Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-11 (2025) 13–18

A Constraints-aware Antagonistic Controller with Disturbance-adaptive Attacks

Shafqat A. Siyyal * **José M. Maestre** ** **Alessandro Freddi** * **Sauro Longhi** *

* *Università Politecnica delle Marche, 60131 Ancona, Italy (e-mails: s.a.siyyal@pm.univpm.it, {a.freddi, sauro.longhi}@univpm.it)*
** *Universidad de Sevilla, 41092 Sevilla, Spain (e-mail: pepemaestre@us.es)*

Abstract: This paper proposes an extension to attack strategies in cyber physical systems, based on adopting input sequences which can maximize an objective function typically designed for minimization. In detail, the proposed method prioritizes the violation of state constraints over cost maximization to directly driving the system into an unsafe region. To achieve this, we reformulate the cost function by introducing a slack variable into the optimization problem, explicitly encouraging constraint violations. The framework also accounts for external disturbances that may counteract the controller's objectives. To guarantee a certain level of damage despite such disturbances, the problem is formulated as max-min optimization where the attacker optimizes for the worst-case scenario. Given that the maximization subproblem is NP-hard, we address this computational challenge using a vertex enumeration method. The effectiveness of the proposed approach is validated in a simulation scenario based on an autonomous aerial vehicle using a Model Predictive Controller (MPC), showing that constraint violations can be achieved at a reduced cost compared to existing methods.

Keywords: Cyber-security, Model Predictive Controller (MPC), cyber-physical systems (CPS)

1. INTRODUCTION

Cyber-Physical Systems (CPS) are increasingly adopted in industries ranging from unmanned vehicles and automated factories to critical infrastructure such as power grids and smart cities Shah and Yaqoob (2016). However, this integration introduces new vulnerabilities, such as cyber attacks which represent a threat to both digital integrity and physical processes Kong (2021). Due to the interconnected nature of CPS, attacks can lead to cascading failures, impacting not only the targeted system but also interconnected subsystems. A well-known example is the Stuxnet attack on Iran's Natanz nuclear facility, where malicious control input sequences were used to subtly manipulate the rotational speeds of centrifuges, causing mechanical degradation without immediate detection Langner (2013). This case illustrates that beyond data breaches, attackers can engineer control-level interventions to physically destabilize a system. Although the stages of cyber attacks from reconnaissance to execution are well understood, most research has focused on developing detection and mitigation strategies Barboni et al. (2020); Abdelwahab et al. (2020); Barboni et al. (2018); Riverso et al. (2016). In contrast, less attention has been devoted to understanding how attackers can systematically design attacks to destabilize the system or cause operational failures Biju et al. (2019). Recent studies have begun to explore this gap by analyzing how attackers exploit system vulnerabilities for greater impact. One key aspect of understanding such attacks involves examining how the performance of cyber-physical systems is typically evaluated. These systems are normally closed loop, with control actions typically optimizing a convex quadratic cost function, as in Linear Quadratic Regulators (LQR) or MPC regulators which focus on minimizing deviations from a desired equilibrium state to ensure stability and optimality under defined conditions. Consequently, when designing attacks to such systems, it becomes natural to target and maximize the very same cost function. For instance, antagonistic control identifies inputs sequences that maximize the cost function, driving the system state away from the reference point to cause instability, thereby supporting both attack development and worst-case analysis Lipp and Boyd (2016). Due to the non-convex nature of this problem, heuristic approaches, such as the convex–concave procedure and S-procedure based bounds, have been proposed to approximate worst-case scenarios Lipp and Boyd (2016). Similarly, the authors in Guthrie and Mallada (2019) examine the strategic execution of stealthy data injection attacks. By framing the attack as a non-convex optimization problem and employing dynamic programming, they design optimal attack vectors that maximize deviation while avoiding detection. These studies and related strategies have primarily focused on maximizing and solving this non-convex problem, which

⋆ This work was supported by the European Union Next-Generation EU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 1, INVESTIMENTI 3.4 e 4.1 – Decreto del Ministero dell'Università e della Ricerca n.351 del 09/04/2022) within the Italian National Ph.D. Program in Autonomous Systems (DAuSy). Financial support from grant PID2023-152876OB-I00 funded by MCIN/AEI/10.13039/501100011033 and ERDF/EU is also gratefully acknowledged (project C3PO-R3).

10.1016/j.ifacol.2025.09.517

is NP-hard in general, arriving at suboptimal solutions via convex-concave approximations, semi-definite relaxations, or general nonlinear programming methods.

In contrast to existing approaches, this paper introduces a method that incorporates constraints into the optimization problem enabling the prioritization of constraint violations and a more direct exploitation of system vulnerabilities. Building upon our previous work Cavanini et al. (2024), the proposed adversarial attack strategy reformulates the optimization problem to emphasize not only state deviations but also the intentional violation of operational safety limits. This extension introduces a new dimension to attack design, improving its impact on system stability and safety. Additionally, this work accounts for external disturbances by adopting the approach from de la Peña et al. (2005), where the controller is specially designed to handle worst-case uncertainties. The formulation ensures that the attack remains effective by computing inputs that drive the system away from its nominal state, guaranteeing a certain level of damage.

The proposed framework is based on MPC due to its ability to predict the impact of control actions using the system model. At each time step, an optimization problem is solved to determine whether the computed control inputs will cause sufficient disruption by the end of the prediction horizon. If the disruption meets the desired threshold, the inputs are applied; otherwise, the attacker waits to identify more effective control inputs. The proposed algorithm is validated through its application to an academic example and to an Unmanned Aerial Vehicle (UAV).

The structure of the rest of the paper is as follows: Section II presents the problem formulation, introducing the foundation of the antagonistic controller and proposing a reformulation of the original maximization problem. Section III provides numerical examples to demonstrate the effectiveness of the proposed approach. Section IV concludes the paper with final remarks and outlines possible directions for future research.

2. PROBLEM FORMULATION

Consider a discrete time linear time invariant system:

$$x_{k+1} = Ax_k + Bu_k + Dw_k \tag{1}$$

here, $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, and $w_k \in \mathbb{R}^{n_w}$ denote the state, control input, and disturbance vectors, respectively, where n_x, n_u, and n_w, represent the number of state variables, control inputs and disturbance components. The disturbance w_k is assumed to lie within a bounded polyhedral set $\mathcal{W} \subseteq \mathbb{R}^{n_w}$ that contains the origin. The initial state is denoted by x_0. For simplicity, it is assumed that the full state vector is known and measurable.

Given the system dynamics in (1), MPC computes an optimal sequence of control inputs over a finite prediction horizon $N \in \mathbb{N}^+$, where $\mathbb{N}^+$ denotes the set of positive natural numbers. The controller minimizes a predefined quadratic cost function subject to constraints on both the system states and control inputs. See Camacho and Bordons (2007).

The predicted state trajectory over the horizon N can be expressed recursively as:

$$x_k = A^j x_0 + \sum_{i=1}^{j} A^{i-1} B u_{j-i} + \sum_{i=1}^{j} A^{i-1} D w_{j-i}, \quad j = 1, 2, \ldots, N \tag{2}$$

where x_k denotes the predicted state at step k based on the initial state and the sequence of input and disturbance. For a compact representation, the predicted state trajectories are rewritten in matrix form as:

$$X = H_x x_0 + H_u U + H_w W \tag{3}$$

where $X \in \mathbb{R}^{N \cdot n_x}$, $U \in \mathbb{R}^{N \cdot n_u}$, and $W \in \mathbb{R}^{N \cdot n_w}$ are stacked vectors of the predicted states, control inputs, and disturbances, respectively. The matrices $H_x \in \mathbb{R}^{N \cdot n_x \times n_x}$, $H_u \in \mathbb{R}^{N \cdot n_x \times N \cdot n_u}$, and $H_w \in \mathbb{R}^{N \cdot n_x \times N \cdot n_w}$ capture the influence of the initial state, input sequence and disturbance on the system evolution. These matrices are constructed as described in Schwenzer et al. (2021).

The cost function, denoted by $J(X, U, W)$, depends on the predicted state, input and disturbance sequences, and is defined as:

$$J(X, U, W) = \sum_{i=1}^{N} \|Q(x_{k+i|k} - r_{k+i})\|_2^2 + \sum_{j=0}^{N-1} \|R u_{k+j|k}\|_2^2 \tag{4}$$

where, $Q \in \mathbb{R}^{n_x \times n_x}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are positive semidefinite weighting matrices ($Q, R \geq 0$), ensuring convexity of the optimization problem Agrawal et al. (2021). The terms $x_{k+i|k}$ and $u_{k+j|k}$ denote the predicted state and control input at time step $k+i$ and $k+j$, respectively, based on information available at time k. The vector $r_k \in \mathbb{R}^{n_x}$ denotes the reference trajectory.

Based on the predicted dynamics and cost formulation, the standard MPC problem is formulated as:

$$\min_{U} \quad J(X, U, W), \tag{5a}$$

$$\text{s.t.} \quad X = H_x x_0 + H_u U + H_w W \tag{5b}$$

$$A_x X \leq b_x, \quad A_u U \leq b_u \tag{5c}$$

where the control sequence U is the decision variable. The matrices $A_x \in \mathbb{R}^{2n_x N \times n_x N}$ and $A_u \in \mathbb{R}^{2n_u N \times n_u N}$, along with the vectors $b_x \in \mathbb{R}^{2n_x N}$ and $b_u \in \mathbb{R}^{2n_u N}$, define the stacked state and input constraints over the prediction horizon. The disturbance sequence W is treated as an estimated quantity in the optimization and is assumed to lie within a known bounded set $\mathcal{W}^N \subseteq \mathbb{R}^{N n_w}$.

2.1 Antagonistic Controller

The antagonistic control problem modifies the standard MPC formulation by reversing the optimization objective, seeking to maximize rather than minimize $J(X, U)$ Lipp and Boyd (2016):

$$\max_{U} \quad J(X, U) \tag{6a}$$

$$\text{s.t.} \quad X = H_x x_0 + H_u U \tag{6b}$$

$$A_u U \leq b_u \tag{6c}$$

here, maximizing over the control sequence U yields the worst case state trajectory consistent with the system

dynamics. Unlike standard MPC, this formulation involves the maximization of a convex cost function, which results in a non-convex and NP-hard problem Žerovnik (2015). Let $J^*(X,U)$ denote the optimal value of this modified problem, representing the worst case cost value. Since solving NP-hard problems is not the primary focus here, we assume that a solution can be obtained using vertex enumeration, as discussed in later sections. This formulation focuses solely on the maximization problem and neglects the state constraints, under assumption that such constraints are unnecessary for modeling adversarial behavior. While this simplification is commonly adopted, it introduces limitations. In particular, the controller may overlook opportunities to trigger state constraint violations that although not yielding the highest possible cost, could be strategically beneficial for an effective attack. Moreover, maximizing the cost function alone does not quantify the severity of any safety violations. It also lacks a tunable or interpretable mechanism for specifying which states to target and by how much they should be violated. As a results, the formulation remains a pure maximization problem, which may still serve the adversary's purpose but offers limited flexibility in shaping the nature of the attack.

2.2 Reformulated Objective Function

To account for state constraint violations, we reformulate the original problem (6) by introducing a nonnegative slack variable and its associated penalty in the cost function. This allows the adversarial controller to drive the system into unsafe regions in a controlled and quantifiable way. While slack variables are commonly used in optimization problems to handle infeasibility Kerrigan and Maciejowski (2000), they are here adapted to serve the adversary's objective by selectively relaxing state constraints. The reformulated problem introduces an auxiliary variable α and is defined as:

$$\min_{\alpha,s} \quad \alpha + p^\top s, \tag{7a}$$

s.t.

$$\alpha \geq \max_U \min_W J(X,U,W) \tag{7b}$$

$$s \geq 0 \tag{7c}$$

$$A_x X \leq b_x + s, \quad A_u U \leq b_u \tag{7d}$$

where the vectors $p, s \in \mathbb{R}^{n_x}$ are defined elementwise, with each component corresponding to a specific state constraint . Although the problem is reformulated as a minimization, constraint (7b) ensures that α captures the upper bound of the original cost $\max_U \min_W J(X,U,W)$, thus preserving the adversarial objective. The slack condition $s \geq 0$, together with penalty vector p act as tunable design parameters that define the attack configuration. While $s \geq 0$ imposes non negativity, users may enforce higher element wise lower bounds on s to guarantee a minimum level of violation for each state. For instance, setting $s_i \geq 1$ forces the solution to include at least one unit of violation in the i^{th} state. The penalty vector p determines the relative cost of violating individual state constraints. A high penalty discourages slack usage for a particular state, enforcing that the corresponding constraint remain tight. Conversely, assigning a lower penalty to certain states allows optimizer to selectively violate them. This configuration enables the adversary to prioritize critical states for protection or exploitation, offering a more targeted and flexible attack strategy.

The formulation further accounts for bounded disturbances (see Eq. (7b)) by adopting a max-min optimization structure, where the control input is optimized against the worst case disturbance realization. Following the approach in de la Peña et al. (2005), the outer maximization over U captures the adversarial objective, while the inner minimization over W models the disturbance as an opposing agent minimizing the cost. The resulting bilevel optimization problem is given by:

$$\max_U \min_W \quad J(X,U,W) \tag{8a}$$

$$\text{s.t.} \quad A_u U \leq b_u \quad \forall U \in \mathcal{V}^{\mathrm{N}} \tag{8b}$$

To isolate the contribution of the disturbance from the nominal system behavior, the cost function is equivalently decomposed by adding and subtracting the disturbance free term $J(X,U,0)$, to facilitate separate analysis of each component within adversarial design framework. This results in a reformulated cost expression where the nominal cost is maximized over the control input U and the disturbance influence is captured by the residual term $J(X,U,W) - J(X,U,0)$, which is minimized over all admissible W, leading to the following form:

$$\max_{U \in \mathcal{V}(U_N)} \Big(J(X,U,0) + \min_W \big(J(X,U,W) - J(X,U,0) \big) \Big) \tag{9}$$

where,

$$J(X,U,W) - J(X,U,0) = w^\top H_w^\top H_w w + 2 w^\top H_w^\top (H_x x + H_u u) \tag{10}$$

To address the non-convexity of the outer maximization, a vertex enumeration approach is used. Since the maximization is performed over a convex cost function subject to linear input constraints, the optimal solution lies at a vertex of the feasible input set. Specifically, U represents a sequence of control input vectors over the horizon N, where each u_k (for $k = 0, \ldots, N-1$) is selected from the vertices of the admissible input set. The set of all such sequences is denoted $\mathcal{V}^N$, which includes every possible combination of $U = [u_0^\top, u_1^\top, \ldots, u_{N-1}^\top]^\top$, with each $u_k \in \mathcal{V}$, typically corresponding to the upper/lower bounds of input variable. Consider that the maximizer may not be unique (as other vectors U could also yield the maximum), the uniqueness of the solution is not important for the outer maximization problem. However, the number of vertices in $\mathcal{V}$ grows exponentially with the prediction horizon N, which can lead to increased computational complexity.

Remark 2.1. A more conservative yet simpler version of (9) can be derived by explicitly solving for the unconstrained optimizer W^* in (10) as function of U using multiparametric programming. By substituting W^* into the cost function, the inner minimization is eliminated, reducing the problem to the maximization of a convex function. This transformation allows the outer maximization problem to be efficiently solved via vertex enumeration.

Algorithm 2.1 summarizes the procedure for solving reformulated problem, including evaluation of upper bound α, solving the max-min structure and performing vertex enumeration to identify optimal control sequence.

Algorithm 2.1 Reformulated Cost Function with Worst-Case Disturbance

Given: Initial state x_k
Step 0 (Computed Offline): Generate all possible input sequences for N instants:
$Uvertices = \text{generateVertices(u\_bounds)}$
Step 1: For each time instant:
A. Compute upper bound of $J(x, u, w)$: $\bar{J}$
for each $u_{\text{vertex}} \in Uvertices$:
Minimize $J(x, u_{\text{vertex}}, w)$ over w
if $J(x, u_{\text{vertex}}, w_{\text{opt}}) \geq \bar{J}$:
$\bar{J} = J(x, u_{\text{vertex}}, w_{\text{opt}})$
end if
end for
B. Solve the reformulated optimization:
Minimize $\alpha + p^T s$
s.t.
$\alpha \geq J(x, u_{\text{vertex}}, w_{\text{opt}})$,
$s \geq 0$, $A_x X \leq b_x + s$, $A_u U \leq b_u$
C. Extract and apply optimal control input

3. NUMERICAL EXAMPLE

This section applies the proposed framework to two case studies: a quadrotor system and a simplified integrator model, highlighting the differences between the maximization based approach and the proposed method. Simulations were proposed in MATLAB R2024b on a macbook with an apple M1 chip and 16GB RAM, using a 5 steps horizon and a fixed time step of 0.01.

3.1 Case Study: Simple First-Order Model

To highlight the difference between the maximization based approach and the proposed formulation, which prioritizes constraint violation over cost maximization, we consider a simple 1st order LTI discrete model:

$$x_{k+1} = x_k + u_k + w_k \tag{11}$$

where the state x is constrained to the range $[-5, 5]$ and the control input u is subject to input bounds $u \in [-1, 1]$.

Figure 1 illustrates the system evolution under the maximization based approach. Prior to the vertical dashed line, the system evolves under nominal conditions. Beyond this point, an adversarial controller selects control inputs (e.g., $[+1]$) to maximize a predefined cost function. This strategy inherently biases the input selection toward actions that yield higher cost, potentially overlooking inputs such as $[-1]$, which, although suboptimal in terms of cost, would still drive the system into unsafe regions.

In contrast, Figure 2 represents the proposed violation-driven method, where the control input $[-1]$ is selected to intentionally steer the system toward violating state constraints, irrespective of the cost. This distinction is evident in the stage cost plot, which demonstrates that constraint violations can occur along trajectories with lower cost. To evaluate the method's consistency under varying external conditions, Figure 3 presents the state trajectories for varying constraints bounds, which indirectly reflect different disturbance ranges. Despite the tighter feasible regions (e.g., [-3, 3]), the method continues to cause violations, though with varying timing and severity. Considering the

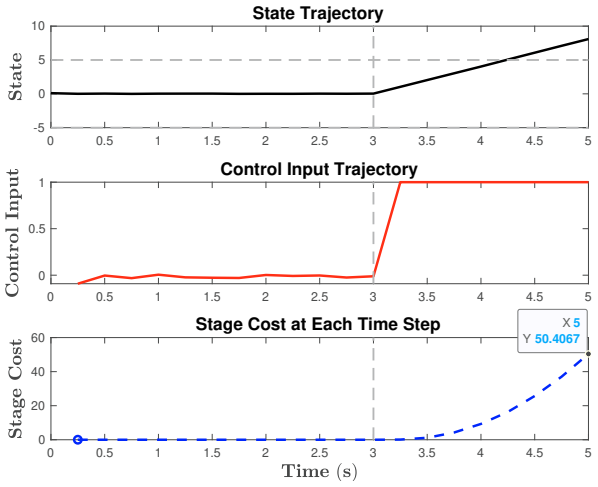

Fig. 1. Maximization-based approach

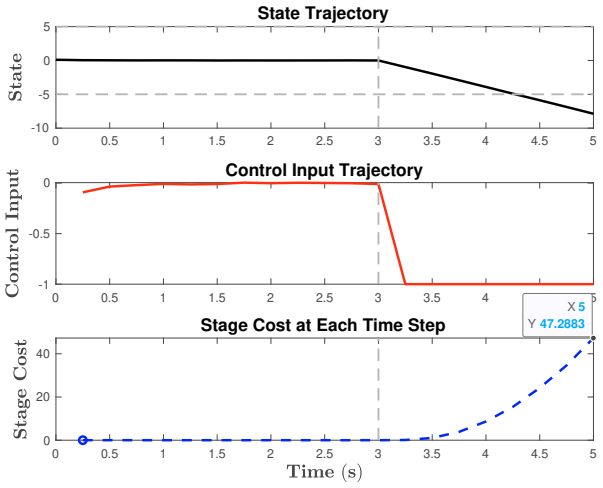

Fig. 2. Violation-based approach

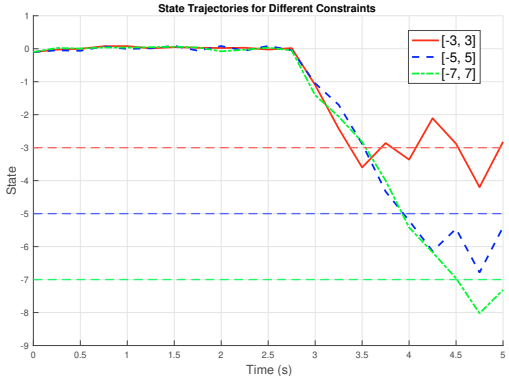

Fig. 3. Violation-based approach under varying constraints

computational complexities, Table 1 represents the exponential growth in both vertex count and computation time with respect to the prediction horizon N.

3.2 Case Study:Quadrotor Dynamics

A multirotor system is considered, where flight controller uses an inner-outer loop architecture for stability and reference tracking, as shown in Fig. 4 Bouabdallah and Siegwart (2007). The inner loop, which operates at 400

Table 1. Computational complexity of the simple integrator model across prediction horizons

Horizon N	Vertices $2^{n_u N}$	Avg. Time (s)	Std. Deviation (s)
3	$2^{1\times3} = 8$	0.012	0.012
5	$2^{1\times5} = 32$	0.026	0.001
7	$2^{1\times7} = 128$	0.106	0.015

Hz, as in autopilot systems like ArduPilot, stabilizes the vehicle by controlling roll, pitch, and yaw, while the outer loop governs slower dynamics (e.g., 40 Hz), focusing on horizontal motion and reference tracking. The quadrotor's physical and aerodynamic parameters (inertia, mass, dimensions, lift and drag coefficients) are taken from Cavanini et al. (2024). The antagonistic attack framework exploits the dual-loop structure of the flight controller, focusing on the inner loop's fast dynamics to destabilize the system. By forcing deviations in attitude angles (ϕ and θ) from their reference values, the attack can push the controller beyond its operational limits, hindering its ability to recover. Attackers can inject destabilizing inputs within brief timeframes, overriding nominal commands and exploiting the inner loop's sensitivity to cause instability. The malicious controller can manipulate the rotational speed of each each rotor (Ω_1, Ω_2, Ω_3, and Ω_4) while adhering to real-world constraints, such as motor speed limits. The proposed antagonistic attack can be embedded through firmware manipulation or exploiting communication vulnerabilities Cui et al. (2013), operating alongside the nominal controller to monitor system states in real time, passively gathering data to devise an optimal attack strategy and waiting for the optimal moment to disrupt the system. However, it is important to emphasize that the focus here is not on identifying the optimal timing; see Cavanini et al. (2024) for attack initiation condition.

To simulate environmental effects like wind gusts, external disturbances w_k are applied, constrained within $[-0.3, +0.3]$, targeting velocity states $(\dot{x}, \dot{y}, \dot{z})$ and orientation states (p, q, r). These disturbances, modeled through a disturbance matrix D, focus on velocity and angular states, which are particularly sensitive to environmental conditions.

In configuring the Antagonistic MPC, a prediction horizon $N_p = 5$ and a control horizon $N_u = 4$ are used. The linearized dynamics of the quadrotor system, represented in state-space form, are characterized by the system matrices A and B Cavanini et al. (2024).

The cost function is designed to prioritize deviations in attitude by assigning high weights to roll and pitch angles ($q_\phi = q_\theta = 1$), presented as:

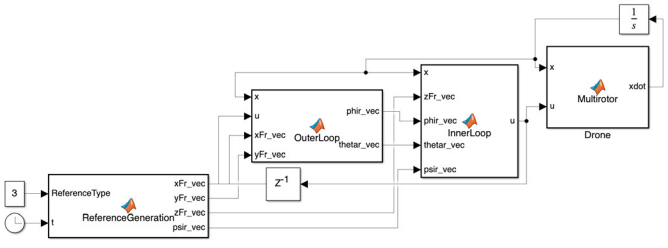

Fig. 4. Quadrotor inner-outer control loop architecture

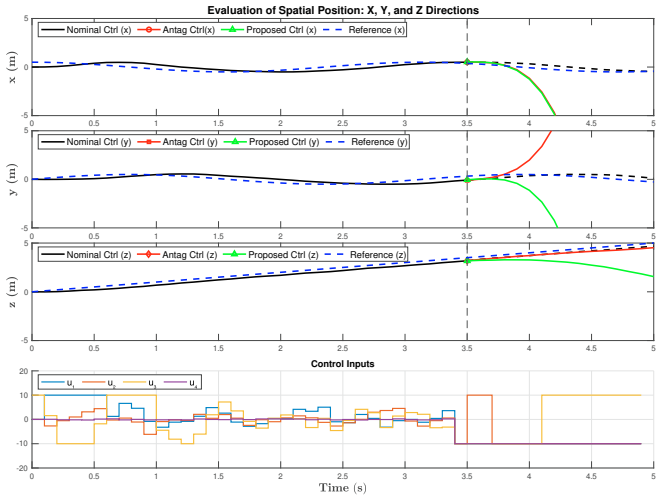

Fig. 5. Outer-loop response of a quadrotor model and corresponding control inputs

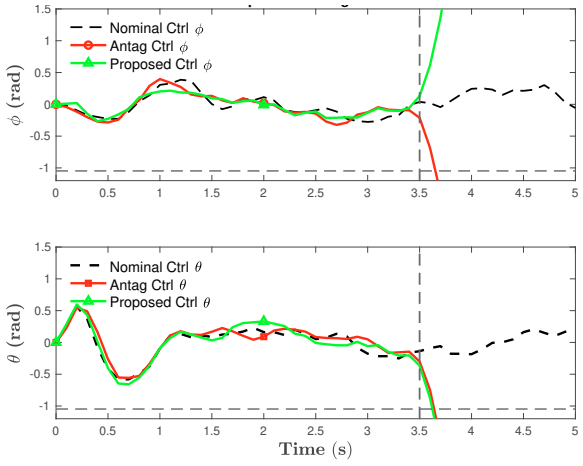

Fig. 6. Inner-loop response of a quadrotor model

$$Q_z = \mathrm{diag}(0, \ldots, 0, q_\varphi, q_\theta, 0), \tag{12}$$

where q_φ and q_θ penalize deviations in roll and pitch angles, while control effort is ignored ($R_u = 0$). The resulting cost function is defined as:

$$J = \sum_{i=0}^{N_p} \left(q_\varphi \varphi_{k+i}^2 + q_\theta \theta_{k+i}^2 \right). \tag{13}$$

This setup allows the adversarial controller to maximize state deviations without restricting control input magnitudes, enabling it to breach safety limits effectively under bounded disturbances.

Figure 5 illustrates the outer-loop response of the quadrotor, comparing its performance under nominal and adversarial control. The blue dashed lines represent the reference trajectory, the black lines depict state variables under nominal control, the red lines and the green lines show state variables under the influence of the attack. Initially, the system operates under a nominal MPC controller, effectively countering random disturbances and closely tracking the reference trajectory. At the vertical dashed line, an adversarial controller is activated, employing either a maximization-based or constraint violation-based strategy to destabilize the system. This leads to

rapid changes in control input and a gradual divergence from the reference trajectory, particularly in the outer loop due to its slower dynamics. Figure 6 shows the inner loop response, where pitch and roll angles, stable under nominal control (black), escalate sharply under attack (red and green), reaching 90° and confirming destabilization. Although the two adversarial controller show similar performance in the figures, their underlying strategies are fundamentally different. The maximization based controller applies the same aggressive input at each step to maximize cost, often leading to repetitive behavior. Meanwhile, the constraint violation based approach adjusts its inputs based on the current state, actively seeking to violate system constraints.This distinction, while not always visually apparent in high dimensional systems, is crucial for understanding their impact on control security. Further, Table 2 reports the computational results for the quadrotor model. With a larger number of control inputs, the total number of vertices grows accordingly with the prediction horizon N, leading to increased computation times. The results align with the expected scaling behavior of the approach in higher-dimensional systems.

Table 2. Computational complexity of the quadrotor model across prediction horizons

Horizon N	Vertices $2^{n_u N}$	Avg. Time (s)	Std. Deviation (s)
3	$2^{4\times3} = 4096$	1.608	0.085
5	$2^{4\times5} = 1048576$	1.9	0.048
7	$2^{4\times7} = 268435456$	2.5	0.085

4. CONCLUSION

This paper extends the original antagonistic control formulation by shifting the objective from cost maximization to explicit prioritization of state constraints violations. In standard formulation, the adversary maximizes a cost function, with constraint violations occurring as a secondary effect. However, this approach may overlook control inputs that, although resulting in lower costs, still lead to constraint violations. By direct targeting constraints, the proposed method aligns more closely with the adversarial intent. Furthermore, this work accounts for external disturbances to ensure that adversarial impact is maintained even under the least favorable conditions. To achieve this, the problem is formulated as a max-min optimization, allowing the attacker to select control inputs that remain effective against worst case disturbances. The proposed approach was first validated on a simple academic example to highlight the difference between the original and proposed formulations. Additionally, a case study on a quadrotor system shows how this method can be applied to high-dimensional models, effectively destabilizing them by targeting critical states. Future work may further explore strategies to reduce the computational burden of the proposed framework for large scale systems and to investigate the optimal timing for initiating attack, considering system condition or the current state, to maximize disruption.

REFERENCES

Abdelwahab, A., Lucia, W., and Youssef, A. (2020). Set-theoretic control for active detection of replay attacks with applications to smart grid. In *2020 IEEE Conference on Control Technology and Applications (CCTA)*.

Agrawal, A., Barratt, S., and Boyd, S. (2021). Learning convex optimization models. *IEEE/CAA J. Autom. Sinica*, 8(8).

Barboni, A., Boem, F., and Parisini, T. (2018). Model-based detection of cyber-attacks in networked mpc-based control systems. *IFAC-PapersOnLine*, 51(24).

Barboni, A., Rezaee, H., Boem, F., and Parisini, T. (2020). Detection of covert cyber-attacks in interconnected systems: A distributed model-based approach. *IEEE Trans. Autom. Control*, 65(9).

Biju, J.M., Gopal, N., and Prakash, A.J. (2019). Cyber attacks and its different types. *Int. Res. J. Eng. Technol.*, 6(3).

Bouabdallah, S. and Siegwart, R. (2007). Full control of a quadrotor. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Camacho, E.F. and Bordons, C. (2007). *Constrained model predictive control*.

Cavanini, L., Felicetti, R., Ferracuti, F., Freddi, A., Longhi, S., and Siyyal, S.A. (2024). Antagonistic model predictive control for quadrotor cyber attacks. In *Proc. IEEE/ASME Int. Conf. Mechatronic and Embedded Systems and Applications (MESA)*.

Cui, A., Costello, M., and Stolfo, S. (2013). When firmware modifications attack: A case study of embedded exploitation.

de la Peña, D.M., Alamo, T., Ramírez, D., and Camacho, E. (2005). Min-max model predictive control as a quadratic program. *IFAC Proceedings Volumes*, 38(1). doi:https://doi.org/10.3182/20050703-6-CZ-1902.00988.

Guthrie, J. and Mallada, E. (2019). Adversarial model predictive control via second-order cone programming. In *2019 IEEE 58th Conference on Decision and Control (CDC)*.

Kerrigan, E.C. and Maciejowski, J.M. (2000). Soft constraints and exact penalty functions in model predictive control. In *Control 2000 Conference, Cambridge*.

Kong, P.Y. (2021). A survey of cyberattack countermeasures for unmanned aerial vehicles. *IEEE Access*, 9.

Langner, R. (2013). To kill a centrifuge: A technical analysis of what stuxnet's creators tried to achieve. *The Langner Group*, 37.

Lipp, T. and Boyd, S. (2016). Antagonistic control. *Syst. Control Lett.*, 98.

Riverso, S., Boem, F., Ferrari-Trecate, G., and Parisini, T. (2016). Plug-and-play fault detection and control-reconfiguration for a class of nonlinear large-scale constrained systems. *IEEE Trans. Autom. Control*, 61(12).

Schwenzer, M., Ay, M., Bergs, T., and Abel, D. (2021). Review on model predictive control: An engineering perspective. *Int. J. Adv. Manuf. Technol.*, 117(5).

Shah, S.H. and Yaqoob, I. (2016). A survey: Internet of things (iot) technologies, applications and challenges. In *2016 IEEE Smart Energy Grid Engineering (SEGE)*.

Žerovnik, J. (2015). Heuristics for np-hard optimization problems-simpler is better!? *Logist. Supply Chain Sustain. Glob. Chall.*, 6(1).