Experimental Results in Cyber-Physical Transportation Systems: A Case Study in Cybersecurity

Won Yong Ha, Sayan Chakraborty, Kaan Ozbay, and Zhong-Ping Jiang

Presented at the 2025 IEEE IV Conference

Experimental Results in Cyber-Physical Transportation Systems: A Case Study in Cybersecurity

Won Yong Ha
Control and Networks Lab
Dept of Electrical Engineering, NYU
Brooklyn, NY 11201, U.S.A.
wh784@nyu.edu

Sayan Chakraborty
Control and Networks Lab
Dept of Electrical Engineering, NYU
Brooklyn, NY 11201, U.S.A.
sc8804@nyu.edu

Kaan Ozbay
C2SMARTER Center
Dept of Civil and Urban Engineering, NYU
Brooklyn, NY 11201, U.S.A.
kaan.ozbay@nyu.edu

Zhong-Ping Jiang
Control and Networks Lab
Dept of Electrical Engineering, NYU
Brooklyn, NY 11201, U.S.A.
zjiang@nyu.edu

***Abstract*—This paper presents experimental results from a learning-based control framework for cyber-physical transportation systems. Building on theoretical guarantees that establish an upper bound on denial-of-service (DoS) attack durations to maintain closed-loop stability, we deploy a resilient learning-based lane-changing control algorithm on a remote-controlled (RC) autonomous vehicle equipped with GPS, IMU, and camera sensors, interfaced with an Nvidia Jetson AGX Xavier board. The algorithm leverages real-time sensor data to make suboptimal yet robust lane-change decisions while enduring intermittent DoS attacks that disrupt communication. Our experiments confirm the resilience of this learning-based approach, demonstrating safe and efficient maneuvers under adversarial conditions in obstacle-rich driving scenarios. By highlighting these experimental findings, this work underscores the importance of cybersecurity in next-generation vehicle control algorithms for autonomous transportation applications.**

***Index Terms*—Learning-based Control, resilient reinforcement learning, Perception-in-the-loop**

I. Introduction

The rapid advancements in communication and networking technologies have greatly influenced cyber-physical systems (CPSs), which now serve as a cornerstone of modern autonomous driving [1], [2]. However, this increased connectivity exposes CPSs to more frequent and sophisticated cyber-physical attacks. Among these, DoS attacks pose a critical threat to control system stability by disrupting the flow of essential information, often resulting in severe operational failures [3], [4]. The urgency to develop robust and resilient control mechanisms is particularly pronounced in autonomous driving, where safe operation hinges on the real-time exchange of data between vehicles and infrastructure [5]. Recently, several researchers have proposed various resilient control strategies to mitigate these risks, but many of these approaches assume complete knowledge of system dynamics [6], [7].

Reinforcement learning (RL) offers a promising alternative in scenarios where the system dynamics are unknown. RL-based approaches enable agents to learn optimal control policies in complex and dynamic environments by leveraging real-time data. While RL has been applied to problems of stabilization and output regulation in control systems [8], [9], its application in the presence of DoS attacks is still an emerging area of research. Recently, the authors of [10] introduced resilient RL frameworks, showcasing the potential of this approach.

Lane-changing in autonomous driving presents a complex challenge, where vehicles must make real-time decisions based on sensor data from surrounding vehicles and road conditions [11], [12]. Recent research has focused on combining learning-based methods with trajectory planning and vehicle-to-infrastructure (V2I) communication for safe lane-changing maneuvers [13]. Despite these advancements, the robustness of these systems under DoS attacks remains underexplored.

Our study builds on the RL-based control approaches to address these vulnerabilities by incorporating learning-based lane-changing strategies for autonomous vehicles in the presence of DoS attacks. Unlike previous works, such as [14], which focus primarily on lane-changing without considering cyber-physical threats, our work integrates real-time sensor data to learn an optimal lane-changing policy under DoS attacks. It achieves an upper bound on the DoS duration that guarantees closed-loop stability under DoS attacks.

The experimental study of DoS attacks on vehicle platooning systems by the authors of [15] emphasizes the impact of such attacks on connected vehicle systems. In their work, platoons are subjected to DoS and jamming attacks, revealing the significant risks associated with network disruptions in intelligent transportation systems. While their approach focuses on platooning, our approach broadens the scope by addressing the more dynamic and complex challenge of learning-based lane-changing control for autonomous vehicles. Furthermore, we leverage RL to enable a data-driven control mechanism capable of learning from real-time environmental interactions without relying on a system model while explicitly addressing the impact of DoS attacks. To our knowledge, this is the first time the challenge of DoS attacks in learning-based control for experimental study in lane-changing maneuvers has been addressed in the literature. Unlike our previous theoretical work [16], which focused on resilience strategies in control systems, this paper emphasizes ensuring operational safety during lane-changing maneuvers in the presence of DoS attacks. Our approach integrates advanced perception technologies like GPS and cameras with adaptive dynamic programming to develop a resilient control framework. This framework makes real-time decisions to ensure safe and stable lane-changing maneuvers, even under adversarial conditions like DoS attacks. Our experiments demonstrate that the learned RL-based controller effectively ensures safe lane-changing operations, even amidst moderate DoS attacks. However, we also observed that when subjected to more severe or prolonged DoS attacks, the system loses its ability to maintain safe operations, challenging us to develop improved vehicle control methods. Findings along this line of research will be reported in a companion paper.

The rest of this paper is structured as follows: Section II describes the learning-based control design. Section III presents our experimental results and analysis of the developed control algorithm for automated lane change under DoS attacks. Finally, Section IV provides the concluding remarks.

Facts and notations: $\mathbb{R}_+$ denotes the set of non-negative real numbers. $\mathbb{Z}_+$ the set of non-negative integers. $|x|$ denotes the Euclidean norm of a vector $x \in \mathbb{R}^n$. $|A|$ denotes the induced matrix norm for a matrix $A \in \mathbb{R}^{m\times n}$. For a square matrix A, $\sigma(A)$ denotes the spectrum of A. For a real symmetric matrix A, $\lambda_m(A)$ and $\lambda_M(A)$ denote the minimum and maximum eigenvalues of A, respectively. $\otimes$ indicates the Kronecker product, $\mathrm{vec}(T) = \left[t_1^{\mathrm{T}}, t_2^{\mathrm{T}}, \cdots, t_m^{\mathrm{T}}\right]^{\mathrm{T}}$ with $t_i \in \mathbb{R}^r$ being the columns of $T \in \mathbb{R}^{r\times m}$. For a symmetric matrix $P \in \mathbb{R}^{m\times m}$, $\mathrm{vecs}(P) = [p_{11}, 2p_{12}, \cdots, 2p_{1m}, p_{22}, 2p_{23}, \cdots, 2p_{(m-1)m}, p_{mm}]^{\mathrm{T}} \in \mathbb{R}^{(1/2)m(m+1)}$, for a column vector $v \in \mathbb{R}^n$, $\mathrm{vecv}(v) = [v_1^2, v_1v_2, \cdots, v_1v_n, v_2^2, v_2v_3, \cdots, v_{n-1}v_n, v_n^2]^{\mathrm{T}} \in \mathbb{R}^{(1/2)n(n+1)}$. For any two sequence of vectors $\{a_i\}_{i=k_0}^{k_s}$, $\{b_i\}_{i=k_0}^{k_s}$, define $\Xi_a = \left[\mathrm{vecv}(a_{k_0+1}) - \mathrm{vecv}(a_{k_0}), \cdots, \mathrm{vecv}(a_{k_s}) - \mathrm{vecv}(a_{k_s-1})\right]^{\mathrm{T}}$, $J_{a,b} = \left[a_{k_0} \otimes b_{k_0}, \cdots, a_{k_s} \otimes b_{k_s}\right]^{\mathrm{T}}$, $J_a = \left[\mathrm{vecv}(a_{k_0}), \cdots, \mathrm{vecv}(a_{k_s})\right]^{\mathrm{T}}$. I_n and 0_n is the identity matrix and the zero matrix of dimension $n \times n$, respectively.

II. Learning-Based Controller Design

A. Preliminary results

Consider the following discrete-time linear system:

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

where $k \in \mathbb{Z}_+$, $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}$ is the control input, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^n$ are unknown matrices. It is assumed that the system is stabilizable. To minimize state deviations and control effort, our goal is to develop a linear optimal control law of the form

$$u_k = -Kx_k, \tag{2}$$

that minimizes the following cost function:

$$\min_u \quad J = \sum_{k=0}^{\infty} (x_k^{\mathrm{T}} Q x_k + u_k^2), \tag{3}$$

where $Q = Q^{\mathrm{T}} \succ 0$, and $(A, \sqrt{Q})$ is observable. If A and B are completely known, the solution to this problem can be found by solving the following discrete-time algebraic Riccati equation

$$A^{\mathrm{T}}PA - P + Q - A^{\mathrm{T}}PB(1 + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA = 0. \tag{4}$$

By the assumptions mentioned above, (4) has a unique solution $P^* = P^{*T} \succ 0$, and the optimal feedback gain K^* can be found as follows

$$K^* = (1 + B^{\mathrm{T}}P^*B)^{-1}B^{\mathrm{T}}P^*A. \tag{5}$$

It is important to highlight that (4) is nonlinear to P, which complicates the direct computation of P, particularly in high-dimensional systems. Following [17], this work implements policy iteration technique to iteratively compute the solution of the Riccati equation by using policy evaluation

$$A_j^{\mathrm{T}} P_j A_j - P_j + Q + K_j^{\mathrm{T}} K_j = 0, \tag{6}$$

and policy improvement

$$K_{j+1} = (1 + B^{\mathrm{T}}P_jB)^{-1}B^{\mathrm{T}}P_jA, \tag{7}$$

where $A_j = A - BK_j$.

B. Resilience analysis under DoS Attacks

Data transmission and reception are interrupted during the onset of DoS attacks. Let $\{h_m\}_{m\in\mathbb{Z}+}$ represent the sequence of off/on transitions corresponding to DoS events, where $h_0 \geq 0$. The m^{th} DoS attack interval, with duration τ_m, is defined as $\mathscr{I}_m := [h_m, h_m + \tau_m)$. For any interval $[k_1, k_2]$, let $\Lambda_D(k_1,k_2) := \bigcup_{m\in\mathbb{Z}_+} \mathscr{I}_m \bigcap [k_1,k_2]$ and $\Lambda_N(k_1,k_2) := [k_1,k_2] \setminus \Lambda_D(k_1,k_2)$ denote the sets of time instants during which communication is denied and permitted, respectively. The following assumptions are necessary for DoS frequency and DoS duration.

Assumption II.1. *(DoS Frequency) There exist $\eta > 1$ and $\tau_D > 0$ such that $\forall k_2 > k_1 \geq 0$,*

$$n(k_1,k_2) \leq \eta + \frac{k_2 - k_1}{\tau_D}, \tag{8}$$

where $n(k_1,k_2)$ denotes the number of DoS off/on transitions occurring on the interval $[k_1,k_2]$. □

Assumption II.2. *(DoS Duration) There exist $T > 1$ and $\kappa > 0$ such that $\forall k_2 > k_1 \geq 0$,*

$$|\Lambda_D(k_1,k_2)| \leq \kappa + \frac{k_2 - k_1}{T}, \tag{9}$$

where $|\Lambda_D(k_1, k_2)|$ denotes the Lebesgue measure of the set $\Lambda_D(k_1, k_2)$. □

When the system is under DoS attack, the control input can be expressed as

$$u_k = -K^\star x_{k_{m(k)}}, \quad \forall k \in \mathbb{Z}_+ \tag{10}$$

where $k_{m(k)}$ represents the most recent instant when the updated information is received. Let $\varepsilon_k = x_{k_{m(k)}} - x_k$ be the error values between the last successfully received values and actual values. Using the optimal controller (10), the following closed-loop system is obtained

$$x_{k+1} = (A - BK^\star)x_k - BK^\star\varepsilon_k. \tag{11}$$

In this work, we seek to give a lower bound on the DoS duration parameter T so that the closed-loop system is globally asymptotically stable at the origin under DoS attacks.

Theorem II.1. *The system described in* (11) *is globally asymptotically stable at the origin if the following condition on the DoS duration parameter T holds*

$$T > 1 + \frac{log(1+\omega_2)}{-log(1-\omega_1)} := T^\star, \tag{12}$$

where

$$\omega_1 = \frac{\lambda_m(Q)}{\lambda_M(P^\star)}, \omega_2 = \frac{\alpha_1 + 4\alpha_2}{\lambda_m(P^\star)}, \alpha_2 = \frac{3}{4} + 3|K^{\star\mathrm{T}}B^{\mathrm{T}}P^\star BK^\star|^2$$
$$\alpha_1 = \frac{1}{2} + |K^{\star\mathrm{T}}B^{\mathrm{T}}P^\star BK^\star|^2 + |A^{\mathrm{T}}P^\star A|^2.$$

Proof. see [16]. □

C. Data-driven formulation

Since A and B are unknown, we develop a learning-based technique that implements the policy iteration using real-time input-state data in this section. Consider the modified system equation as follows:

$$x_{k+1} = A_j x_k + B(u_k + K_j x_k). \tag{13}$$

Along the trajectories of (13), we have

$$x_{k+1}^{\mathrm{T}} P_j x_{k+1} = \big[A_j x_k + B(u_k + K_j x_k)\big]^{\mathrm{T}} P_j \big[A_j x_k + B(u_k + K_j x_k)\big]. \tag{14}$$

Then, using (6) we have:

$$\begin{aligned} x_{k+1}^{\mathrm{T}} P_j x_{k+1} - x_k^{\mathrm{T}} P_j x_k + x_k^{\mathrm{T}} Q_j x_k = 2x_k^{\mathrm{T}} A^{\mathrm{T}} P_j B u_k \\ + 2x_k^{\mathrm{T}} A^{\mathrm{T}} P_j B K_j x_k - x_k^{\mathrm{T}} K_j^{\mathrm{T}} B^{\mathrm{T}} P_j B K_j x_k + u_k^{\mathrm{T}} B^{\mathrm{T}} P_j B u_k \end{aligned} \tag{15}$$

where $Q_j = Q + K_j^{\mathrm{T}} K_j$. Now, by the property of Kronecker product that $\mathrm{vec}(XYZ) = (Z^{\mathrm{T}} \otimes X)\mathrm{vec}(Y)$, we have:

$$\begin{aligned} &\big[(x_{k+1}^{\mathrm{T}} \otimes x_{k+1}^{\mathrm{T}}) - (x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}})\big]\mathrm{vec}(P_j) = \big[2(x_k^{\mathrm{T}} \otimes u_k^{\mathrm{T}}) \\ &+ 2(x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}})(I_n \otimes K_j^{\mathrm{T}})\big]\mathrm{vec}(\Gamma_{1j}) + \big[-(K_j x_k)^{\mathrm{T}} \otimes (K_j x_k)^{\mathrm{T}} \\ &+ (u_k^{\mathrm{T}} \otimes u_k^{\mathrm{T}})\big]\mathrm{vec}(\Gamma_{2j}) - (x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}})\mathrm{vec}(Q_j), \end{aligned} \tag{16}$$

where $\Gamma_{1j} = B^{\mathrm{T}} P_j A$ and $\Gamma_{2j} = B^{\mathrm{T}} P_j B$. Under Asusmption II.2, there exists a sequence $\{k_s\}_{s=0}^{\infty}$, where communication is allowed. The, by collecting online data for the time sequence $k_0 < k_1 < \cdots < k_s$, we obtain

$$\Psi_j \theta_j = -I_{x,x}\mathrm{vec}(Q_j), \tag{17}$$

where $\Psi_j = \Big[\Xi_x, -2J_{x,u} - 2J_{x,x}(I_n \otimes K_j^{\mathrm{T}}),\ J_{K_j x} - J_u\Big]$,

$\theta_j = \Big[\mathrm{vecs}(P_j)^{\mathrm{T}}, \mathrm{vec}(\Gamma_{1j})^{\mathrm{T}}, \mathrm{vecs}(\Gamma_{2j})^{\mathrm{T}}\Big]^{\mathrm{T}}$.

Assumption II.3. *There exists a $s^* \in \mathbb{Z}_+$ such that for all $s > s^*$:*

$$\mathrm{rank}([I_{x,x}, I_{x,u}, I_{u,u}]) = \frac{n(n+1)}{2} + n + 1. \tag{18}$$

Remark 1. *Notice that $s^* \geq \frac{n(n+1)}{2} + n + 1$ to guarantee the feasibility of (18).*

Remark 2. *Under Assumption II.3, Ψ_j has full column rank for all $j \in \mathbb{Z}_+$ [18].*

Algorithm 1 Model-Free Policy Iteration

```
1: Employ $u_k = -K_0 x_k + \eta_k$ as the input on the time interval $[k_0, k_s]$, where $K_0$ is an initial stabilizing control gain and $\eta_k$ is the exploration/probing noise.
2: Compute $\Xi_x, J_x, J_{x,u}, J_u$ until the rank condition in (18) is satisfied. Let $j = 0$.
3: Solve for $\theta_j$ from (17). Then, $K_{j+1} = (1 + \Gamma_{2j})^{-1}\Gamma_{1j}$.
4: Let $j \leftarrow j + 1$ and repeat Step 3 until $|P_j - P_{j-1}| \leq \bar{\varepsilon}$ for $j \geq 1$, where the constant $\bar{\varepsilon} > 0$ is a predefined small threshold.
```

Remark 3. *Note that (18) is like the persistency of excitation condition in adaptive control, which is used in previous algorithms by adding an exploration signal to the input to satisfy (18) [18].*

Remark 4. *Note that $T^\star$ in* (12) *can be obtained by solving for θ_j using Algorithm 1.*

III. Experimental Results and Analysis

A. Developing the RC car [14]

For our experiments, we developed a small-scale RC car model equipped with GPS, IMU, and camera sensors for lane following and changing. The system uses the Nvidia Jetson AGX Xavier for real-time processing. Built on the Traxxas TRX-4 platform [19], the car features PWM-controlled motors, differential gears, and precise steering. With added sensors and batteries (over 2.1kg), it still performs accurate autonomous maneuvers (Fig. 1). The Marvelmind Indoor GPS, combined with an IMU and Kalman filter, provides localization with ±2cm accuracy. A wide-angle camera aids in lane detection.

The Intel RealSense D435 offers RGB and depth data at 1280×720 and 90fps with a wide FOV, supporting obstacle detection even in low light. Sensor data is processed by the Xavier, which generates PWM signals via a PCA9685 board.

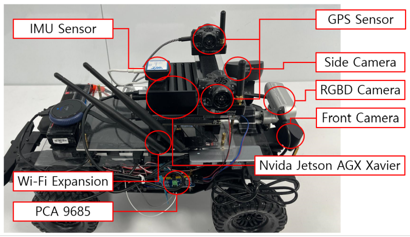


Figure 1: The RC car is fully equipped with battery-powered hardware components, all carefully positioned to ensure balanced weight distribution and optimize safety.

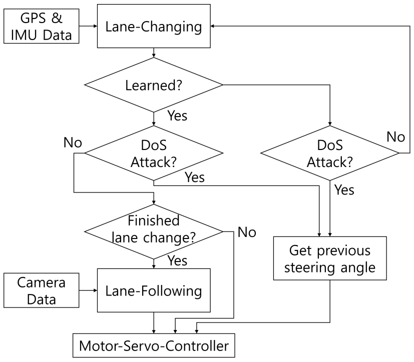


Figure 2: This figure illustrates the basic architecture of autonomous driving algorithms, with each algorithm crafted to deliver its output to the motor exclusively under certain predetermined circumstances.

A Logitech G29 Racing Wheel is used for manual control during tests.

At the core of our car model is the Nvidia Jetson AGX Xavier Board, a compact yet powerful computing platform. Sensors connect directly to the Jetson via serial ports, reducing latency and enhancing scalability, with three distinct programs running simultaneously. The Motor-Servo-Controller manages all motors (acceleration and servo) using parameters such as steering angle, velocity, and driving mode, which are received via UDP packets. Significantly, it processes these independently of the Lane-Following program. The Motor-Servo-Controller interfaces with the PCA9685 driver.

The Lane-Following programs control direction and speed based on sensor inputs relayed through serial ports. After data analysis, results like steering angle and speed are sent to the Motor-Servo Controller using local host addresses and unique port numbers.

Fig. 1 shows our car model, which is $50cm$ wide, $24cm$ long, and $26cm$ tall. The steering angle is capped at 30 degrees for safety, with a motor output of $46W$ and a top speed of $15km/h$, maintaining speed until the battery drops below 80%.

B. Perception

We use the Intel RealSense D435 RGBD camera for obstacle detection and lane changes. Mounted at the front, it processes depth data in key regions—center and sides—based on distance thresholds, while MobileNet-SSD enhances perception with object detection via bounding boxes. UDP signals are sent at 50Hz for real-time control. The system was validated through various obstacle and lane-change scenarios. If a potential collision is detected in the current lane and the adjacent lane is clear, the system sends a UDP signal to initiate a lane change. Otherwise, if the current lane is deemed safe, it instructs the vehicle to maintain its path.

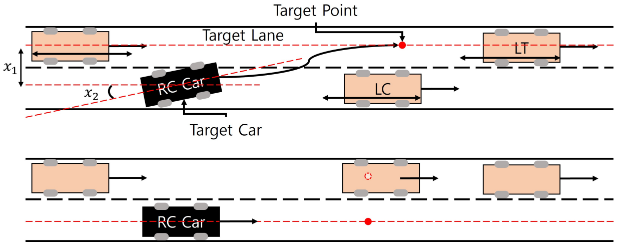


Figure 3: This diagram illustrates two scenarios. From top to bottom: (1) Lane Change to the Next Lane and (2) Lane Following. And the extreme scenario is same as the Lane Change to the Next Lane.

C. Experiments Environment

We built a 5-meter, two-lane test track (each lane 0.3m wide) for experiments, as shown in Fig. 3. The setup includes an RC car and a second vehicle—either a leading car in the current lane (LC) or in the target lane (LT). If the LC stops, the RC car must detect a safe adjacent lane to change into; if the LT blocks the adjacent lane, the RC car must stay in its lane to avoid a collision.

In Fig. 3, x_1 is the lateral offset to the target lane center, and x_3 is the orientation error. These are measured via GPS and IMU. The state vector is $x_k = [x_{1k}, x_{2k}, x_{3k}, x_{4k}]^{\mathrm{T}}$, where x_{2k} and x_{4k} are their respective rates. The control input u_k is the steering angle. The RC car aims to change lanes safely while avoiding obstacles. The decision-making and vehicle dynamics models are detailed in [20], [21].

We aim to develop a data-driven optimal controller that computes the steering angle, enabling the RC car to transition safely from its current lane to the target lane. Our proposed algorithms also ensure the safety of surrounding vehicles. Specifically, we use the data-driven Algorithm 1 to compute the optimal steering angle, where the data matrices in Algorithm 1 are constructed by collecting real-time state x_k and input u_k data from the RC car using the GPS and IMU sensors. The initial stabilizing gain in Algorithm 1 is set to $K_0 = [0.0047, -0.0447, 2.0002, 0.0002]$. The weight matrices are selected as $Q = 0.01 \times I_4$ and $R = 1$. The exploration noise η_k is also chosen as a sum of sinusoidal waves.

D. Result and Analysis

Fig. 4 illustrates the learning phase of the experimental study. To train the optimal controller, input-state data were collected over 200-time steps in the presence of DoS attacks with parameters $\kappa = 50$, $\tau_D = 15$, $T = 4$, and $\eta = 1$. However, only 100 input-output samples were available for learning the controller using Algorithm 1. The experiment was repeated across 10 trials to determine the optimal gain. The lower

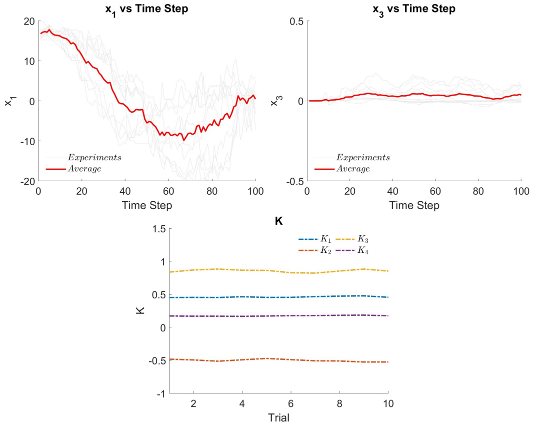


Figure 4: This figure shows the learning phase. State variables x_1 and x_3 represent lateral displacement (cm) and orientation error (rad). The gray line represents the raw data from each trial, while the red line indicates the averages over 10 trials. The K vs. Trial plot displays the gain matrix components $K_i, i = 1, \cdots, 4$ after training.

graph in Fig. 4 shows each trial's components of the learned control gain vector K. The learned controller remains stable across trials, with only minor fluctuations due to sensor noise, demonstrating the robustness of the proposed method. Using Algorithm 1, the critical value of the DoS duration parameter can be obtained as $T^{\star} = 1.1967 \times 10^3$. Similar to [22] and [10], these are sufficient conditions to guarantee the resilience and stability of the closed-loop system. In practice, $T^{\star}$ can be much smaller. Our experimental study demonstrates this by applying stronger DoS attacks with $T = 4$.

Table I: Comparison of Initial and Trained K Values

	K_1	K_2	K_3	K_4
Initial K_0	0.0047	-0.0447	2.0002	0.0002
Average Trained K	**0.4514**	$\mathbf{-0.4819}$	**0.8365**	**0.1724**

By equipping the RC car with our proposed algorithm for learning the optimal controller, we conducted tests under two distinct traffic scenarios and one extreme DoS attack scenario to assess the RC car's adaptability to varying conditions in the presence of DoS attacks. Despite these challenges, the RC car successfully executed lane changes and maintained its trajectory, demonstrating the effectiveness of the control algorithms in managing diverse traffic conditions, even under DoS attacks. In a third scenario, we introduced prolonged DoS attacks to evaluate further the robustness of our methodology against such persistent threats.

1) Scenario 1: Lane change to empty lane: In the first scenario, a slow vehicle in the current lane obstructs the path. As shown in Fig. 5a, x_1 exhibits a significant peak as the RC car transitions to the left into the target lane. The smoothness of the maneuver is reflected by the gradual changes in x_3 and x_4, indicating a controlled adjustment in orientation and position. The position graph confirms the successful lane change, capturing the RC car's shift into the target lane.

2) Scenario 2: Maintaining the Current Lane: Fig. 5b shows the state of the RC car during a scenario where the target lane is blocked, but the current lane remains clear. As illustrated, both x_1 and x_3 remain steady, indicating that the RC car continues to maintain its position in the current lane. The lateral position plot shows the vehicle's trajectory, closely adhering to the current path without merging into the target lane.

3) Scenario 3: Prolonged DoS Attack: In Fig. 5c, the state of the RC car is shown during a prolonged DoS attack scenario, characterized by several long consecutive attacks. Due to these consecutive attacks, the car deviated significantly off track. This experiment highlights that the system may lose stability and operational safety if the attack persists too long. Such extreme experiments are particularly well-suited for using a prototype car model, as demonstrated in our research, since they pose no risk of human casualties.

IV. Conclusion

In this work, we have conducted experiments using an RC car equipped with online learning capabilities from real-time interactions while explicitly addressing the impact of DoS attacks during lane-changing maneuvers. Although this paper focuses on a case study in cybersecurity, this is the first experimental study to address DoS attacks in the context of learning-based control for lane-changing. Our approach integrates advanced perception technologies like GPS and cameras with reinforcement learning and adaptive dynamic programming to develop a resilient control framework. Through experiments using an RC car, we demonstrated that it is possible to ensure safe lane-changing operations in the presence of moderate DoS attacks. This ability to maintain operational safety under attack scenarios highlights the potential of a learning-based control framework for resilient vehicle control in adversarial environments. However, our findings also revealed that the system could not guarantee consistent, safe operations under prolonged DoS attacks. This finding underscores the need for future research to investigate active learning-based controller designs that can adapt to evolving DoS attack conditions and ensure consistent operational safety.

V. ACKNOWLEDGEMENT

This work has been supported in part by the NSF grants CNS-2148309 and CPS-2227153.

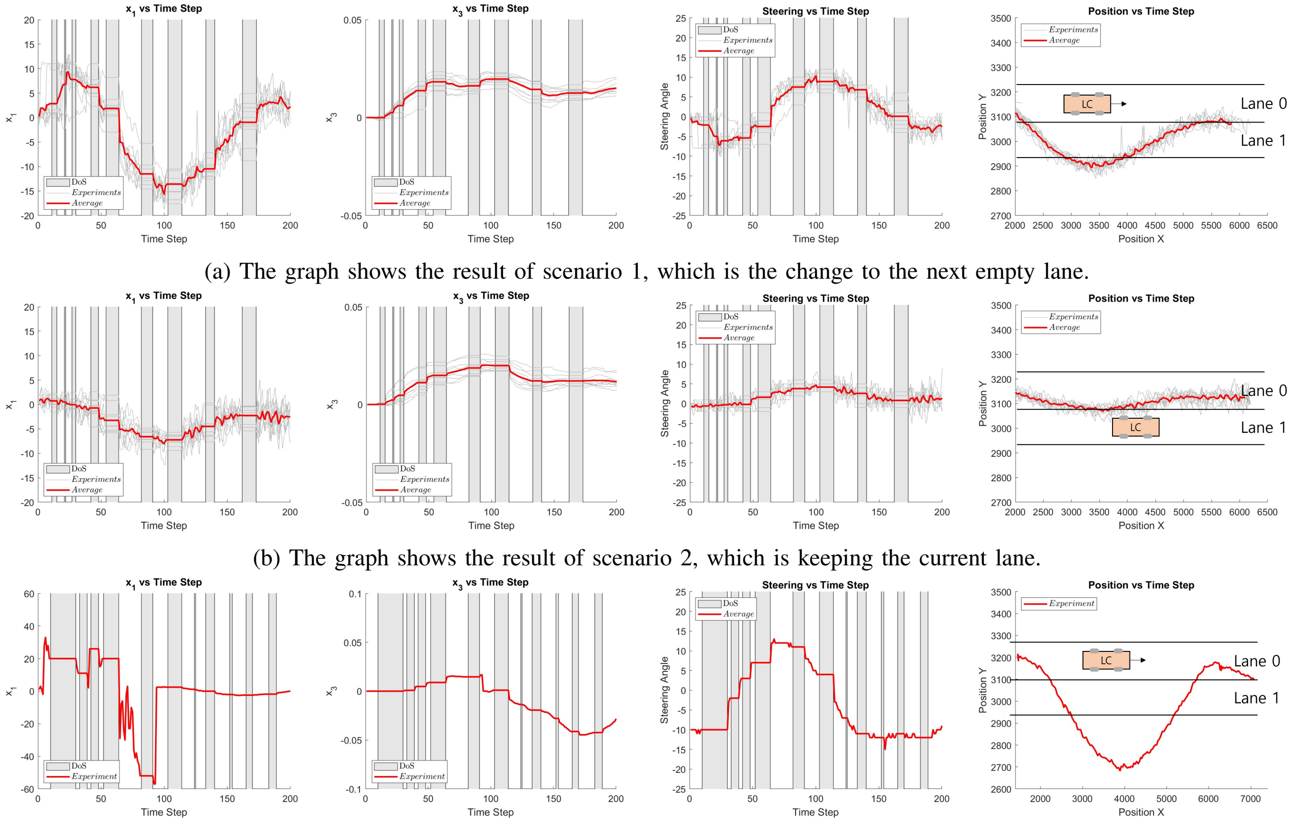
References

[1] Z. A. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3893–3902, 2018.

[2] D. Lee, C. Eom, and M. Kwon, "Ad4rl: Autonomous driving benchmarks for offline reinforcement learning with value-based dataset," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8239–8245.

[3] M. Zhu and S. Martinez, "On the performance analysis of resilient networked control systems under replay attacks," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 804–808, 2013.

[4] A. Cetinkaya, H. Ishii, and T. Hayakawa, "An overview on denial-of-service attacks in control systems: Attack models and security analyses," *Entropy*, vol. 21, no. 2, p. 210, 2019.



(a) The graph shows the result of scenario 1, which is the change to the next empty lane.

(b) The graph shows the result of scenario 2, which is keeping the current lane.

(c) The graph shows the result of scenario 3, which is the change to the next empty lane with a prolonged DoS attack.

Figure 5: The figure shows the results of the experiments, which are categorized into three conditions: Lane Change Required, Unobstructed Current Lane, and prolonged DoS attack.

[5] S.-S. Sun, Y.-X. Li, and Z. Hou, "Prescribed performance-based resilient model-free adaptive control for cpss against aperiodic dos attacks," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 5, pp. 3335–3350, 2024.

[6] M. Wakaiki, A. Cetinkaya, and H. Ishii, "Stabilization of networked control systems under dos attacks and output quantization," *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3560–3575, 2019.

[7] W. Liu, J. Sun, G. Wang, F. Bullo, and J. Chen, "Resilient control under quantization and denial-of-service: Codesigning a deadbeat controller and transmission protocol," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 3879–3891, 2021.

[8] K. G. Vamvoudakis, N.-M. T. Kokolakis *et al.*, "Synchronous reinforcement learning-based control for cognitive autonomy," *Foundations and Trends® in Systems and Control*, vol. 8, no. 1–2, pp. 1–175, 2020.

[9] S. Chakraborty, L. Cui, K. Ozbay, and Z.-P. Jiang, "Automated lane changing control in mixed traffic: An adaptive dynamic programming approach," *Transportation Research Part B: Methodological*, vol. 187, p. 103026, 2024.

[10] W. Gao, C. Deng, Y. Jiang, and Z.-P. Jiang, "Resilient reinforcement learning and robust output regulation under denial-of-service attacks," *Automatica*, vol. 142, p. 110366, 2022.

[11] Y. Liu, B. Zhou, X. Wang, L. Li, S. Cheng, Z. Chen, G. Li, and L. Zhang, "Dynamic lane-changing trajectory planning for autonomous vehicles based on discrete global trajectory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8513–8527, 2021.

[12] X. Li, "Trade-off between safety, mobility and stability in automated vehicle following control: An analytical method," *Transportation research part B: methodological*, vol. 166, pp. 1–18, 2022.

[13] S. Alagumuthukrishnan, S. Deepajothi, R. Vani, and S. Velliangiri, "Reliable and efficient lane changing behaviour for connected autonomous vehicle through deep reinforcement learning," *Procedia Computer Science*, vol. 218, pp. 1112–1121, 2023, international Conference on Machine Learning and Data Engineering.

[14] W. Y. Ha, S. Chakraborty, Y. Yu, S. Ghasemi, and Z.-P. Jiang, "Automated lane changing through learning-based control: An experimental study," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 4215–4220.

[15] S. Malik, P. Bandi, and W. Sun, "An experimental study of denial of service attack against platoon of smart vehicles," in *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*. IEEE, 2021, pp. 23–30.

[16] S. Chakraborty, W. Gao, K. G. Vamvoudakis, and Z.-P. Jiang, "Resilient learning-based control under denial-of-service attacks (accepted for presentation at CDC 2024)," *arXiv preprint arXiv:2409.07766*, 2024.

[17] G. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Transactions on Automatic Control*, vol. 16, no. 4, pp. 382–384, 1971.

[18] Z.-P. Jiang, T. Bian, W. Gao *et al.*, "Learning-based control: A tutorial and some recent results," *Foundations and Trends® in Systems and Control*, vol. 8, no. 3, pp. 176–284, 2020.

[19] L. Traxxas, "The new traxxas summit 16.8 v electric extreme terrain monster truck," 2008.

[20] S. Chakraborty, L. Cui, K. Ozbay, and Z.-P. Jiang, "Automated lane changing control in mixed traffic: An adaptive dynamic programming approach," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1823–1828.

[21] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[22] C. De Persis and P. Tesi, "Input-to-state stabilizing control under denial-of-service," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2930–2944, 2015.