# Estimation of Safe Sensor Measurements of Autonomous System Under Attack

Raj Gautam Dutta
University of Central Florida
rajgautamdutta@knights.ucf.edu

Xiaolong Guo
University of Central Florida
guoxiaolong@knights.ucf.edu

Teng Zhang
University of Central Florida
teng.zhang@ucf.edu

Kevin Kwiat
Air Force Research Lab., Information Directorate, Cyber Assurance Branch
kevin.kwiat@us.af.mil

Charles Kamhoua
Air Force Research Lab., Information Directorate, Cyber Assurance Branch
charles.kamhoua.1@us.af.mil

Laurent Njilla
Air Force Research Lab., Information Directorate, Cyber Assurance Branch
laurent.njilla@us.af.mil

Yier Jin
University of Central Florida
yier.jin@eecs.ucf.edu

## Abstract

The introduction of automation in cyber-physical systems (CPS) has raised major safety and security concerns. One attack vector is the sensing unit whose measurements can be manipulated by an adversary through attacks such as denial of service and delay injection. To secure an autonomous CPS from such attacks, we use a challenge response authentication (CRA) technique for detection of attack in active sensors data and estimate safe measurements using the recursive least square algorithm. For demonstrating effectiveness of our proposed approach, a car-follower model is considered where the follower vehicle's radar sensor measurements are manipulated in an attempt to cause a collision.

## 1 Introduction

The age of autonomous cyber-physical systems (CPS) is upon us and their influence is gradually increasing in our lives. Currently, many ground vehicles have automated features such as adaptive cruise control (ACC), lane keeping control (LKC), and intelligent parking assist, which takes input from sensors to carry out the desired task. However, many instances have been reported, where these automated systems have performed undesired task due to sensor errors. Such failures can occur due to internal or environmental disturbances. Additionally, intentional sensor measurement errors could be caused by attacks targeting sensing software, hardware, or even its analog signals. While a lot of work has been done on ensuring safety of systems under standard sensing errors, much less attention has been given on securing it and its sensors from attacks. Recently, researchers have demonstrated successful spoofing of GPS, radar, lidar, and ultrasonic signals along with attack on cameras [9]. As such, autonomous CPS, which rely heavily on sensing units for decision making, remain vulnerable to such attacks.

In this paper, we propose a solution for detecting attacks on analog signals of *active sensors*. We will also provide an algorithm for estimating sensor measurements when it is under attack. Our detection method and estimation algorithm are implemented before the measurements enter the digital domain of the system. Thus, we are able to defend the system before corrupted measurements could effect its operations. The main contributions of this paper are summarized as follows.

- We use a challenge response authentication (CRA) method for detecting attacks on active sensors. This method does not require any redundant sensors and it does not produce any false positives or false negatives.
- We develop an algorithm to estimate correct sensor measurements using recursive least square (RLS) approach, after an attack has been detected. Consequently, it enables the autonomous CPS to recover from an attack.
- We also modify the standard intelligent-driver car following model (IDM) by integrating it with an adaptive cruise control (ACC) system.

## 2 Related Work

State estimation methods have been used recently to detect sensor attacks on linear and non-linear systems [1, 3, 8, 11, 12]. In [3], secure states were first estimated of a linear control system under sensor spoofing attack by using a decoder. Then, a linear static controller was designed based on the set of secure states. In [12], a learning mechanism was used to construct a set of invariants called "safety envelope" from collected sensor data. When the system violated these constrains, an alarm was raised depending on whether it was an attack or noise. Chow et al. [1] provided two sound and complete state estimation algorithms for multi-output continuous-time linear systems. They also provided conditions to check whether the system was observable under attack. In [8], a detection algorithm was developed based on sensor fusion, which detects malfunctioning of sensors on an autonomous ground vehicle. In [11], an event-triggered projected gradient descent algorithm was used to recover state of discrete-time linear time invariant system under attack.

However, all these works rely on availability of multiple sensors for detection of attack and estimation of safe states of the system. In addition, existing sensor fusion methods combine redundant measurements to provide correct data to the system. Redundancy is useful for ensuring accurate sensor measurements, but it increases cost of the system. To avoid redundancy, Shoukry et al. [10] relied on Chi-square detector to identify spoofing attempts on sensors. However, they did not provide any solution for recovery of CPS

©2017 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the United States Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
Approved for Public Release; Distribution Unlimited : 88ABW-2016-4645 ; Dated 20 Sep 2016
*DAC '17, 2017*
© 2017 ACM. 978-1-4503-4927-7/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3061639.3062241

system from attacks, but only detection. Our solution overcomes this limitation by enabling autonomous recovery in the presence of spoofing and DoS attack.

## 3 System Model

In this section, the dynamics of autonomous CPS is modeled as a discrete-time linear time-invariant (LTI) system without process noise and is given by the following equations:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{1}$$

$$\mathbf{y}_k = C\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

where, $\mathbf{x}_k \in \mathbb{R}^n$ is a real-valued system state vector at time $k$, $\mathbf{u}_k \in \mathbb{R}^m$ is a real-valued control input at $k$, and $\mathbf{y}_k \in \mathbb{R}^p$ is a real-valued sensor measurement vector at time $k$. $\mathbf{v}_k \sim N(0, R)$ is Gaussian *measurement* noise with zero *mean* and *covariances* $R = E[\mathbf{v}_k\, \mathbf{v}_k^T]$. Also, $A$ is the system matrix, $B$ is the control matrix, and $C$ is the output matrix. We assume that matrices *A, B, C* are known.

## 4 Attack Model

Attack on an autonomous CPS, such as a ground-vehicle, can be carried out either via physical access or remotely [7, 9, 10]. In our paper, we consider remote attacks, as attackers getting physical access of the vehicles internal operation may be an infeasible assumption. Remote exploitation can be carried out either on communication networks or on sensing units of the vehicle [7, 9, 10]. In this paper, we have considered attack on the sensing unit.

An autonomous vehicle has many sensors for collecting data of its internal and external environment. Compromising the data gathered by these sensors can impact the decisions made by the motion control units of the vehicle. By following the attack models described in [9, 10], we assume that the non-invasive attacker target the *external active sensors*, has limited knowledge of sensors firmware or software, and is in the vicinity of the attacked system. We also assume that they use resources that can conceal their remote spoofing attacks. Furthermore, the attack can be mounted on the target vehicle while its stationary or in motion.

Under these assumptions, we consider an autonomous vehicle, whose *active sensors* such as *ultrasonic, radar, or lidar* are under *Denial of Service (DoS)* attack or *delay injection based spoofing* attack. As delay is an inherent property of received signals of active sensors and is essential for making various measurements, distinction between original signal and received signal based on timing characteristics is not possible. Thus, to find the presence of such attacks, a detection method should be developed, which can differentiate original signals from the delay injected counterfeit signals and it should have high sensitivity.

The autonomous CPS model under sensor attack can be represented using the following equations:

$$\mathbf{x}'_{k+1} = A\mathbf{x}'_k + B\mathbf{u}'_k \tag{3}$$

$$\mathbf{y}'_k = C\mathbf{x}'_k + \mathbf{y}^a_k + \mathbf{v}_k \tag{4}$$

where,

$$\mathbf{y}^a_k = \begin{cases} 0 & \text{if } \textit{delay injection attack}, \\ r \in \mathbb{R}^p & \text{if } \textit{DoS attack} \end{cases}$$

In case of *delay injection attack*, $\mathbf{y}'_k$ is a counterfeit signal, which is similar to the normal signal except with a longer delay. An attacker can use inexpensive hardware and adversarial machine learning techniques to analyze the actual sensor signals and generate counterfeit signals. Whereas, in DoS attack, correct sensor measurements are suppressed with a stronger signal, $\mathbf{y}^a_k = r$, by method such as *jamming*. As such the system receives corrupted sensor measurements $\mathbf{y}'_k$. Next, we describe a model of radar sensor of a vehicle and also explain the effects of DoS attack and the delay injection attack on the sensor.

### 4.1 Attacks on Radar Sensor

We consider a model of a long-range automotive radar, which uses mm-wave for object detection. Such a radar measures distance and relative velocity to the target object. It has a carrier frequency of 77 GHz and is operated as Frequency Modulated Continuous Wave (FMCW) system for better sensitivity and simplicity of design [5].

For detecting an object, a mm-wave radar continuously transmits triangular frequency modulated waveforms. Due to Doppler effect, the signals received from the reflecting object (moving or stationary) by the radar are shifted in frequency from the transmitted signal by a delay, $\tau$. Subsequently, the received signal is mixed with a portion of the transmitted signal in a *mixture* of the radar's FMCW system. From the mixed output signal, two beat frequencies are extracted: $f_{b+}$ is for the positive portion of the slope and $f_{b-}$ is for negative portions of the slope of the mixed signal [5]. These two frequencies are determined using the following equations:

$$f_{b+} = \frac{2d}{c}\frac{Bs}{Ts} - \frac{2\dot{d}}{\lambda} \tag{5}$$

$$f_{b-} = \frac{2d}{c}\frac{Bs}{Ts} + \frac{2\dot{d}}{\lambda} \tag{6}$$

where $d$ is distance to the target object in meters, $c$ is the speed of light, delay $\tau = \frac{2d}{c}$, sweep bandwidth $B_s = 150$ Mhz, sweep time $T_s = 2$ msec, and wavelength $\lambda = 3.89$ mm.

From these two beat frequencies, distance and relative velocity to the object can be calculated as following:

$$d = \frac{cT_s}{4B_s}(f_{b+} + f_{b-}) \tag{7}$$

$$\Delta v = \frac{\lambda}{4}(f_{b-} - f_{b+}) \tag{8}$$

We can also calculate the power of the signal received from the target object by using the following equation:

$$P_r = \frac{P_t\, G\, \lambda^2 \sigma_s}{(4\pi)^3\, d^4\, L} \tag{9}$$

where, $P_r$ is the received signal power; $P_t = 10$ mW is the maximum transmitted signal power; $G = 28$ dBi is the antenna gain; $\sigma_s$ is the scattering cross section of the target object located at the distance $d$; and $L = 0.10$ db is the radar system losses.

To implement our challenge response authentication method, we modify the modulation unit of the radar to enable probing of the environment at random times. Subsequently, by implementing the Algorithm 2, we can detect the attack before it enters the domain of automotive control system.

**Denial of Service attack.** In this case, an attacker can use a self-screening jammer to transmit a signal with more power than the original signal received by the ACC's mm-wave radar. The power of the jamming signal is given by the following equation:

$$P_{jammer} = \frac{P_J\, G_J\, \lambda^2\, G\, B}{(4\pi)^2\, d^2\, B_J\, L_J} \tag{10}$$

where, $P_J$, $G_J$, $B_J$, $L_J$ represents jammer's peak power, antenna gain, operating bandwidth, and losses respectively. $B$ is the operating bandwidth of the mm-wave radar. Rest of the parameters, $\lambda$, $G$, and $d$, have same value as that of the radar. To carry out a successful attack, the following power ratio should be less than unity.

$$\frac{P_r}{P_{jammer}} = \frac{P_t\,\sigma_s\,B_J\,L_J}{4\pi\,P_J\,G_J\,d^2\,B\,L} \tag{11}$$

When the condition is valid, the radar will start receiving corrupted measurements, which can lead to vehicular collision.

**Delay injection attack.** In this case, an attacker replays a counterfeit signal with additional physical delay ($\tau$) to create an illusion that the object is further away than the actual distance. To carry out this attack, an adversary should have special hardware, which generates a signal with similar characteristics as the original reflected signal, except with more delay. Such a reflected signal will change values of beat frequencies (Eqn. 5, 6) extracted by the radar's receiver, which then will effect distance and velocity measurements. In the absence of true distance measurements, a vehicle will not be able to slow down or accelerate as desired.

# 5 Estimation of Sensor measurements and Attack Detection

We develop Algorithm 2 for attack detection and sensor measurement estimation. The detection method is based on a challenge response authentication technique and is particularly useful for detecting delay injection attacks, where the attacked signal has the same characteristics as that of the original signal, except with longer delay. Such an increase of delay in sensor measurements can disrupt normal operation of the system. Moreover, this method is also capable of detecting DoS attacks. Once an attack is detected, our estimation method, which is based on *recursive least square* algorithm (Algorithm 1), predicts sensor measurements for the duration of attack. With these estimated measurements, the controller determines an optimal control input for the CPS system. Unlike [10], our method enables recovery of the system from attack. Such a recovery mechanism is particularly useful for CPS that cannot be brought to a hault. Before explaining our method in details, we define the attack detection and estimation problem.

## 5.1 Problem Definition

Given a close loop system whose control input, $\mathbf{u}_k$, is determined according to sampled sensor measurements, $\mathbf{y}_k$ and the sensors are under DoS or delay injection attack producing corrupted measurements $(\mathbf{y}'_{k_1}, \ldots, \mathbf{y}'_{k_n})$ over a finite interval $[k_1, k_n], k_1 \neq 0, k_n < \infty$, we want to design a detector that can find the presence of an attack and an estimator that can predict outputs $(\hat{\mathbf{y}}_{k_1}, \ldots \hat{\mathbf{y}}_{k_n})$ during the duration of attack.

## 5.2 Detection of Attacks

For the challenge response authentication technique, we consider sensors, which are active e.x. radar, ultrasonic, lidar. Such sensors probe the environment with self-generated signals for gathering information. To incorporate a challenge on transmitted/probing signals of active sensors, we modify its modulation system with a pseudo-random binary modulation unit. As such, the transmitted signal, $p'(t)$, of the sensor is modulated as

$$p'(t) = m(t).p(t), \qquad m(t) \in [0, 1]$$

where, $m(t)$ is the binary modulation signal and $p(t)$ is the actual signal. Now, the modulated signal $p'(t)$ of the sensor changes values according to the following conditions:

$$p'(t) = \begin{cases} 0 & \text{if } m(t) = 0 \text{ for } t \in T_c, \\ p(t) & \text{if } m(t) = 1 \text{ for } t \in \mathbb{N} \setminus T_c. \end{cases}$$

where, $T_c$ is the set of time points $(t)$ at which outgoing probing signals are suppressed. At the corresponding sample time point $k$, received sensor measurements, $\mathbf{y}'_\mathbf{k}$ (sampled signal), of the receiving unit of an active sensor should be zero. For all the other time points, the modulated signal is same as the actual signal and the receiver produces a non-zero output. Only in the case of an attack, the receiver gives a non-zero output for $p'(t) = 0$. By comparing the expected output of the receiver at sampled time points $k \in T_k \subseteq T_c$ against the actual output, we can detect the presence or absence of an attack. A *DoS attack* can be easily detected by this method as the receiving unit of sensor will produce a large non-zero output ($\mathbf{y}^a_k = r$) when no signal was transmitted at time points $t \in T_c$. However, in the case of *delay injection attack*, a smart adversary attempting to transmit counterfeit signals with additional delay could conceal their attack when the modulated signal $p'(t)$ is zero at time points $T_c$. Now, due to the unavoidable time delay incurred by adversaries hardware, the time required to carry out the attack is always more than zero. As a result, attackers spoofing attempts can be detected using a simple detector that compares value of expected signal with the received signal.

Our method can also be used on passive sensors, but with additional hardware. Lines 7-9 of the Algorithm 2 represents the steps of our detection method.

## 5.3 Estimation of Sensor Measurements

After detecting the attack, we estimate future values of the active sensor by using the Recursive Least Square (RLS) estimation method as shown in Algorithm 1 [4]. The RLS algorithm estimate recursively in time the measurement values, $\{w_0, w_1, \ldots, w_n\}$.

**Algorithm 1** Recursive Least Square Estimation (RLSEstimate)

**Input:**
1: $h_i$ ▷ Entries of measurement matrix
2: $y'_i$ ▷ Corrupted Sensor Measurement
**Output:** $W = \{w_i\}$ ▷ Estimated values
3: Initialize: $w_0 \leftarrow 0, P_0 \leftarrow \delta I$;
4: **for** Each time instant, $k$ **do**
5: $\quad g \leftarrow h_k^T P_{(k-1)}$;
6: $\quad \gamma \leftarrow \lambda + g h_k$;
7: $\quad j_k \leftarrow \frac{g^T}{\gamma}$;
8: $\quad e_k \leftarrow y'_k - w^T_{(k-1)} h_k$;
9: $\quad w_k \leftarrow w_{(k-1)} + j_k e_k$;
10: $\quad P' \leftarrow j_k g$;
11: $\quad P_k \leftarrow \frac{P_{(k-1)} - P'}{\lambda}$;
12: **end for**
13: return $W$;

where, $\lambda$ is the *forgetting/weighting* factor taking values between $(0, 1)$, $\delta$ is a positive number (we consider it as 1), $g$ is the *gain vector*, $\gamma$ is the *conversion factor*, and $P$ is the *correlation matrix*. Here, $e_k$ is the error signal with $y'_k$ being the measurements from

attacked sensor, $w_{(k-1)}$ being the predicted values from the RLS estimator, and $h_k$ is the measurement matrix entries at time $k$. In this algorithm, given $y_i'$ and $h_i$, we find estimated values, $w_i$, at each time point $k = 0, 1, 2, \ldots n$, that minimizes *weighted* sum of square error between predicted output and the sensor measurements. At the end of each iteration, the RLS algorithm updates the estimation error covariance matrix $P_k$. During the duration of attack, we compute the control input, $\mathbf{u}_k$, with the estimated values, $w_i$. As such, we ensure that the system operates within the safety bounds given by us. Overall complexity of the RLS algorithm is $O(n^2)$ and it provides excellent performance while operating in real-time systems.

Prior to using the RLS algorithm, we detect the attack and perform pre-processing of data as shown in lines (6-10) of the Algorithm 2.

**Algorithm 2** Algorithm for attack detection and measurement estimation

**Input:**
1: $list_{zero}$ ▷ Time points $t \in T_c$ of zero sensor outputs.
2: $y_i'$ ▷ Corrupted Sensor Measurement
3: $h_i$ ▷ Entries of Measurement Matrix
**Output:** $attack_{detect}, list_{\hat{y}'}$ ▷ attack detection time ($t_{ad}$), and list of estimated sensor outputs ($\hat{y}_i'$).
4: $list_{y'}, list_{\hat{y}'}$;
5: $attack_{detect} \leftarrow False$;
6: **for** Each $y_i'$ Input **do**
7: add $y_i'$ to $list_{y'}$;
8: **if** $attack_{detect} == False$ **then**
9: **if** $y_i' \in list_{zero} \& Val(y_i') \neq 0$ **then**
10: $attack_{detect} \leftarrow True$;
11: $list_{\hat{y}'} \leftarrow RLSEstimate(h_i, y_i')$;
12: **end if**
13: **else**
14: $attack_{detect} \leftarrow False$;
15: $list_{y'}, list_{\hat{y}'}$;
16: **end if**
17: **end for**

## 6 Case Study on Car-Following

For demonstration, we build a car-following model in which a follower vehicle is equipped with an adaptive-cruise control (ACC) system and it follows a leader vehicle on the same lane. The ACC system uses *mm* wave radar sensor to measure relative distance and velocity to a preceding vehicle (presumably an attacker vehicle). We use the radar sensor model of Section 4.1 for this purpose. For our experiments, we consider parameters of Bosch LRR2 long-range ($2 \leq d \leq 200$ *meter*) mm-wave radar. To implement our challenge-response authentication (CRA) method in this radar, we modify its modulation unit. We assume that the sensor measuring velocity of the follower vehicle ($v_{F_v}$) is trusted. Our car-following model, modified CRA radar, and the detection as well as estimation methods are shown in Figure 1. Subsequently, simulation results of attacks on the follower vehicle and of our detection and estimation methods are shown in Figures 2 and 3.

### 6.1 Car-Following Model of Ground Vehicles

The ACC system of the follower vehicle operates in two modes (i) speed control and (ii) spacing control. In the absence of preceding vehicles, the ACC system operates the vehicle in the speed control mode, where it drives at a user-set speed ($v_{set}$). When a preceding vehicle is detected on the road by the radar, the ACC system of the follower vehicle decides on whether to continue driving in the speed control mode based on distance between the vehicles. If the distance is less than a desired value ($d_{des}$), given by Eqn. 12, the ACC system switches to spacing mode. In this mode, the desired distance, which is proportional to the headway time ($\tau_h$ = 3 sec) between the vehicles, minimum stopping distance ($d_0$ = 5 m), and speed of the follower vehicle ($v_{F_v k}$), is maintained by controlling both throttle and brakes.

$$d_{des_{(k)}} = d_0 + \tau_h v_{F_{v(k)}} \tag{12}$$

We model longitudinal control of the ACC equipped follower vehicle as a hierarchical control architecture, consisting of an upper level controller and a lower level controller, shown in Figure 1 [6]. The upper level controller determines the desired longitudinal acceleration ($a_{des}$) according to distance ($d$), relative velocity $\triangle v$ between a leader vehicle ($L_v$) and a follower vehicle ($F_v$) (measured using a radar) and speed of the follower vehicle ($v_{F_v}$). The upper level linear output feedback controller is implemented based on a constant time headway (CTH) policy, which states that the desired speed of the follower vehicle ($v_{des}$) should be proportional to the inter-vehicular distance ($d$) and inversely proportional to the headway time ($\tau_h$). The controllers output dynamics based on CTH policy and transfer function of lower level controller is given by the following discrete time equation [2],

$$v_{des_{(k+1)}} = \frac{1}{K_L} v_{F_{v(k)}} + \frac{T_L}{\tau_h . K_L} \Delta v_{(k)} + \frac{T_L}{k . \tau_h . K_L} \Delta d_{(k)} \tag{13}$$

where, $K_L$ = 1.0 is the system gain and $T_L$ = 1.008 is the time constant for the follower vehicle [6]. The clearance error between the vehicles is $\triangle d_{(k)} = d_{(k)} - d_{des_{(k)}}$, relative speed is $\triangle v_{(k)} = v_{L_{v(k)}} - v_{F_{v(k)}}$, $k$ is discrete time in seconds. Here, $v_{L_v}$ is speed of the leader vehicle. From the value of $v_{des}$, we derive the output, $a_{des}$, of the upper level controller.

The lower level controller of the ACC system determines the acceleration of pedal ($a_{pedal}$) and brake pressure ($P_{brake}$) of the follower vehicle to ensures the desired acceleration $a_{des}$ is tracked by actual acceleration $a_{F_v}$. The closed loop transfer function of this controller with follower vehicle as the plant is given by the following first-order equation:

$$a_{F_v} = \frac{K_L}{T_L s + 1} a_{des} \tag{14}$$

While designing the upper level controller, internal and external disturbances are neglected to ensure the lower level controller works correctly and satisfy dynamics of Eqn. 14. Similarly, nonlinearity at the lower level controller are compensated using inverse longitudinal dynamics. Now, our car-following model to simulate vehicular traffic flow (longitudinal) dynamics is built by enhancing the intelligent-driver model (IDM) with the hierarchical control model of ACC equipped follower vehicle, as shown in Figure 1. With the help of our model, we can describe acceleration and deceleration among vehicles in a satisfactory way. To find continuously changing velocity of the leader vehicle, we use the following equation

$$v_{F_{l(k+1)}} = v_{F_{l(k)}} + a_{F_{l(k+1)}} \tag{15}$$

where, $a_{F_l}$ and $v_{F_l}$ are acceleration and velocity of the leader vehicle respectively. Similarly, we derive values of actual and desired

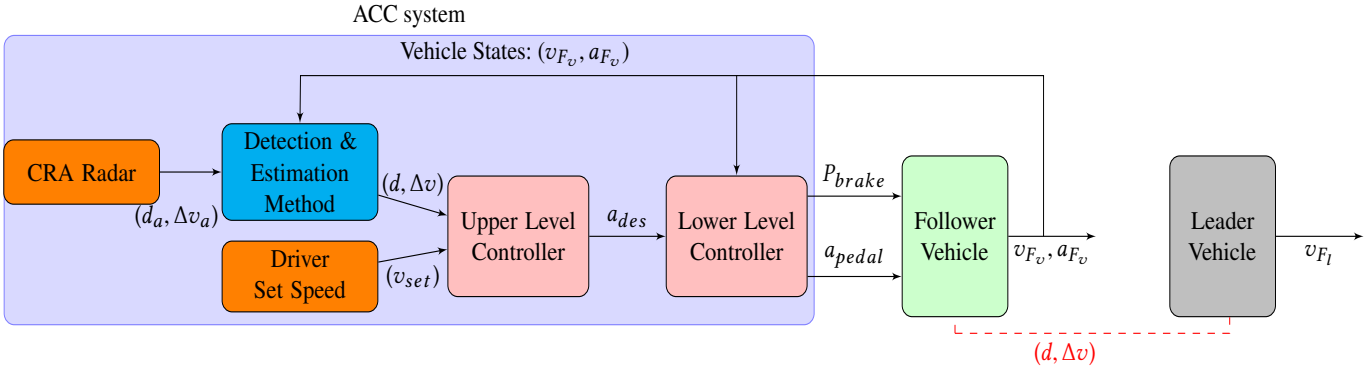



**Figure 1.** Car-following model with hierarchical control architecture of ACC system & Detection, Estimation Method.

acceleration ($a_{F_{v,des}}$) of the follower vehicle by using the following equation

$$a_{F_{v,des(k+1)}} = v_{v,des_{(k+1)}} - v_{v,des_{(k)}} \tag{16}$$

To find positions of leader ($x_{F_l}$) and follower ($x_{F_v}$) vehicles, we use the following equation

$$x_{F_{l,v(k+1)}} = x_{F_{l,v(k)}} + v_{F_{l,v(k+1)}} + \frac{1}{2} a_{F_{l,v(k+1)}} \tag{17}$$

In our simulation, we use Eqn.(17) to measure distance, $d_{(k+1)} = x_{F_{l(k+1)}} - x_{F_{v(k+1)}}$, between the leader and the follower vehicles. In an actual scenario, values of $d$ and $\triangle v$ of the car-following model are calculated using the radar Eqn. 7 and Eqn. 8. From Figure 1, we can see that the internal states of the ACC equipped follower vehicle are $a_{des}, P_{brake}$ and $a_{pedal}$. Corrupted distance and relative velocity measurements of radar, effects calculation of state, $a_{des}$, which then influences output ($v_{F_v}$) of the system.

### 6.2 Simulation and Results

The goal of our simulation is to demonstrate the attacks on an ACC equipped follower vehicle and show effect of our detection and estimation methods on velocity of the vehicle. We consider two car-following scenarios: (i) the leader vehicle decelerates at a constant acceleration of -0.1082 $m/sec^2$ (Figure 2) and (ii) the leader vehicle decelerates and accelerates at -0.1082 $m/sec^2$ and +0.012 $m/sec^2$ respectively (Figure 3). The follower vehicle has to slow down accordingly to ensure the inter-vehicular distance is greater than the desired distance ($d_{des}$) to avoid rear end collision. We consider 65 miles/hr and $v_{set}$ = 67 miles/hr as the initial velocities of the leader and the follower vehicles respectively. The leader starts slowing down when the initial distance between the vehicles is 100m. For such a scenario, an adversaries intention is to provide corrupted data ($d_a, \triangle v_a$ of Figure 1) to the ACC system's radar sensor for causing a collision.

We simulate the attacks and the car-following scenario in MATLAB. For design, simulation, and analysis of the radar sensor, we use the Phased Array System Toolbox. The root MUSIC algorithm is used to extract beat frequencies from radar data. We derive values of distance and relative velocity between vehicles from the measured beat frequencies.

• *Follower vehicle under DoS attack*

To carry out this attack, we consider a self-screening jammer whose $P_J$ = 100 mW, $G_J$ = 10 dbi, $B_J$ = 155 Mhz, and $L_J$ = 0.10 db. When the follower vehicle is attacked with such a jammer, measurements of the radar sensor becomes corrupted. In the Figure 2a and 3a, we show that the DoS attack begins at time $k$ = 182 sec, after which the sensor receives very high value of corrupted distance and velocity measurements. Prior to it, the attacked signal follows the actual reflected signal. Due to incorporation of our CRA based detection method on the radar, it receives measurements whose values are zero at certain time instances such as at $k$ = 15, 50, 175 sec [1] of Figures 2a and 3a. This occurs because the radar do not transmit any data at those time points. As, at $k$ = 182 sec, the attacked signal value is not zero, our detector could detect the attack and notify it to the system. Subsequently, after $k$ = 182 sec till the end of attack, our estimation method provides distance and relative velocity data to the upper level controller of the ACC system. As such, our method prevents the vehicle from performing task with undesired consequences.

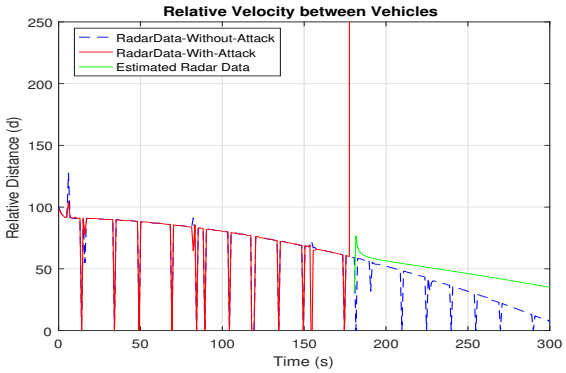
• *Follower vehicle under Delay injection attack*

For the delay injection attack, we consider a scenario where the adversary generates counterfeit signals to increase distance between vehicles. In our case, the distance measurements received by the radar after $k$ = 180 sec are 6 meter more than the actual distance. Such corrupted values of distance effects the desired acceleration ($a_{des}$) measurements of the ACC controller. As a result, the follower vehicle does not slow down as desired, which can be seen in Figures 2b and 3b. On using our detection method in the same way as was in the DoS attack, we observe that the attack occurs at $k$ = 182 sec. Due to the attack, the velocity of the follower increases and the distance reduces between the vehicles. To make the ACC equipped follower vehicle drive properly during the duration of the attack, we use our estimation method, which corrects the distance and relative velocity measurements after the attack is detected.

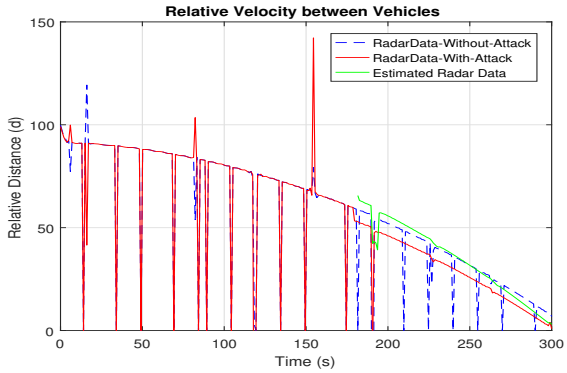
• *Results*

With the help of the detection method, we were able to detect both the attacks at $k$ = 182 sec. Subsequently, we use recursive least square algorithm to predict distance and relative velocity values for the duration of the attack ($k$ = 182 sec to $k$ = 300 sec). The algorithm had run-times of 1.2e+7 nanoseconds and 1.3e+7 nanoseconds

[1] The spikes going to zero in Figure 2 are indication of radar sensor not producing any output at challenge times $k$ = 15, 50, 175 etc. They do not imply relative velocity between the vehicles going to zero at those times. Any other value of spikes in the Figure 2 are indications of noise in radar data.

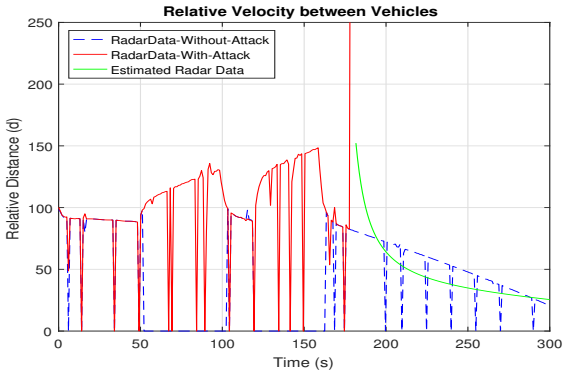



**(a)** DoS Attack on reflected signal of radar and Detection, Estimation Output

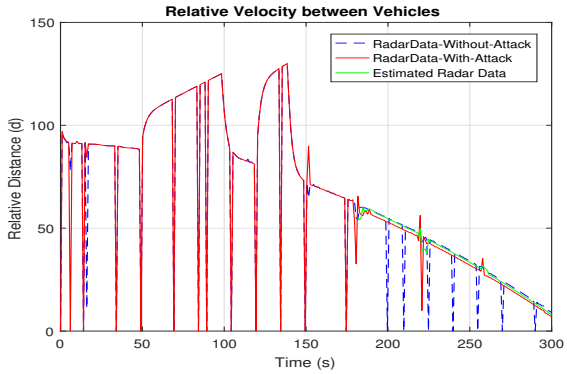



**(b)** Delay Attack on reflected signal of radar and Detection, Estimation Output

**Figure 2.** Plots of Attacks and Detection, Estimation Outputs with constant deceleration of leader vehicle




**(a)** DoS Attack on reflected signal of radar and Detection, Estimation Output




**(b)** Delay Attack on reflected signal of radar and Detection, Estimation Output

**Figure 3.** Plots of Attacks and Detection, Estimation Outputs with acceleration and deceleration of leader vehicle

for both cases of jamming and delay injection attacks respectively. Our detection method did not produce any false positives or false negatives for both the attack scenarios.

## 7 Conclusion

In this paper, we introduce a challenge-response authentication based method for detection of two types of attacks: the Denial of Service (DoS) and the delay injection, on active sensors of autonomous systems. The recursive least square approach is used for estimation of sensor measurements when it is under attack. With these estimated measurements, safe control inputs of the autonomous CPS are derived, which enables the system to recover and operate safely in the presence of attacks. A case study was presented to show resiliency of adaptive cruise control system of ground vehicle, leveraging our proposed solutions to counter these attacks. However, the detection method fails when an adversary with adequate resources can sample the incoming signals from active sensors faster than the defender. Our future research will address this limitation and we will provide defence mechanisms to prevent such adversaries from attacking active sensors of autonomous systems. We will also extend our case study on autonomous ground vehicle to include a non-linear system model with lateral dynamics.

## References


[1] M. S. Chong, M. Wakaiki, and J. P. Hespanha. 2015. Observability of linear systems under adversarial attacks. In *2015 ACC*. 2439–2444.

[2] Gerald H Engelman and et al. 2001. Adaptive vehicle cruise control system and methodology. (May 15 2001).

[3] H. Fawzi, P. Tabuada, and S. Diggavi. 2014. Secure Estimation and Control for Cyber-Physical Systems Under Adversarial Attacks. *IEEE TAC* 59, 6 (June 2014), 1454–1467.

[4] Simon Haykin. 1996. *Adaptive Filter Theory (3rd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[5] Vipul Jain and Payam Heydari. 2013. Radar Fundamentals. In *Automotive Radar Sensors in Silicon Technologies*. Springer New York, 5–11.

[6] Shengbo Li, Keqiang Li, Rajesh Rajamani, and Jianqiang Wang. 2011. Model predictive multi-objective vehicular adaptive cruise control. *IEEE TCST* 19, 3 (2011), 556–566.

[7] Chung-Wei Lin and et al. 2015. Security-Aware Design Methodology and Optimization for Automotive Systems. *ACM TODAES* 21, 1 (2015), 18.

[8] Junkil Park, Radoslav Ivanov, James Weimer, Miroslav Pajic, and Insup Lee. 2015. Sensor attack detection in the presence of transient faults. In *ACM/IEEE Sixth ICCPS*. ACM, 1–10.

[9] Jonathan Petit and et al. November, 2015. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR. In *Black Hat Europe*.

[10] Yasser Shoukry and et al. PyCRA: Physical Challenge-Response Authentication For Active Sensors Under Spoofing Attacks. In *22Nd ACM SIGSAC CCS, 2015*. 1004–1015.

[11] Y. Shoukry and P. Tabuada. 2014. Event-triggered projected Luenberger observer for linear systems under sparse sensor attacks. In *53rd IEEE CDC*. 3548–3553.

[12] Ashish Tiwari and et al. 2014. Safety Envelope for Security. In *HiCoNS*. 85–94.