H-Infinity Tracking for Intelligent Edge-Controlled Systems over Fading Channels in AI-RAN

Minjie Tang, Chenyuan Feng, Geyong Min, and Tony Q. S. Quek, *Fellow, IEEE*

***Abstract*—In this work, we investigate the H-infinity tracking control problem for linear systems operating within an AI-driven radio access network (AI-RAN), where communication between the remote tracking controller and the dynamic plant occurs over random wireless multiple-input multiple-output (MIMO) fading channels. We formulate the problem as a stochastic zero-sum game and derive the corresponding coupled optimality condition that characterizes the Nash equilibrium. To address the curse of dimensionality, we introduce a structured reduced-order optimality condition that significantly simplifies the solution process. We further develop an online learning algorithm based on structured stochastic approximation (SA) that asymptotically learns the Nash equilibrium in real time. Extensive simulations validate the proposed method, demonstrating superior performance in tracking accuracy, convergence speed, and computational efficiency compared to state-of-the-art methods.**

***Index Terms*—Robust control, H-infinity control, stochastic game, online learning, stochastic approximation.**

I. INTRODUCTION

The rapid proliferation of intelligent applications, such as autonomous vehicles and UAV swarms, is driving the evolution of Radio Access Networks (RANs) towards greater intelligence and responsiveness. AI-enabled RANs (AI-RANs), by embedding AI capabilities at the network edge, enable real-time sensing, inference, and control between edge nodes and user devices [1], [2]. This architectural shift not only reduces latency but also facilitates the deployment of closed-loop control systems over wireless networks, where robust control becomes essential to ensure stability and performance under uncertain and dynamic conditions [3], [4]. In such AI-RAN scenarios, control systems must operate over time-varying multiple-input multiple-output (MIMO) fading channels, which are inherently unreliable. These uncertainties, compounded by potential adversarial attacks on communication channels, significantly challenge the design of reliable feedback controllers. Robust control, particularly H-infinity ($\mathcal{H}_\infty$) tracking control, has become a critical tool for enabling safe and effective control in RAN environments [5], [6].

Several prior works have investigated robust control strategies in networked systems. For instance, heuristic Proportional-Integral-Derivative (PID)-based frequency-domain methods have been applied in [7], though their performance is constrained by the lack of systematic optimization. Linear Quadratic Tracking (LQT) approaches [8] compute optimal gains under idealized communication assumptions, but perform poorly in realistic fading conditions. Moreover, most existing approaches [6]–[9] either neglect stochastic channel variation or fail to account for adversarial threats. This leaves a critical gap when deploying such controllers in adversarial and stochastic AI-RAN environments. Recent efforts have explored $\mathcal{H}_\infty$ control in simplified wireless settings. For example, [10] cast the problem as a zero-sum game under constant channel assumptions, while [11] introduces AWGN-based channel models but ignores real-time channel state information (CSI), which is crucial for decision-making in AI-RANs.

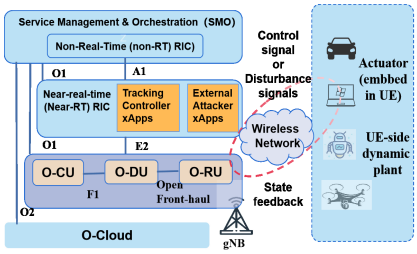


Fig. 1: The architecture of the H-infinity tracking control system over the random MIMO fading channels within AI-RAN.

To address these challenges, this paper investigates robust $\mathcal{H}_\infty$ tracking control over general wireless MIMO fading channels with stochastic dynamics and adversarial interference. The key contributions are as follows: i) We first propose a novel AI-RAN framework for robust control, and then formulate the $\mathcal{H}_\infty$ tracking control problem as a stochastic zero-sum game over the internal plant state, target state, and random channel state. To mitigate the *curse of dimensionality*, we derive a structured reduced-order optimality condition applicable to general random fading channels with an uncountable state space. ii) Using the structured reduced-order optimality condition, we derive a closed-form expression for the Nash equilibrium in the $\mathcal{H}_\infty$ tracking control problem. iii) Utilizing the structured stochastic approximation (SA) framework, we propose an efficient online learning algorithm that asymptotically converges to the Nash equilibrium. The proposed methods provide a principled and scalable approach to robust wireless control in AI-RANs, paving the way for secure and real-time feedback systems over dynamic and adversarial communication environments. The main notations are listed in Table I.

M. Tang is with the Department of Communication Systems, EURECOM, France (email: Minjie.Tang@eurecom.fr); C. Feng and G. Min are with the Department of Computer Science, University of Exeter, U.K. (emails: {c.feng, g.min}@exeter.ac.uk); T. Q. S. Quek is with Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore (email: tonyquek@sutd.edu.sg). *Corresponding author: Chenyuan Feng*.

This work was funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee MSCA postdoctoral fellowships (grant number: EP/Z53433X/1), and in part by the National Natural Science Foundation of China under Grant 62301328.

TABLE I: Main Notations & Definitions

Notation	Definition / Physical Meaning
$\mathbf{x}_k$, $\mathbf{r}_k$	Plant and target state at timeslot k
$\mathbf{u}_{1,k}$	Control input of the tracking controller at timeslot k
$\mathbf{u}_{2,k}$	Control input of the external attacker at timeslot k
$\delta_{i,k}$	Channel access indicator for controller i at timeslot k
$\mathbf{H}_{i,k}$	MIMO fading channel matrix between Controller i and the plant at timeslot k
$\mathbf{w}_k$,$\mathbf{v}_k$	Additive plant and channel noise at timeslot k
$\mathbf{A}, \mathbf{B}$	System matrices of the dynamic plant
$\mathbf{Q}, \mathbf{R}_1, \mathbf{R}_2$ $\mathbf{M}_1, \mathbf{M}_2$	Weighting matrices for tracking error cost, communication cost, and actuation cost
γ, ξ	Non-cooperative penalty coefficient and Discount factor
$\mathbf{S}_k$	Aggregated state at timeslot k
π_1, π_2	Policies of tracking controller and attacker at timeslot k
$\mathbf{P}$	Kernel matrix of the reduced-state value function
α_k	Step size for SA update at timeslot k
α	Correlation coefficient of the channel fading
N_t, N_r	Number of transmit and receive antennas
S	Dimension of plant/target state

II. System Model

As Fig. 1 shows, we reinterpret the classical control loop—comprising a *dynamic plant*, a co-located *actuator*, a remote *tracking controller*, and an *external attacker*—within the modular AI-RAN architecture [1]. The dynamic plant and actuator are co-located on a User Equipment (UE), while the controller and attacker are modeled as intelligent agent-based xApps deployed within the near-real-time RAN Intelligent Controller (Near-RT RIC). Both xApps transmit control and disturbance signals, respectively, through the E2 interface to the RAN node, which then forwards them over the 5G NR wireless links to the UE [4].

A. System Components and AI-RAN Mapping

1) Dynamic Plant (UE-side physical system): The plant represents a real-world physical process, such as an autonomous vehicle, or UAV, hosted on a UE. Its discrete-time dynamics are governed by a linear state-space model:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\hat{\mathbf{u}}_k + \mathbf{w}_k, k = 0, 1, 2, ..., \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^{S\times 1}$ is the plant state, $\hat{\mathbf{u}}_k \in \mathbb{R}^{N_r\times 1}$ is the received noisy control signal at the actuator, $N_r \in \mathbb{Z}_+$ is the number of receiving antennas at the actuator, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}_{S\times 1}, \mathbf{W})$ is the additive plant noise with finite noise covariance matrix $\mathbf{W} \in \mathbb{S}_+^S$. $\mathbf{A} \in \mathbb{R}^{S\times S}$ is the potentially unstable plant dynamics, i.e., $\|\mathbf{A}\| > 1$ and $\mathbf{B} \in \mathbb{R}^{S\times N_r}$ is the actuation matrix. We assume the plant system $(\mathbf{A}, \mathbf{B})$ is controllable.

2) Actuator (embedded in UE): The actuator is co-located with the dynamic plant within the same UE. It executes control signals $\hat{\mathbf{u}}_k$ received from the RAN infrastructure and applies them to the physical system in real time.

3) Remote Tracking Controller (xApp in Near-RT RIC): The controller is deployed as an xApp within the Near-Real-Time RAN Intelligent Controller (Near-RT RIC). It observes plant state feedback and computes control policies based on $\mathcal{H}_\infty$ optimization. These control signals are transmitted to RAN nodes via the standardized E2 interface, and then sent to the target user equipment (UE) over the 5G NR downlink.

4) External Attacker (adversarial xApp): To evaluate robustness, we introduce a virtual attacker, also hosted as an xApp in the RIC. It injects disturbance signals over the air interface to interfere with the control process. This models real-world threats such as adversarial signal injection, policy spoofing, or jamming.

B. Control and Communication Architecture

The control loop spans the RAN and UE layers, as follows: *Step 1:* The Near-RT RIC receives periodic feedback from the UE (e.g., CQI, or plant state estimates) via E2 interface; *Step 2:* The controller computes a robust control action based on the current state and the target state, while the adversarial xApp may generate a disturbance, intended to disrupt the control process; *Step 3:* Both control and disturbance signals are transmitted over the wireless MIMO fading channel to the UE. *Step 4:* The UE-side actuator applies the noisy composite signal to the plant in real time.

Denote the remote tracking controller and the external attacker as Controller 1 and Controller 2, respectively. We model the wireless network connecting the dynamic plant and the remote controllers as an $N_r \times N_t$ MIMO fading channel, where $N_t \in \mathbb{Z}_+$ represents the number of transmission antennas at the remote controllers. At each timeslot, the remote controllers generate the signal $\mathbf{u}_{i,k} \in \mathbb{R}^{N_t\times 1}$ and randomly access the wireless network. The active signal $\mathbf{u}_{i,k}$ is transmitted to the actuator through the wireless communication channels. The received signal $\hat{\mathbf{u}}_k \in \mathbb{R}^{N_r\times 1}$ at the actuator is given by:

$$\hat{\mathbf{u}}_k = \delta_{1,k}\mathbf{H}_{1,k}\mathbf{u}_{1,k} + \delta_{2,k}\mathbf{H}_{2,k}\mathbf{u}_{2,k} + \mathbf{v}_k, \tag{2}$$

where $\mathbf{H}_{i,k} \sim \mathcal{N}(\mathbf{0}_{N_r\times 1}, \mathbf{I}_{N_r})$ is the wireless MIMO fading realization between the dynamic plant and the remote controllers. It remains constant within each timeslot and is independently and identically distributed (i.i.d.) across controllers and timeslots. $\delta_{i,k} \in \{0, 1\}$ is used to model the random access activity of the i-th remote controller. Moreover, $\delta_{i,k}$ is i.i.d. distributed across timeslots and the remote controllers satisfies $\Pr(\delta_{1,k} = 1) = \Pr(\delta_{2,k} = 1) = p \in [0, 1]$. We assume i.i.d. Bernoulli channel access for both the tracking controller and the attacker, justified by its analytical tractability, the independence of their uncoordinated access decisions, and its consistency with randomized scheduling in Medium Access Control (MAC) protocols and energy-efficient access of stealthy attackers.

C. Problem Formulation

In the presence of random wireless channels between the remote controllers and the actuator, the system evolves as a linear time-varying (LTV) system. By substituting Eq. (2) into Eq. (1), the equivalent plant dynamics can be expressed as:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \delta_{1,k}\mathbf{B}\mathbf{H}_{1,k}\mathbf{u}_{1,k} + \delta_{2,k}\mathbf{B}\mathbf{H}_{2,k}\mathbf{u}_{2,k} + \mathbf{B}\mathbf{v}_k + \mathbf{w}_k. \tag{3}$$

Let $\mathbf{r}_k \in \mathbb{R}^{S\times 1}$ denote the prior-known target state, evolving according to $\mathbf{r}_{k+1} = \mathbf{G}\mathbf{r}_k$, where $\mathbf{G} \in \mathbb{R}^{S\times S}$ represents the reference dynamics. The goal of the remote tracking controller is to generate the control action $\mathbf{u}_{1,k}$ so that the real-time state $\mathbf{x}_k$ closely follows the target trajectory $\mathbf{r}_k$. In practice, this objective is hindered by (i) worst-case disturbances $\mathbf{u}_{2,k}$ from an external attacker, and (ii) unreliable MIMO fading channels causing stochastic and intermittent control delivery. We address Challenge 1 via an $\mathcal{H}_\infty$ zero-sum formulation

explicitly modeling the attacker, and Challenge 2 by embedding channel state information (CSI) $\{\delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}\}$ into the control to adapt actions to real-time fading. The tracking control problem is thus a stochastic zero-sum game over $\mathbf{S}_k = \{\mathbf{x}_k, \mathbf{r}_k, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}\}$:

Problem 1 ($\mathcal{H}_\infty$ Tracking Control over MIMO Channels):

$$\min_{\pi_1}\max_{\pi_2} \mathcal{J}^{\pi_1,\pi_2} \qquad s.t.\ \ (3),\ \ \xi \in (0,1), \tag{4}$$

where $\mathcal{J}^{\pi_1,\pi_2} = \limsup_{K\to\infty} \frac{1}{K}\mathbb{E}[\sum_{k=0}^{K-1} \xi^k r(\mathbf{S}_k, \mathbf{u}_{1,k}, \mathbf{u}_{2,k})]$. The control policy $\pi_i : \mathcal{S} \to \mathcal{U}$ maps the aggregated state $\mathbf{S}_k \in \mathcal{S}$ to the control action $\mathbf{u}_{i,k} \in \mathcal{U}$. The per-stage cost function $r(\mathbf{S}_k, \mathbf{u}_{1,k}, \mathbf{u}_{2,k})$ is formulated as:

$$\begin{aligned} r(\mathbf{S}_k, \mathbf{u}_{1,k}, \mathbf{u}_{2,k}) &= (\mathbf{x}_k - \mathbf{r}_k)^T\mathbf{Q}(\mathbf{x}_k - \mathbf{r}_k) + \mathbf{u}_{1,k}^T\mathbf{R}_1\mathbf{u}_{1,k} \\ &- \gamma^2\mathbf{u}_{2,k}^T\mathbf{R}_2\mathbf{u}_{2,k} + \delta_{1,k}(\mathbf{B}\mathbf{H}_{1,k}\mathbf{u}_{1,k})^T\mathbf{M}_1(\mathbf{B}\mathbf{H}_{1,k}\mathbf{u}_{1,k}) - \\ &\gamma^2\delta_{2,k}(\mathbf{B}\mathbf{H}_{2,k}\mathbf{u}_{2,k})^T\mathbf{M}_2(\mathbf{B}\mathbf{H}_{2,k}\mathbf{u}_{2,k}), \end{aligned} \tag{5}$$

where $\mathbf{Q} \in \mathbb{S}_+^S$, $\mathbf{R}_1 \in \mathbb{S}_+^{N_t}$, $\mathbf{R}_2 \in \mathbb{S}_+^{N_t}$, $\mathbf{M}_1 \in \mathbb{S}_+^S$ and $\mathbf{M}_2 \in \mathbb{S}_+^S$ are weighting matrices. The positive constant $\gamma > 0$ serves as a penalty coefficient, accounting for non-cooperation between controllers. $\xi \in (0,1)$ is a discounting factor. In Problem 1, $\mathbb{E}[\cdot]$ is over plant noise $\mathbf{w}_k$, channel noise $\mathbf{v}_k$, access indicators $\delta_{i,k}$, and MIMO fading $\mathbf{H}_{i,k}$, $i \in \{1,2\}$. This expectation is not computed explicitly; instead, optimal policies are learned via stochastic approximation (SA) using real-time $\{\delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}\}$. In Problem 1, the optimization variables are the control policies π_1 and π_2, generating the tracking control $\{\mathbf{u}_{1,k}\}$ and attack $\{\mathbf{u}_{2,k}\}$ sequences, respectively. The objective $\limsup_{K\to\infty} \frac{1}{K}\mathbb{E}[\sum_{k=0}^{K-1} \xi^k r(\mathbf{S}_k, \mathbf{u}_{1,k}, \mathbf{u}_{2,k})]$ is the long-term average cost, where $r(\cdot)$ sums: (i) *tracking error* $(\mathbf{x}_k - \mathbf{r}_k)^\top\mathbf{Q}(\mathbf{x}_k - \mathbf{r}_k)$; (ii) *communication cost* $\mathbf{u}_{1,k}^\top\mathbf{R}_1\mathbf{u}_{1,k} - \gamma^2\mathbf{u}_{2,k}^\top\mathbf{R}_2\mathbf{u}_{2,k}$; and (iii) *actuation cost* $\delta_{1,k}(\mathbf{B}\mathbf{H}_{1,k}\mathbf{u}_{1,k})^\top\mathbf{M}_1(\mathbf{B}\mathbf{H}_{1,k}\mathbf{u}_{1,k}) - \gamma^2\delta_{2,k}(\mathbf{B}\mathbf{H}_{2,k}\mathbf{u}_{2,k})^\top\mathbf{M}_2(\mathbf{B}\mathbf{H}_{2,k}\mathbf{u}_{2,k})$. Here, π_1 minimizes this cost to steer $\mathbf{x}_k$ toward $\mathbf{r}_k$ efficiently, while π_2 maximizes it to drive $\mathbf{x}_k$ away within its own resource limits. The optimization is subject to (3), and $\xi \in (0,1)$ emphasizes near-term performance. The optimal solution to Problem 1 is referred to as the *Nash equilibrium*, defined as follows.

Definition 1 (Nash equilibrium): The control policies of the remote controllers, $\{\pi_1^*, \pi_2^*\}$, constitute the Nash equilibrium of Problem 1 if $\mathcal{J}^{\pi_1^*,\pi_2} \le \mathcal{J}^{\pi_1^*,\pi_2^*} \le \mathcal{J}^{\pi_1,\pi_2^*}, \forall\{\pi_1,\pi_2\}$.

III. Nash Equilibrium in $\mathcal{H}_\infty$ Tracking Control

A. Problem Reformulation

Problem 1 can be equivalently formulated as a virtual ergodic stochastic game, defined as follows: *Problem 2 (Reformulation of Problem 1):*

$$\min_{\pi_1}\max_{\pi_2}\limsup_{K\to\infty}\frac{1}{K}\mathbb{E}[\sum_{k=0}^{K-1}\xi^k\hat{r}(\mathbf{S}_k,\mathbf{u}_{1,k},\mathbf{u}_{2,k})]$$
$$s.t.\ \ \hat{\mathbf{x}}_{k+1} = \hat{\mathbf{A}}\hat{\mathbf{x}}_k + \hat{\mathbf{B}}_{1,k}\mathbf{u}_{1,k} + \hat{\mathbf{B}}_{2,k}\mathbf{u}_{2,k} + \hat{\mathbf{w}}_k, \tag{6}$$

where the equivalent per-stage reward function $\hat{r}(\mathbf{S}_k, \mathbf{u}_{1,k}, \mathbf{u}_{2,k})$ is defined as:

$$\begin{aligned} \hat{r}(\mathbf{S}_k, \mathbf{u}_{1,k}, \mathbf{u}_{2,k}) &= \hat{\mathbf{x}}_k^T\hat{\mathbf{Q}}\hat{\mathbf{x}}_k + \mathbf{u}_{1,k}^T\mathbf{R}_1\mathbf{u}_{1,k} - \gamma^2\mathbf{u}_{2,k}^T\mathbf{R}_2\mathbf{u}_{2,k} \\ &+ \delta_{1,k}(\mathbf{B}\mathbf{H}_{1,k}\mathbf{u}_{1,k})^T\mathbf{M}_1(\mathbf{B}\mathbf{H}_{1,k}\mathbf{u}_{1,k}) - \\ &\gamma^2\delta_{2,k}(\mathbf{B}\mathbf{H}_{2,k}\mathbf{u}_{2,k})^T\mathbf{M}_2(\mathbf{B}\mathbf{H}_{2,k}\mathbf{u}_{2,k}), \end{aligned} \tag{7}$$

where $\hat{\mathbf{x}}_k = [\mathbf{x}_k^T, \mathbf{r}_k^T]^T \in \mathbb{R}^{2S\times 1}$, $\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & -\mathbf{Q} \\ -\mathbf{Q} & \mathbf{Q} \end{bmatrix} \in \mathbb{S}^{2S}$, $\hat{\mathbf{A}} = \mathrm{Diag}(\mathbf{A}, \mathbf{G}) \in \mathbb{R}^{2S\times 2S}$, $\hat{\mathbf{B}}_{i,k} = [\delta_{i,k}\mathbf{H}_{i,k}^T\mathbf{B}^T, \mathbf{0}_{N_t\times S}]^T \in \mathbb{R}^{2S\times N_t}$, and $\hat{\mathbf{w}}_k = [\mathbf{v}_k^T\mathbf{B}^T + \mathbf{w}_k^T, \mathbf{0}_{1\times S}]^T \in \mathbb{R}^{2S\times 1}$. As a result, the Nash equilibrium of Problem 1 can be obtained by solving the equivalent Problem 2.

B. Structured Nash Equilibrium

Conventionally, the Nash equilibrium of Problem 1 can be obtained by solving the coupled ergodic Bellman optimality equation for Problem 2, as stated in the following theorem.

Theorem 1 (Coupled Bellman Optimality Equation for Problem 2): If a Nash equilibrium of Problem 1 exists, it can be determined by solving the coupled Bellman optimality equation:

$$\begin{aligned} \rho + V(\mathbf{S}_k) = &\min_{\mathbf{u}_{1,k}}\max_{\mathbf{u}_{2,k}}[\hat{r}(\mathbf{S}_k,\mathbf{u}_{1,k},\mathbf{u}_{2,k}) \\ &+ \xi\mathbb{E}[V(\mathbf{S}_{k+1})|\mathbf{S}_k,\mathbf{u}_{1,k},\mathbf{u}_{2,k}]], \end{aligned} \tag{8}$$

where $V(\mathbf{S}_k)$ is the optimal value function over the state space $\mathbf{S}_k = \{\mathbf{x}_k, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}\}$. The Nash equilibrium policies $\{\pi_1^*, \pi_2^*\} = \left\{\mathbf{u}_{1,k}^*, \mathbf{u}_{2,k}^*\right\}$ correspond to the minimizer and maximizer of the right-hand side (R.H.S.) of (8), respectively. $\rho > 0$ is a positive constant. Due to space constraints, we omit the detailed proof, which is similar to Chapter 6.7 of [12].

Traditionally, one may consider employing iterative methods such as value iteration or Q-learning to solve the Bellman optimality equations (8). However, these methods suffer from the *curse of dimensionality* due to the continuous state space $\mathbf{S}_k$, which has a total dimension of $2S+2N_tN_r+2$. A brute-force approach necessitates prior knowledge of the optimal value function $V(\mathbf{S}_k)$ or the Q-function $Q(\mathbf{S}_k, \mathbf{u}_{1,k}, \mathbf{u}_{2,k})$, requiring the computation of up to $2S + 2N_tN_r + 2 + 2N_t$ parameters. This results in excessive computational overhead, making learning-based approaches impractical for large values of S, N_t, or N_r. To achieve a low-complexity implementation, we exploit the i.i.d. properties of the CSI, $\{\delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}\}$, in Eq. (8) and derive an equivalent structured reduced-order optimality equation, formulated as follows.

Theorem 2 (Structured Reduced-Order Optimality Equation): If the Nash equilibrium of Problem 1 exists, it can be determined by solving the equivalent structured reduced-order optimality equation, given by:

$$\begin{aligned} \hat{\rho} + \hat{V}(\hat{\mathbf{x}}_k) = &\mathbb{E}[\min_{\mathbf{u}_{1,k}}\max_{\mathbf{u}_{2,k}}[\hat{r}(\mathbf{S}_k,\mathbf{u}_{1,k},\mathbf{u}_{2,k}) \\ &+ \xi\mathbb{E}[\hat{V}(\hat{\mathbf{x}}_{k+1})|\hat{\mathbf{x}}_k,\delta_{1,k}\mathbf{H}_{1,k},\delta_{2,k}\mathbf{H}_{2,k}]]], \end{aligned} \tag{9}$$

where $\hat{V}(\hat{\mathbf{x}}_k) = \hat{\mathbf{x}}_k^T\mathbf{P}\hat{\mathbf{x}}_k$ represents the structured reduced-order value function with a kernel $\mathbf{P} \in \mathbb{S}^{2S}$. The notation $\hat{\rho} = \rho = \mathcal{J}^{\pi_1^*,\pi_2^*} = \mathrm{Tr}(\xi\mathbf{P}_{1:S}\mathbf{W} + \xi\mathbf{B}^T\mathbf{P}_{1:S}\mathbf{B})$, where $\mathbf{P}_{1:S}$ denotes the leading principal submatrix of order S in $\mathbf{P}$. The Nash equilibrium policies $\{\pi_1^*, \pi_2^*\} = \left\{\mathbf{u}_{1,k}^*, \mathbf{u}_{2,k}^*\right\}$ are obtained as the optimizers of Eq. (9), characterized by:

$$\mathbf{u}_{i,k}^* = \mathbf{K}_{i,k}(\mathbf{P})\hat{\mathbf{x}}_k, \tag{10}$$

where the gain matrix $\mathbf{K}_{i,k} \in \mathbb{R}^{N_t\times S}$ is given by:

$$\begin{aligned} \mathbf{K}_{1,k}(\mathbf{P}) = -(&\mathbf{R}_1 + \mathbf{H}_{1,k}^T\mathbf{B}_1^T\mathbf{M}_1\mathbf{B}_1\mathbf{H}_{1,k} \\ &+ \xi\hat{\mathbf{B}}_{1,k}^T\widetilde{\mathbf{P}}_{1,k}\hat{\mathbf{B}}_{1,k})^{-1}\xi\hat{\mathbf{B}}_{1,k}^T\widetilde{\mathbf{P}}_{1,k}\hat{\mathbf{A}}, \end{aligned} \tag{11}$$

$$\mathbf{K}_{2,k}(\mathbf{P}) = (\gamma^2\mathbf{H}_{2,k}^T\mathbf{B}^T\mathbf{M}_2\mathbf{B}\mathbf{H}_{2,k} + \gamma^2\mathbf{R}_2 - \xi\hat{\mathbf{B}}_{2,k}^T\widetilde{\mathbf{P}}_{2,k}\hat{\mathbf{B}}_{2,k})^{-1}\hat{\mathbf{B}}_{2,k}^T\widetilde{\mathbf{P}}_{2,k}\hat{\mathbf{A}}, \quad (12)$$

where $\widetilde{\mathbf{P}}_{1,k} = (\mathbf{P}^{-1} - \gamma^{-2}\hat{\mathbf{B}}_{2,k}(\mathbf{R}_2 + \mathbf{H}_{2,k}^T\mathbf{B}^T\mathbf{M}_2\mathbf{B}\mathbf{H}_{2,k})^{-1}\times \hat{\mathbf{B}}_{2,k}^T)^{-1}$ and $\widetilde{\mathbf{P}}_{2,k} = (\mathbf{P}^{-1} - \gamma^{-2}\hat{\mathbf{B}}_{1,k}(\mathbf{R}_1 + \mathbf{H}_{1,k}^T\mathbf{B}^T\mathbf{M}_1\mathbf{B}\mathbf{H}_{1,k})^{-1}\hat{\mathbf{B}}_{2,k}^T)^{-1}$.

As shown in Theorem 2, $\mathbf{u}_{1,k}^*$ and $\mathbf{u}_{2,k}^*$ scale with the instantaneous CSI $\{\delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}\}$ via the gains $\mathbf{K}_{1,k}(\mathbf{P})$ and $\mathbf{K}_{2,k}(\mathbf{P})$. A better channel for the controller (larger $\|\delta_{1,k}\mathbf{H}_{1,k}\|$) increases $\|\mathbf{K}_{1,k}\|$, prompting stronger control (larger $\|\mathbf{u}_{1,k}\|$) to exploit the favorable link, while a better channel for the attacker (larger $\|\delta_{2,k}\mathbf{H}_{2,k}\|$) increases $\|\mathbf{K}_{2,k}\|$, enabling more disruptive injection. Moreover, each side anticipates that the other becomes more effective when its own channel is good, and thus further amplifies its action in such cases. This bidirectional adaptation to both links embodies the non-cooperative game nature of the problem.

Unlike solving the Bellman optimality equations (8) by learning the optimal value function $V(\mathbf{S}_k)$, which involves infinitely many unknowns, the equivalent reduced-order Bellman optimality equations (9) require learning only the structured reduced-order value function $\hat{V}(\hat{\mathbf{x}}_k)$ with a single unknown, $\mathbf{P}$. This effectively mitigates the *curse of dimensionality* caused by the uncountable space of the $\mathbf{S}_k$, as learning a single unknown for the Nash equilibrium is computationally feasible. In the following section, we develop an online learning algorithm to compute the Nash equilibrium of Problem 1 by learning the structured kernel $\mathbf{P}$ for the reduced-order value function $\hat{V}(\hat{\mathbf{x}}_k)$.

IV. Online Learning Algorithm for Nash Equilibrium

Leveraging the structured form of the reduced-order value function $\hat{V}(\hat{\mathbf{x}}_k)$, the Nash equilibrium policies $\{\pi_1^*, \pi_2^*\}$, and the bias term $\hat{\rho}$ from Theorem 2, the reduced-order optimality equation (9) can be reformulated as a coupled nonlinear matrix equation, given by:

$$\mathbf{P} = \mathbb{E}[g(\mathbf{P}, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k})], \quad (13)$$

where the nonlinear operator $g(\mathbf{P}, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k})$ is defined as follows:

$$g(\mathbf{P}, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}) = \hat{\mathbf{Q}} + \xi\hat{\mathbf{A}}^T\mathbf{P}\hat{\mathbf{A}} - \xi^2\hat{\mathbf{A}}^T \times \begin{bmatrix}\hat{\mathbf{B}}_{1,k}^T\mathbf{P} \\ \hat{\mathbf{B}}_{2,k}^T\mathbf{P}\end{bmatrix}^T \begin{bmatrix}\mathcal{N}_{11,k} & \mathcal{N}_{12,k} \\ \mathcal{N}_{21,k} & \mathcal{N}_{22,k}\end{bmatrix}^{-1} \begin{bmatrix}\hat{\mathbf{B}}_{1,k}^T\mathbf{P} \\ \hat{\mathbf{B}}_{2,k}^T\mathbf{P}\end{bmatrix} \hat{\mathbf{A}}, \quad (14)$$

and

$$\mathcal{N}_{11,k} = \mathbf{R}_1 + \delta_{1,k}\mathbf{H}_{1,k}^T\mathbf{B}^T\mathbf{M}_1\mathbf{B}\mathbf{H}_{1,k} + \xi\hat{\mathbf{B}}_{1,k}^T\mathbf{P}\hat{\mathbf{B}}_{1,k}, \quad (15)$$

$$\mathcal{N}_{12,k} = \xi\hat{\mathbf{B}}_{1,k}^T\mathbf{P}\hat{\mathbf{B}}_{2,k}, \quad (16)$$

$$\mathcal{N}_{21,k} = \mathcal{M}_{12,k}^T, \quad (17)$$

$$\mathcal{N}_{22,k} = \xi\hat{\mathbf{B}}_{2,k}^T\mathbf{P}\hat{\mathbf{B}}_{2,k} - \gamma^2\delta_{2,k}\mathbf{H}_{2,k}^T\mathbf{B}^T\mathbf{M}_2\mathbf{B}\mathbf{H}_{2,k} - \gamma^2\mathbf{R}_2. \quad (18)$$

Since Eq. (13) is a fixed-point equation with respect to (w.r.t.) the unknown variable $\mathbf{P}$, we can leverage SA theory [13] to develop an online learning algorithm for estimating $\mathbf{P}$ based on Eq. (13). The learned unknown variable $\mathbf{P}$ is then used to derive the optimal reduced-order value function $\hat{V}(\hat{\mathbf{x}}_k)$ and the optimal control solution $\mathbf{u}_{i,k}$, enabling computation of the Nash equilibrium for Problem 1.

Specifically, Eq. (9) can be rewritten in the standard form $f(\mathbf{P}) = \mathbf{0}_S$, where $f(\mathbf{P})$ is expressed as follows:

$$f(\mathbf{P}) = \mathbb{E}[g(\mathbf{P}, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k})] - \mathbf{P}. \quad (19)$$

To find the root of $f(\mathbf{P}) = \mathbf{0}_S$, we employ the SA algorithm outlined in Algorithm 1. Specifically, the estimated root $\mathbf{P}_k$ at each k-th timeslot is updated as follows:

$$\mathbf{P}_k = \mathbf{P}_{k-1} + \alpha_k(g(\mathbf{P}_{k-1}, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}) - \mathbf{P}_k), \quad (20)$$

Here, $\{\alpha_k\}$ denotes the step-size sequence, satisfying $\sum_{k=0}^{\infty}\alpha_k = \infty$ and $\sum_{k=0}^{\infty}\alpha_k^2 < \infty$. The term $g(\mathbf{P}_{k-1}, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k})$ serves as an unbiased estimator of $\mathbb{E}[g(\mathbf{P}, \delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k})]$ in Eq. (19).

Algorithm 1 Online Learning for Nash Equilibrium

Initialization: Set $\mathbf{P}_{-1} = [\mathbf{P}_{\text{init}}, -\mathbf{P}_{\text{init}}; -\mathbf{P}_{\text{init}}, \mathbf{P}_{\text{init}}] \in \mathbb{S}^{2S}$ for $\hat{V}_{-1}(\hat{\mathbf{x}}) = \hat{\mathbf{x}}^T\mathbf{P}_{-1}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} \in \mathbb{R}^{2S\times 1}$ and $\mathbf{P}_{-1} \in \mathbb{S}_+^S$. Initialize the plant state $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}_{S\times 1}, \mathbf{I}_S)$.

For $k = 0, 1, ...$**:**

- **Step 1: (Update at Tracking Controller)**
  - $\mathbf{P}_k \leftarrow$ based on Eq. (20);
  - $\mathbf{u}_{1,k} \leftarrow \mathbf{K}_{1,k}(\mathbf{P}_k)\hat{\mathbf{x}}_k$;
  - $V_k(\hat{\mathbf{x}}) \leftarrow \hat{\mathbf{x}}^T\mathbf{P}_k\hat{\mathbf{x}}$.
- **Step 2: (Update at Attacker)**
  - $\mathbf{P}_k \leftarrow$ based on Eq. (20);
  - $\mathbf{u}_{2,k} \leftarrow \mathbf{K}_{2,k}(\mathbf{P}_k)\hat{\mathbf{x}}_k$;
  - $V_k(\hat{\mathbf{x}}) \leftarrow \hat{\mathbf{x}}^T\mathbf{P}_k\hat{\mathbf{x}}$.

End

The per-step complexity of Algorithm 1 is dominated by (i) the SA update of the kernel matrix $\mathbf{P}_k$ and (ii) the computation of the control gains $\mathbf{K}_{i,k}(\cdot)$, involving matrix multiplications of size $2S \times 2S$ or $2S \times N_t$ and inversions of $2S \times 2S$ and $N_t \times N_t$ matrices. Thus, the cost scales as $\mathcal{O}(S^3 + S^2N_t + N_t^3)$. In memory, only a single kernel $\mathbf{P}_k$ is stored, independent of the CSI dimension. By contrast, conventional value iteration for Theorem 1's Bellman equation must store and update $V(\mathbf{S}_k)$ over the full CSI space. If each channel coefficient $[\mathbf{H}_{i,k}]_{m,n}$, $i \in \{1, 2\}$, $m = 1, \ldots, N_r$, $n = 1, \ldots, N_t$, is quantized into q levels, the CSI state space has size $q^{2N_rN_t}$. Consequently, the computational and memory complexities scale as $\mathcal{O}(q^{2N_rN_t})$, i.e., exponentially with the antenna numbers (N_r, N_t). Leveraging the reduced-state formulation and closed-form SA updates removes this exponential dependence, ensuring scalability w.r.t. the antenna dimension.

Remark 1 (Online Learning for Non-Cooperative Players): Algorithm 1 provides an online procedure to learn the Nash equilibrium of Problem 1, yielding the optimal tracking control policy $\pi_1^* = \{\mathbf{u}_{1,k}^*\}$ and the worst-case disturbance policy $\pi_2^* = \{\mathbf{u}_{2,k}^*\}$. It supports two deployment modes: (i) *single-agent*, where only the tracking controller (Step 1) or only the attacker (Step 2) learns its own policy; and (ii) *simultaneous*, where both agents run their respective steps in parallel. In the simultaneous mode, each agent i updates its kernel $\mathbf{P}_k$ via SA and computes its action $\mathbf{u}_{i,k}$ from the aggregated state $\mathbf{S}_k$ (plant state and real-time CSI) fed back by the plant. The

agents observe the same $\mathbf{S}_k$ but operate independently, without sharing actions or policies.

Remark 2 (CSI Requirement): Note that Step 1 and Step 2 of Algorithm requires the CSI $\{\delta_{1,k}\mathbf{H}_{1,k}, \delta_{2,k}\mathbf{H}_{2,k}\}$. This can be acquired through standard channel estimation at the dynamic plant, based on pilot signals transmitted by the remote controllers, followed by channel feedback to them [14].

We conclude this section with a theorem on the convergence of Algorithm 1, stated as follows.

Theorem 3 (Convergence of Algorithm 1): Under the step-size condition $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, the control policy $\mathbf{u}_{i,k}$ in Algorithm 1 converges almost surely to the optimal solution of Problem 1. That is, $\Pr(\limsup_{k\to\infty} \mathbf{u}_{i,k} = \mathbf{u}_{i,k}^*) = 1$, where $\mathbf{u}_{i,k}^*$ is the optimal solution given in Theorem 2.

Proof: Note that the evolution of Eq. (20) w.r.t. $\mathbf{P}$ can be approximated by the ordinary differential equation (ODE) trajectory $\dot{\mathbf{P}} = f(\mathbf{P})$. The almost-sure convergence of Algorithm 1 is ensured by proving the convergence of $\dot{\mathbf{P}} = f(\mathbf{P})$, as detailed in Chapter 2 of [13]. Due to space constraints, we omit the detailed proof. ■

V. Numerical Results

A. Experiment Setup & Baselines

We consider a dynamic plant with 12 states, i.e., $\mathbf{x}_k \in \mathbb{R}^{12\times 1}$. The system dynamics $\mathbf{A} \in \mathbb{R}^{12\times 12}$ is randomly generated, with each element independently drawn from a Gaussian distribution with zero mean and unit variance. The matrices are set as $\mathbf{B} = \mathbf{G} = \mathbf{W} = \mathbf{R}_i = \mathbf{M}_1 = \mathbf{M}_2 = \mathbf{Q} = \mathbf{I}_{12}$. $p = \xi = 0.7$. $\gamma = 10$. $\mathbf{r}_0 = 50\mathbf{I}_{12\times 1} \in \mathbb{R}^{12\times 1}$, while $\mathbf{x}_0 \sim \mathcal{N}(10\mathbf{I}_{12\times 1}, 10\mathbf{I}_{12})$ is randomly generated. $\mathbf{I}_{12\times 1}$ is a 12×1 vector with all elements to be 1.

- **Baseline 1** *(Prior-Known Nash Equilibrium):* The optimal solution $\mathbf{u}_{1,k}^*$ from Theorem 2 is known and implemented at the remote tracking controller.
- **Baseline 2** *(PID-based Tracking Control):* The control input is given by $\mathbf{u}_{1,k} = \mathbf{K}_p(\mathbf{x}_k - \mathbf{r}_k) + \mathbf{K}_i \sum_{t=0}^{k}(\mathbf{x}_t - \mathbf{r}_t) + \mathbf{K}_d\left[(\mathbf{x}_k - \mathbf{r}_k) - (\mathbf{x}_{k-1} - \mathbf{r}_{k-1})\right]$, where the gain matrices $\mathbf{K}_p, \mathbf{K}_i, \mathbf{K}_d \in \mathbb{R}^{12\times 12}$ are tuned offline using pole placement, and $\mathbf{x}_{-1} = \mathbf{r}_{-1} = \mathbf{0}_{12\times 1}$.
- **Baseline 3** *(LQT-based Tracking Control over Fading Channels):* $\mathbf{u}_{1,k} = \mathbf{K}_k(\mathbf{x}_k - \mathbf{r}_k)$, where the time-varying gain $\mathbf{K}_k \in \mathbb{R}^{12\times 12}$ is obtained by solving the Riccati equation over the fading channels in an online manner.
- **Baseline 4** *($\mathcal{H}_\infty$ Tracking Control by Brute-force Value Iteration):* $\mathbf{u}_{1,k}$ is obtained by solving the Bellman equation in (8) using a brute-force approach.
- **Baseline 5** *(Distributionally Robust Reinforcement Learning (RL)):* π_1 is trained offline under parameter uncertainty (controller channel access probability and plant noise covariance) using the EPOpt-ε framework. The policy is updated via Soft Actor-Critic (SAC) using only the worst $\lfloor \varepsilon N \rfloor$ trajectories. In simulations, we set $N = 100$, $\varepsilon = 0.3$, and training horizon $K = 1000$. The policy/value networks have three fully connected layers $[128, 128, 64]$ with ReLU activations.
- **Baseline 6** *(Adversarial RL):* π_1 is trained offline together with an auxiliary disturbance generator in a minimax game using Proximal Policy Optimization (PPO) with Gaussian policies. The training horizon and the policy/value network architecture are the same as in Baseline 5.

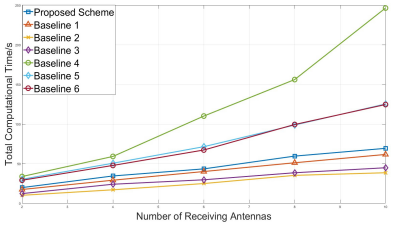


Fig. 2: CPU runtime vs. number of receiving antennas in Scenario 1.

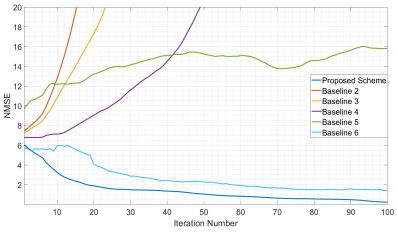


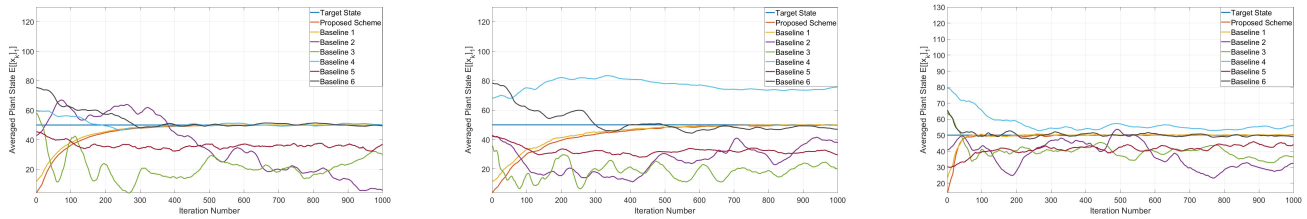
Fig. 3: Convergence behavior in Scenario 1.

The considered communication models and corresponding attacker actions $\mathbf{u}_{2,k}$ are summarized as follows.

- **Scenario 1** *(Default):* The wireless fading gain $\mathbf{H}_{i,k} \sim \mathcal{N}(\mathbf{0}_{12\times 12}, \mathbf{I}_{12})$ is i.i.d. across controllers $i \in \{1, 2\}$ and timeslots k. The attacker applies the worst-case disturbance in Theorem 2.
- **Scenario 2** *(Markovian Channels):* $\mathbf{H}_{i,k}$ is i.i.d. across controllers but evolves over time via $\mathbf{H}_{i,k+1} = \alpha\mathbf{H}_{i,k} + \sqrt{1-\alpha^2}\,\mathbf{V}_{i,k}$, where $\alpha \in [0, 1]$ is the correlation coefficient set to be 0.3 in the simulation. $\mathbf{V}_{i,k} \sim \mathcal{N}(\mathbf{0}_{12\times 12}, \mathbf{I}_{12})$. The initial state $\mathbf{H}_{i,0} \sim \mathcal{N}(\mathbf{0}_{12\times 12}, \mathbf{I}_{12})$. The attacker uses the same worst-case disturbance as in Scenario 1.
- **Scenario 3** *(Spoofing Attacker):* Same channel model as Scenario 1. At each timeslot, the attacker estimates $\mathbf{u}_{1,k}^*$ via Algorithm 1, denoted $\bar{\mathbf{u}}_{1,k}$, and sets $\mathbf{u}_{2,k} = \bar{\mathbf{u}}_{1,k} + \Delta_k$, where $\Delta_k = 10\sin(6k)\,\phi_k$, and $\phi_k \sim \mathcal{N}(\mathbf{0}_{12\times 1}, \mathbf{I}_{12})$.

B. Performance Comparison & Analysis

1) CPU Computational Time v.s. Number of Receiving Antennas: We adopt CPU execution time as the primary efficiency metric, as it reflects scalability w.r.t. antenna dimension and enables fair comparison with existing methods. By solving a structured reduced-state optimality equation over the plant state only, the proposed scheme avoids high-dimensional CSI dependence and achieves much lower CPU time than value-iteration- and RL-based algorithms (Baselines 4–6), which require optimization over a black-box Bellman equation (Fig. 2). It also stores only the reduced-order kernel matrix $\mathbf{P}_k$, making memory usage independent of antenna dimension and suitable for resource-constrained RIC xApps. Reduced memory access, together with CPU time savings, further implies lower overall energy consumption than computationally intensive Baselines 4–6. While Baselines 2 and 3 have slightly lower runtime by using overly simplified control algorithms



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 4: Tracking performance in different scenarios.

that ignore both the attacker and channel randomness, they suffer poor tracking performance in dynamic and adversarial environments. Baseline 1 has lower complexity as it assumes prior knowledge of the Nash equilibrium, whereas our method learns it online via SA.

2) NMSE between Applied and Optimal Control Actions: Fig. 3 plots the normalized MSE $\frac{\mathbb{E}[\|\mathbf{u}_{1,k}-\mathbf{u}_{1,k}^*\|^2]}{\mathbb{E}[\|\mathbf{u}_{1,k}^*\|^2]}$ versus iteration number. In all cases, the iteration number corresponds to online execution timeslots for all schemes. The expectation is approximated by averaging over 200 runs with randomly initialized seeds. For RL-based baselines (Baselines 5 and 6), it denotes offline training iterations of the policy/value network before deployment. The proposed scheme converges asymptotically to the optimal solution, while the baselines deviate due to ignoring CSI (Baseline 2), neglecting disturbances (Baseline 3), or suffering from dimensionality issues (Baseline 4). Baseline 5 (EPOpt-ϵ) inherently departs from the equilibrium, and Baseline 6 (adversarial RL) converges more slowly owing to deep network training, whereas our method exploits structured closed-form updates.

3) Plant State vs. Iteration Number: Figs. 4 plot the averaged plant state $\mathbb{E}[[\mathbf{x}_k]_1]$ under three scenarios. In all cases, the iteration number corresponds to online execution timeslots for all schemes. The expectation is approximated by averaging over 200 runs with randomly initialized seeds. In Scenario 1, the proposed scheme, Baseline 1, and Baseline 6 track the target, while Baselines 2–4 fail due to ignoring CSI, neglecting the attacker, or suffering from dimensionality. Among the convergent schemes, the proposed method is faster and more stable than Baseline 6, which is affected by approximation errors and variance from finite-time RL training, while Baseline 1 is slightly faster thanks to prior knowledge of the Nash equilibrium. In Scenario 2, the relative performance is unchanged as convergent schemes adapt to CSI and non-convergent ones remain ineffective. In Scenario 3, all schemes improve under weaker attacks, while their relative convergence behavior remains unchanged.

VI. Conclusion

This work investigated $\mathcal{H}_\infty$ tracking control for linear systems over random wireless MIMO fading channels within the AI-RAN framework. We first formulated the problem as a stochastic zero-sum game over the plant state, channel state, and target state. By leveraging the structure of the reduced-order optimality condition, we developed a structured online learning algorithm that asymptotically attains the Nash equilibrium of the stochastic game using SA. Numerical results validate the superiority of the proposed scheme over baseline methods in terms of tracking performance, computational efficiency, and convergence behavior. Future work includes extending the framework to handle temporally correlated or adaptive attacks (e.g., correlated jamming, channel access spoofing), integrating precoding and detection to enhance control signal fidelity under severe channels or stringent accuracy requirements, and incorporating practical uncertainties such as E2 interface delays, CSI estimation errors, and multi-user contention as bounded or stochastic variations. These extensions will be validated through hardware-in-the-loop experiments on real RIC platforms to assess memory, energy efficiency, and overall system performance under realistic conditions.

References

[1] B. Brik, H. Chergui, L. Zanzi, F. Devoti, A. Ksentini, M. S. Siddiqui, X. Costa-Pèrez, and C. Verikoukis, "Explainable AI in 6G O-RAN: A tutorial and survey on architecture, use cases, challenges, and future research," *IEEE Commun. Surv. Tut.*, pp. 1–1, 2024.

[2] AI-RAN Alliance, "AI-RAN alliance vision and mission white paper," AI-RAN Alliance, White Paper, Dec. 2024.

[3] J. Groen *et al.*, "Implementing and evaluating security in O-RAN: Interfaces, intelligence, and platforms," *IEEE Network*, vol. 39, no. 1, pp. 227–234, 2025.

[4] O-RAN Alliance, "O-RAN E2 general aspects and principles (E2GAP), version v4.01," O-RAN Alliance, Technical Specification, 2023, version v4.01.

[5] A. S. Kolarijani, S. C. Bregman, P. M. Esfahani, and T. Keviczky, "A decentralized event-based approach for robust model predictive control," *IEEE Trans. Autom. Control*, vol. 65, no. 8, pp. 3517–3529, 2020.

[6] M. Pezzutto *et al.*, "Remote mpc for tracking over lossy networks," *IEEE Control Systems Lett.*, vol. 6, pp. 1040–1045, 2021.

[7] G.-P. Liu, "Tracking control of multi-agent systems using a networked predictive PID tracking scheme," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 1, pp. 216–225, 2023.

[8] R. Moghadam and F. L. Lewis, "Output-feedback h_∞ quadratic tracking control of linear systems using reinforcement learning," *Int. J. Adaptive Control Signal Process.*, vol. 33, no. 2, pp. 300–314, 2019.

[9] M. F. Miranda and K. G. Vamvoudakis, "Online optimal auto-tuning of PID controllers for tracking in a special class of linear systems," in *American Control Conf. (ACC)*, 2016, pp. 5443–5448.

[10] B. Farzanegan and S. Jagannathan, "Continual reinforcement learning formulation for zero-sum game-based constrained optimal tracking," *IEEE Trans. Systems Man Cybern.: Systems*, vol. 53, no. 12, pp. 7744–7757, 2023.

[11] S. Nakamori, "H-infinity tracking controller for linear discrete-time stochastic systems with uncertainties," *WSEAS Trans. Circuits Systems*, vol. 21, pp. 238–248, 2022.

[12] D. P. Bertsekas *et al.*, "Dynamic programming and optimal control 3rd edition, volume II," *Belmont, MA: Athena Scientific*, 2011.

[13] V. S. Borkar and V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint, the 9th edition*. Springer, 2008.

[14] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.