In the reported experiments, this fallback mechanism was not activated during nominal mission phases.

The resulting constrained optimization problem is formulated primarily as a quadratic program and solved repeatedly online throughout closed-loop operation. The use of linearized CW dynamics together with quadratic cost terms and linearized predictive safety constraints preserves computational tractability and enables real-time execution within the competition environment.

At each control update, the controller solves a constrained finite-horizon optimization problem using the current spacecraft state, the active prediction model, and the current constraint set. Although selected optimization parameters may be modified online by the adaptive supervisory layer, the underlying MPC problem formulation, orbital dynamics model, actuator constraints, and predictive safety constraints remain embedded within the optimization problem throughout execution. Consequently, the adaptive mechanism modifies the tuning of the predictive controller rather than replacing the underlying constrained MPC structure itself.

C. Safety Constraints and Feasibility Handling

Safety preservation was treated as a primary design requirement of the proposed architecture. Rather than allowing the adaptive supervisory layer to directly generate control commands, all maneuvering actions remained constrained by the MPC optimization problem, actuator limitations, and predictive safety constraints throughout execution.

Predictive KOZ constraints were introduced to reduce unsafe proximity interactions between the *Bandit* and *Guard* spacecraft, as conceptually illustrated in Fig. 2. Both the KOZ and the surrounding safety buffer were modeled as spherical proximity regions centered on the predicted *Guard* position. The inner KOZ defined the minimum allowable separation distance, while the larger surrounding safety buffer defined the region within which predictive safety analysis became active.

The *Guard* relative trajectory was propagated over the MPC prediction horizon using the same CW-based prediction structure employed for the controlled *Bandit* dynamics. At each prediction step, the optimizer evaluated the predicted

relative separation between the two vehicles in order to determine whether future safety violations could occur over the receding horizon.

Importantly, predictive KOZ constraints were not enforced uniformly during all phases of the maneuver. Instead, the controller activated predictive KOZ analysis only when the predicted *Guard* trajectory entered a larger predefined safety buffer region surrounding the active keep-out zone. Consequently, the safety buffer acted as an early-warning region within which the controller began evaluating future KOZ penetration risk before direct violation of the minimum-separation boundary could occur.

When the predicted *Guard* trajectory remained outside the safety buffer region, the controller operated using nominal tracking-oriented MPC behavior without activating additional predictive KOZ constraint handling. As the *Guard* entered the safety buffer region, predictive safety constraints and associated slack-variable handling became active over the dynamically relevant subset of the prediction horizon.

The use of a predictive safety buffer reduced unnecessary conservatism during nominal rendezvous phases while enabling earlier evasive maneuver planning during rapidly evolving adversarial interactions. Consequently, the controller could maintain more aggressive target-approach behavior when the *Guard* spacecraft remained sufficiently distant, while progressively prioritizing safety-aware maneuvering as predicted *Guard* proximity increased.

Once the safety buffer becomes active, predictive KOZ constraints are enforced over the subset $i = 1, \dots, K_{\text{KOZ}}$ of the prediction horizon associated with potentially unsafe future interaction intervals. $\mathbf{r}_{BG}^{\text{nom}}(k+i)$ denotes the predicted *Bandit-Guard* relative position at prediction step i under the current nominal open-loop trajectory. The corresponding line-of-sight unit vector $\hat{\mathbf{n}}_i$ together with the nominal safety-margin quantity h_0^i are defined in Eq. 19.

$$\hat{\mathbf{n}}_i = \frac{\mathbf{r}_{BG}^{\text{nom}}(k+i)}{\|\mathbf{r}_{BG}^{\text{nom}}(k+i)\|}, \quad h_0^i = \|\mathbf{r}_{BG}^{\text{nom}}(k+i)\| - d_{\text{KOZ}}. \quad (19)$$

A first-order expansion along $\hat{\mathbf{n}}_i$ yields the linearized predictive safety condition reported in Eq. 20. In this formulation, $M_i \in \mathbb{R}^{3 \times 3H_u}$ maps the decision controls to the predicted *Bandit* position deviation at prediction step i , $\gamma_K > 0$ is a decay coefficient regulating the rate at which the linearized constraint may approach the boundary, and $s_i \geq 0$ is the corresponding slack variable.

$$-\hat{\mathbf{n}}_i^T M_i \mathbf{U}_k - s_i \leq -\hat{\mathbf{n}}_i^T \mathbf{r}_{BG}^{\text{nom}}(k+i) - \gamma_K h_0^i. \quad (20)$$

The original nonlinear distance condition $d_{BG}(k+i) + s_i \geq d_{\text{KOZ}}$ is preserved logically, while the linearized form of Eq. 20 preserves the convex structure of the underlying quadratic program required for real-time online solution. The slack variables are introduced only over the K_{KOZ} prediction intervals with active predictive safety enforcement, thereby reducing optimization dimensionality while localizing constraint softening to dynamically relevant safety-critical interaction intervals.

The slack variables soften selected predictive safety constraints in situations where strict feasibility cannot be maintained due to highly adversarial geometric configurations, limited maneuver authority, or conflicting tracking and safety objectives. These slack variables are directly penalized through the MPC cost function, ensuring that constraint relaxation is minimized whenever feasible solutions exist.

Importantly, the slack variables do not remove the safety constraints from the optimization problem. Instead, they provide controlled and explicitly penalized feasibility relaxation in order to avoid optimizer infeasibility, unstable control behavior, or excessively aggressive evasive maneuvers during adversarial multi-agent interactions. This formulation allows the controller to preserve continuous closed-loop operation even when perfect simultaneous satisfaction of all tracking and safety objectives cannot be achieved.

Consequently, the adaptive supervisory layer was restricted to modifying high-level optimization parameters while the constrained MPC structure itself remained unchanged. The adaptive layer therefore could not directly bypass the orbital dynamics model, actuator limitations, or predictive safety constraints embedded within the optimization problem.

D. Control Barrier Function safety filter

As an optional inner-most safety mechanism, the MPC-computed control $\mathbf{u}_{\text{cmd}}$ can be passed through the discrete-time CBF filter defined in Eq. 23 before being applied to the actuator [24]. The safety barrier function $h(\mathbf{x})$, positive whenever the *Bandit* spacecraft remains outside the KOZ, is defined as $h(\mathbf{x}) = d_{BG} - d_{\text{KOZ}}$. Its time derivative along the relative dynamics is given in Eq. 21.

$$\dot{h} = \hat{\mathbf{n}}^T(\mathbf{v}_B - \mathbf{v}_G), \quad \hat{\mathbf{n}} = (\mathbf{r}_B - \mathbf{r}_G)/\|\mathbf{r}_B - \mathbf{r}_G\|. \quad (21)$$

A discrete-time higher-order CBF condition with gains $k_0, k_1 > 0$ is imposed according to Eq. 22, where $\mathbf{a}_G^{\text{now}}$ denotes the current PD-pursuit estimate of the adversary acceleration.

$$\hat{\mathbf{n}}^T \mathbf{u}_{\text{safe}} \geq \beta(\mathbf{x}_k), \quad \beta(\mathbf{x}_k) \triangleq -\hat{\mathbf{n}}^T \mathbf{a}_G^{\text{now}} - k_1 \dot{h} - k_0 h. \quad (22)$$

The filtered control input is obtained from the projection quadratic program reported in Eq. 23.

$$\mathbf{u}_{\text{safe}} = \arg \min_{\mathbf{u} \in \mathbb{R}^3} \|\mathbf{u} - \mathbf{u}_{\text{cmd}}\|^2 \text{ s.t. } \hat{\mathbf{n}}^T \mathbf{u} \geq \beta(\mathbf{x}_k), \quad -u_{\text{max}} \leq \mathbf{u} \leq +u_{\text{max}}. \quad (23)$$

When the condition of Eq. 22 is already satisfied by $\mathbf{u}_{\text{cmd}}$, the filter acts as the identity map. The primary safety guarantees in the implemented framework are provided by the constrained MPC formulation of Eq. 15 together with predictive safety evaluation and feasibility-preserving slack handling; the CBF filter is used as an additional defense layer.

Formal recursive feasibility and closed-loop stability guarantees under arbitrary adaptive parameter updates remain outside the scope of the present work. However, two design choices already provide partial guarantees. First, the parameter set $\mathcal{P}$ over which the supervisory layer adapts is bounded by construction (Section IV.E), so the inner-loop QP data of Eq. 15 depend continuously on a compact set of parameters at every step. Second, the soft-constraint formulation with bounded non-negative slacks introduced in Eq. 14 helps preserve optimization feasibility under strongly adversarial interaction conditions [23]. Beyond these structural guarantees, robustness is validated empirically through the Monte Carlo campaign of Section V.C. The proposed framework therefore follows an engineering-oriented robustness validation approach rather than a theorem-driven stability analysis.

E. Adaptive Parameter Layer

The proposed framework incorporates an outer-loop adaptive layer designed to modify selected MPC optimization parameters online during closed-loop operation. Unlike end-to-end learned control architectures, the adaptive layer does not directly generate thrust commands or bypass the constrained optimization process. Instead, adaptation is restricted to high-level parameters governing the MPC objective and predictive behavior.

A representative subset of MPC parameters considered in the supervisory tuning framework is summarized in Table 1. The implementation includes additional bounded parameters associated with KOZ horizon selection, control effort weighting, final-approach switching, guard-response shaping, and safety-constraint decay. All adapted parameters live in a bounded set $\mathcal{P}$ specified a priori, which ensures that the inner-loop QP remains well-posed under every supervisory decision.

Table 1 Representative subset of MPC parameters considered within the supervisory tuning framework.

Parameter Category	Adaptive Parameters	Functional Role
Horizon handling	H_p, H_u	Adjust the prediction and control horizons used by the receding-horizon optimizer.
Tracking objective	$w_{L,\text{pos}}, w_{L,\text{vel}}$	Set the relative position and velocity tracking priorities with respect to the <i>Lady</i> spacecraft.
Safety and KOZ handling	$w_{\text{KOZ}}, d_{\min}$	Tune the penalty and minimum-separation behavior associated with <i>Guard</i> avoidance.
Terminal weighting	$\text{term}_{\text{base}}, \text{term}_{\text{final}}$	Adjust terminal weighting during nominal and final approach phases.
Barrier filtering	k_0, k_1	Shape the optional CBF-style distance and closing-rate safety response.

The resulting parameter updates influence the strategic behavior of the controller while preserving the underlying constrained MPC structure described in Sections IV.B and IV.C.

1. Offline stage

A base parameter vector $P^* \in \mathcal{P}$ is obtained by minimizing the expected closed-loop episode return according to Eq. 24.

$$P^* \in \arg \min_{P \in \mathcal{P}} \mathbb{E}_{\xi} [J(\pi_{\text{MPC}}(\cdot; P), \xi)] . \quad (24)$$

In Eq. 24, ξ denotes an initial-condition seed drawn from the KSPDG environment distribution, $\pi_{\text{MPC}}(\cdot; P)$ is the receding-horizon control law of Eq. 18 instantiated with parameter P , and J is the mission cost returned by the environment (lower is better). The expectation of Eq. 24 is non-convex in P and not analytically differentiable; it is therefore approximated by sample average over n_{ep} episodes and optimized using derivative-free search methods. Two optimization approaches were implemented: the Cross-Entropy Method (CEM) [25, 26] and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [27]. CEM is a population-based stochastic optimization method that iteratively samples candidate parameter vectors from a probability distribution, evaluates their closed-loop performance, selects an elite subset of the best-performing candidates, and updates the sampling distribution toward the statistics of this elite set. In this work, CEM was used to search over the bounded parameter set $\mathcal{P}$ and identify a baseline MPC parameter vector with favorable empirical closed-loop performance; interested readers are referred to the original CEM literature for further details [25]. For the results reported in this work, the offline parameter vector P^* was obtained using CEM, while CMA-ES was retained as an implemented alternative optimization strategy.

2. Online stage

During deployment, the controller operates using the offline-optimized parameter set P^* together with a set of geometry-driven supervisory adaptation mechanisms. These mechanisms modify selected MPC optimization parameters according to the evolving interaction geometry while preserving the underlying constrained MPC formulation, orbital dynamics model, actuator constraints, and predictive safety constraints.

The implemented online adaptation logic consists of three geometry-dependent supervisory mechanisms:

- **Proximity-based minimum-separation inflation.** The effective minimum-separation objective $d_{\min}$ is inflated only when the current predicted *Bandit–Guard* separation d_{BG} falls below $1.25 d_{\min}$. A normalized proximity factor is first computed according to Eq. 25.

$$\text{prox} = \text{clip} \left(\frac{1.25 d_{\min} - d_{BG}}{1.25 d_{\min}}, 0, 1 \right) . \quad (25)$$

The effective minimum-separation objective is then updated according to Eq. 26.

$$d_{\min, \text{eff}} = d_{\min} (1 + 0.08 \text{prox}) . \quad (26)$$

This mechanism increases the desired safety margin only when the *Guard* spacecraft is meaningfully close to the *Bandit*.

- **Phase-dependent tracking-weight scaling.** The relative-position and relative-velocity tracking weights are scaled according to the current *Bandit–Lady* separation d_{BL} . Specifically, multipliers of (4, 25), (8, 50), and (12, 80) are applied to $(w_{L,pos}, w_{L,vel})$ when d_{BL} falls below 500 m, 200 m, and 80 m, respectively. An additional final-approach condition applies intermediate multipliers of (2.5, 12). The resulting effective weights enter the block-diagonal MPC weighting matrix Q used in the quadratic cost.
- **Conditional KOZ-weight modulation.** The KOZ safety weight w_{KOZ} is adjusted during QP construction and relaxation according to the active interaction geometry and solver feasibility. If the capture lane is open, the KOZ horizon is disabled and the effective KOZ weight is set to zero. If the nominal QP is infeasible, a first relaxation level reduces the effective KOZ weight to $0.5 w_{KOZ}$; if infeasibility persists, a second relaxation level reduces it to $0.3 w_{KOZ}$. As a last-resort guidance mode, the KOZ terms are disabled by setting the effective KOZ weight to zero. This rule reduces excessive conservatism only when required to recover QP feasibility while preserving the nominal safety-aware formulation whenever feasible.

These supervisory mechanisms operate directly on interpretable MPC optimization parameters and are evaluated throughout closed-loop execution. Consequently, adaptive behavior emerges through geometry-dependent parameter modulation rather than through direct modification of the underlying control law.

F. Real-Time Control Architecture

In deployment, the controller operates according to the closed-loop architecture illustrated in Fig. 1.

- 1) Acquisition of updated relative spacecraft state information;
- 2) Propagation of the MPC prediction model over the receding horizon;
- 3) Evaluation of predictive safety constraints and keep-out zone conditions;
- 4) Online solution of the constrained parametric QP in Eq. 15;
- 5) Optional CBF projection of the first optimized control action via Eq. 23;
- 6) Application of the projected control to the *Bandit* spacecraft;
- 7) Geometry-driven supervisory update of the active MPC parameters according to the mechanisms described in Section IV.E.

This closed-loop process was repeated continuously throughout the maneuvering scenario. Prior to deployment, the baseline parameter vector P^* was obtained through offline optimization across multiple simulation episodes generated under varying initial orbital configurations and adversarial encounter geometries, as described in Section IV.E. This process exposed the optimization procedure to a diverse set of interaction conditions and mission outcomes, allowing the selection of a parameter set with robust closed-loop performance across the considered scenario distribution.

During execution, the controller operates using the offline-optimized parameter set together with the geometry-driven supervisory adaptation mechanisms described previously. The online supervisory layer modifies selected MPC tuning parameters according to the current interaction geometry while preserving the underlying constrained MPC formulation.

The computational complexity of the optimization problem remained compatible with real-time execution due to the use of linearized CW dynamics, quadratic objective functions, and moderate prediction horizon sizes. The constrained MPC formulation, together with the safety handling mechanisms described in Section IV.C, remained active throughout all controller updates independently of the adaptive parameter modifications. The implemented architecture preserves a layered autonomy structure in which geometry-driven supervisory adaptation, constrained optimization, orbital dynamics propagation, and safety evaluation remain functionally separated. This separation improves interpretability while maintaining the constraint-aware behavior required for safety-critical orbital operations.

V. Experimental Setup

A. KSPDG Capture-the-Satellite Scenario

KSPDG is the official environment for the 2026 AIAA Capture the Satellite challenge [7]. The considered scenario consists of an adversarial *Lady-Bandit-Guard* RPO configuration involving three spacecraft interacting simultaneously within a closed-loop orbital environment. The controlled vehicle, denoted as the *Bandit*, represented the autonomous pursuing spacecraft implementing the proposed adaptive MPC architecture. The *Lady* spacecraft acted as the rendezvous target, while the *Guard* spacecraft represented an adversarial vehicle attempting to intercept or obstruct the *Bandit* during proximity operations with undisclosed control policy [7].

Each evaluation trial consisted of a complete KSPDG simulation run in which the *Bandit* spacecraft attempted to safely approach the *Lady* spacecraft while avoiding unsafe interactions with the *Guard* vehicle using the proposed adaptive MPC framework described in Section IV.

The orbital guidance problem was formulated entirely in relative coordinates within the LVLH frame associated with the target spacecraft. At each control update, the controller received updated relative state information from the simulation environment and solved the corresponding constrained finite-horizon MPC problem online.

B. Controller Configurations

Three controller configurations were evaluated in order to compare the effect of adaptive supervisory parameter tuning on closed-loop rendezvous performance. Table 2 summarizes the primary characteristics of the evaluated controller configurations used throughout the experimental campaign.

The first configuration consisted of a naive MPC controller operating with nominal fixed optimization parameters and without adaptive parameter updates. This configuration relied exclusively on the baseline MPC formulation throughout execution and provided a reference case for evaluating the effect of predictive safety handling and adaptive supervisory

Table 2 Summary of evaluated controller configurations.

Configuration	Adaptive Tuning	Safety Handling	Control Characteristics
Naive MPC	No	Minimal baseline safety handling.	Basic constrained MPC with fixed nominal parameters.
Fixed Agent	No	Predictive KOZ constraints + slack handling.	Manually tuned constrained MPC with fixed optimization parameters.
Adaptive Agent	Yes	Predictive KOZ constraints + slack handling.	Adaptive constrained MPC with CEM-tuned base parameters and geometry-driven online supervisory modulation.

parameter tuning.

The second configuration employed a fixed-parameter constrained MPC controller incorporating predictive KOZ constraints, slack-variable feasibility handling, and manually tuned optimization parameters. The controller therefore combined constrained predictive safety handling with fixed supervisory tuning, while all optimization parameters remained constant throughout execution.

The third configuration employed the proposed adaptive MPC framework in which selected optimization-level controller parameters were updated online according to the evolving interaction conditions and supervisory adaptation logic described in Section IV.E. The adaptive layer modified parameters including tracking weights, safety penalties, prediction horizon settings, and minimum KOZ separation objectives while preserving the underlying constrained MPC structure.

All controller configurations operated using the discrete-time CW prediction model introduced in Section IV.A together with repeated online quadratic program optimization over a receding prediction horizon.

Parameter adaptation was performed through a combination of offline derivative-free tuning (Eq. 24) and online geometry-driven supervisory parameter modulation based on the evolving orbital interaction conditions.

C. Evaluation Protocol

A total of 1000 closed-loop simulation runs were executed, within the KSPDG environment, using varying initial interaction conditions and adversarial encounter geometries. Aggregate Monte Carlo outcome statistics and representative trajectory comparisons are reported in Section VI. In addition, the controller-configuration comparison was performed over 25 runs per controller configuration, while the large-scale robustness analysis focused on the 1000-run adaptive MPC Monte Carlo campaign. During each simulation run, the *Bandit* spacecraft attempted to safely rendezvous with the *Lady* spacecraft while avoiding interception or unsafe proximity interactions with the *Guard* vehicle.

Table 3 summarizes the primary evaluation objectives together with the corresponding Monte Carlo performance metrics used throughout the experimental campaign.

Table 3 Monte Carlo evaluation objectives and representative performance metrics.

Evaluation Objective	Representative Metric
Rendezvous performance	Success rate, target-relative distance evolution
Safety preservation	Minimum <i>Guard</i> separation distance, KOZ violations
Robustness under adversarial interactions	Failure rate across varying encounter geometries
Controller adaptability	Adaptive supervisory parameter evolution
Closed-loop feasibility	Feasible completion of closed-loop simulation runs
Numerical stability	Stable closed-loop trajectory behavior

Performance evaluation included both mission-level outcome metrics and trajectory-level safety analysis. The resulting dataset provided a large-scale empirical assessment of the proposed framework under realistic orbital interaction conditions. The Monte Carlo evaluation campaign was performed using controller configurations obtained after the offline parameter-optimization phase described in Section IV.E. During Monte Carlo evaluation, the controller operated in deployment mode and no additional offline parameter training was performed across simulation runs.

Importantly, the Monte Carlo campaign was not intended to provide formal proof of recursive feasibility or closed-loop stability under arbitrary adaptive parameter updates. Instead, the evaluation was designed to provide engineering-oriented robustness validation through repeated closed-loop execution across a wide range of interaction scenarios.

D. Logged Quantities and Interpretability Analysis

During each closed-loop simulation run, the framework recorded spacecraft states, control actions, MPC optimization parameters, predictive safety constraint activity, KOZ interactions, QP relaxation events, and mission outcome statistics for subsequent robustness and interpretability analysis.

Because the adaptive supervisory layer operated on interpretable MPC tuning parameters rather than directly on low-level control actions, the resulting parameter trajectories could be directly correlated with evolving orbital geometry, adversarial interactions, and safety conditions throughout execution.

The resulting dataset enabled post-run analysis of adaptive parameter evolution, safety-performance trade-offs, constraint activation behavior, and closed-loop robustness characteristics under dynamically evolving adversarial interaction conditions.

Algorithm 1 summarizes the interpretability-oriented logging and post-processing workflow used throughout the evaluation campaign. The recorded quantities include relative spacecraft geometry, control actions, supervisory MPC parameters, predictive safety activity, and mission outcomes, enabling post-run analysis of the relationship between interaction geometry, adaptive parameter evolution, and closed-loop controller behavior.

Algorithm 1 Interpretability-Oriented Logging and Analysis Procedure.

- 1: **for** each Monte Carlo simulation run **do**
 - 2: Record relative spacecraft states $\mathbf{r}_{BG}, \mathbf{v}_{BG}, \mathbf{r}_{BL}, \mathbf{v}_{BL}$ and control actions $\mathbf{u}_k$.
 - 3: Record MPC optimization parameters $w_{KOZ}, d_{\min}, w_{L, \text{pos}}, w_{L, \text{vel}}$ and supervisory parameter updates.
 - 4: Record predictive safety-constraint activity and KOZ interactions.
 - 5: Record slack-variable activations and mission outcome statistics.
 - 6: **end for**
 - 7: Analyze adaptive supervisory parameter-evolution trends.
 - 8: Analyze controller behavior under varying interaction conditions.
 - 9: Evaluate closed-loop robustness and safety behavior.
-

VI. Results and Discussion

A. Closed-Loop Behavior and Monte Carlo Outcome Characteristics

Monte Carlo outcome statistics and representative robustness characteristics generated by the proposed adaptive MPC framework are shown in Fig. 3. Across the 1000-run evaluation campaign, the adaptive MPC controller achieved 940 successful runs, corresponding to a 94.0% mission success rate, with 44 timeout cases and 16 *Guard*-capture cases. The green dots represent success cases, the brown dots timeout cases, and the purple dots *Guard*-capture cases. Panel (A) shows a success rate of 94.0%, a *Guard*-capture rate of 1.6%, and a timeout rate of 4.4%. Panels (B) and (C) show the closest *Bandit-Lady* and *Bandit-Guard* approach distances, respectively.

The outcome distributions and closest-approach statistics shown in Fig. 3(B)–(D) indicate substantial variability in the relationship between rendezvous performance and adversarial separation behavior. In Fig. 3(B), successful rendezvous cases achieve closest-approach distances below the 5 m target threshold, whereas timeout and *Guard*-capture cases remain farther from the *Lady*. In Fig. 3(C), successful and timeout cases typically maintain closest *Guard* separations between approximately 10 and 50 m, while *Guard*-capture outcomes correspond to separations below the 10 m capture threshold. Successful runs consistently achieved close *Bandit-Lady* proximity, while exhibiting a broad range of closest *Guard* separation distances ranging from near-threshold avoidance behavior to substantially larger safety margins.

Panel (D) shows the combined distribution of the two closest-approach distances. *Guard*-capture cases are concentrated in the region characterized by small *Bandit-Guard* separation distances, whereas successful and timeout cases occupy a broader range of *Guard* separations.

As shown in Fig. 3(D), most *Guard*-capture failures were additionally associated with relatively large final *Bandit-Lady* distances, suggesting that highly adversarial interaction geometries could simultaneously degrade rendezvous progression and safety preservation. Timeout outcomes exhibited more dispersed behavior across the safety–performance trade-space illustrated in Fig. 3(D), including cases that remained relatively far from both spacecraft as well as cases that approached both the *Lady* and *Guard* vehicles more closely during the interaction. Several timeout cases additionally exhibited partial rendezvous progression while preserving relatively safe *Guard* separation distances, suggesting that

the controller occasionally favored conservative safety-preserving maneuvering behavior over aggressive terminal interception attempts under unfavorable interaction geometries.

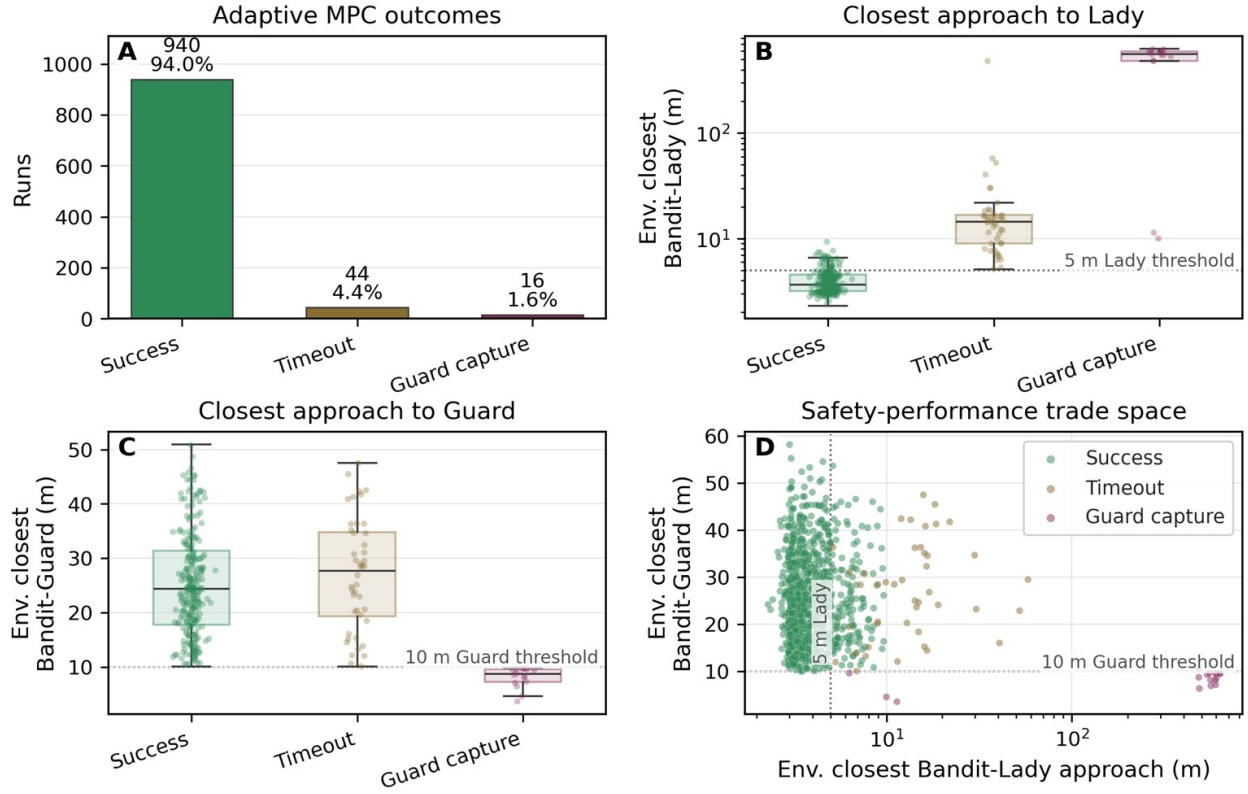


Fig. 3 Monte Carlo robustness evaluation for the 1000-run adaptive MPC campaign.

The adaptive supervisory layer modified the strategic behavior of the controller through optimization-level parameter adaptation while preserving the constrained receding-horizon MPC structure responsible for generating the control actions. Consequently, all maneuvering behavior remained consistent with the imposed actuator limits, predictive safety constraints, and orbital dynamics model.

The resulting trajectory evolution further indicates that the adaptive controller maintained relatively smooth maneuvering behavior despite rapidly evolving adversarial conditions. The imposed control penalties and predictive horizon structure reduced excessive oscillatory thrust behavior while preserving sufficient responsiveness for collision avoidance and target pursuit.

The mission-relative distance evolution observed during representative simulation runs is shown in Fig. 4. The adaptive MPC framework progressively reduced the relative distance to the *Lady* spacecraft while dynamically modifying the optimization priorities of the predictive controller according to the evolving adversarial interaction geometry.

The normalized mission-progress histories shown in Fig. 4(A)–(B) correspond to population-level distance histories grouped by mission outcome and plotted over normalized mission progress for successful, timeout, and *Guard*-capture

trajectories. Successful runs typically exhibited monotonic reduction of *Bandit-Lady* distance while simultaneously delaying aggressive *Guard* proximity until late mission phases. In contrast, timeout and *Guard*-capture cases often displayed early safety-dominant interaction behavior followed by reduced rendezvous progression efficiency.

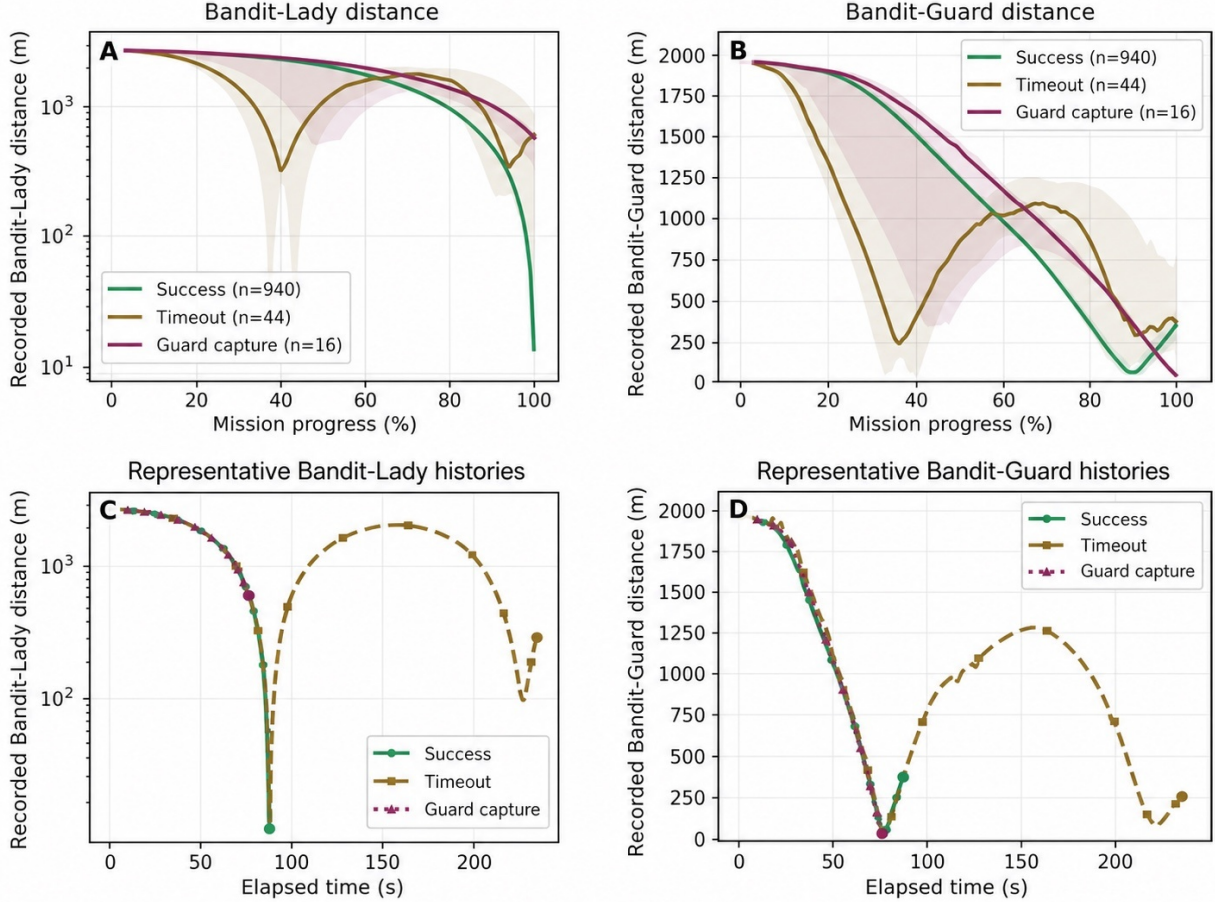


Fig. 4 Recorded *Bandit-Lady* and *Bandit-Guard* distance histories from the 1000-run adaptive MPC robustness dataset.

The representative histories shown in Panels (C) and (D) of Fig. 4 correspond to individual closed-loop simulation runs selected from the successful, timeout, and *Guard*-capture outcome classes. These representative trajectories illustrate the different safety–performance trade-offs observed during the Monte Carlo campaign, including successful rendezvous progression, safety-preserving timeout behavior, and unsuccessful adversarial interactions leading to *Guard* capture. Circular markers indicate the final recorded state of each trajectory. In several regions of the plots, overlapping trajectories partially obscure individual curves due to the large number of similar interaction histories contained within the dataset.

As illustrated by the representative single-run trajectories in Fig. 4(C)–(D), temporary increases in target-relative distance were associated with elevated *Guard* proximity and increased collision-risk conditions. Several timeout

trajectories exhibited temporary increases in target-relative distance accompanied by recovery of *Guard* separation margins, indicating that the predictive controller occasionally sacrificed short-term rendezvous efficiency in order to preserve longer-term safety feasibility. Representative successful trajectories shown in Panels (C) and (D) of Fig. 4 additionally showed that periods of decreasing *Bandit-Guard* separation were frequently followed by rapid recovery of adversarial distance margins during the terminal rendezvous phase. In contrast, several *Guard*-capture cases exhibited sustained reduction of *Bandit-Guard* separation without comparable recovery behavior, indicating unsuccessful evasive response during highly adversarial interaction phases. During these phases, the supervisory adaptation layer increased safety-oriented optimization priorities, including keep-out zone penalties and minimum-separation objectives, resulting in more conservative maneuvering behavior intended to preserve safety margins.

As the *Guard* threat level decreased and the interaction geometry became more favorable, the adaptive supervisory layer progressively reduced safety-dominant weighting behavior and restored more aggressive rendezvous-oriented optimization priorities. This interpretation is consistent with the supervisory MPC parameter trends shown in Fig. 6, which provide additional evidence of the transition from safety-oriented to rendezvous-oriented behavior.

Overall, the recorded distance histories indicate that the adaptive MPC framework was capable of dynamically balancing rendezvous progression and adversarial avoidance behavior under rapidly evolving orbital interaction conditions. The resulting closed-loop trajectories exhibited coordinated transitions between target-pursuit and safety-preserving maneuvering regimes while maintaining stable and physically consistent behavior throughout the evaluated interaction scenarios.

B. Controller Configuration Comparison

Representative controller-comparison experiments were performed in order to evaluate the effect of supervisory adaptation on closed-loop rendezvous behavior under adversarial interaction conditions. Comparative results obtained over 25 simulation runs per controller configuration are shown in Fig. 5. The figure illustrates qualitative differences in closed-loop trajectory behavior and observed mission outcomes under adversarial rendezvous conditions.

Figure 5 compares the three controller configurations over 25 simulation runs per configuration. The adaptive MPC controller achieved the highest success rate (72%), followed by the fixed-parameter MPC controller (60%) and the naive MPC controller (28%). Correspondingly, the adaptive controller produced fewer unsuccessful mission outcomes and demonstrated more consistent rendezvous performance across the evaluated adversarial encounter conditions. These results illustrate the benefit of supervisory parameter adaptation under adversarial rendezvous conditions.

The adaptive supervisory layer enabled the controller to dynamically modify optimization priorities according to the evolving interaction geometry, producing visibly different maneuvering behavior relative to the fixed-parameter configurations during adversarial encounters. In contrast, fixed-parameter MPC configurations exhibited reduced adaptability during rapidly evolving adversarial encounters due to static weighting structures and fixed predictive

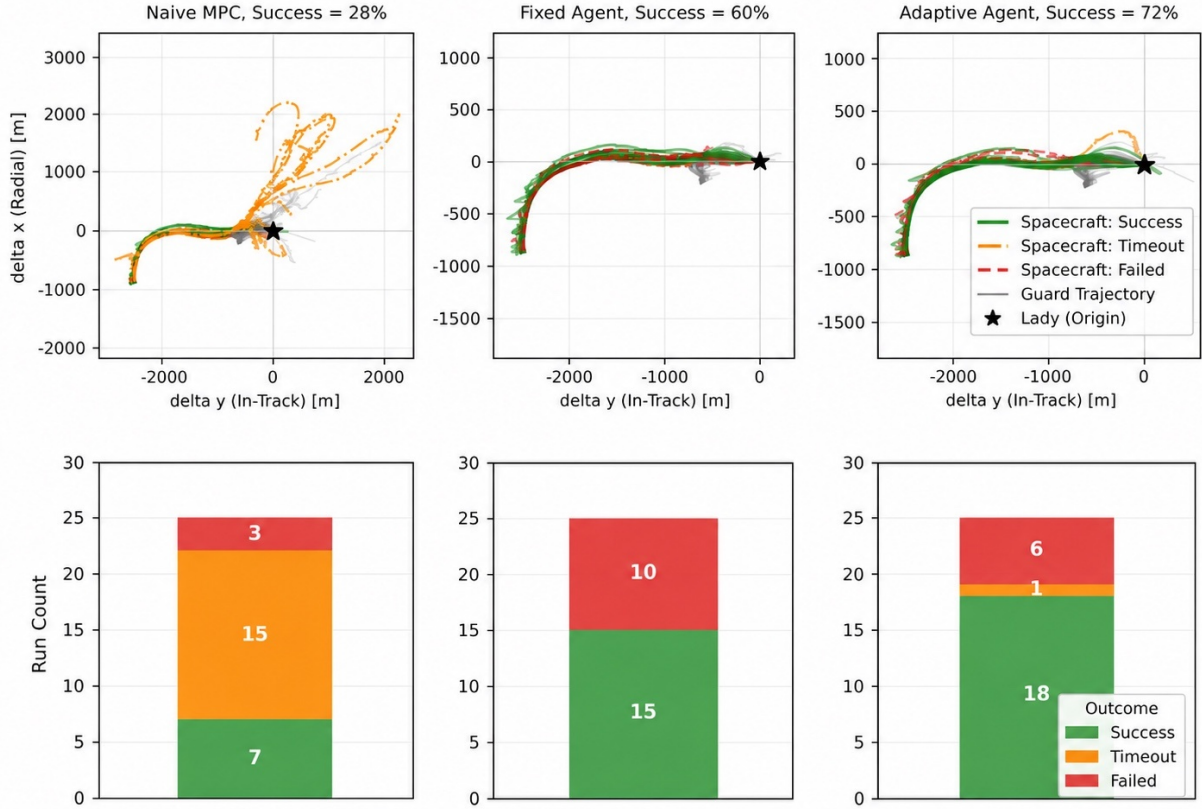


Fig. 5 Representative controller-configuration comparison across naive MPC, fixed-parameter MPC, and adaptive MPC cases.

behavior.

Across the evaluated comparison runs, the adaptive controller consistently maintained feasible closed-loop operation throughout dynamically evolving engagements. The imposed actuator limitations, predictive safety constraints, and feasibility-preserving constraint-relaxation handling remained active independently of the adaptive supervisory updates.

Differences between controller configurations were particularly visible during interaction phases involving aggressive *Guard* interception trajectories. Under these conditions, the adaptive controller was able to adjust selected optimization parameters according to the evolving interaction geometry, enabling adaptive maneuvering behavior under rapidly changing adversarial conditions.

Importantly, the evaluation campaign was not intended to provide formal proof of recursive feasibility or closed-loop stability under arbitrary adaptive parameter adaptation. Instead, the reported results provide representative empirical evaluation through repeated closed-loop execution under dynamically evolving adversarial conditions.

No catastrophic controller divergence or persistent unstable oscillatory behavior was observed during the reported

Monte Carlo campaigns. The resulting robustness characteristics therefore emerged from persistent constrained optimization, feasibility-preserving safety handling, and bounded supervisory adaptation rather than from theorem-driven stability guarantees.

C. Adaptive Parameter Evolution and Interpretability

Representative adaptive supervisory parameter evolution trends are shown in Fig. 6. Panel (A) reports the population-level evolution of the effective KOZ safety weight w_{KOZ} as a function of normalized mission progress for successful, timeout, and *Guard*-capture outcomes. Panel (B) shows the corresponding evolution of the effective minimum-separation objective d_{min} . Shaded regions indicate the 10th–90th percentile band computed across trajectories within each outcome class, while the solid curves represent the corresponding population-level trends. The figure illustrates how supervisory safety priorities evolve differently across successful and unsuccessful interaction scenarios. The adaptive MPC parameters summarized in Section IV.E correspond to physically meaningful controller quantities, including the prediction horizon H_p , the control horizon H_u , the KOZ safety weight w_{KOZ} , and the minimum-separation objective d_{min} . Since the adaptive layer operates directly on interpretable MPC optimization parameters, the resulting controller behavior can be analyzed through the evolution of physically meaningful supervisory quantities including w_{KOZ} and d_{min} .

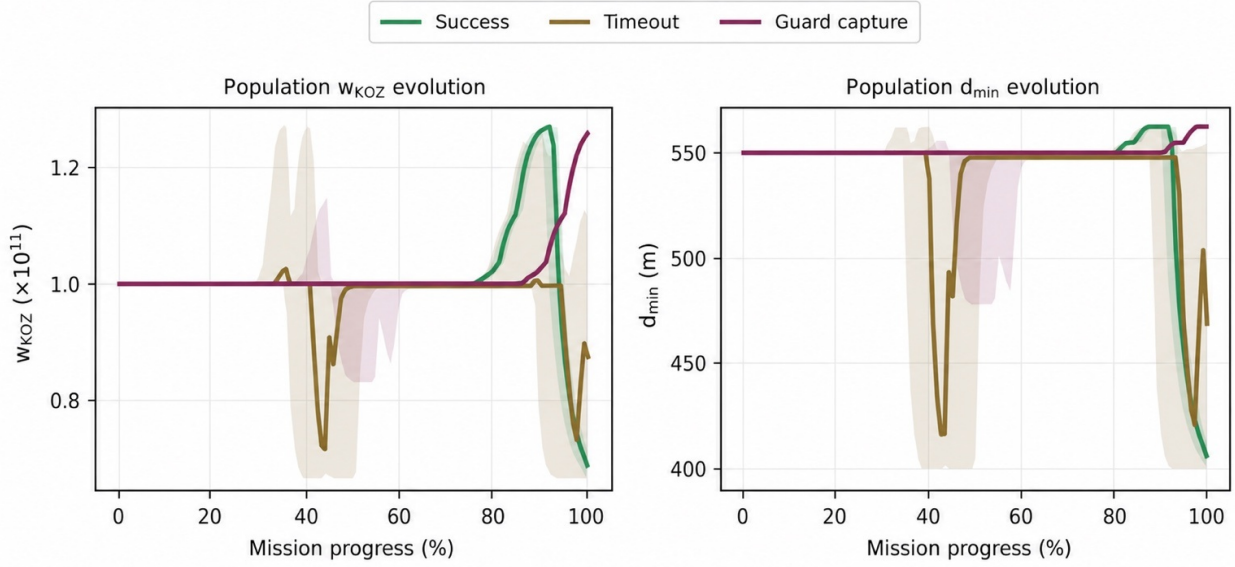


Fig. 6 Adaptive evolution of w_{KOZ} and d_{min} across successful, timeout, and *Guard*-capture trajectories from the 1000-run adaptive MPC robustness dataset.

The recorded parameter-evolution trends in Fig. 6 demonstrate how the supervisory adaptation layer dynamically modified safety-oriented optimization priorities according to the evolving adversarial interaction geometry. In the

reported deployment results, the dominant online adaptive behavior was primarily associated with variation of the predictive KOZ safety weight w_{KOZ} and the minimum-separation objective d_{min} , both of which directly influence the safety-performance balance of the constrained MPC problem.

There are distinct adaptive behaviors across successful, timeout, and *Guard*-capture outcomes. In the successful outcome class (green curves), both w_{KOZ} and d_{min} frequently increased during late-mission interaction phases associated with reduced *Bandit-Guard* separation, consistent with the distance-history trends previously observed in Fig. 4. As the *Bandit* spacecraft approached the terminal rendezvous region near the *Lady* spacecraft, the supervisory layer temporarily increased predictive safety priorities in order to preserve adversarial separation margins during the most collision-sensitive interaction phases. Following successful recovery of *Guard* separation distance, both w_{KOZ} and d_{min} subsequently decreased, indicating a transition back toward more rendezvous-oriented optimization behavior once the immediate adversarial threat had been mitigated.

Timeout cases exhibited more prolonged conservative adaptation behavior, including sustained increases in minimum-separation objectives and delayed recovery toward more rendezvous-oriented optimization priorities. These trends are consistent with the previously observed distance-history evolution, where several timeout trajectories temporarily sacrificed rendezvous progression in order to preserve adversarial separation margins and maintain longer-term safety feasibility under unfavorable interaction geometries.

In contrast, several *Guard*-capture trajectories exhibited comparatively delayed safety-parameter adaptation relative to the evolving adversarial geometry. In many of these cases, increases in w_{KOZ} occurred only after substantial reduction of *Bandit-Guard* separation had already developed, suggesting that the timing of supervisory adaptation played an important role in determining whether adversarial avoidance behavior could be successfully recovered during late interaction phases.

Importantly, the adaptive layer modified interpretable optimization-level MPC quantities rather than directly generating low-level spacecraft thrust commands. Consequently, the observed adaptive behavior remained physically interpretable in terms of evolving predictive safety priorities, minimum-separation objectives, and rendezvous aggressiveness within the constrained MPC formulation.

The recorded parameter trajectories further provide detailed post-run insight into the interaction between adversarial geometry evolution, supervisory adaptation, and constrained MPC behavior throughout closed-loop execution. Because the adaptive variables directly correspond to physically meaningful MPC optimization quantities, changes in controller behavior can be directly associated with evolving safety conditions and predictive maneuvering priorities. A representative geometry-parameter coupling analysis is shown in Fig. 7, including the interaction geometry, adaptive safety parameters, controller activity, and the resulting safety-performance response.

Panel (A) of Fig. 7 shows the relative orbital interaction geometry during a representative successful rendezvous scenario. As the *Guard* spacecraft progressively reduced separation distance relative to the *Bandit* spacecraft prior

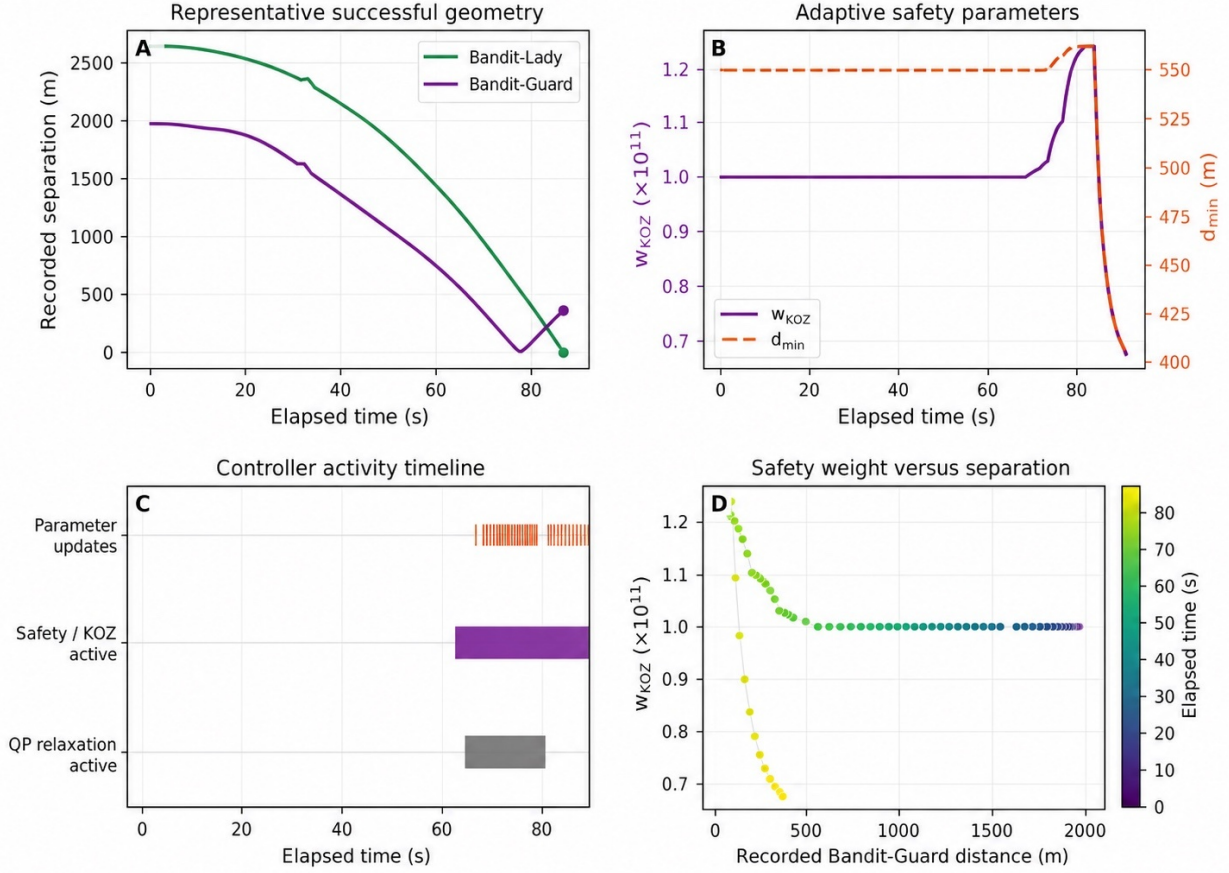


Fig. 7 Interpretability analysis of a representative successful adaptive MPC run showing geometry, supervisory adaptation, and controller behavior.

to approximately $t \approx 80$ s, the supervisory adaptation layer simultaneously increased both the predictive KOZ safety weight w_{KOZ} and the minimum-separation objective d_{min} , as shown in Panel (B). These coordinated parameter increases indicate that the controller temporarily shifted the optimization problem toward more safety-dominant behavior during the most adversarial interaction phase.

As shown in Panel (B) of Fig. 7, both w_{KOZ} and d_{min} decreased following recovery of *Bandit-Guard* separation distance after the closest-approach event, indicating that the supervisory adaptation layer progressively reduced safety-oriented optimization priorities once the immediate adversarial interaction became less critical. This transition was accompanied by simultaneous deactivation of QP relaxation activity together with coordinated reduction of both w_{KOZ} and d_{min} , as shown in Panel (B) and Panel (C). The resulting behavior suggests that, following recovery of adversarial separation margins, the supervisory adaptation layer interpreted the restored interaction geometry as sufficiently safe to transition the optimization priorities back toward more aggressive rendezvous behavior with the *Lady* spacecraft.

Panel (D) further illustrates the direct coupling between w_{KOZ} and *Bandit-Guard* separation distance throughout

the interaction. The resulting relationship demonstrates that the adaptive supervisory behavior remained strongly correlated with physically meaningful safety geometry rather than latent internal policy representations that may be more difficult to interpret directly in terms of physical safety objectives. This constitutes an important distinction relative to fully end-to-end learned control architectures, where internal policy evolution may not directly correspond to interpretable control objectives or operational safety quantities [28]. In the proposed framework, adaptive behavior can be directly traced to interpretable MPC optimization parameters associated with predictive safety preservation, adversarial avoidance behavior, and rendezvous aggressiveness, thereby enabling physically grounded post-run analysis of controller decision evolution throughout complex orbital interaction scenarios.

Across the evaluated Monte Carlo campaigns, the constrained MPC formulation remained active throughout execution despite the online supervisory adaptation process, allowing safety-aware predictive optimization behavior to be preserved under dynamically evolving adversarial interaction conditions.

VII. Conclusion

This work presented an adaptive MPC framework for autonomous RPO in adversarial multi-agent orbital environments. The proposed architecture combined constrained receding-horizon optimization, linearized relative orbital dynamics, predictive safety handling, and supervisory adaptive parameter tuning within a unified closed-loop guidance framework.

Instead of directly generating spacecraft thrust commands as in some end-to-end learned control architectures, the proposed adaptive layer modifies interpretable optimization-level MPC parameters governing tracking performance, safety behavior, prediction-horizon selection, terminal weighting, and keep-out-zone handling within the constrained MPC formulation. Consequently, the constrained MPC structure, orbital dynamics model, actuator limitations, and predictive safety constraints remained active throughout execution.

The proposed framework was evaluated within the official KSPDG Capture-the-Satellite competition environment through large-scale simulation campaigns involving dynamically evolving adversarial interactions between the *Bandit*, *Lady*, and *Guard* spacecraft. The resulting closed-loop simulations demonstrated adaptive adversarial maneuvering behavior and favorable mission-outcome trends relative to representative fixed-parameter MPC configurations while preserving constraint-aware operation and real-time computational feasibility. Moreover, the presented adaptive MPC architecture was developed as part of the Embry-Riddle XDLab team submission, which achieved first place in the 2026 international AIAA Differential Game Capture the Satellite competition.

The proposed framework provided a physically meaningful interpretation of the adaptive controller behavior. The recorded supervisory parameter trajectories enabled physically grounded post-run analysis of controller behavior by directly relating adaptive optimization priorities to evolving orbital geometry and adversarial interaction conditions. The ability to directly relate adaptive supervisory behavior to physically meaningful MPC parameters improves transparency compared to fully end-to-end learned control policies and aligns with broader efforts toward interpretable and trustworthy

autonomous systems for safety-critical applications.

Robustness was tested through persistent constrained optimization, feasibility-preserving safety handling, and extensive empirical Monte Carlo validation under adversarial interaction conditions. This engineering-oriented approach enabled the investigation of adaptive constrained autonomy while preserving the interpretability and modularity of the underlying MPC architecture. The present work does not provide formal recursive feasibility or closed-loop stability guarantees under arbitrary adaptive supervisory updates.

More broadly, the proposed framework supports the perspective that adaptive supervisory parameter tuning can operate effectively on top of validated numerical control structures rather than directly replacing the underlying constrained guidance and control architecture.

Future work will investigate higher-fidelity nonlinear orbital dynamics models, additional adversarial interaction scenarios, onboard implementation considerations, and more advanced supervisory adaptation strategies together with stronger formal safety and stability guarantees for autonomous constrained guidance and control in complex multi-agent orbital environments.

Conflict of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

LS proposed the adaptive MPC research concept, designed and implemented the control architecture, contributed to the adaptive supervisory training framework, conducted data processing and analysis activities, and wrote the manuscript. TB developed the adaptive supervisory training framework, conducted the simulation campaigns, and generated the manuscript figures. CK and DW supervised the research activities, contributed through technical guidance and scientific discussions, and provided feedback and revision support for the manuscript. All authors reviewed and approved the manuscript.

Data Availability Statement

The datasets, simulation framework, and code developed for this study were generated within the research activities conducted at Embry-Riddle Aeronautical University. Access to portions of the simulation framework and associated implementation code may require institutional approval. Requests regarding data and research materials availability can be directed to the corresponding author.

Acknowledgments

The authors would like to thank Ross E. Allen of MIT Lincoln Laboratory, the organizers of the AIAA Capture the Satellite challenge, and the Kerbal Space Program Differential Game development team for providing the competitive simulation environment used throughout this work and for technical discussions that helped motivate aspects of the present research. The authors also thank the reviewers and competition participants whose feedback and interactions contributed to the development and evaluation of the proposed framework. The authors acknowledge the use of ChatGPT (OpenAI) for limited language editing and manuscript polishing; all scientific content and final text were reviewed and approved by the authors.

References

- [1] Rawlings, J. B., Mayne, D. Q., and Diehl, M. M., *Model Predictive Control: Theory, Computation, and Design*, 2nd ed., Nob Hill Publishing, 2017.
- [2] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P., “Constrained model predictive control: Stability and optimality,” *Automatica*, Vol. 36, No. 6, 2000, pp. 789–814. [https://doi.org/10.1016/S0005-1098\(99\)00214-9](https://doi.org/10.1016/S0005-1098(99)00214-9).
- [3] Camacho, E. F., and Bordons, C., *Model Predictive Control*, Springer, 2007.
- [4] Pesce, V., Colagrossi, A., and Silvestrini, S. (eds.), *Modern Spacecraft Guidance, Navigation, and Control: From System Modeling to AI and Innovative Applications*, Elsevier, 2022. <https://doi.org/10.1016/C2020-0-03563-2>.
- [5] Starek, J. A., Schmerling, E., Maher, G. D., Barbee, B. W., and Pavone, M., “Fast, Safe, Propellant-Efficient Spacecraft Motion Planning Under Clohessy–Wiltshire–Hill Dynamics,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 418–438. <https://doi.org/10.2514/1.G001913>.
- [6] Izzo, D., Märtens, M., and Pan, B., “A Survey on Artificial Intelligence Trends in Spacecraft Guidance Dynamics and Control,” 2018. URL <https://arxiv.org/abs/1812.02948>.
- [7] Allen, R. E., Rachlin, Y., Ruprecht, J., Loughran, S., Varey, J., and Viggh, H., “SpaceGym: Discrete and Differential Games in Non-Cooperative Space Operations,” *2023 IEEE Aerospace Conference*, 2023, pp. 1–12. <https://doi.org/10.1109/AERO55745.2023.10115968>.
- [8] Carson, J. M., and Acikmese, B., “A Model-Predictive Control Technique with Guaranteed Resolvability and Required Thruster Silent Times for Small-Body Proximity Operations,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006. <https://doi.org/10.2514/6.2006-6780>.
- [9] Curtis, H. D., *Orbital Mechanics for Engineering Students*, 4th ed., Butterworth-Heinemann, Oxford, UK, 2019.
- [10] Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Cambridge Aerospace Series, Cambridge University Press, 2003.

- [11] Richards, A., Feron, E., How, J., and Schouwenaars, T., “Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming,” *Journal of Guidance Control and Dynamics*, Vol. 25, 2002. <https://doi.org/10.2514/2.4943>.
- [12] Breger, L., and How, J., “Safe Trajectories for Autonomous Rendezvous of Spacecraft,” *Journal of Guidance Control and Dynamics*, Vol. 31, 2008. <https://doi.org/10.2514/1.29590>.
- [13] Federici, L., Benedikter, B., and Furfaro, R., “Reinforcement-Learning-Enhanced Model Predictive Control with Application to Autonomous Planetary Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 49, No. 3, 2026, pp. 788–805. <https://doi.org/10.2514/1.G009534>.
- [14] Capra, L., Brandonisio, A., and Lavagna, M. R., “Reinforced Model Predictive Guidance and Control for Spacecraft Proximity Operations,” *Aerospace*, Vol. 12, No. 9, 2025. <https://doi.org/10.3390/aerospace12090837>.
- [15] He, H., Boyd-Graber, J. L., Kwok, K., and III, H. D., “Opponent Modeling in Deep Reinforcement Learning,” *CoRR*, 2016. URL <https://arxiv.org/abs/1609.05559>.
- [16] García, J., and Fernández, F., “A Comprehensive Survey on Safe Reinforcement Learning,” *Journal of Machine Learning Research*, Vol. 16, No. 42, 2015, pp. 1437–1480. URL <http://jmlr.org/papers/v16/garcia15a.html>.
- [17] Brunke, L., Greeff, M., Hall, A., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A., “Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning,” 2021. <https://doi.org/10.48550/arXiv.2108.06266>.
- [18] Hewing, L., Wabersich, K., Menner, M., and Zeilinger, M., “Learning-Based Model Predictive Control: Toward Safe Learning in Control,” *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 3, 2020. <https://doi.org/10.1146/annurev-control-090419-075625>.
- [19] Koller, T., Berkenkamp, F., Turchetta, M., and Krause, A., “Learning-Based Model Predictive Control for Safe Exploration,” *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6059–6066. <https://doi.org/10.1109/CDC.2018.8619572>.
- [20] Kiumarsi, B., Vamvoudakis, K. G., Modares, H., and Lewis, F. L., “Optimal and Autonomous Control Using Reinforcement Learning: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 6, 2018, pp. 2042–2062. <https://doi.org/10.1109/TNNLS.2017.2773458>.
- [21] Fridovich-Keil, D., Ratner, E., Peters, L., Dragan, A. D., and Tomlin, C. J., “Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games,” *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1475–1481. <https://doi.org/10.1109/ICRA40945.2020.9197129>.
- [22] Clohessy, W. H., and Wiltshire, R. S., “Terminal Guidance System for Satellite Rendezvous,” *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658. <https://doi.org/10.2514/8.8704>.
- [23] Kerrigan, E. C., and Maciejowski, J. M., “Soft Constraints and Exact Penalty Functions in Model Predictive Control,” *Proceedings of the UKACC International Conference on Control*, 2000. URL https://www-control.eng.cam.ac.uk/Homepage/papers/cued_control_53.pdf.

- [24] Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P., “Control Barrier Function Based Quadratic Programs for Safety Critical Systems,” *IEEE Transactions on Automatic Control*, Vol. 62, No. 8, 2017, pp. 3861–3876. <https://doi.org/10.1109/TAC.2016.2638961>.
- [25] de Boer, P.-T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y., “A Tutorial on the Cross-Entropy Method,” *Annals of Operations Research*, Vol. 134, No. 1, 2005, pp. 19–67. <https://doi.org/10.1007/s10479-005-5724-z>.
- [26] Moss, R. J., “Cross-Entropy Method Variants for Optimization,” *CoRR*, Vol. abs/2009.09043, 2020. <https://doi.org/10.48550/arXiv.2009.09043>.
- [27] Hansen, N., and Ostermeier, A., “Completely Derandomized Self-Adaptation in Evolution Strategies,” *Evolutionary Computation*, Vol. 9, 2001, pp. 159–195. <https://doi.org/10.1162/106365601750190398>.
- [28] Doshi-Velez, F., and Kim, B., “Towards A Rigorous Science of Interpretable Machine Learning,” *arXiv: Machine Learning*, 2017. <https://doi.org/10.48550/arXiv.1702.08608>.