# Security for Safety: A Path Toward Building Trusted Autonomous Vehicles

***(Invited Paper)***

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

Feng Yu
University of Central Florida
yfeng@knights.ucf.edu

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

Yaodan Hu
University of Florida
yaodan.cindy.hu@gmail.com

Yier Jin
University of Florida
yier.jin@ece.ufl.edu

## ABSTRACT

Automotive systems have always been designed with safety in mind. In this regard, the functional safety standard, ISO 26262, was drafted with the intention of minimizing risk due to random hardware faults or systematic failure in design of electrical and electronic components of an automobile. However, growing complexity of a modern car has added another potential point of failure in the form of cyber or sensor attacks. Recently, researchers have demonstrated that vulnerability in vehicle's software or sensing units could enable them to remotely alter the intended operation of the vehicle. As such, in addition to safety, security should be considered as an important design goal. However, designing security solutions without the consideration of safety objectives could result in potential hazards. Consequently, in this paper we propose the notion of *security for safety* and show that by integrating safety conditions with our system-level security solution, which comprises of a modified Kalman filter and a Chi-squared detector, we can prevent potential hazards that could occur due to violation of safety objectives during an attack. Furthermore, with the help of a car-following case study, where the follower car is equipped with an adaptive-cruise control unit, we show that our proposed system-level security solution preserves the safety constraints and prevent collision between vehicle while under sensor attack.

## KEYWORDS

Cyber-Physical System, Self-Driving Car, Security, Safety, Estimation, Sensor attack

**ACM Reference Format:**
Raj Gautam Dutta, Feng Yu, Teng Zhang, Yaodan Hu, and Yier Jin. 2018. Security for Safety: A Path Toward Building Trusted Autonomous Vehicles *(Invited Paper)*. In *IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (ICCAD '18), November 5–8, 2018, San Diego, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3240765.3243496


*ICCAD '18, November 5–8, 2018, San Diego, CA, USA*

ACM ISBN 978-1-4503-5950-4/18/11…$15.00
https://doi.org/10.1145/3240765.3243496

## 1 INTRODUCTION

Safety-critical systems comprising of computers and network components have increasingly found their applications in various industrial sectors such as healthcare, manufacturing, energy, transportation, and space. Such systems are required to operate safely as their failure could lead to loss of life, significant damage of property, or the environment. However, as these systems get larger, more complex, and sophisticated, the range of vectors via which attack and errors can be introduced into the system increases, which makes ensuring safety and security difficult. Furthermore, these systems are built by integrating components designed by various vendors, which adds to the challenge of verifying safety and security of each component.

One application which is extremely complex and consist of heterogeneous components from different vendors is the self-driving car. Due to economic, social, and environmental benefits, government and industry are investing a lot of effort on this technology. Many autonomous features already exist in modern cars such as adaptive cruise control (ACC), automatic forward collision braking, automatic parking, autopilot, and lane-keep assistant. However, as a result of these enhancements, these vehicles have become susceptible to various safety and security threats. Some of the implication of these hazards are unwanted steering, suppression of evasive maneuvers, and faulty navigation.

Consequently, the ISO 26262 international standard for functional safety in passenger vehicles was introduced to guide the design of safety solutions. While these approaches enable the design of systems that are protected against hazards arising from random hardware or software failures, the potential for the safety of the system to be compromised by attacks is currently not considered. Recently, researchers have demonstrated successful spoofing of sensor signals such as of GPS, radar, lidar, and ultrasonic along with attack on cameras [8]. As such, autonomous CPS, which rely heavily on sensing units for decision making, remain vulnerable to such attacks. Thus, additional measures are required to prevent vehicles from malicious attacks on sensors.

Thus, to ensure safety of the vehicular system during attack on sensor measurements, we propose a system-level security solution with integrated safety constraints, that is inspired from the linear dynamical systems literature. In our approach, we model the vehicular dynamics as a stochastic linear system with zero mean, white Gaussian noise. Malicious False Data Injection (FDI) attack

corrupt the measurements of the sensors of the considered system. Consequently, we develop an attack resilient estimator in the Bayesian framework and combine it with the Chi-squared detector to address the problem of simultaneous attack detection and state estimation that preserve safety constraints. For our method, the estimation error asymptotically converges to zero when there is no attack and has an upper bound on the error during attack. By bounding the estimation error, we were able to obtain *approximate* state estimates, when sensors of the system were compromised. The unique features of our method are:

- Our method can recursively estimate states by considering safety constraints and perform better than the standard Kalman filter and Robust Kalman filter of [6] against DoS and FDI attacks.
- Our method can *approximately* (within an error bound) reconstruct the states when the sensors were attacked.

The rest of the paper is organized as follows: The state-of-the-art is discussed in Section 2. In Section 3, we formulate the problem and describe the stochastic linear model of the system and the attack model. Our resilient state estimation algorithm is explained in Section 4. The effectiveness of our estimation method is demonstrated on a car-following case study in Section 5. Finally, conclusions are drawn in Section 6.

## 2 RELATED WORK

Integrating security with safety involves many challenges [5]. Due to different objectives, security and safety solutions for vehicles are often designed independently. To mitigate this issue, recent efforts have been made to integrate them at the design stage. Burton *et al.* [3] extended the ISO 26262 safety standard by considering attacks as the third source of hazard. In their work, the lane keeping assist system was considered as an application of interest. Hazards causing lane departure were first identified and safety goals were decided accordingly. Then, possible causes of violations of safety goals were identified and new goals to prevent them were defined. By repeating this procedure, security was integrated with safety during the system design stage. Plosz *et al.* [9] proposed a method for combining security with safety and demonstrated it on a remote engine testing system. In their approach, a Data Flow Diagram (DFD) was used to represent interaction among system components, thereby enabling the assessment of security and safety in the same system model. This was done to avoid independent assessment of security and safety. Schmittner *et al.* [12] proposed Failure Mode, Vulnerabilities and Effect Analysis (FMVEA) method, which is an extention of the Failure Mode Effect Analysis (FMEA) approach with a security model that captured failure of security attributes associated with components.

Our proposed method is different from the existing approaches in a way that we leverage estimation theory to design a system-level security solution that preserves safety goals of the system. Along these lines, robust and resilient state estimation methods, which can withstand sensor measurement attack have been developed. The set of literature relevant to our work consider additive zero mean, Gaussian white noise with unbounded attack signal in the system model [4, 6, 7]. Forti *et al.* [4] designed a computationally expensive hybrid Bernoulli filter (in Bayesian Framework) to simultaneously detect attacks (signal, packet substitution, and extra packet injection) and estimate system states. Their filter could recursively update in real-time the joint posterior density of the attacks and of the state vector, provided all measurement were available upto that time. A Robust Kalman filter (RKF) for estimating states during sparse sensor disturbances was developed in [6]. They modified the measurement update equation of the standard Kalman filter with the solution of $\ell_1$-based convex optimization problem. However, they did not provide any optimality guarantee. Mishra *et al.* [7] used a bank of Kalman filters for secure state estimation of noisy linear dynamical system subjected to sparse data injection attack. They identify the subset of sensors that are not attacked by using a *block residue test* and use their outputs to calculate the secure estimate. However, they assume that all sensors data are not corrupted and the number of Kalman filters used in their method is dependent on the number of attacked sensors.

## 3 PRELIMINARIES

### 3.1 System Model

We model the dynamics of the CPS as a linear time-invariant (LTI) system with process and measurement noise, which is described by the following equations:

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

$$y_k = Cx_k + v_k \tag{2}$$

where, $x_k \in \mathbb{R}^n$ is a real-valued system state vector at time $k$, $u_k \in \mathbb{R}^m$ is a real-valued control input vector, and $y_k \in \mathbb{R}^q$ is a real-valued sensor measurement vector, $w_k \sim \mathcal{N}(0, \Sigma_w)$ is the additive white Gaussian *system* noise with zero mean and covariance $\Sigma_w$, and $v_k \sim \mathcal{N}(0, \Sigma_v)$ is the additive white Gaussian *measurement* noise with zero mean and covariance $\Sigma_v$ (in this work, $\mathcal{N}(\mu, \Sigma)$ represents a Gaussian distribution with mean $\mu$ and covariance $\Sigma$). Both the noises are assumed to be independent of each other. Here, $A$ is the system matrix and $B$, $C$ are the transformation matrices. We assume that the time-invariant matrices $A$, $B$, $C$ are known.

To mitigate the impact of noise on the estimation accuracy, a Kalman Filter (KF) first predicts the state of the system then combines the prediction with latest measurements to obtain the final estimation. The prediction stage is represented by the following equations:

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k; \qquad P_{k+1|k} = AP_kA^T + \Sigma_w$$

where, $P_k$ is the estimation error covariance matrix.

The measurement update equations of the KF are:
$z_{k+1} = y_{k+1} - C\hat{x}_{k+1|k}$; $K_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + \Sigma_v)^{-1}$;
$\hat{x}_{k+1} = \hat{x}_{k+1|k} + K_{k+1}z_{k+1}$, and $P_{k+1} = (I - K_{k+1}C)P_{k+1|k}$

where, $z_k \in \mathbb{R}^q$ is the estimation residue (without attack) at time $k$ with Gaussian distribution $z_k \sim \mathcal{N}(0, CP_{k+1|k}C^T + \Sigma_v)$ and $K_k$ is the time-varying Kalman gain matrix. When the matrices $(A, B)$ and $(A, C)$ of the system are assumed to be stabilizable and detectable respectively, we get a *steady-state KF*, whose time-varying error covariance matrices $P_{k|k-1}$ converges to $P$ i.e. $P = \lim_{k\to\infty} P_{k|k-1}$ and the time-varying Kalman gain converges to a constant value i.e. $K = \lim_{k\to\infty} K_k$ and thus it reduces to $K = PC^T(CPC^T + \Sigma_v)^{-1}$. The estimation error of the filter is given by $e_k = (x_k - \hat{x}_k)$.

The $\chi^2$ detector is generally used with KF in a control system to detect attacks [2]. Based on the estimation residue of the *steady-state* KF, we can define a function, $g_k = z_k^T(CPC^T + \Sigma_v)z_k$, whose value is greater than a user specified threshold ($\eta$) i.e. $g_k > \eta$, then an alarm is triggered by the $\chi^2$ detector. Now, the probability of triggering an alarm at time $k$ can be given as $\beta_k \triangleq P(g_k > \eta)$. When the system is operating normally, the value of $\beta_k$ is a constant, $\alpha$, which is the false alarm rate of the detector. Any other value of $\beta_k$ correspond to an attack.

### 3.2 Attack Model

We consider an adversary whose intention is to make the system violate safety constraints such as safe following distance among vehicles. We assume that the adversary has knowledge of the model of the plant and the filtering algorithm used during the feedback control i.e. static matrices of the system A, B, and C, filter gain K, and distribution of noises are known. We also assume that an adversary can manipulate any number of sensors and actuators of the system. For simplicity of analysis, we consider attacks on sensors of the system and restrict our attention to the measurement/output equation: $y_k = Cx_k + v_k$. However, our results can easily be extended to the case where attacks are on actuator inputs.

In this paper, we consider False Data Injection (FDI) attack and make the following assumptions about it:

- Attacks do not corrupt all the measurements after its initiation;
- Any of these attacks can be carried out at any time for a finite duration, but not simultaneously.

Both of these assumptions are made to correspond to a real-world attack scenario, where an adversary needs to operate within the constrains of limited resource and access to measurements and consider continuously changing dynamics of the autonomous system with respect to its environment.

### 3.3 Problem Definition

Therefore, our objective stated formally is:

*Given a stochastic linear time-invariant system whose sampled sensor measurements $y_k$ are under FDI attack, producing corrupted measurements $y'_k$ over a finite interval $[k_1, k_n], k_1 \neq 0, k_n < \infty$, we want to use a detector that can find the presence of an attack and design a filter that can estimate system states with bounded estimation error to prevent safety constraint violation during the duration of attack.*

## 4 RESILIENT STATE ESTIMATION

In this section, we explain our Bayesian inspired computationally efficient recursive algorithm, which is combined with the Chi-squared ($\chi^2$) detector to simultaneously detect and estimates states that are resilient against FDI attack and prevent violation of safety constraints. In our algorithm design, we assume that the computational delays incurred by the chi-squared detector prior to estimation is minimal and hence are neglected [1].

The standard Kalman filter (KF) is robust to noise, but it has not been designed to mitigate the effect of adversarial attacks on state estimates. The $\chi^2$ detector has been used with KF to detect attack, but to the best of our knowledge, there has been few attempts on recursively estimating $x_k$ during an attack. Toward this objective, we propose the following computationally efficient algorithm, that combines the $\chi^2$ detector with our Bayesian inspired estimator.

First, let us briefly review the Bayesian interpretation of KF, which uses Bayes rule, $p(a|b)p(b) = p(b|a)p(a)$, to express the posterior probability in terms of the likelihood and the prior. Assuming that the distribution of $x_k$ follows the Gaussian distribution $\mathcal{N}(\hat{x}_k, P_k)$, we obtain a prior distribution that is $P(x_{k+1}|x_k) \sim \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k})$. By combining this prior information with $y_{k+1} \sim \mathcal{N}(Cx_{k+1}, \Sigma_v)$ and Bayes rule, we can show that the posterior distribution is $x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1}, P_{k+1})$, where $\hat{x}_{k+1}$ and $P_{k+1}$ are as defined in the KF.

As a result, given $x_k \sim \mathcal{N}(\hat{x}_k, P_k)$, $y_{k+1} = Cx_{k+1} + v_k$ should follow the distribution $\mathcal{N}(C\hat{x}_{k+1|k}, CP_{k+1|k}C^T + \Sigma_v)$. To detect the attack in $y_{k+1}$, a $\chi^2$ detector is applied to $z_{k+1}^T(CP_{k+1|k}C^T + \Sigma_v)^{-1}z_{k+1}$.

To apply the Bayes rule for resilient estimation, we again assume that for each $k$, $x_k \sim \mathcal{N}(\hat{x}_k, P_k)$. However, depending on whether there is an attack at $y_{k+1}$, the estimation of the posterior distribution of $x_{k+1}$ would be different.

We first derive the prior distribution of $x_{k+1}$. Let

$$P_{k+1|k} = AP_kA^T + \Sigma_w, \hat{x}_{k+1|k} = A\hat{x}_k + Bu_k, \tag{3}$$

then the prior information of $x_{k+1}$ is $x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k})$. As a result, when $x_k$ is given, to detect whether $y_{k+1}$ is attacked, we should apply the $\chi^2$ detector to $g_{k+1} = (y_{k+1} - C\hat{x}_{k+1|k})^T P_{k+1|k}^{-1}(y_{k+1} - C\hat{x}_{k+1|k})$. As the residue, $g_{k+1}$, satisfies $\chi^2$ distribution with $q-1$ degrees of freedom, we can determine a threshold, $\eta$, (based on our chosen probability of interest) for the detector such that an alarm for attack is triggered when $g_{k+1} > \eta$.

When an attack is not detected at $y_{k+1}$, then we combine the prior distribution of $x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k})$ with the information $y_{k+1} \sim \mathcal{N}(Cx_{k+1}, \Sigma_v)$ to obtain the posterior distribution of $x_{k+1}$. Applying Bayes rule, the posterior distribution of $x_{k+1}$ is proportional to the product of these two probability density functions:

$$x_{k+1} \sim \exp\Big( -\frac{1}{2}\Big[(y_{k+1} - Cx_{k+1})^T\Sigma_v^{-1}(y_{k+1} - Cx_{k+1}) + (x_{k+1} - \hat{x}_{k+1|k})^T P_{k+1|k}^{-1}(x_{k+1} - \hat{x}_{k+1|k})\Big]\Big)$$

By calculation, we obtain $x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1}, P_{k+1})$ with

$$P_{k+1} = \left(C^T\Sigma_v^{-1}C + P_{k+1|k}^{-1}\right)^{-1} \tag{4}$$

$$\hat{x}_{k+1} = P_{k+1}\left(C^T\Sigma_v^{-1}y_{k+1} + P_{k+1|k}^{-1}\hat{x}_{k+1|k}\right). \tag{5}$$

When the detector suggests that there is an attack at $y_{k+1}$, then we propose to drop the information of $y_{k+1}$. As a result, the posterior distribution of $x_{k+1}$ is the same as its prior distribution, that is, $x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1}, P_{k+1})$, where $\hat{x}_{k+1} = \hat{x}_{k+1|k}$, $P_{k+1} = P_{k+1|k}$.

Our procedure is summarized in Algorithm 1 and Figure 1 and its computational time complexity is $O(\max(n, q)^3)$, same as of KF. The proposed method resembles an "event-triggered" approach: when a detection alarm is triggered and safety constraint is violated, update the Kalman filter in open-loop; otherwise, update the KF as usual. Our solution is simple, but effective against sensor attacks. We also use the Bayesian perspective to derive $\hat{x}_k$ and $P_k$ during filters open-loop operation.

**Algorithm 1** Attack Detection & Resilient State Estimation

**Input:** Observation $\{y'_k\}_{k\geq 1} \in \mathbb{R}^q$; detection threshold $\eta$; model parameters $A, B, C, \Sigma_v, \Sigma_w, \{u_k\}_{k\geq 0}$; safety constraints $c$.
**Output:** Estimated values $\hat{x}_k, k \geq 1$
**Initialize:** $\hat{x}_0 \in \mathbb{R}^n$ and $P_0 \in \mathbb{R}^{n\times n}$;

1: **for** $k = 0, 1, 2, \cdots$ **do**
2: Calculate $\hat{x}_{k+1|k}$ and $P_{k+1|k}$ using (3).
3: Apply $\chi^2$ detector to
4: $g_{k+1} = (y_{k+1} - C\hat{x}_{k+1|k})^T (CP_{k+1|k}C^T + \Sigma_v)^{-1}(y_{k+1} - C\hat{x}_{k+1|k})$
5: **if** $g_{k+1} > \eta$ and $\hat{x}_{k+1|k} \geq c$ **then** $P_{k+1} = P_{k+1|k}$ and
6: $\hat{x}_{k+1} = \hat{x}_{k+1|k}$.
7: **else** Calculate $P_{k+1}$ and $\hat{x}_{k+1}$ using (4) and (5)
8: **end if**
9: **Return**: $\hat{x}_{k+1}$.
10: **end for**

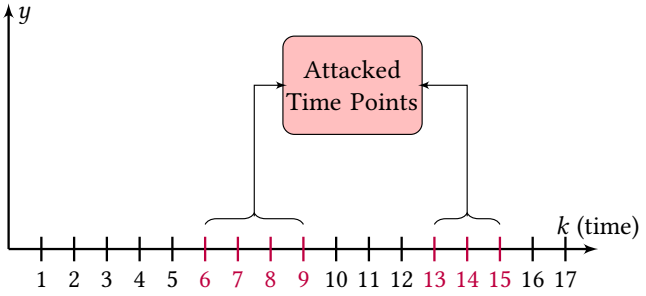


**Figure 1: Idea behind estimation in Algorithm 1: After $\chi^2$ detect the attack at $k$ = (6-9) and (13-15) and safety condition is violated, sensor measurements at these time points are discarded and state estimation is done using the last good sensor value (at time points 5 and 12). For other values of $k$, Algorithm 1 considers the measurements for estimation.**

Compared to an $\ell_1$ optimization based RKF [6], our method is simpler and more computationally efficient, as we do not solve an optimization problem. In addition, RKF fixes $P_k$ to be the $P$ of the steady-state KF, which is different from our setting where $P_k$ is derived from the Bayesian perspective and could be different for different $k$. As shown later in Section 5.2, our proposed algorithm outperforms RKF by producing smaller estimation error.

## 5 CASE STUDY

For demonstration, we consider a car-following scenario (resembling a platoon of two vehicles), shown in Figure 2, in which a follower vehicle ($f$) is equipped with an adaptive-cruise control (ACC) unit and it follows a leader vehicle ($l$) on the same lane. The ACC system uses mm wave radar (external) such as Bosch LRR2 long-range ($2 \leq d \leq 200$ *meter*) and internal sensors to measure position of the preceding and follower vehicle. We consider attacks that corrupt position ($x^{(f)}_{attack}$) measurements of the follower vehicle, as shown in Figure 3. As such, the goal of our Algorithm 1 is to minimize the effect of corrupted sensor data on the inputs (relative distance) of the ACC controller. Our method can be also extended to multiple vehicle platoon. In such a case, our Algorithm 1 will operate on each vehicle by considering details of the state-space model of the platoon and network topology. In the following subsections, we elaborate on the car-following setup and discuss our simulation results.

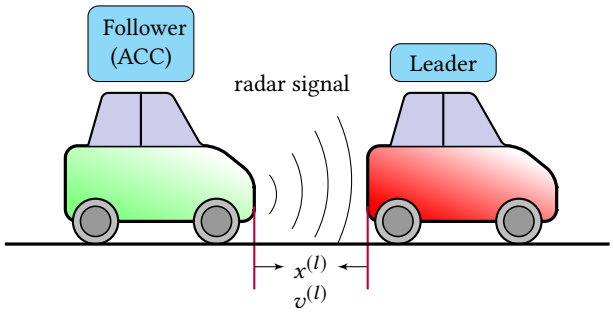


**Figure 2: Car-following with ACC equipped follower whose radar measures position ($x^{(l)}$) and velocity ($v^{(l)}$) of the leader**

### 5.1 Leader and Follower Vehicle Models

We use kinematic equations to describe leader vehicle ($l$) dynamics. Changing velocity of the leader is generated using,

$$v^{(l)}_{k+1} = v^{(l)}_k + a^{(l)}\Delta t \tag{6}$$

where, $k \in \{0, 1, 2, \dots\}$ is the number of time iteration and $a^{(l)}, v^{(l)}$ are constant acceleration and velocity respectively. Here, $\Delta t = 0.01$ is the size of time increment. The position of the leader ($x^{(l)}$) at any time can be determined using the equation,

$$x^{(l)}_{k+1} = x^{(l)}_k + v^{(l)}_k \Delta t + \frac{1}{2}a^{(l)}(\Delta t)^2 \tag{7}$$

These velocity and position measurements of the leader are captured by the radar of the follower vehicle.

The ACC system (shown in Figure 3) drives the follower vehicle ($f$) at a user-set speed ($v_{set}$) in the absence of a preceding/leader vehicle. When a vehicle is detected, the ACC unit uses the constant time-gap (CTH) spacing policy (8) for maintaining desired inter-vehicular distance ($d^{(l,f)}$) to its preceding vehicle,

$$d^{(l,f)} = d_r + hv^{(f)} \tag{8}$$

where, $h$ is the headway time ($h = 3$ sec) between the vehicles, $d_r$ is the minimum stopping distance ($d_r = 5$ m), and speed of the follower vehicle is $v^{(f)}$. According to [11], such a spacing-policy improves traffic throughput and safety. Here, (8) is the *safety constraint.*

Autonomous controller designed based on the CTH policy has a hierarchical architecture [10]. The upper level controller of the architecture determines the desired longitudinal acceleration ($a^{(fd)}$) according to speed of the follower vehicle ($v^{(f)}$), relative velocity ($\dot{u}$), and relative distance ($u$) between the leader and the follower vehicles. As such, the control law is given by the following equation,

$$a^{(fd)} = -\frac{1}{h}(\dot{u} + \gamma u + \gamma h v^{(f)}) \tag{9}$$
$$u = x^{(l)} - x^{(f)}$$

where, $\gamma = 0.9$ is a system parameter and $x^{(f)}$ is position of the follower at any time.

The lower level controller of the architecture determines the acceleration of pedal ($a_{pedal}$) and brake pressure ($P_{brake}$) of the vehicle. Due to the presence of actuator dynamics and the lower controller, acceleration ($a^{(f)}$) obtained is not same as the desired value ($a^{(fd)}$). This is shown by the following equation,

$$a^{(f)} + \tau \dot{a}^{(f)} = a^{(fd)} \tag{10}$$

where, $\tau = 1.008$ is the time constant, and $\dot{a}^{(f)}$ is the jerk. While designing the upper level controller, internal and external disturbances are neglected to ensure the lower level controller works correctly and satisfy dynamics of (10). Similarly, non-linearity at the lower level controller are compensated using inverse longitudinal dynamics.

Based on (9) and (10), the ACC vehicle dynamics can be represented in the following discrete time, multi-input multi-output state-space form,

$$\mathbf{X}_{k+1}^{(f)} = H_k \mathbf{X}_k^{(f)} + G u_k + w_k^{(f)} \tag{11}$$

$$\mathbf{Y}_k^{(f)} = C \mathbf{X}_k^{(f)} + v_k^{(f)} \tag{12}$$

where, $\mathbf{X}^{(f)} = [x^{(f)}, v^{(f)}, a^{(f)}] \in \mathbb{R}^{3\times 1}$ is the state vector, relative distance, $u$, is the input, $\mathbf{Y}^{(f)} = [x^{(f)}, v^{(f)}] \in \mathbb{R}^{2\times 1}$ is the output vector, and the matrices are $H_k = I + \Delta t A_k$, $G = \Delta t B$, and

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Notice, that $A_k \in \mathbb{R}^{3\times 3}$ is a time-varying matrix that changes according to acceleration ($A_a$) and deceleration ($A_d$) of the vehicle, where

$$A_a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{(1+\frac{1}{h})\gamma}{\tau} & \frac{(1+\frac{1}{h})}{\tau} \end{bmatrix}$$

$$A_d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{(1+\frac{1}{h})\gamma}{\tau} & -\frac{(1+\frac{1}{h})}{\tau} \end{bmatrix}.$$

The input matrix, $B = [\frac{(1+\frac{1}{h})}{\tau^2 h} - \frac{\gamma}{\tau h}, -\frac{1}{\tau h}, 0] \in \mathbb{R}^{3\times 1}$ and time-varying Gaussian process and measurement noises are given by $w_k^{(f)}$ and $v_k^{(f)}$ respectively. During attacks, the output (12) changes and it produces malicious measurements ($x_{attack}^{(f)}, v_{attack}^{(f)}$). To mitigate the effect of attacks, we use our Algorithm 1, which produces estimated values of position ($\hat{x}^{(f)}$) and velocity ($\hat{v}^{(f)}$) (shown in Figure 3). With these estimates, we obtain corrected values of inputs: relative distance ($u$) and relative velocity ($\dot{u}$), of the ACC controller.

## 5.2 Simulation and Results

The car-following scenario consisting of leader and follower vehicle models, False Data Injection (FDI) attack, and our resilient state estimation and detection algorithm with safety measure are simulated in MATLAB. In the car-following scenario, we consider the leader vehicle decelerates and accelerates at -0.1082 $m/sec^2$ and +0.012 $m/sec^2$ respectively. The follower vehicle has to slow down accordingly to ensure the inter-vehicular distance is greater than the desired distance ($d^{(l,f)}$) to avoid rear end collision (*safety constraint*). We consider 65 miles/hr and 60 miles/hr as the initial velocities of the leader and the follower vehicles respectively. The leader starts slowing down when the distance between the vehicles is 10 $m$. For such a scenario, an adversaries intention is to corrupt measurements of the internal sensors of the follower vehicle so that it leads to undesired consequences.

• *Setup for Algorithm 1*

The threshold of the $\chi^2$ detector of the algorithm is fixed for both the attacks at $\eta = c_d$, where we choose the constant $c_d = 20$. The initial state for the leader vehicle is set to $\mathbf{X}_0^{(l)} = [12, 29.05, -0.108]^T$. For the follower vehicle it is $\mathbf{X}_0^{(f)} = [2, 26.82, 0.112]^T$ and covariance of the process and measurement noises are assumed to be $\Sigma_w = \text{diag}(1, 1, 1)$ and $\Sigma_v = \text{diag}(1, 1)$ respectively. The estimation results of our method is compared against the standard Kalman filter and the Robust Kalman filter [6]. The estimator [6] requires a parameter $\lambda$ and we pick the one that gives optimal performance.

• *Case 1: Attack free scenario*

We first evaluate our algorithm against the standard Kalman filter and Robust Kalman filter in the attack free case. For our experiment, we consider a time frame of (0 - 1.5) *sec* with step size of 0.01 *sec* and $k = \{1, 2, \ldots, 150\}$ iterations. Figures 4 compares true and corrected values of relative distance. We observe that the three estimators minimizes the effect of noise in both the measurements. To highlight the performance of our filter, we calculate the estimation error using $\sqrt{\frac{1}{150}\sum_{i=1}^{150} \|\mathbf{X}_{\text{true}} - \hat{\mathbf{X}}_{\text{estimate}}\|^2}$ and found that the error produced by our algorithm for followers' position (3.604 m) is less than the Robust Kalman filter: 3.673 m (followers' position).

• *Case 2: False Data Injection attack*

We consider a scenario where an adversary corrupt measurements of internal sensors of the follower vehicle after a certain time point. In case of FDI attack on our experimental system, malicious data of random value are added to the position ($x^{(f)}$) outputs at random time points and duration. We assume that the signal attack vector $y_k^{af}$ is injected into the output data at random time points after $k = 5$, but the attack does not corrupt all the measurements after its initiation. For instance, in our experiment, the attack occurs at time points such as $0.05, 0.11, 0.23 - 0.25, 0.44, 0.5 - 0.52, 1.18 - 1.2$ and 1.46. The $\chi^2$ detector of our Algorithm 1 promptly detect the attacks by comparing the value of the function $g_k$ against the threshold $\eta$. Subsequently, the estimated values of position ($\hat{x}^{(f)}$) generated by our algorithm is used for calculating relative position of the ACC controller. Figures 5 provides comparison between the true and corrected values of relative distance. Note that the effect of follower vehicles position estimate on the ACC controller inputs, ($u, \dot{u}$), generated by our method at attack time points outperform the results of the traditional Kalman filter and the Robust Kalman filter. Most significantly, unlike the other two methods, the position estimates from our algorithm ensures that the relative distance values are always positive, which indicate that the leader and the follower vehicles never collide at any time. At all other time points (when there is no attack), our method performs as well as the optimal Kalman Filter. We also calculated the estimation error during FDI attack and found that the error produced by our algorithm for followers'

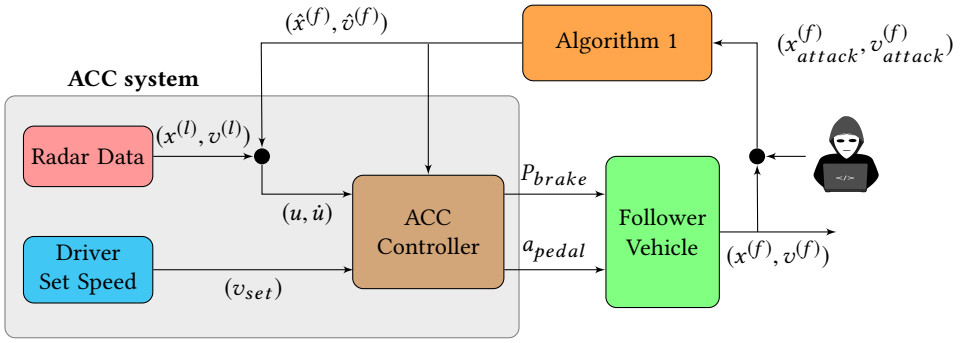


**Figure 3: Attack on sensor measurements of ACC system**

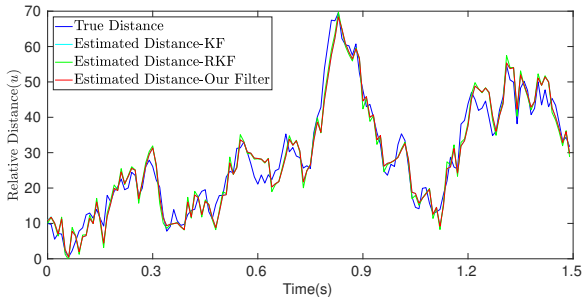


**Figure 4: Plots of relative distance between leader and follower vehicles for the attack free case. Estimation of Our Filter is as good as the Kalman Filter (KF) and the Robust Kalman Filter (RKF)**

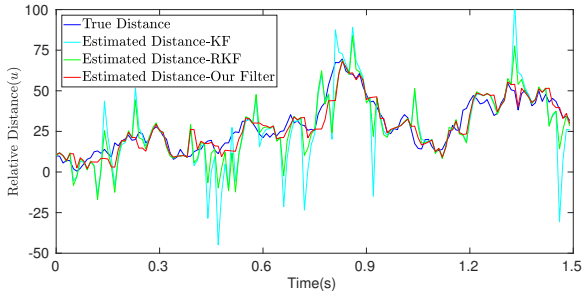


**Figure 5: False Data Injection attack: comparison of true and corrected values of relative distance by Kalman Filter (KF), Robust Kalman Filter (Robust KF) and Our Filter.**

position (5.75 m) measurement were again less than the Robust Kalman filter (RKF): 10.97 m (followers' position). Consequently, since the minimum stopping distance, $d_r = 5\,m$, the likelihood of our algorithm preventing a collision is higher than the RKF.

## 6 CONCLUSION

In this paper, we have proposed a novel attack resilient filter that can recursively estimate states within an error bound and preserve safety constraints, when sensors of the system are compromised. Our approach leverages Bayesian interpretation of the Kalman filter and combines it with the $\chi^2$ detector to ensure safety of CPS against Denial of Service and False Data Injection attacks. The computational complexity of our method is $O(\max(n, q)^3)$, which is same as that of the Kalman filter and it performs better than the standard and the Robust Kalman filters during attack as was shown in the car-following case study. In future, we intend to extend our algorithm toward other attacks such as Denial of Service and Replay and include extensive security analysis.

## ACKNOWLEDGMENT

The work presented in this paper is partially supported by National Science Foundation (NSF 1818500).

## REFERENCES

[1] Nasser Abouzakhar and Abu Bakar. 2010. A Chi-square testing-based intrusion detection Model. In *Procs 4th International Conference on Cybercrime Forensics Education & Training*.
[2] B Brumback and M Srinath. 1987. A chi-square test for fault-detection in Kalman filters. *IEEE Trans. Automat. Control* 32, 6 (1987), 552–554.
[3] Simon Burton, Juergen Likkei, Priyamvadha Vembar, and Marko Wolf. 2012. Automotive functional safety= safety+ security. In *Proceedings of the First International Conference on Security of Internet of Things*. ACM, 150–159.
[4] Nicola Forti, Giorgio Battistelli, Luigi Chisci, and Bruno Sinopoli. 2016. A Bayesian approach to joint attack detection and resilient state estimation. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 1192–1198.
[5] Benjamin Glas, Carsten Gebauer, Jochen Hänger, Andreas Heyl, Jürgen Klarmann, Stefan Kriso, Priyamvadha Vembar, and Philipp Wörz. 2015. Automotive safety and security integration challenges. *Automotive-Safety & Security 2014* (2015).
[6] J. Mattingley and S. Boyd. 2010. Real-Time Convex Optimization in Signal Processing. *IEEE Signal Processing Magazine* 27, 3 (May 2010), 50–61. https://doi.org/10.1109/MSP.2010.936020
[7] S. Mishra, Y. Shoukry, N. Karamchandani, S. N. Diggavi, and P. Tabuada. 2017. Secure State Estimation Against Sensor Attacks in the Presence of Noise. *IEEE Transactions on Control of Network Systems* 4, 1 (March 2017), 49–59.
[8] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. November, 2015. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR. In *Black Hat Europe*.
[9] Sándor Plósz, Christoph Schmittner, and Pál Varga. 2017. Combining safety and security analysis for industrial collaborative automation systems. In *International Conference on Computer Safety, Reliability, and Security*. Springer, 187–198.
[10] Rajesh Rajamani. 2011. *Vehicle dynamics and control*. Springer Science & Business Media.
[11] Rajesh Rajamani and Chunyu Zhu. 2002. Semi-autonomous adaptive cruise control systems. *IEEE Transactions on Vehicular Technology* 51, 5 (2002), 1186–1192.
[12] Christoph Schmittner, Zhendong Ma, and Paul Smith. 2014. FMVEA for safety and security analysis of intelligent and cooperative vehicles. In *International Conference on Computer Safety, Reliability, and Security*. Springer, 282–288.