2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Attack Resilient State Estimation for Autonomous Robotic Systems

Nicola Bezzo, James Weimer, Miroslav Pajic, Oleg Sokolsky, George J. Pappas, Insup Lee

Abstract— **In this paper we present a methodology to control ground robots under malicious attack on sensors. Within the term attack we intend any malicious disturbance injection on sensors, actuators, and controller that would compromise the safety of a robot. In order to guarantee resilience against attacks, we use a control-level technique implemented within a recursive algorithm that takes advantage of redundancy in the information received by the controller. We use the case study of a vehicle cruise-control, however, the strategy we present in this work is general for several applications. Our methodology relays on redundancy in the sensor measurements: specifically we consider N velocity measurements and use a recursive filtering technique that estimates the state of the system while being resilient against sensor attacks by acting on the variance of the measurements noise. Finally, we move our focus on hardware validation demonstrating our algorithm through extensive outdoor experiments conducted on two unmanned ground robots.**

I. Introduction

Modern vehicular and robotic systems are equipped with several sensors and Electronic Control Units (ECUs) that interact with each other over a complex network. This availability of technology and especially networking has led to an overall higher comfort of driving, an increase of the safety of the driver and passengers, and the introduction of new services such as remote diagnosis and vehicle-to-vehicle communication. However, this increase in functionality and communication may introduce security vulnerability and compromise the integrity of the system. For instance an attacker who is able to spoof the GPS could mislead the vehicle to unsafe regions [1]; similarly if a vehicle is in cruise control and an attacker compromises the speedometer reading, the vehicle speed could change drastically inducing an higher probability of collisions and accidents. These risks increase even more with the new generation of unmanned ground vehicles (UGVs) and self driving cars.

To address these issues, we have introduced a design framework for development of high-confidence vehicular control systems that can be used in adversarial environments [2]. The framework employs system design techniques that guarantee that the vehicle will maintain control, possibly at a reduced efficiency, under several classes of attacks. In this paper we focus primarily on estimation design schemes and address attacks on sensors for autonomous ground vehicles (Fig. 1). We utilize a security-aware attack-resilient estimator that identifies an attack and allows the controller to pursue a mitigation strategy.

The contribution of this paper is threefold: *i)* we develop an estimator that is easy to implement by using a recursive approach, *ii)* we compare it with other techniques, and *iii)* we run extensive hardware evaluations to validate the proposed technique. Specifically our framework is inspired by the Linear Quadratic Estimator in which together with the update and predict steps we add a *shield* procedure to cancel the effects due to possible attacks on sensors. Our technique adds an extra weighted variance to the measurement error whenever there is a mismatch between the updated state and the measurement from each of the sensors. By using this technique all sensors are always considered, however the one that contains a corrupted measurement will have a large error variance and thus will count less when estimating the desired state.

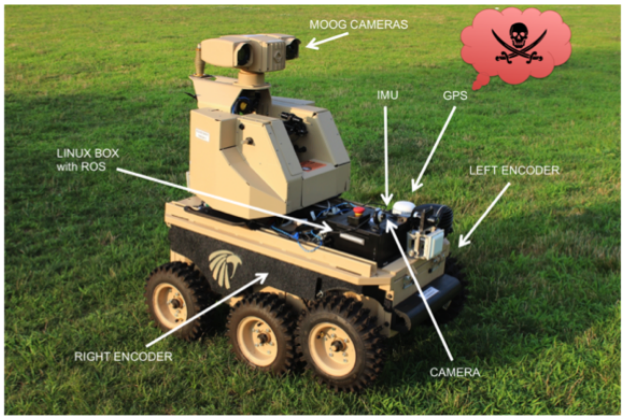


Fig. 1. The LandShark robot [3]: one of the platforms used to study malicious attacks on sensors.

A. Related Work

The study of high assurance vehicular systems is a recent topic that is attracting several researchers in both the control and computer science communities. Malicious attacks are defined as adversarial actions conducted against a system or part of it and with the intent of compromising the performance, operability, integrity, and safety of the system. The main difference between failure and malicious attack is that the former is usually not coordinated while an attack is usually camouflaged or stealthy and behaves and produces results similar or expectable by the dynamics of the systems. Attack vectors can be classified in the following groups: *i)* attacks on the vehicle's sensors and actuators; *ii)* attacks on RF communication and over the local network (i.e., shared bus); *iii)* attacks on the system's maintenance mechanisms and physical interfaces. The focus of this paper is on control-level defenses (i.e., the first group of attacks). These attacks include attacks on sensors measurements transmitted over a common bus (network) to other system components, and injection of malformed data from corrupted system components with access to the bus or a faulty sensor. The ability to attack sensors in vehicles was demonstrated in [1] in which a GPS was spoofed misguiding a yacht off route. Similarly in [4], the authors presented the steps and equipment necessary to spoof a GPS. A more general assessment on car security was recently performed by authors in [5] in which under

N. Bezzo, James Weimer, Miroslav Pajic, Oleg Sokolsky, George J. Pappas, and Insup Lee are with the PRECISE Center, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA {nicbezzo, weimerj, pajic, pappasg}@seas.upenn.edu {sokolsky, lee}@cis.upenn.edu

some circumstances it was demonstrated how to attack steering, braking, acceleration, and display on two vehicles.

Even though this area of study is still at an early stage, some preliminary work on vehicular security was performed in [6] in which the authors showed through intensive experiments on common cars, that an attacker could take over a vehicle and compromise its safety. Specifically it was shown that the CAN bus system is unprotected and several functionality of a car can be controller and accessed by different devices in the vehicle. The main attacks on sensors in ground vehicles generally involve the speedometer and GPS. In our work we will present experimental results dealing with attacks on these sensors.

Standing from a control perspective, authors in [7] propose a resilient consensus algorithm based on receding-horizon control to deal with replay attacks between an operator and a remotely controlled unmanned ground vehicle. In [8] the authors use plant models for attack detection and monitor in cyber-physical systems. In [9] the authors consider wireless sensor networks and the problem of attacks on state estimation performed by a Kalman Filter. An ellipsoidal algorithm is proposed to estimate the resilience of the system against such attacks. Our work in this paper is motivated by the previous results in [10] in which a strategy for resilient attack detection is formulated based on redundancy in the sensor measurements. During the experimental implementation we use the Robot Operating System (ROS) by Willow Garage [11] on different UGVs. It is worth mentioning that a preliminary study and assessment about the security of ROS was performed in [12]. The authors showed that a key security issue is that messages are not authenticated and can be easily decipher and spoofed. In our work we do not solve this problem and use it as a motivation and to create attacks that compromise the sensors.

The remainder of this paper is organized as follows. In Section II we open our discussion with the problem formulation. In Section III we present the recursive estimator algorithm. In Section IV we show the architecture and model used for the vehicles under analysis followed by simulations in Section V. Hardware/software implementations on two ground robots are presented in VI and finally we draw conclusions in Section VII.

II. Problem Statement & Preliminaries

Within this work we are interested in finding a strategy to guarantee that a robot under malicious attack can reach a desired state without being hijacked.

We assume that the robotic system is a discrete-time linear time-invariant (LTI) of the following form

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + w_k + B\mathbf{u}_k, \\ \mathbf{z}_k &= C\mathbf{x}_k + \nu_k, \\ \mathbf{y}_k &= \mathbf{z}_k + \lambda_k, \end{aligned} \tag{1}$$

with $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{u}_k \in \mathbb{R}^m$ are the state vector and control inputs at time k respectively. $\mathbf{z}_k$ is the set of measurements without the effects of attack while $\mathbf{y}_k = [y_{k,1}, y_{k,2} \dots y_{k,N}] \in \mathbb{R}^N$ is the sensor measurements vector with attack where $y_{k,i} \in \mathbb{R}$ is the measurement taken by the ith sensor at time k. λ_k is the attack vector in which each term represents attack of magnitude $\|\lambda_{k,i}\|$. The attack vector is assumed to be arbitrary, i.e., we assume no prior knowledge of statistical properties of bounds on the values of the attack vector. Finally, both the process noise $w_k = \mathcal{N}(0, W)$ and the sensor measurement noise $\nu_k = \mathcal{N}(0, V)$ are expressed as Gaussian random variables.

Specifically we are interested in designing a linear recursive estimator consisting of the following prediction and update steps:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= A\hat{\mathbf{x}}_{k-1|k-1} + B\mathbf{u}_{k-1}, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k\left(\mathbf{y}_k - C\hat{\mathbf{x}}_{k|k-1}\right). \end{aligned} \tag{2}$$

We define $\hat{\mathbf{y}}_{k|k-1} = C\hat{\mathbf{x}}_{k|k-1} + \lambda_k$ and such that $\hat{\mathbf{y}}_{k|k-1} = \bar{\mathbf{z}}_k + \mathbf{s}_k$ where $\bar{\mathbf{z}}_k$ is an *unknown* minimum mean square error (MMSE) estimate of $\mathbf{z}_k$ corresponding to a *known* covariance matrix $\hat{\Sigma}_k$ [1]. $\mathbf{s}_k = C(\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k) + \lambda_k$ is the deviation of the actual measurement estimate from the MMSE estimate and it is caused by potential attacks, previous attacks residuals, and counter measurements. Ideally we would choose K_k such that $K_k C\mathbf{s}_k = 0$. However this require mixed integer programming as demonstrated in [13]. Thus as a relaxation, in this work we use a recursive implementation and address the following problem

Problem 2.1: **Bounding the Attack Innovation** Given $\mathbf{s}_k$ the effect of the attack on the system and ζ a constant small positive value, find the optimal gain value K_k such that the following inequality holds:

$$E\left[\|K_k\mathbf{s}_k\|^2\right] \leq \zeta, \tag{3}$$

where $E[\cdot]$ is the expected value and $K_k\mathbf{s}_k$ is the *attack innovation*.

Through out this work we use the following assumption on the maximum number of sensors under attack.

Assumption 2.2: **Attack Feasibility** The maximum tolerable number of sensors under malicious attack is less than $N/2$.

III. Design of a Resilient Control Scheme

A. Resilient Recursive Adaptive Estimator (RAE)

Most of the techniques available in the current literature are very efficient in detecting and removing sensors under attacks if we consider deterministic systems with bounded small noise or time invariant sensor models [7], [10]. However, in real world applications the sensor noise profile is not always fixed and variations due to non-attack effects such as biases and environmental effects, can occur. Secondly, often some sensors, even if under attack or compromised, can be still used and fused to obtain useful information and improve the state estimation. Instead of using a binary selection criterium, we propose a strategy that assigns weights to each sensor based on how close each measurement is from the estimated state.

Before showing our algorithm we introduce an oracle estimation to consider the optimal solution in the case that everything was known a priori, including the attack vector.

1) **Oracle Estimator:** In the literature [14], [15] we find that the Linear Quadratic Estimator has been widely used in engineering applications because it combines measurements of the same variable but from different sensors and it combines inexact forecast of a system's state with an inexact measurement of the state. However, an adversarial attack could destabilize the system and drive the vehicle to undesired states. If we are always able to predict a priori

[1] $\hat{\Sigma}_k$ is known despite $\bar{\mathbf{z}}_k$ being unknown since it only depends on the number of sensors

when and where an attack will happen (e.g., an oracle), the prediction step takes the following form:

$$\hat{\mathbf{x}}^o_{k|k-1} = A\hat{\mathbf{x}}^o_{k-1|k-1} + B\mathbf{u}_{k-1}, \tag{4}$$

where we have use the upper script o to represent the oracle estimation parameters. The predicted estimate covariance follows as

$$P^o_{k|k-1} = AP^o_{k-1|k-1}A^T + W. \tag{5}$$

The oracle update will take then the following form

$$\hat{\mathbf{x}}^o_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K^o_k\left(\mathbf{y}_k - C\hat{\mathbf{x}}_{k|k-1}\right), \tag{6}$$

$$P^o_{k|k} = (I - K^o_k C)P_{k|k-1}, \tag{7}$$

with

$$K^o_k = P_{k|k-1}CQ\left(Q^T\left(CP_{k|k-1}C^T + V\right)Q\right)^{-1}Q^T, \tag{8}$$

where $Q \in \{Q|QQ^T = I, Q_{i,j} \in \{0,1\}\}$ is the selection matrix given by the oracle, where I is the identity matrix and $Q\mathbf{y}_k$ are the non-attacked measurements.

Clearly in a realistic scenario we will not have an oracle; thus, next we propose a strategy that attempts to solve this problem by implementing a recursive filter in which, together with updating and predicting the state of the vehicle, we introduce a resilience step (called *Shield*) to consider and isolate attacks on sensors. The oracle presented above will be used later on to prove property of the developed recursive resilient state estimator.

2) **Resilient Adaptive Estimator:** Our recursive algorithm is motivated by the well established results found in the Linear Quadratic Estimator implementation with some modifications to accommodate the possible presence of an attack in one of the sensors.

The generalized form of our resilient filter is

- *PREDICT.* The predicted state estimate becomes

$$\hat{\mathbf{x}}_{k|k-1} = A\mathbf{x}_{k-1|k-1} + B\mathbf{u}_{k-1}, \tag{9}$$

  and the predicted estimate covariance follows

$$P_{k|k-1} = AP_{k-1|k-1}A^T + W. \tag{10}$$

- *UPDATE.* In the update phase, we introduce the following modified gain

$$\hat{K}_k = P_{k|k-1}C^T\left(CP_{k|k-1}C^T + V + D_k\right)^{-1}, \tag{11}$$

  where D is the *Shielding Gain* matrix (described below) introduced to consider and remove attacks on the sensor measurements.
  The updated state and covariance become

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{y}_k - C\hat{\mathbf{x}}_{k|k-1}), \tag{12}$$

$$P_{k|k} = (I - K_kC)P_{k|k-1}. \tag{13}$$

- *SHIELD.* If an attack is present and such that one of the measurements is corrupted, the goal is to remove it or mitigate its effect. Since the attack vector is generally unknown, one strategy we can implement is to change the covariance matrix associated with the measurement error in order to increase the uncertainty where the measurement is different from the predicted state estimate. To this end, let us define the *Shielding Gain* $D_k = \text{diag}\{d_{k,1}, d_{k,2}, \ldots, d_{k,N}\}$ as a positive semidefinite diagonal matrix, then we can write

$$d_{k,j} = d_{k-1,j} + \eta_{k,j}\left(\left|\left|\frac{\mathbf{y}_{k,j} - C_j\hat{\mathbf{x}}_{k|k-1}}{\sigma_{k,j}}\right|\right|^2 - 1\right), \tag{14}$$
$$\text{with } \sigma_{k,j} = C_jP_{k|k-1}C_j^T + V_{jj},$$

  where $\eta_{k,j}$ is a gain factor of the measurement error. We call $d_{k,j}$ the *shielding factor* since its purpose is to increase the covariance of the measurement noise associated with the sensor under attack, thus "shielding" the malicious effects on the system. Notice that we impose (14) to be always positive semidefinite. In fact, a negative value would imply that we are able to improve the performance of a sensor (which is not possible).

We are now interested to show that the proposed strategy is guaranteed to satisfy Problem 2.1 if we choose correctly η in (14). Before showing this, we introduce the *attack-to-noise ratio*, a new measure which relates the sensor noise to the attack effects on the robotic agent, as follow:

Definition 3.1: **Attack-to-noise Ratio (ANR)** We define the attack-to-noise ratio τ as a measure that compares the attack effects to the noise level of a sensor. In formula:

$$\tau = ANR = ((\Sigma_y + D_k)^{-1}\mathbf{s}_k)^2, \tag{15}$$

where $\Sigma_y = CP_{k|k-1}C^T + V$ (see (11)) and $\mathbf{s}_k$ is as defined in Section II.

Theorem 3.2: **The ANR is bounded** For some $\zeta \geq 0$, there exists a η in (14) independent of s_k such that the expected value of ANR $E[\|\tau\|] \leq \zeta$.

Proof: To prove Theorem 3.2 we start by considering the average ANR for the worst case scenario, and then calculate its boundaries with respect to η. Let's consider the expression for ANR in (15):

$$\|\tau\| = \left|\left|((\Sigma_y + D_k)^{-1}\mathbf{s}_k)^2\right|\right| \leq \left|\left|((\sigma^2_{\min}I + D_k)^{-1}\mathbf{s}_k)^2\right|\right| = \sum_{j=1}^{N}\frac{s^2_{k,j}}{(\sigma^2_{\min} + d_{k,j})^2}. \tag{16}$$

Thus, bounding $E\left[\left(\frac{s_{k,j}}{(\sigma^2_{\min}+d_{k,j})}\right)^2\right]$ implies $E[\|\tau\|]$ is bounded.

To this end, by Jensen's inequality, we have that

$$E\left[\left(\frac{s_{k,j}}{(\sigma_{\min} + d_{k,j})}\right)^2\right] \leq \left[\frac{s_{k,j}}{E\left[\sigma^2_{\min} + d_{k,j}\right]}\right]^2. \tag{17}$$

Now we observe that

$$\left[\frac{s_{k,j}}{E\left[\sigma^2_{\min} + d_{k,j}\right]}\right]^2 =$$

$$= \left[\frac{s_{k,j}}{E\left[\sigma^2_{\min} + d_{k-1,j} + \eta_{k,j}\left(\frac{(\mathbf{y}_{k,j} - C_j\hat{\mathbf{x}}_{k|k-1})^2}{\sigma^2_{k,j}}\right) - 1\right]}\right]^2$$

$$= \left[\frac{s_{k,j}}{E\left[\sigma^2_{\min} + d_{k-1,j} + \eta_{k,j}\left(\frac{(\mathbf{z}_{k,j} - \bar{\mathbf{z}}_k + s_{k,j})^2}{\sigma^2_{k,j}}\right) - 1\right]}\right]^2$$

$$= \left[\frac{s_{k,j}}{(\sigma^2_{\min} + d_{k-1,j} + \eta_{k,j}\left(\frac{\bar{\sigma}^2_{k,j}}{\sigma^2_{k,j}} + \frac{s^2_{k,j}}{\sigma^2_{k,j}} - 1\right)} \right]^2 = f(s), \tag{18}$$

where $\bar{\sigma}^2_{k,j} = E[\mathbf{z}_{k,j} - \bar{\mathbf{z}}_k]$.

If we now take the derivative of $f(s)$ with respect to s $\left(\frac{d}{ds}f(s) = 0\right)$ we obtain a maximum at

$$s^2_{k,j} = \frac{\sigma^2_{k,j}}{\eta_j}\left(\sigma^2_{\min} + d_{k-1,j} + \eta_{k,j}\left(\frac{\bar{\sigma}^2_{k,j}}{\sigma^2_{k,j}} - 1\right)\right). \tag{19}$$

Now substituting $s^2_{k,j}$ in (17) we obtain

$$E\left[\frac{s^2_{k,j}}{(\sigma^2_{\min} + d_{k,j})^2}\right] \leq \frac{\sigma^2_{k,j}}{4\eta_{k,j}\left(\sigma^2_{\min} + d_{k-1,j} + \eta_{k,j}\left(\frac{\bar{\sigma}^2_{k,j}}{\sigma^2_{k,j}} - 1\right)\right)} \tag{20}$$

which does not depend on $s_{k,j}$, concluding the proof. ■

Thus, Theorem 3.2 and the proof above show that there exists an upper bound on the ANR that depends on $\eta_{k,j}$ in the recursive adaptive estimator defined in (9) - (14).

We are now ready to show that Problem 2.1 holds, given the bounds calculated on the ANR in Theorem 3.2.

Lemma 3.3: **Bounded Attack Innovation** Given the recursive adaptive estimator outlined in (9) - (14) there exists a $\eta \geq 0$ such that

$$E\left[\|K_k\mathbf{s}_k\|^2\right] \leq \zeta \tag{21}$$

Proof: Given $\Sigma_y = (CP_{k|k-1}C^T + V)$ and $\Sigma_{xy} = P_{k|k-1}C^T$,

$$\begin{aligned} E\left[\|K_k\mathbf{s}_k\|^2\right] =& E\left[\|\Sigma_{xy}(\Sigma_y + D_k)^{-1}\mathbf{s}_k\|^2\right] \\ \leq& \|\Sigma_{xy}\|^2 E\left[\|(\Sigma_y + D_k)^{-1}\mathbf{s}_k\|^2\right] \\ =& \|\Sigma_{xy}\|^2 E[\tau], \end{aligned} \tag{22}$$

where we notice that the second part of equation (22) is the ANR, $\tau = ((\Sigma_y + D_k)^{-1}\mathbf{s}_k)^2$.

Given that

$$\eta_j = \frac{\sigma^2_{\min} + d_{k-1,j}}{1 - \frac{\bar{\sigma}_{k,j}}{\sigma^2_{k,j}}} \tag{23}$$

is substituted into (20) and summed over all the sensors, then (21) is satisfied when

$$\zeta \geq Tr\left[\left(\sigma_{\min^2}I + D_{k-1}\right)^{-1}\left(\hat{\Sigma}_k - \bar{\Sigma}_k\right)\left(\sigma_{\min^2}I + D_{k-1}\right)^{-1}\right]. \tag{24}$$

■

If (24) is not satisfied, then we need to reset the estimator by assigning $\hat{\Sigma}_k = \bar{\Sigma}_k$.

B. Properties of the Shielding Gain

In the following lines we present some other properties of the proposed recursive algorithm focusing primarily on the existence, expectation, and evolution of the shielding factors.

Lemma 3.4: **Existence of the Shielding Gain** Given the recursive implementation composed of the prediction, shield, and update steps of (9) - (14), there exists a shielding gain such that the estimated state is equivalent to the oracle one. In formula

$$\exists D_k \text{ s.t. } \forall P_{k|k-1}, \hat{K}_k = K^o_k. \tag{25}$$

Proof: Given $\hat{K}$ defined in (11) and assuming K^o in (8) is the optimal gain, we would like to show that there is a D_k such that the following equality is satisfied

$$\begin{aligned} Q\left(Q^T\left(CP_{k|k-1}C^T + V\right)Q\right)^{-1}Q^T =& \\ = \left(CP_{k|k-1}C^T + V + D_k\right)^{-1}. \end{aligned} \tag{26}$$

Let

$$P_y = (CP_{k|k-1}C^T + V) = \begin{pmatrix} P_n & P_{nd} \\ P^T_{nd} & P_d \end{pmatrix}, \tag{27}$$

then the left hand side of (26) becomes

$$Q(Q^TP_yQ)^{-1}Q^T = \begin{pmatrix} P^{-1}_n & 0 \\ 0 & 0 \end{pmatrix}. \tag{28}$$

Now for the right hand side, let us select

$$D_k = \begin{pmatrix} 0 & 0 \\ 0 & rI \end{pmatrix}, \tag{29}$$

with r a constant. Then it follows that

$$\begin{aligned} (P_y + D)^{-1} =& \begin{pmatrix} P_n & P_{nd} \\ P^T_{nd} & P_d + rI \end{pmatrix}^{-1} = \\ =& \begin{pmatrix} P^{-1}_n - P^{-1}_nP_{nd}\Psi^{-1}P^T_{nd}P^{-1}_n & -\Psi^{-1}P_{nd}P^{-1}_n \\ -P^{-1}_nP^T_{nd}\Psi^{-1} & \Psi^{-1}, \end{pmatrix} \end{aligned} \tag{30}$$

where $\Psi = P_d + rI$.

Choosing r very large is equivalent to unselecting the sensors that are under attack, thus leaving us with only the correct measurements. Therefore, as r goes to infinity we obtain

$$\lim_{r\to\infty}(P_y + D)^{-1} = \begin{pmatrix} P^{-1}_n & 0 \\ 0 & 0 \end{pmatrix}, \tag{31}$$

which is equivalent to the left hand side in (28). ■

Lemma 3.5: **Expectation of the Shielding Gain** Given the recursive algorithm described in (9) - (14), if the magnitude of the attacks is always non-zero, then the expected value of the shield factor is always non-negative and non-decreasing. In formula, given N_λ the number of sensors under attack and $j = 1, \ldots N_\lambda$

$$\forall\ N_\lambda \leq N/2, \text{if } \|\lambda_{k,j}\| > 0 \text{ then } E[D_k] \geq D_{k-1}. \tag{32}$$

Proof: The proof is straightforward. In the presence of an attack:

$$\begin{aligned} E[D_k|_{\text{attack}}] =& D_{k-1} + \eta\left(1 + \left\|\frac{\mathbf{s}_k}{\sigma_{k,j}}\right\|^2 - 1\right) = \\ =& D_{k-1} + \eta\left(\left\|\frac{\mathbf{s}_k}{\sigma_{k,j}}\right\|^2\right) \geq D_{k-1}, \end{aligned} \tag{33}$$

proving the statement of the lemma. ■

Lemma 3.6: **Probabilistic Evolution of the Shielding Factors** Given an attack λ_j, by using the recursive algorithm (9)-(14), the probability P that $d_k \geq d_{k-1}$ increases by increasing η.

Proof: To prove Lemma 3.6 we show that the probability that $d_k \leq d_{k-1}$ under attack depends on η in (14).

$$\begin{aligned} \mathrm{P}\left[d_{k-1} \geq d_k|_{\text{attack}}\right] \leq E\left[\frac{d_{k-1}}{d_k}\right] = d_{k-1}E\left[1/d_k\right] \leq \\ \leq \frac{d_{k-1}}{E\left[d_k\right]} = \frac{1}{1 + \frac{\eta}{d_{k-1}}\|\frac{\mathbf{s}_k}{\sigma_{k,j}}\|^2}. \end{aligned} \tag{34}$$

(34) demonstrates that the probability that $d_{k-1} \geq d_k$ under attack is inversely proportional to η. In particular, we notice that for $\eta \to \infty$ the probability in (34) goes to 0, which, in turns, means that the probability of increasing d_k at every step converges to 1.

■

In the following sections we discuss simulation results and experimental implementations on ground wheeled robots. First we give an overview of the architecture of our control system and the robots dynamical model used during the simulations and hardware implementations.

IV. System Architecture

Our formulation is hierarchical and use feedback to control the motion of the vehicle and achieve the desired state. Fig. 2 shows a block diagram representing all the control components used in our framework.

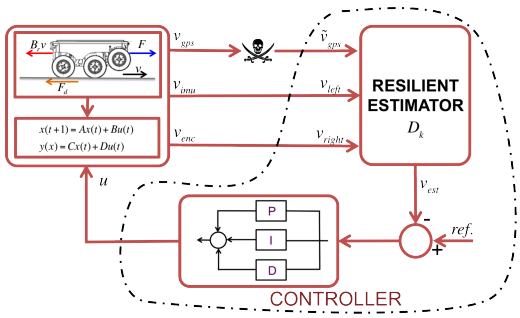


Fig. 2. Block diagram representing the control system architecture employed for secure cruise control.

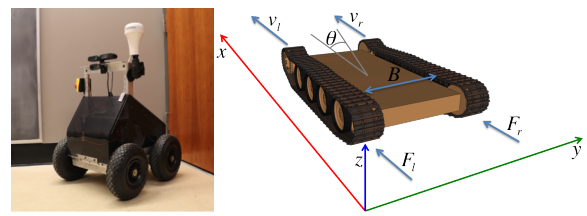
A. Dynamical Model

We illustrate the development framework on a design of secure cruise control of two fully electric unmanned ground vehicles (UGV): the Black-I LandShark [3] shown in Fig. 1, and a custom made robot built at the University of Pennsylvania (UPenn) which we call MiniShrak, shown in Fig. 3(a). Both vehicles are equipped with encoders, IMUs, GPS, and vision sensors. From a computation perspective, both UGVs are equipped with quad-core processors running Ubuntu with ROS.

To obtain a dynamical model of the vehicle we have used the standard differential drive vehicle model (Fig. 3(b)) [16]. Here, F_l and F_r denote forces on the left and right set of wheels respectively and B_r is the mechanical resistance of the wheels to rolling. The vehicle position is specified by its p_x and p_y coordinates, θ denotes the heading angle of the vehicle measured from the x axis, while v is the speed of the vehicle in this direction. Both the vehicles in our testbed employs skid steering, meaning that in order to make a turn it is necessary to generate enough torque to overcome the sticking force S_l. Consequently, if we assume that the wheels do not slip, the dynamical model of the vehicle can be specified as:

$$\begin{aligned}
\dot{v} &= \begin{cases} \frac{1}{m}(F_l + F_r - (B_s + B_r)v, & \text{if } turning \\ \frac{1}{m}(F_l + F_r - B_r v, & \text{if } not\ turning \end{cases} \\
\dot{\omega} &= \begin{cases} \frac{1}{J_t}(\frac{B}{2}(F_l - F_r) - B_l\omega, & \text{if } turning \\ 0, & \text{if } not\ turning \end{cases} \\
\dot{\theta} &= \omega \\
\dot{p}_x &= v\sin(\theta),\ \dot{p}_y = v\cos(\theta)
\end{aligned} \tag{35}$$

Also, $w = 0$ if the vehicle is not turning. Finally, to estimate the state of the vehicle for cruise control (i.e., its speed) we use three sensors: typically the wheel encoders on both sets of wheel, inertial sensors such as the IMU, and the GPS. We have also derived a 6-state linear model of the low-level electromechanical system, which is then used to derive a local controller that provides the desired F_l, F_r levels.



(a) (b)

Fig. 3. Skid-steering ground vehicle; (a) The MiniShark unmanned ground vehicle; (b) Coordinate system and variables used to derive the model.

V. Simulation Results

In this section we show simulation results for the resilient adaptive estimator discussed in Section III in comparison with a well established resilient state estimator presented in [17], [18].

A. Overview of the Resilient State Estimator (RSE)

To illustrate our development framework we compare our technique with the work from [17], [18], where recent results on error correction over the reals and compressed sensing are used to derive a technique to develop secure state estimators when system sensors or actuators are under attack.

In [17] it is shown that for linear systems the state estimate can be obtained from the previous N sensor measurements and actuator inputs as the minimization argument of the following optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|Y^N - \Phi^N \mathbf{x}\|_{l_1/l_2}. \tag{36}$$

Here, for a matrix $M \in \mathbb{R}^{p\times N}$, $\|M\|_{l_1/l_2}$ denotes the sum of l_2 norms of the rows of the matrix, and $\Phi^N = [C\mathbf{x}|CA\mathbf{x}|\ldots|CA\mathbf{x}^{N-1}]$. Furthermore, $Y^N = [\tilde{\mathbf{y}}_{k-N+1}|\tilde{\mathbf{y}}_{k-N+2}|\ldots|\tilde{\mathbf{y}}_k]$ aggregates the sensor measurements while taking into account applied inputs – i.e., $\tilde{\mathbf{y}}_k = \mathbf{y}_k - \sum_{i=k-N+1}^{k} CA^i Bu_{k-1-i}$.

B. Software Implementations

For ease of discussion we abbreviate the resilient adaptive estimator as RAE and the resilient state estimator in [17], [18] as RSE. In all simulations presented in this section the robot is set to first maintain a cruise control speed of 4 m/s and after 50 seconds to switch to 10 m/s. The state space representation of the vehicle has been deduced from careful measurements on the real robot.

Following the architecture and the dynamical model for skid-steering vehicles described in Section IV we developed a ROS based simulator that emulates the same electro-mechanical and dynamical behavior of the real robot. The top subfigure in Fig. 4 displays the normalized voltage supplied to the motors. The middle plot shows the true velocity of the simulated robot, and finally on the bottom of Fig. 4 the three sensor measurements with applied an attack on the GPS measurement are displayed. The attack in this case is modeled as a 10 Hz pulse with pick amplitude of 5 m/s.

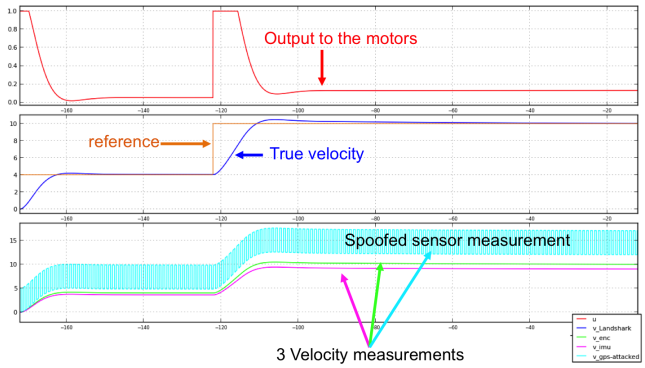


Fig. 4. Rqt-plot of the results from the ROS Simulator with GPS data corrupted.

The developed ROS simulator was used in debugging phase before using the real robot. Notice that both the simulator and the real robots use exactly the same ROS code.

We now show a comparison between the RAE and the RSE, with simulations run in Matlab/Simulink. The same simulation of Fig. 4 was run with the RSE. In Fig. 5 we report the error between reference and true velocity values for both strategies. Each strategy behaves correctly converging to 0 every time we select a new reference velocity.

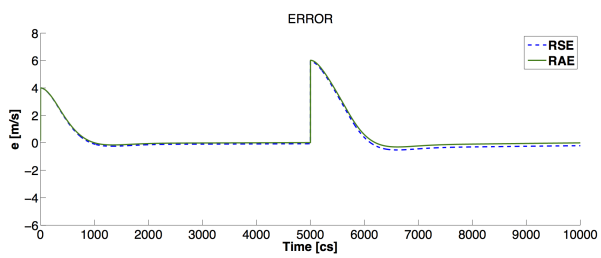


Fig. 5. Comparison between RSE and RAE state estimation errors for low noise situation.

Fig. 6 displays the state estimation error comparison between RSE and RAE when the sensor noise is increased ($\sigma = 1$ m/s). The attack on the GPS is modeled as a pulse with amplitude alternating between 3 m/s and 0 m/s every 10 s. Again we notice that both estimators errors converge to 0 within the noise profile of the sensors. The RAE has slightly less oscillations than the RSE.

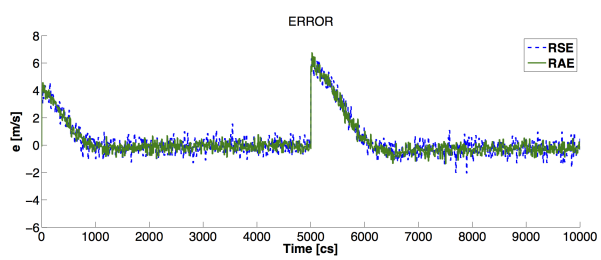


Fig. 6. Comparison between RSE and RAE state estimation errors for noisy measurements with large attacks

Finally, in the simulation displayed in Fig. 7 we increase the noise of each sensor to $\sigma = 2$ m/s. In this case we hide the attack within the noise profile of the GPS measurement but keep it around the boundary of the noise, that is, the attack is a pulse with magnitude alternating between 2 m/s and 0 m/s every 10 s. Because the noise is large and the attack is within the noise profile and on its boundary, we see that the error grows when an attack is inserted and decreases when it is removed creating oscillation in the velocity.

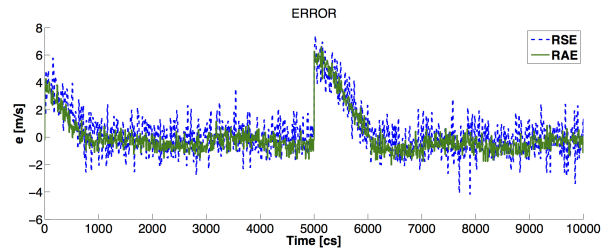


Fig. 7. Comparison between RSE and RAE state estimation errors for large noisy measurements with attacks within the noise profile.

Table I shows numerical results that compare four different strategies: an Oracle estimator (Oracle), a Kalman Filter (KF), the Resilent State Estimator (RSE) of [17], [18], and the Resilient Adaptive Estimator (RAE) proposed in this paper. We use the same three case studies run above within Figs. 5-7. Each entry on the table contains two values: the upper number is the average error between estimated and true state, while the lower number is the ratio between the estimated and the oracle errors. As expected, the Kalman Filter approach is not able to deal with attacks whose value exceeds the noise profile of the sensor measurements, while it can be used in the case that the attack is camouflaged within the error noise of the sensor (last column of the table). As already discussed, the RAE behaves slightly better than the RSE with a maximum recorded error of 18% against the 21% of the RSE in the case of noisy measurements with attack vector outside the noise profile (second column of the table).

TABLE I

STATE ESTIMATE ERROR UNDER ATTACK

Approach	$\sigma = 0.01$ $\lambda = 5$	$\sigma = 1$ $\lambda = 3$	$\sigma = 2$ $\lambda = 2$
Oracle	0.005 1.00	0.13 1.00	0.18 1.00
KF	1.67 334	0.99 7.62	0.35 1.94
RSE	0.005 1.00	0.21 1.62	0.32 1.78
RAE	0.005 1.00	0.18 1.38	0.28 1.56

VI. HARDWARE/SOFTWARE IMPLEMENTATION

The resilient state estimator and the recursive adaptive estimator strategies have been implemented through several experiments on the two robots described in Section IV-A and on different types of surfaces. To facilitate the experimental evaluation, we have built a remote User Interface (UI) (see subfigures inside Figs. 8 and 9) which allows us to start/stop processes, attack each sensor, read/save data, and visualize plots of important information such as the speed calculated from the sensor measurements and the input sent to the

actuators. For these hardware implementations we decided to use GPS and the left and right encoders to obtain three independent speed measurements.

In Fig. 8 it is shown a snapshot of a cruise control experiment run on the LandShark on a tiled pathway inside the UPenn campus. As noted from the UI, the robot can reach and maintain the desired reference speed even when one of the sensors is under attack. Next experiment in Fig. 9(a) displays the MiniShark running with the resilient state estimator described in Section V-A. Finally in Fig. 9(b) we present a snapshot of the implementation of the recursive estimator derived in Section III-A, showing similar results as with the LandShark in Fig. 8. Once an attack is injected in one of the sensors, a weight is added to the noise variance of the corrupted measurement decreasing its trustworthiness. All experimental results show that the two methods can be used to efficiently estimate the state of the system, although the RAE is computationally more efficient and does not require a history of measurements and control inputs.

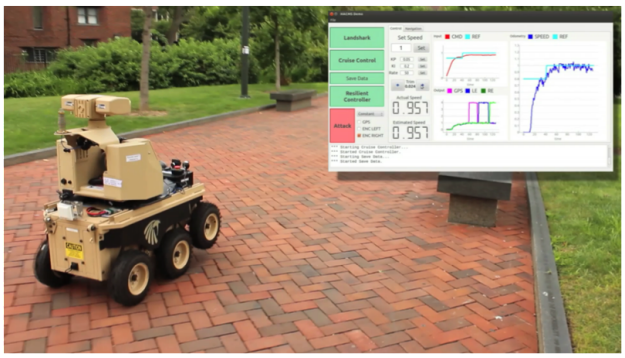
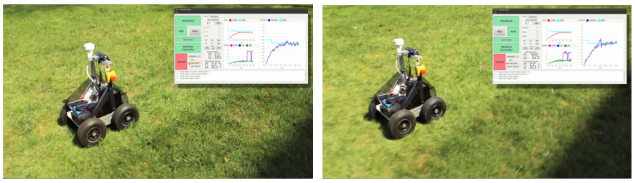


Fig. 8. Experimental result. Snapshot of the deployment of the LandShark on a tiled pathway inside the University of Pennsylvania. The picture in the picture displays the user interface used during the experiments.



(a) (b)

Fig. 9. Snapshots of the deployment of the MiniShark UGV on a grass field inside the University of Pennsylvania. The robot is in cruise control and one by one each sensor is compromised with a constant attack. Figure (a) shows the implementation of the RSE, while (b) depicts the RAE strategy.

VII. Conclusion & Future Work

In this paper we have presented a method to estimate the state of a system under malicious attack on the sensors on a vehicle. The proposed recursive state estimator compares the estimated state with redundant measurements coming from different sources and returns a higher variance of the measurement noise if a sensor is under attack. This strategy allows to consider noisy measurements to estimate the correct state of the system and can be used for attacks that act outside the noise profile of the sensors. However there are few limitations: the algorithm needs an accurate selection of the noise profile and weights in order to converge to the correct state. A too small bound on the error noise implies that the estimator may reject most of the measurements while a too large bound on the error can lead more attacks going through the system because within the error noise profile. In real applications these boundaries on the noise profiles are usually given or can be calculated through hardware testing.

Future work will be centered on: *i)* implementing the proposed strategies on unstable systems (e.g., quadrotors) and more evolved experiments involving obstacle avoidance and way-point navigation; *ii)* running more complex coordinated attacks, and *iii)* developing supervisory capabilities to alternate between different strategies.

Acknowledgments

This work is based on research sponsored by DARPA under agreement number FA8750-12-2-0247.
The authors would like to thank Prof. Paulo Tabuada for useful discussions about the resilient state estimator, Peter Gebhard for creating the UI used in the experiments, and Prof. Daniel Lee and his students for their help in building the MiniShark robot.

References

[1] "Spoofers Use Fake GPS Signals to Knock a Yacht Off Course. http://www.technologyreview.com/news/517686/spoofers-use-fake-gps-signals-to-knock-a-yacht-off-course."
[2] M. Pajic, N. Bezzo, J. Weimer, R. Alur, R. Mangharam, N. Michael, G. J. Pappas, O. Sokolsky, P. Tabuada, S. Weirich *et al.*, "Towards synthesis of platform-aware attack-resilient control systems," in *Proceedings of the 2nd ACM international conference on High confidence networked systems*. ACM, 2013, pp. 75–76.
[3] "Black-I Robotics LandShark UGV. http://www.blackirobotics.com/LandShark_UGV_UC0M.html."
[4] J. S. Warner and R. G. Johnston, "A simple demonstration that the global positioning system (gps) is vulnerable to spoofing," *Journal of Security Administration*, vol. 25, no. 2, pp. 19–27, 2002.
[5] "Car Hacking http://blog.ioactive.com/2013/08/car-hacking-content.html."
[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. of USENIX Security*, 2011.
[7] M. Zhu and S. Martinez, "On resilient consensus against replay attacks in operator-vehicle networks," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 3553–3558.
[8] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, 2012, submitted.
[9] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *49th IEEE Conference on Decision and Control (CDC), 2010*. IEEE, 2010, pp. 5967–5972.
[10] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *arXiv preprint arXiv:1205.5073*, 2012.
[11] "Robotic Operating System. http://www.ros.org."
[12] J. McClean, C. Stull, C. Farrar, and D. Mascareñas, "A preliminary cyber-physical security assessment of the robot operating system (ros)," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, pp. 874 110–874 110.
[13] J. Weimer, N. Bezzo, M. Pajic, O. Sokolsky, and I. Lee, "Attack-resilient minimum-variance estimation," in *American Control Conference (ACC), 2014 (to appear)*. IEEE, 2014.
[14] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice using MATLAB*. Wiley. com, 2011.
[15] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Wiley. com, 2006.
[16] J. J. Nutaro, *Building Software for Simulation: Theory and Algorithms, with Applications in C++*. Wiley, 2010.
[17] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure state-estimation for dynamical systems under active adversaries," in *Proceedings of the 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2011, pp. 337–344.
[18] ——, "Security for control systems under sensor and actuator attacks," in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012.