Designing and Testing A Secure Cooperative Adaptive Cruise Control under False Data Injection Attack

Jonas Cunningham-Rush, James Holland, Shirin Noei, Arman Sargolzaei *Senior Member, IEEE*

2023 IEEE Conference on Dependable and Secure Computing (DSC) | 979-8-3503-8211-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/DSC61021.2023.10354170

***Abstract*— Cooperative adaptive cruise control (CACC) is one of the many advanced driver assistance systems (ADAS) that utilize communication between nearby vehicles to maintain speed while maintaining safe following distances between vehicles. Current CACC algorithms are designed assuming that the communication channel is secure; however, using communication channels make CACC susceptible to adversarial injection attacks, such as False data injections (FDI). This paper validates a developed, novel secure controller which can detect and estimate FDI attacks in real-time. Experimental results show that the designed controller and state estimation techniques ensure accurate tracking under FDI attacks. The effectiveness of the developed controller and detection algorithm is shown in a simulation and tested further on a golf-cart-based vehicle-in-the-loop (ViL) platform.**

***Index Terms*— Secure control design; false data injection attack; Lyapunov stability; Cooperative Adaptive Cruise Control; Testing and verification;**

I. Introduction

The National Highway and Traffic Safety Administration (NHTSA) estimates that human error can be blamed for 94%-96% of the 6 million traffic collisions that occur each year in the United States [1], [2]. Automated vehicles (AVs) use sensors to perceive the world around them and support the driver to improve safety. Connected automated vehicles (CAVs) improve upon the foundation of AVs by enabling communication with each other and infrastructure to maximize efficiency in terms of traffic, energy, and safety [3].

CAVs are expected to provide numerous benefits, from improved efficiency and traffic flow to safer and less stressful commutes. CAVs can achieve higher efficiency by forming platoons that coordinate movements to reduce accelerations and decelerations and follow closely to take advantage of lower aerodynamic drag [4], [5]. Platoons also have the benefit of increasing roadway utilization which effectively increases capacity.

Adaptive cruise control (ACC) is an advanced driver assistance system (ADAS) that adjusts the speed of a vehicle based upon feedback from sensors (i.e., radar, lidar, and cameras) to maintain a safe following distance from the leader vehicle. ACC, however, proves unsuitable for forming vehicle strings and platoons. The average data transmission delay for AVs is 1.5s per vehicle length due to onboard sensors, processors, control, and actuation [6]. To address this issue, vehicle-to-vehicle (V2V) connectivity was added to the foundation of ACC to create cooperative adaptive cruise control (CACC) [7], [8]. Data is broadcast continuously to other vehicles in the loop to provide information in real-time [9].

CACC enables shorter following distances, time gaps, and improved stability against oscillations in the flow of traffic [10]. ACC is susceptible to oscillations in traffic flows that are compounded further along in the vehicle string. CACC mitigates the majority of this problem. In a favorable environment, CACC vehicles will acquire information sent from the leading vehicle and adjust accordingly, which greatly reduces the delay between vehicles, resulting in improved energy efficiency, reduced travel time, and reduced collisions [11].

Even with all the benefits of CACC, this technology is unproven and vulnerable to unique threats caused by connectivity and continuous data transmission. These systems are cutting-edge and uncommon in the real-world. Of the many reasons for this, cost, complexity, and infrastructure readiness are some of the key hurdles to traverse. Furthermore, the V2V communication that CAVs and CACC rely upon is susceptible to attacks that could result in widespread disruption. The three primary attack vectors for CACC systems include time delay switch (TDS), denial of service (DoS), and false data injection (FDI). FDIs attacks aim to compromise the integrity of a sensor's data in order to give false readings [12]. TDS attacks will delay the transmission of sensor readings by a given time, relaying outdated information to the control system [13]. DoS attacks aim to render the CACC system unavailable and unable to process the flow of information from other vehicles [14]. This paper focuses on FDI attacks, which are the most probable to occur on a CACC system, which has been an active area of study in recent years [15]–[23]. Existing literature has focused on the detection of FDI attacks. However, unlike the existing works, this research designs and implements a Lyapunov-based controller that combines model-based and learning-based algorithms. This results in a CACC that uses an observer and controller to maintain real-time tracking of a lead vehicle while the follower vehicle is under FDI attack. Understanding why and how these systems fail is vital to ensuring that this technology fulfills its promises of safety.

Unlike other papers in the literature, this paper aims to develop a secure Lyapunov-based controller. We combined model-based and learning-based algorithms to design a novel

J. Cunningham-Rush is with the Department of Mechanical Engineering at Tennessee Technological University, Cookeville, TN. J. Holland, and A. Sargolzaei are with the Department of Mechanical Engineering at the University of South Florida, Tampa, Florida. S. Noei is with the Department of Civil and Coastal Engineering, University of Florida, Gainesville, FL. Email: holland33@usf.edu, a.sargolzaei@gmail.com For papers in which all authors are employed by the US government, the copyright notice is: U.S. Government work not protected by U.S. copyright observer and controller, which are able to maintain the real-time tracking of the lead vehicle while the following vehicle is under FDI attack. Additionally, this paper, unlike others, will also validate its controller and detection accuracy through the use of a real-time target machine. The contributions of this paper are summarized as follows: (i) a novel control strategy is developed which is resilient under FDI attacks, (ii) an FDI attack estimation technique is developed which is able to detect abrupt FDI attacks in real-time with high accuracy, (iii) the stability of the developed controller is illustrated using Lyapunov stability.

A. *Testing and Validation Methods*

For testing CAV's and ADAS, there are several approaches [24]–[28]. Under the umbrella of simulated testing, there are offline and real-time methods. Offline testing seeks to maximize the execution speed of tests while real-time simulations focus on testing accuracy in a bounded response time [24]. Real-time methods also ensure access to testing data, accelerating the validation phase of the testing process and increasing certainty during development [25]. Real-time simulations consist of Software-in-the-Loop (SiL), Vehicle-in-the-Loop (ViL), and Hardware-in-the-Loop (HiL) testing.

SiL testing is often employed in situations where traditional testing is infeasible due to cost, safety, or other limiting factors [26]. SiL, however, is simply unable to match the accuracy and resolution of other methods that incorporate physical hardware and components. HiL, on the other hand, subjects physical components to virtual stimuli in order to explore how a system reacts in real-time [27]. Due to the inclusion of the physical system, HiL is capable of higher quality results compared to purely simulated testing. In terms of vehicle testing, however, HiL lacks the incorporation of the real-time dynamics and motion of the test subject. ViL fills this gap by integrating the entirety of the physical vehicle into the simulated testing environment. The downside of ViL testing, however, is that this method can only consider the events and data generated by the simulation, resulting in lower accuracy and resolution compared to real-world testing. While real-world testing is regarded as the best in terms of accuracy and resolution due to the inclusion of vehicle dynamics and motion, this method is costly in regard to time and capital [29]–[31]. Real-world testing also proves incapable of investigating the myriad of unsafe or infeasible scenarios, particularly edge cases that involve pedestrians.

II. Dynamic Model of CACC under FDI attacks

This paper describes the linear model for CACC. The acceleration, velocity, and position data from the lead vehicle is assumed to be relayed to the following vehicles. For a string of homogeneous vehicles with CACC systems following a leader using a dynamic velocity profile; the dynamics model of vehicles are described as

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = -\frac{b_i}{m_i} v_i(t) + u_i(t), \end{cases} \tag{1}$$

where $i \in \{1, \cdots n\}$ and denotes the follower vehicles, n is the maximum number of follower vehicles, and $i-1$ indicates the leader vehicle. In addition, $m_i \in \mathbb{R}$ is the vehicle's mass and $b_i \in \mathbb{R}$ is the friction force between the road and tires. Also $v_i \in \mathbb{R}$, $x_i \in \mathbb{R}$, and $u_i \in \mathbb{R}$ represent the velocity, position, and control input, respectively.

A. *FDI Attack Representation*

Adversaries inject FDI attacks into the communication network of connected vehicles, so that, vehicles accessing that information are obtaining incorrect data. This causes instability in a platoon of vehicles, resulting in possible collisions. In this paper we assume that acceleration is the only parameter affected by the attack, understood as equation (2). We also assume that the leading vehicle's velocity, position, and acceleration data is sent to its followers. The attack affects the output, which converts it into the observed output

$$\pi_i(a_{i-1}(t)) \triangleq a_{i-1}(t) + d_i(t), \tag{2}$$

where $\pi_i \in \mathbb{R}$ is the attack function, $d_i \in \mathbb{R}$ is the bounded, unknown, continuous, and time-varying FDI attack, and a_{i-1} is the leader acceleration.

Assumption 1. The FDI attack, d_i, is assumed to be bounded and differentiable such that $|d_i(t)| \leq \bar{d}_i$, where $t \geq t_0$ and $\bar{d}_i$ is a positive constant.

III. Problem Statement

The main objective of this paper is to take a designed secure controller that regulates CACC under FDI attacks and implement in a ViL scenario. The controller is designed such that safe distance between the leader and follower vehicles is maintained under the presence of an FDI attack. The CACC algorithm requires an acceleration signal from the lead vehicle in real-time. However, this process is challenged by adversary manipulation, which can lead to potential collisions. Therefore, our second objective is to design an observer and FDI attack detection mechanism to estimate the FDI attack in real-time. To quantify these objectives we define an error signal, $e_i : [t_0, \infty) \to \mathbb{R}$ as

$$e_i(t) \triangleq x_i(t) - x_{i-1}(t) + D_i + x_{d_i}(t), \tag{3}$$

where $D_i \in \mathbb{R}$ is the length of vehicle$_i$, and $x_{d_i} \in \mathbb{R}$ is the desired distance between vehicles.

Assumption 2. The desired distance, its first, and second derivatives are assumed to be bounded by positive known constants, $x_{d_i}, \dot{x}_{d_i}, \ddot{x}_{d_i} \in \mathcal{L}_\infty$ [32].

To facilitate the stability analysis and design process, an auxiliary error equation is proposed as

$$r_i(t) \triangleq \dot{e}_i(t) + \alpha_i e_i(t), \tag{4}$$

where $\alpha_i \in \mathbb{R}_{>0}$, is a user-specified known gain.

The follower vehicles are relayed false information from the leader during an FDI attack. Therefore, the accuracy of the observer needs to be measured and maintained. A state estimate error $\tilde{x}_{i-1} : [t_0, \infty) \to \mathbb{R}$, can be described as

$$\tilde{x}_{i-1}(t) \triangleq x_{i-1}(t) - \hat{x}_{i-1}(t), \tag{5}$$

where $\hat{x}_{i-1} \in \mathbb{R}$ denotes the estimated position of vehicle.

To facilitate the stability analysis for the state estimation, another auxiliary error signal $\tilde{r}_{i-1} : [t_0, \infty) \to \mathbb{R}$ is defined as

$$\tilde{r}_{i-1}(t) \triangleq \dot{\tilde{x}}_{i-1}(t) + \alpha_{i-1}\tilde{x}_{i-1}(t), \tag{6}$$

where $\alpha_{i-1} \in \mathbb{R}_{>0}$ is a user-defined gain.

For determining the accuracy of the control signal estimate an estimation error for the control signal, $\tilde{u}_{i-1} : [t_0, \infty) \to \mathbb{R}^{n_i}$, is defined as

$$\tilde{u}_{i-1} \triangleq u_{i-1} - \hat{u}_{i-1}, \tag{7}$$

where $\hat{u}_{i-1} \in \mathbb{R}$ and $u_{i-1} \in \mathbb{R}$ are the estimated and actual control signal of the leader, respectively.

Defining $\bar{u}_{i-1} \triangleq u_{i-1} + d_i$ and $\hat{u}_{i-1} \triangleq \bar{u}_{i-1} - \hat{d}_i$ yields

$$\tilde{u}_{i-1} = u_{i-1} - \bar{u}_{i-1} + \hat{d}_i, \tag{8}$$

where $\hat{d}_i \in \mathbb{R}$ is the estimated FDI attack.

To measure the accuracy of the FDI attack estimation, the estimation error for the FDI attack, $\tilde{d}_i : [t_0, \infty) \to \mathbb{R}^{n_i}$, is defined as

$$\tilde{d}_i(t) \triangleq d_i(t) - \hat{d}_i(t), \tag{9}$$

where $\tilde{d}_i$ is bounded such that $\tilde{d}_i \leq \bar{\tilde{d}}_i$, where $\bar{\tilde{d}}_i \in \mathbb{R}_{>0}$.

IV. Proposed Solution

A. Controller Design

The control signal was designed using the Lyapunov stability analysis in Section V as

$$\begin{aligned} u_i(t) \triangleq & \frac{b_i}{m_i}v_i(t) - \frac{b_{i-1}}{m_{i-1}}v_{i-1}(t) + \bar{u}_{i-1}(t) - \hat{d}_i(t) \\ & - \ddot{x}_d(t) - \alpha_i r_i(t) + \alpha_i^2 e_i(t) - e_i(t) - K_{1_i} r_i(t), \end{aligned} \tag{10}$$

where $K_{1_i} \in \mathbb{R}_{>0}$ is a gain specified by the user.

Taking the derivative of equation (4) and substituting (3) yields the closed loop form of the system as

$$\dot{r}_i(t) = \ddot{x}_i(t) - \ddot{x}_{i-1}(t) + \ddot{x}_d(t) + \alpha_i \dot{e}_i(t). \tag{11}$$

Replacing $\ddot{x}_i$ and $\ddot{x}_{i-1}$ and (8) into (11) produces

$$\begin{aligned} \dot{r}_i(t) = & -\frac{b_i}{m_i}v_i(t) + u_i(t) + \frac{b_{i-1}}{m_{i-1}}v_{i-1}(t) - \bar{u}_{i-1}(t) \\ & + d_i(t) + \ddot{x}_{di}(t) + \alpha_i \dot{e}_i(t). \end{aligned} \tag{12}$$

Substituting (10) into (12) results in

$$\dot{r}_i(t) = \tilde{d}_i(t) - K_{1_i} r_i(t) - e_i(t). \tag{13}$$

B. FDI Attack Estimation

The detailed observer design in the ensuing subsection includes a neural network-based FDI attack detection algorithm and state estimator, based on the work done in [33]. Based on their research N_{n_i} is bounded such that $N_{n_i} \leq \bar{n}_{n_i}$, where $\bar{n}_{n_i} \in \mathbb{R}_{>0}$ [33].

Considering respect to the spatial domain, the NN estimation of FDI attack can be described as

$$\hat{d}_i \triangleq \hat{W}_i^T \sigma(\hat{V}_i^T \delta_i), \tag{14}$$

where $\hat{W}_i \in \mathbb{R}^{(n_i+1)\times n_i}, \hat{V}_i \in \mathbb{R}^{(n_i+1)\times n_n}$ represent the estimated ideals weights, and δ_i is given as

$$\delta_i \triangleq [1, \hat{d}_i^T]^T. \tag{15}$$

A Taylor's series approximation is applied resulting

$$\tilde{d}_i = \tilde{W}_i^T \sigma(\hat{V}_i^T \delta_i) + \hat{W}_i^T \sigma'(\hat{V}_i^T \delta_i)\tilde{V}_i^T \delta_i + N_{n_i}. \tag{16}$$

The updating laws for the NN weights written in [33] are redescribed as

$$\dot{\hat{W}}_i = proj(\Gamma_{1_i}(\hat{V}_i^T \delta_i) r_i), \tag{17}$$

and

$$\dot{\hat{V}}_i = proj(\Gamma_{2_i}^T r_i \hat{W}_i^T \sigma(\hat{V}_i^T \delta_i)), \tag{18}$$

where $\Gamma_{1_i}, \Gamma_{2_i} \in \mathbb{R}^{n_i \times n_i}$ are definite positive matrices.

C. Observer Design

Based on the stability analysis in section V, the observer for vehicle i is designed as

$$\begin{aligned} \ddot{\hat{x}}_{i-1}(t) = & -\frac{b_{i-1}}{m_{i-1}}v_{i-1}(t) + \bar{u}_{i-1}(t) - \hat{d}_i(t) + L_{1_i}\tilde{r}_{i-1}(t) \\ & + \alpha_{i-1}\tilde{r}_{i-1}(t) - \alpha_{i-1}^2\tilde{x}_{i-1}(t) + \tilde{x}_{i-1}(t), \end{aligned} \tag{19}$$

where L_{1_i} represents a user-defined gain.

Taking the derivative of (6) with respect to time yields

$$\dot{\tilde{r}}_{i-1}(t) = \ddot{\tilde{x}}_{i-1}(t) + \alpha_{i-1}\dot{\tilde{x}}_{i-1}(t). \tag{20}$$

More simplification and variable substitution results in

$$\begin{aligned} \dot{\tilde{r}}_{i-1}(t) = & -\frac{b_{i-1}}{m_{i-1}}v_{i-1}(t) + u_{i-1}(t) - \ddot{\hat{x}}_{i-1}(t) + \alpha_{i-1}\tilde{r}_{i-1}(t) \\ & - \alpha_{i-1}^2\tilde{x}_{i-1}(t). \end{aligned} \tag{21}$$

Exchanging (19) the final error equation can be further simplified into

$$\dot{\tilde{r}}_{i-1}(t) = -L_{1_i}\tilde{r}_{i-1}(t) - \tilde{x}_{i-1}(t) - \tilde{d}_i(t). \tag{22}$$

V. Stability Analysis

For simplicity (t) was dropped in further calculations. Consider $V_{L_i} : \mathbb{R}^5 \times [0, \infty) \to \mathbb{R}_{\geq 0}$, a radially unbounded, positive definite, continuously differentiable Lyapunov function displayed as

$$V_{L_i} = \frac{1}{2}e_i^2 + \frac{1}{2}r_i^2 + \frac{1}{2}\tilde{x}_{i-1}^2 + \frac{1}{2}\tilde{r}_{i-1}^2 + H_i, \tag{23}$$

where $H_i : [t_0, \infty) \to \mathbb{R}_{\geq 0}$ is defined as

$$H_i \triangleq \frac{1}{2}tr(\tilde{W}_i^T \Gamma_{1_i}^{-1} \tilde{W}_i) + \frac{1}{2}tr(\tilde{V}_i^T \Gamma_{2_i}^{-1} \tilde{V}_i). \tag{24}$$

Since $\tilde{W}_i$ and $\tilde{V}_i$ are bounded, H_i is bounded by $|H_i| \leq H_{i,max}$ where $H_{i,max} \in \mathbb{R}_{>0}$. Furthermore, let $p_i \in \mathbb{R}^{4ni}$ be define as

$$p_i \triangleq [e_i^T, r_i^T, \tilde{r}_{i-1}^T, \tilde{x}_{i-1}^T]^T, \tag{25}$$

and let ψ_{1_i} and ψ_{2_i} be defined as

$$\psi_{1_i} \triangleq \frac{1}{2}{p_i}^2, \tag{26}$$

and

$$\psi_{2_i} \triangleq {p_i}^2. \tag{27}$$

Taking the derivative of (23) results

$$\begin{aligned} \dot{V}_{L_i} =& e_i \dot{e}_i + r_i \dot{r}_i + \tilde{x}_{i-1}\dot{\tilde{x}}_{i-1} + \tilde{r}_{i-1}\dot{\tilde{r}}_{i-1} \\ & - tr(\tilde{W}_i \Gamma_{1i}^{-1} \dot{\hat{W}}_i) - tr(\tilde{V}_i \Gamma_{2i}^{-1} \dot{\hat{V}}_i). \end{aligned} \tag{28}$$

The Lyapunov function satisfies the following inequality

$$\psi_{1_i} \leq V_{L_i} \leq \psi_{2_i} + H_{i,max}. \tag{29}$$

Substituting (4) and (13) into (28) yields

$$\begin{aligned} \dot{V}_{L_i} =& e_i(r_i - \alpha_i e_i) + r_i(\tilde{d}_i - K_{1_i} r_i - e_i) \\ & + \tilde{x}_{i-1}\dot{\tilde{x}}_{i-1} + \tilde{r}_{i-1}\dot{\tilde{r}}_{i-1} - tr(\tilde{W}_i \Gamma_{1i}^{-1} \dot{\hat{W}}_i) \\ & - tr(\tilde{V}_i \Gamma_{2i}^{-1} \dot{\hat{V}}_i). \end{aligned} \tag{30}$$

Further substitution of (16) in results in

$$\begin{aligned} \dot{V}_{L_i} =& -\alpha_i e_i^2 + r_i(\tilde{W}_i^T \sigma(\hat{V}_i^T \delta_i) + \hat{W}_i^T \sigma'(\hat{V}_i^T \delta_i)\tilde{V}_i^T \delta_i \\ & + N_{n_i}) + r_i(-K_{1_i} r_i) - \tilde{r}_{i-1}\tilde{d}_i - \alpha_i \tilde{x}_{i-1}^2 - L_{1_i}\tilde{r}_{i-1}^2 \\ & - tr(\tilde{W}_i \Gamma_{1i}^{-1} \dot{\hat{W}}_i) - tr(\tilde{V}_i \Gamma_{2i}^{-1} \dot{\hat{V}}_i). \end{aligned} \tag{31}$$

Young's Inequality is applied to select terms in (31) and given as

$$\begin{aligned} r_i N_{n_i} &\leq \frac{1}{2\varepsilon_0}{r_i}^2 + \frac{\varepsilon_0}{2}{N_{n_i}}^2, \\ \tilde{r}_{i-1}\tilde{d}_i &\leq \frac{1}{2\varepsilon_1}\tilde{r}_{i-1}^2 + \frac{\varepsilon_1}{2}\tilde{d}_i^2. \end{aligned} \tag{32}$$

Applying Young's Inequality and the updated laws from (17) and (18), the equation (31) becomes

$$\begin{aligned} \dot{V}_{L_i} \leq & -\alpha_i {e_i}^2 + \frac{1}{2\varepsilon_0}{r_i}^2 + \varphi_i + \frac{1}{2\varepsilon_1}\tilde{r}_{i-1}^2 \\ & - K_{1_i}{r_i}^2 - \alpha_{i-1}\tilde{x}_{i-1}^2 - L_{1_i}\tilde{r}_{i-1}^2, \end{aligned} \tag{33}$$

where φ_i is defined as

$$\varphi_i \triangleq \frac{\varepsilon_0}{2}\bar{n}_{n_i}^2 + \frac{\varepsilon_1}{2}\bar{\tilde{d}}_i^2. \tag{34}$$

Combining like terms results in

$$\begin{aligned} \dot{V}_{L_i} \leq & -(\alpha_{i-1})\tilde{x}_{i-1}^2 - (\alpha_i){e_i}^2 \\ & - (L_{1_i} - \frac{1}{2\varepsilon_1})\tilde{r}_{i-1}^2 \\ & - (K_{1_i} - \frac{1}{2\varepsilon_0}){r_i}^2. \end{aligned} \tag{35}$$

The sufficient conditions are given as

$$\begin{aligned} &\alpha_{i-1} > 0, \\ &\alpha_i > 0, \\ &L_{1_i} > \frac{1}{2\varepsilon_1}, \\ &K_{1_i} > \frac{1}{2\varepsilon_0}, \end{aligned} \tag{36}$$

where ε_0 and ε_1 denote positive known constants.

Based on the sufficient conditions in (36), positive constants, α_{1_i} and α_{2_i} can be written as

$$\alpha_{1_i} \triangleq L_{1_i} - \frac{1}{2\varepsilon_1}, \tag{37}$$

$$\alpha_{2_i} \triangleq K_{1_i} - \frac{1}{2\varepsilon_0}. \tag{38}$$

Knowing the the Lyapunov function is bounded, (35) can be written as

$$\dot{V}_{L_i} \leq -\frac{\alpha_{3_i}}{\psi_{2_i}}V_{L_i} + \frac{\alpha_{3_i}}{\psi_{2_i}}H_{i,max} + \varphi_i, \tag{39}$$

where this ensures semi-globally uniformly bounded tracking and $\alpha_{3_i} \triangleq \min\{\alpha_{i-1}, \alpha_i, \alpha_{1_i}, \alpha_{2_i}\}$.

Stability is assured given the sufficient equations provided in (36) are satisfied.

VI. Results

The following section inspects the performance of the designed controller and detection algorithm under an FDI attack through MATLAB/Simulink simulation. Additional analysis was performed in simulation to explore additional disturbance that was initially unaccounted for. The designed controller and detection algorithm was then implemented on a golf-cart ViL platform.

A. Model of Golf Cart

A golf cart-based ViL research platform was developed specifically to extend the capabilities of testing and experimentation of the proposed CACC. To implement the CACC on the platform, a transfer function for the golf cart was determined using experimental analysis. The first-order linear model was obtained using a power supply to determine the step response for a given input. By applying DC voltages of 1.5V, 2.0V, and 3.0V to the motor controller's signal wire, it was possible to measure the response of the system and determine the best signal to base our transfer function on. Using a signal of 2.5 VDC, several runs were recorded and averaged together, resulting in the plot shown in Fig. 6. From this data, the transfer function was found to be $G(s) = \frac{b}{s+a} = \frac{0.245}{s+.1818}$. This model is implemented into the simulation, altering the dynamic profile shown in equation (1).

B. Testing Setup

Since we are dealing with security of a vehicle, real-world testing is not safe to perform. Hence, we validated the developed technique on SiL environment, HiL, and finally ViL.

1) SiL and HiL setup: The proposed Lyapunov-based controller and neural network detection algorithm was implemented into MATLAB/Simulink for a SiL environment. This simulation in conjunction with a Speedgoat Baseline target machine created a HiL environment where the controller was further tested. Figure 1 displays the flow of information between both MATLAB and the target machine.

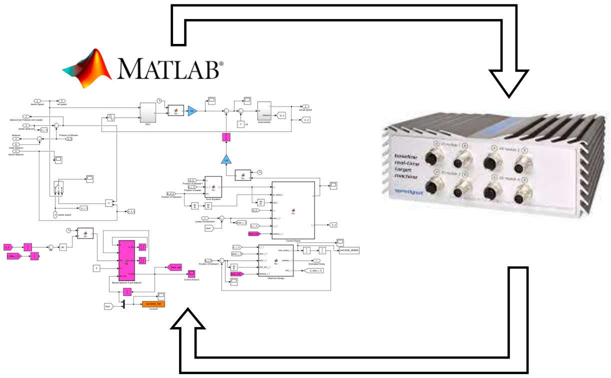

Fig. 1: Hardware-in-the-Loop Connection.

2) Vehicle-in-the-Loop Setup: An automated Club Car Precedent i2 golf cart was developed for testing the developed CACC. The platform is based upon an electric golf cart with extensive modifications to enable programmable control of the vehicle. The vehicle is equipped with an Arduino Uno that feeds inputs to the golf cart to control movement. Under testing, the Arduino is connected to the simulation and sends the resulting voltage commands to the golf cart, depicted in Figure 2. The golf cart then carries out these actions, feeding dynamics and motion back into the simulation in real-time.

The modifications, while extensive, were made easier due to the golf cart's electric drive-train. The original throttle system in the golf cart utilizes a potentiometer to supply variable voltage to the DC motor. In order to replicate the original function, an Arduino was spliced into the signal wire of the potentiometer. The Arduino uses a pulse width modulation (PWM) function to supply the wire with varying voltages to simulate variable speeds. The motor controller listens for analog inputs in the range of 0.3 VDC and 4.5 VDC, allowing for the Arduino to simply inject an analog signal to control velocity.

The golf cart does not have a built-in speedometer; consequently, we used our phone's built-in accelerometer sensor to determine its speed. Figure 2 portrays how the phone was connected to our simulation and the golf cart. The phone was connected via WiFi to MATLAB running on our laptop. The laptop then communicated with another; through user datagram protocol (UDP), which was running the simulation. To run the simulation on the golf cart, an Arduino Uno was connected to its signal wires. We also devised a first-order linear model to convert the controller's output signal into a voltage reading that the golf cart can receive. That output signal is sent to the Arduino, which would power the motor.

With this platform, the value of HiL and ViL, combined, will be demonstrated. Relative to a full-size vehicle, our golf cart-based platform is simple, compact, easily accessible, and low-cost. This allows for ADAS, such as the developed CACC, to be tested on the golf cart prior to scaling testing up to a real, full-size vehicle.

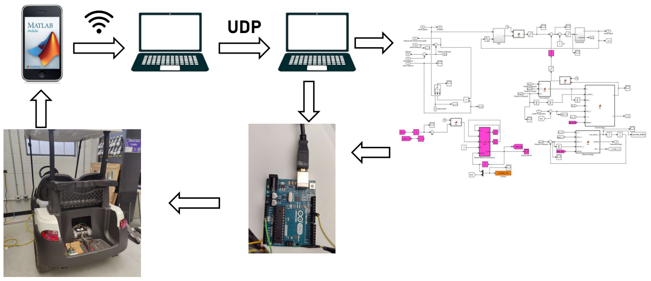

Fig. 2: Vehicle-in-the-Loop Connection.

C. SiL Testing

Figure 3 shows a calculation of the distance between the leader and follower vehicles. As the speed increases, consequently, the distance between vehicles increases as well, shown in Figure 3. This explains why the distance between vehicles in the SiL scenario is slightly greater than HiL. Throughout the entirety of the simulation it remains a positive value indicting no collision occurring. The follower vehicle's speed is depicted in figure 4. The lead vehicle had an input of 0 m/s at 60 seconds which caused the following vehicle to deviate from maintaining the desired speed and, instead, begin slowing down to avoid a collision. As both vehicles reach 0 m/s, the final distance between them is a positive constant dictating that no collision occurred. The accuracy of the detection algorithm is portrayed in figure 5. Notably, the lower estimated value for the FDI attack does not cause a collision.

Using Simulink, additional analysis was performed by injecting signals directly into the leader's acceleration data, which is processed by the follower vehicle. These signals were sinusoidal and pseudo-random waveforms with amplitudes from 1 to 10 and a final run with an amplitude of 20. During 100 iterations of each test signal, the developed controller was resilient to these additional disturbances.

D. HiL Testing

The HiL simulation displays a different scenario, where the leader vehicle continues at a positive speed instead of stopping. Figure 4 shows that the follower vehicle deviates from its desired speed and slows to follow the leader's speed.

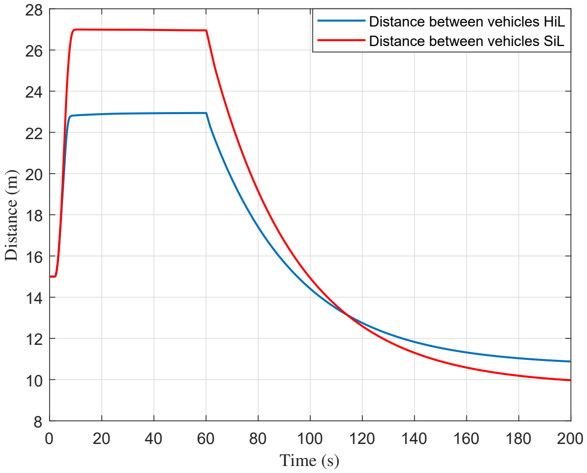

Fig. 3: Distance between Follower and Leader Vehicle.

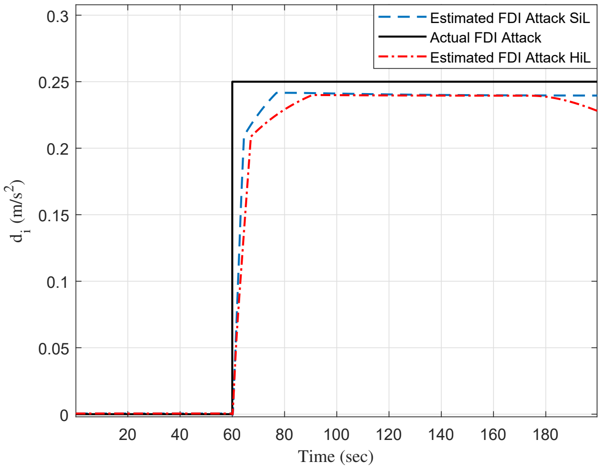

Fig. 5: FDI Attack Estimation.

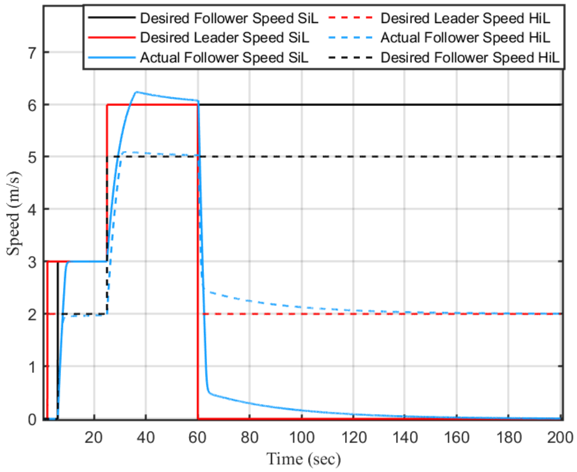

Fig. 4: Follower Vehicle's Speed Profile.

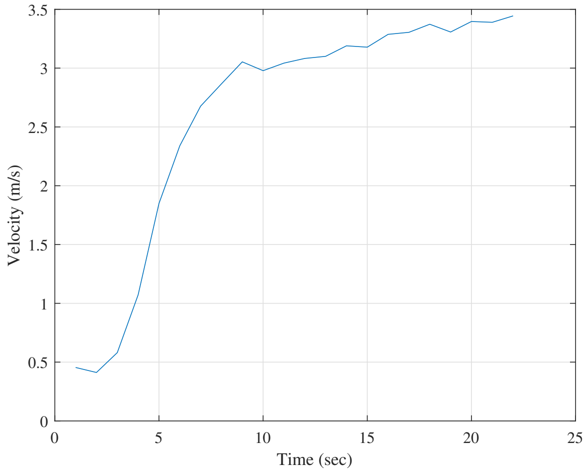

Fig. 6: Golf Cart Data Used for Determining the Model.

E. ViL Testing

This section displays the performance of the controller and detection algorithm when implemented on a ViL platform. The speed of the golf car is shown in Figure 7. It takes approximately 3 seconds for the golf cart to respond to the signal being sent. The golf cart overshoots the desired speed at approximately 9 seconds and then slows to correct itself. Once the desired speed rises at 15 seconds, the golf cart follows shortly after, settling at a value lower than desired. The golf cart then continues to attempt to follow the desired speed, but fluctuates around the value due to the unconventional ViL setup. The estimation is depicted in Figure 8. The estimator oscillates around the actual FDI attack value. A calculation of the RMSE of the FDI estimation resulted in a value of 0.0277.

VII. Conclusion

CACC is an advanced driver-assistance system that collects acceleration data from a leading car, along with its own onboard sensor data to adjust the vehicle's speed in order to maintain a safe distance between both vehicles. An FDI attack occurs when incorrect data is injected into the system, with the goal of causing instability and collisions. In order to negate the effects of an FDI attack on a CACC system, both a secure and resilient controller and detection algorithm were designed. The proposed designs accurately detected the FDI attack and negated its effects on the vehicle, causing it to maintain a safe distance throughout the entire simulation. The simulation was run through MATLAB/Simulink, hardware-in-the-loop (HIL) using a Speedgoat Baseline real-time target machine, and vehicle-in-the-loop using an electric golf cart based platform. The ViL scenario maintained a safe distance between vehicles and accurately detected the FDI attack. However, due to the process by which we constructed the ViL environment, the golf cart's response was delayed, and its speed fluctuated moderately around the desired. Further testing using an IMU sensor to relay the speed in real-time could potentially reduce the delay.

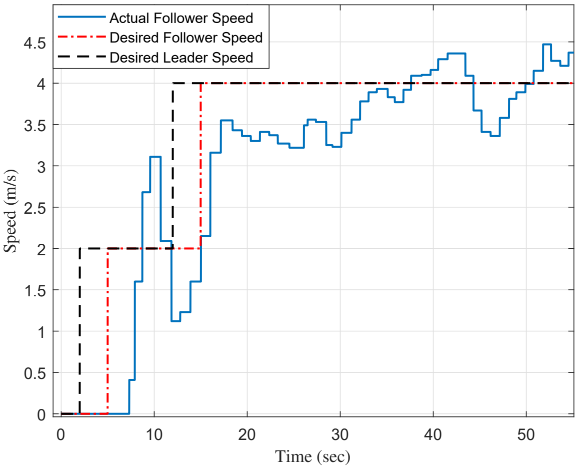

Fig. 7: Follower Vehicle's Speed Profile using ViL.

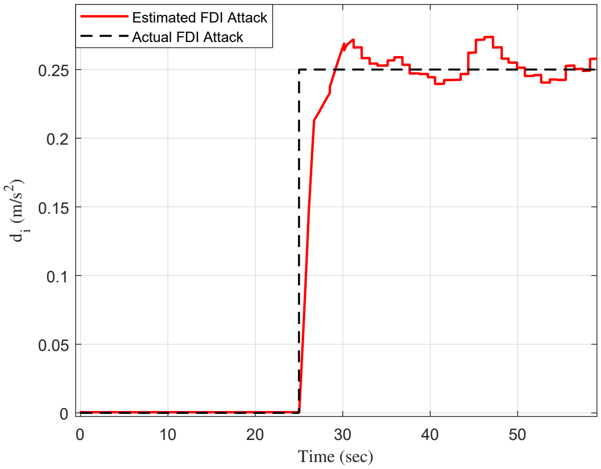

Fig. 8: FDI Attack Estimation using ViL.

Also, the Speedgoat Baseline real-time machine that we used for the HiL environment includes an IO module that could replace the Arduino. These modifications could improve the delay and overall performance of the designed controllers and estimators.

Implementing such as controller, however, comes with some challenges. While our implementation used commonly available off-the-shelf components, one that would see widespread adoption would likely be drastically more expensive and rigorously analyzed. Secondly, there is the major issue of an infrastructure that possesses V2X capabilities to support CACC.

VIII. Acknowledgement

Partial support of this research was provided by the National Science Foundation under Grant No. CNS-1919855 and ECCS-EPCN-2241718. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency. This work was also sponsored partially by a contract from the Southeastern Transportation Research, Innovation, Development and Education Center (STRIDE), a Regional University Transportation Center sponsored by a grant from the U.S. Department of Transportation's University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein.

References

[1] N. Highway and T. S. Administration, "2016 fatal motor vehicle crashes: Overview," 2017-10.

[2] N. Highway and T. S. Administration, "Crash report sampling," 2017.

[3] H. B. Almobayedh, *Simulation of the Impact of Connected and Automated Vehicles at a Signalized Intersection*. PhD thesis, University of Dayton, 2019.

[4] O. of Energy Efficiency Renewable Energy, "Smart mobility connected and automated vehicles capstone report," 2020-07.

[5] J. Cerutti, G. Cafiero, and G. Iuso, "Aerodynamic drag reduction by means of platooning configurations of light commercial vehicles: A flow field analysis," *International Journal of Heat and Fluid Flow*, vol. 90, p. 108823, 2021.

[6] T. Guo, "Cooperatie adaptive cruise control (CACC) in the context of vehicle to vehicle communications: an overview," tech. rep., UC Davis, 2017.

[7] F. H. A. NU.S. Department of Transportation, *Vehicle-to-Infrastructure Program, Cooperative Adaptive Cruise Control*. FHWA-JPO-16-257, 2015.

[8] S. Noei, M. Parvizimosaed, and M. Noei, "Longitudinal control for connected and automated vehicles in contested environments," *Electronics*, vol. 10, no. 16, p. 1994, 2021.

[9] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control: Definitions and operating concepts," *Transportation Research Record*, vol. 2489, no. 1, pp. 145–152, 2015.

[10] K. C. Dey, L. Yan, X. Wang, Y. Wang, H. Shen, M. Chowdhury, L. Yu, C. Qiu, and V. Soundararaj, "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 491–509, 2015.

[11] E. Semsar-Kazerooni, J. Verhaegh, J. Ploeg, and M. Alirezaei, "Cooperative adaptive cruise control: An artificial potential field approach," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 361–367, IEEE, 2016.

[12] R. A. Biroon, P. Pisu, and Z. Abdollahi, "Real-time false data injection attack detection in connected vehicle systems with pde modeling," in *2020 American Control Conference (ACC)*, pp. 3267–3272, IEEE, 2020.

[13] A. Sargolzaei, K. K. Yen, and M. N. Abdelghani, "Preventing time-delay switch attack on load frequency control in distributed power systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1176–1185, 2015.

[14] 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.

[15] R. A. Biroon, Z. A. Biron, and P. Pisu, "False data injection attack in a platoon of cacc: Real-time detection and isolation with a pde approach," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[16] M. Wolf, A. Willecke, J.-C. Müller, K. Garlichs, T. Griebel, L. Wolf, M. Buchholz, K. Dietmayer, R. W. van der Heijden, and F. Kargl, "Securing cacc: Strategies for mitigating data injection attacks," in *2020 IEEE Vehicular Networking Conference (VNC)*, pp. 1–7, IEEE, 2020.

[17] R. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (cacc)," in *2017 IEEE Vehicular Networking Conference (VNC)*, pp. 45–52, IEEE, 2017.

[18] C. Zhao, J. S. Gill, P. Pisu, and G. Comert, "Detection of false data injection attack in connected and automated vehicles via cloud-based sandboxing," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[19] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, 2015.

[20] A. Bezemskij, G. Loukas, D. Gan, and R. J. Anthony, "Detecting cyber-physical threats in an autonomous robotic vehicle using bayesian networks," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 98–103, IEEE, 2017.

[21] A. Petrillo, A. Pescapé, and S. Santini, "A collaborative control strategy for platoons of autonomous vehicles in the presence of message falsification attacks," in *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 110–115, IEEE, 2017.

[22] A. Sargolzaei, C. D. Crane, A. Abbaspour, and S. Noei, "A machine learning approach for fault detection in vehicular cyber-physical systems," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 636–640, IEEE, 2016.

[23] S. Noei, A. Sargolzaei, A. Abbaspour, and K. Yen, "A decision support system for improving resiliency of cooperative adaptive cruise control systems," *Procedia computer science*, vol. 95, pp. 489–496, 2016.

[24] J. Bélanger and P. Venne, "The what, where and why of real-time simulation."

[25] O. Gietelink, J. Ploeg, B. D. Schutter, and M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations," *Vehicle System Dynamics*, vol. 44, p. 569–590, Jul 2006.

[26] "Advantages and disadvantages of simulation." Available at `https://www.concentricmarket.com/blog/advantages-and-disadvantages-of-simulation`.

[27] D. Bullock, B. Johnson, R. Wells, M. Kyte, and Z. Li, "Hardware-in-the-loop simulation," *Transportation Research Part C: Emerging Technologies*, vol. 12, p. 73–89, Feb 2004.

[28] S. S. Banerjee, S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer, "Hands off the wheel in autonomous vehicles?: A systems perspective on over a million miles of field data," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, p. 586–597, Jun 2018.

[29] A. J. Hawkins, "Toyota will test self-driving car 'edge cases' at new proving ground in michigan." Available at `https://www.theverge.com/2018/5/3/17314778/toyota-self-driving-car-test-proving-ground-michigan`.

[30] A. J. Hawkins, "Waymo gave me a ride in a car with no driver." Available at `https://www.theverge.com/2017/10/31/16579180/waymo-self-driving-test-facility-castle-google`.

[31] D. Muoio, "Uber built a fake city in pittsburgh with roaming mannequins to test its self-driving cars." Available at `https://www.businessinsider.com/ubers-fake-city-pittsburgh-self-driving-cars-2017-10`.

[32] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and rise feedback control structure," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2180–2185, 2008.

[33] A. Sargolzaei, B. C. Allen, C. D. Crane, and W. Dixon, "Lyapunov-based control of a nonlinear multi-agent system with a time-varying input delay under false-data-injection attacks," *IEEE Transactions on Industrial Informatics*, 2021.