State Consistencies for Cyber-Physical System Recovery[*]

Fanxin Kong
Syracuse University
fkong03@syr.edu

Oleg Sokolsky, James Weimer, Insup Lee
University of Pennsylvania
{sokolsky,weimerj,lee}@cis.upenn.edu

ABSTRACT

Becoming to open architectures has been making Cyber-Physical Systems (CPS) vulnerable to malicious attacks that are beyond conventional cyber attacks. Recently, a procedure, named cyber-physical system checkpointing and recovery, has been proposed to improve system resilience against CPS attacks. While the feasibility of CPS recovery is well demonstrated, one issue not fully addressed is that what kind of states should be used for the recovery. In this paper, we address this issue and claim to use consistent states. Yet, to define state consistency is a challenging task because CPS systems usually have both functional and real-time requirements. To address the challenge, we formally define state consistency by taking account of both requirements. Specially, we define two consistencies in the intersection of the cyber and physical world: cyber-physical logic-consistency and cyber-physical timing-consistency, based on whether the physical world can be accurately reflected by the corresponding cyber part. We build a simulator of PID controlled DC motor to evaluate how control performance is affected by these consistencies when performing recovery.

1 INTRODUCTION

Cyber-Physical Systems (CPS) tightly integrate computing and communication processes with sensing and actuation components that interact with the physical world. The ever increasing functionalities and network interoperability have been advancing CPS from isolated control systems to more open interacting architectures. This development enables various new services and applications, but meanwhile, it also introduces potential security vulnerabilities that are easily exploitable.

The interaction between information technology and the physical world makes CPS vulnerable to malicious attacks that are beyond the traditional cyber attacks [2]. For example, the authors in [3] demonstrate how to disrupt the operation of a car and even disable the vehicle using some simple methods. Further, the authors in [4] present a case study, where they can launch a Denial-of-Service attack to compromise the CAN bus and the functionalities dependent on the CAN bus. Exclusively utilizing cyber-security techniques to secure CPS is inadequate. This is indicated especially by non-invasive sensor attacks, that is, when the physical environment is compromised to allow injecting malicious signals to sensors [7, 9]. For instance, the authors in [10] demonstrate remote attacks on sensors including camera and LiDAR that are usually mounted in autonomous vehicles. The authors in [11] demonstrate attacks on GPS sensors to misguide a yacht off its course.

These results have motivated many efforts that study the problem of improving attack-resilience under the cases of various attacks on sensors, actuators and communication networks [5, 6, 9]. An effective way to address this problem is to develop methods that can estimate system states accurate enough for control regardless of the compromised components. One advantage of this way is that it allows a system to use the same controller as in the case without attacks. Along this way, a procedure, called cyber-physical system checkpointing and recovery, has been recently proposed in [7].

In [7], we divide CPS recovery into two different operations: roll-back recovery for cyber states and roll-forward recovery for physical-states. Cyber-states are computing information of a controller, such as values of data variables, while physical-states are defined as physical information of a plant, such as speed of a motor. Roll-forward recovery is defined as rolling the system forward to the current time, starting from a historical physical-state. It can be seen as a general method that handles failed estimated states (caused by attacks or faults). We further demonstrate the feasibility of CPS recovery by utilizing a case study of sensor faults or attacks.

One issue not fully addressed in [7] is that what kind of states should be used for CPS recovery. We study this issue and claim to use consistent states for CPS recovery. The focus of this paper is to define state consistency for roll-forward recovery (the definition for roll-back recovery is left as future work). However, to define state consistency is a challenging task. First, CPS systems are usually subject to not only functional requirements, e.g., guaranteeing the correctness of results, but also real-time constraints, e.g., producing results within a certain time frame. Recovery based on a functional correctness state may not yield a result meeting timing requirements, and vice versa. Further, attacks can compromise both value and timing properties of a state, which also requires definitions that take account of both aspects. Second, state consistency is no longer confined to cyber world for CPS systems, because of the interaction between cyber components and the physical world. Thus, consistency definitions applicable to CPS systems should involve both cyber and physical parties.

To address these challenges, we formally define consistencies for cyber-physical states (i.e., cyber information that reflects the physical world). Cyber-physical consistency is in the intersection of the cyber and physical world, i.e., whether the physical world can be accurately reflected by the corresponding cyber part. We define 1) cyber-physical logic-consistency based on whether cyber values can accurately reflect corresponding physical states, 2) cyber-physical syn-timing-consistency based on whether individual elements of estimated states and control inputs are synchronized, and 3) cyber-physical exp-timing-consistency based on the freshness of cyber-physical states to guarantee a certain degree of control performance when performing recovery. We build a simulator of PID controlled DC motor to evaluate how control performance is affected by these consistencies when conducting recovery.

[*] Acknowledgement: This work was supported in part by NSF CNS-1505799 and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy and in part by Global Research Laboratory Program (2013K1A1A2A02078326) through NRF, and the DGIST Research and Development Program (CPS Global Center) funded by the Ministry of Science, ICT & Future Planning.

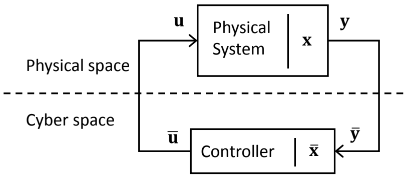


Figure 1: A System Diagram of CPS.

2 SYSTEM MODEL AND PRELIMINARY

Fig. 1 shows a system diagram of CPS. Notations are described as follows. Notation $\mathbf{y}$ denotes an actual output of the plant, while $\bar{\mathbf{y}}$ denotes a measurement used by a state estimator. Notation $\mathbf{x}$ denotes a physical state of the plant/the physical system and $\bar{\mathbf{x}}$ denotes an estimated state. (If the system has no state estimator, $\bar{\mathbf{x}}$ can be seen as a cyber value that represents the physical state.) Notation $\mathbf{u}$ represents an input actually applied by actuators to the plant, while $\bar{\mathbf{u}}$ represents an output of the control system, i.e., a control input produced by the control system. Notation $\mathbf{x}_r$ represents a desired or reference state. We use $t(\cdot)$ to denote a parameter's time stamp. For example, for an element $\bar{x}_i$, $t(\bar{x}_i)$ denote its time stamp; for an estimated state $\bar{\mathbf{x}}$, $t(\bar{\mathbf{x}})$ denotes the time stamp vector that consists of the time stamp of each element in $\bar{\mathbf{x}}$. We call the combination of $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ as a cyber-physical state $\bar{\mathbf{c}}$, which is denoted as $\bar{\mathbf{c}} = \{\bar{\mathbf{x}}, \bar{\mathbf{u}}\}$.

We use $|\mathbf{P}|$ to denote the matrix whose elements are absolute values of the initial matrix $\mathbf{P}$. For matrices $\mathbf{P}$ and $\mathbf{Q}$, $\mathbf{P} \leq \mathbf{Q}$ means that matrix $\mathbf{P}$ is element-wise less than or equal to matrix $\mathbf{Q}$.

3 CYBER-PHYSICAL CONSISTENCY

Cyber-physical consistency defined here is in the intersection of the cyber and physical world. It describes whether the physical world can be accurately reflected by the corresponding cyber part. We consider both logic and timing aspects and a consistent cyber-physical state must be both logic-consistent and timing-consistent.

3.1 Cyber-Physical Logic-Consistency

Definition 1 (Cyber-Physical Logic-Consistency). *A cyber-physical state $\bar{\mathbf{c}} = \{\bar{\mathbf{x}}, \bar{\mathbf{u}}\}$ is logic-consistent if*

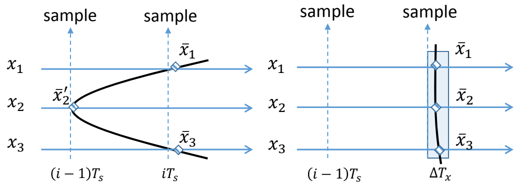
$$\{|\bar{\mathbf{x}} - \mathbf{x}| \leq \Delta\mathbf{V_x}\} \tag{1}$$

$$\wedge\ \{|\bar{\mathbf{u}} - \mathbf{u}| \leq \Delta\mathbf{V_u}\}, \tag{2}$$

where $\Delta\mathbf{V_x}$ and $\Delta\mathbf{V_u}$ denote the given estimation error and actuation error, respectively, that a system can tolerate.

Eqn. (1) checks whether the cyber values, i.e., estimated states $\bar{\mathbf{x}}$, can accurately capture the physical state $\mathbf{x}$. Eqn. (2) checks whether the cyber values, i.e., output $\bar{\mathbf{u}}$ of the control system, can be accurately actuated to the plant. Some faults or attacks can make cyber-physical states violate this logic-consistency. For example, injection attacks [12], such as injecting malicious signals to sensors or injecting malicious packets to the communication between sensors and processors, can compromise sensor measurements, which thus can cause estimated states far from real physical states.

The logic-consistency defined in Def. 1 is confined to values of cyber-physical states. This is not enough for a control system, where the correctness of results also relies on timing requirements. For example, as shown in Fig. 2(a), we consider estimated state values $\bar{\mathbf{x}}' = [\bar{x}_1, \bar{x}_2', \bar{x}_3]^\intercal$ and physical state values $\mathbf{x}' = [x_1, x_2', x_3]^\intercal$, where $\bar{x}_2'$ and x_2' are of $(i-1)^{th}$ sampling period and other individual elements are of i^{th} sampling period. Even if it satisfies logic-consistency, i.e., $|\bar{\mathbf{x}}' - \mathbf{x}'| \leq \Delta\mathbf{V_x}$, individual elements of $\bar{\mathbf{x}}'$ are of different sampling periods and thus $\bar{\mathbf{x}}'$ may be not usable for control. To address this, we will define consistency in terms of timing aspects in the following.



(a) $\bar{\mathbf{x}}'$ does not satisfy Eqn. (3).

(b) $\bar{\mathbf{x}}$ satisfies Eqn. (3).

Figure 2: An example illustrating Eqn. (3).

3.2 Cyber-Physical Timing-Consistency

Definition 2 (Cyber-Physical Timing-Consistency). *A cyber-physical state $\bar{\mathbf{c}} = \{\bar{\mathbf{x}}, \bar{\mathbf{u}}\}$ is timing-consistent if it satisfies*

(1) *Syn-Timing-Consistency:*

$$\{|\max_{\forall i} t(\bar{x}_i) - \min_{\forall j} t(\bar{x}_j)| \leq \Delta T_x\} \tag{3}$$

$$\wedge\{|\max_{\forall j}\ t(\bar{u}_j) - \min_{\forall i}\ t(\bar{x}_i)| \leq T_s\}, \tag{4}$$

where ΔT_x denotes the maximum difference of states' time stamps that a system can tolerate; T_s is the sampling period.

(2) *Exp-Timing-Consistency:*

$$q(\bar{\mathbf{c}}) \geq h, \tag{5}$$

where $q(\cdot)$ is the expire time of a cyber-physical state and h denotes the current time.

We consider a discrete time model of the system. To deal with a continuous-time plant, it is necessary to discretize the plant. Here, the plant's output is sampled (i.e., measured) with a period of T_s, and actuators apply the newly calculated input in each sampling period. Based on this, Def. 2 is interpreted as follows.

1) Syn-Timing-Consistency. This consistency describes synchronization between individual elements of a cyber-physical state. Eqn. (3) expresses that time stamps of individual states (i.e., individual element of $\bar{\mathbf{x}}$) should be close enough to each other. Their difference should be not greater than the threshold (i.e., ΔT_x) that a system can tolerate.

Fig. 2 shows an example that illustrates Eqn. (3). In Fig. 2(a), for the estimated state $\bar{\mathbf{x}}'$, we know that $|t(\bar{x}_1) - t(\bar{x}_3)| < \Delta T_x$, but meanwhile, $|t(\bar{x}_1) - t(\bar{x}_2')| > \Delta T_x$ and $|t(\bar{x}_3) - t(\bar{x}_2')| > \Delta T_x$. Thus, it is not syn-timing-consistent. In Fig. 2(b), for the estimated state $\bar{\mathbf{x}}$, we know that $|t(\bar{x}_1) - t(\bar{x}_2)| < \Delta T_x$, $|t(\bar{x}_1) - t(\bar{x}_3)| < \Delta T_x$, and $|t(\bar{x}_2) - t(\bar{x}_3)| < \Delta T_x$. Thus, it satisfies Eqn. (3). Yet, whether the corresponding cyber-physical state is syn-timing-consistent is still in question. We need to further check Eqn. (4).

Eqn. (4) expresses that a control input should be produced within the same sampling period with the used estimated state. Fig. 3 shows an example that illustrates Eqn. (4) of the syn-timing-consistency. Control input $\bar{u}_j$ is produced at time $t(\bar{u}_j)$. Then, it is applied to the actuator until the new control input $\bar{u}_j'$ is produced and applied. We can see that $|t(\bar{u}_j') - t(\bar{x}_i)| > T_s$, i.e., $\bar{u}_j'$ and $\bar{x}_i$ are in different

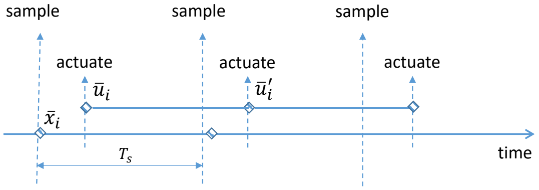


Figure 3: An example illustrating Eqn. (4).

sampling periods, and thus it violates Eqn. (4) and is not syn-timing-consistent. By contrast, $\bar{u}_j$ and $\bar{x}_i$ are in the same sampling period, i.e., $|t(\bar{u}_j) - t(\bar{x}_i)| < T_s$. If a cyber-physical state $\bar{\mathbf{c}} = \{\bar{\mathbf{x}}, \bar{\mathbf{u}}\}$ satisfies both Eqn. (3) and Eqn. (4), it is syn-timing-consistent.

As mentioned above, even though an estimated state is cyber-physical logic-consistent, it may not satisfy syn-timing-consistency. Some attacks can create such estimated states. For example, we consider replay attacks [12] that capture a valid sequence of sensor measurements and then retransmit that valid measurements with some delay. By this manner, an attacker can feed legitimate looking measurements while performing an attack that makes estimated states violate syn-timing-consistency. Time stamps of individual elements are different because of replay attacks.

2) Exp-Timing-Consistency. This timing-consistency is defined based on the expiration time of a cyber-physical state. The rationale behind this definition is that it is usually better to utilize cyber-physical states that are freshly recorded when carrying out recovery. (Refer to [7] for CPS recovery.) Using cyber-physical states stored far from the current time may result in considerable control performance degradation or violating time requirements. Thus, we use a timing property $q(\bar{\mathbf{c}})$, i.e., an expiration time, to capture the freshness of a cyber-physical state $\bar{\mathbf{c}}$. The expiration time is an absolute time, after which the cyber-physical state cannot be used for recovery. States stored earlier have earlier time stamps and thus have smaller expiration times than states that are stored later.

We illustrate how to determine the expiration time of a cyber-physical state with an example based on the CPS recovery framework proposed in [7]. In that paper, we propose a roll-forward recovery framework that handles compromised sensor measurements or incorrect estimated states. The framework is defined as rolling the system to the current time, starting from a historical cyber-physical state. The essential operation is state prediction, i.e., predicting the current state based on a historical state. Please refer to [7] for details of roll-forward recovery. We use a function $g(\bar{\mathbf{c}}, \lambda)$ to denote the predicted state of an amount λ of time later and based on a cyber-physical state $\bar{\mathbf{c}}$. We further use a function $\epsilon(\bar{\mathbf{c}}, \lambda)$ to denote the corresponding prediction error. Then, the expiration time of the cyber-physical state $\bar{\mathbf{c}}$ is

$$q(\bar{\mathbf{c}}) = \min_{\epsilon(\bar{\mathbf{c}},\lambda) \npreceq \mathbf{E}} \lambda + t(\bar{\mathbf{c}}), \tag{6}$$

where $\mathbf{E}$ is the maximum error that a system can tolerate. As shown in Fig. 4, the expiration time $q(\bar{\mathbf{c}})$ is the time point when some element of $\epsilon(\bar{\mathbf{c}}, \lambda)$ becomes just greater than the corresponding element of $\mathbf{E}$.

Some distance metrics quantifying value discrepancy and timing discrepancy have also been presented in previous works such as [1, 8]. The difference is that they are focused on distance between traces while our work targets that of states. In future work, we will study the expiration time of a cyber-physical state for more concrete scenarios.

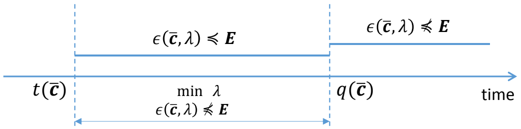

$\epsilon(\bar{\boldsymbol{c}},\lambda) \preccurlyeq \boldsymbol{E}$
$\epsilon(\bar{\boldsymbol{c}},\lambda) \npreceq \boldsymbol{E}$
$t(\bar{\boldsymbol{c}})$
$\min_{\epsilon(\bar{\boldsymbol{c}},\lambda) \npreceq \boldsymbol{E}} \lambda$
$q(\bar{\boldsymbol{c}})$
time

Figure 4: Illustration of determining the expiration time for a cyber-physical state $\bar{\mathrm{c}}$. $q(\bar{\mathrm{c}})$ is the expiration time.

4 EVALUATION

To validate consistency definitions and highlight their utility when carrying out recovery, we conduct extensive simulations and analysis. We use Simulink to build a simulator, where a DC motor drives an inertial load. DC motors are widely used in electric vehicles and many autonomous car prototypes. We use the dynamic model of the DC motor given as follows

$$\begin{bmatrix} \dot{i} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_m}{J} & -\frac{K_f}{J} \end{bmatrix} \begin{bmatrix} i \\ w \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v, \tag{7}$$

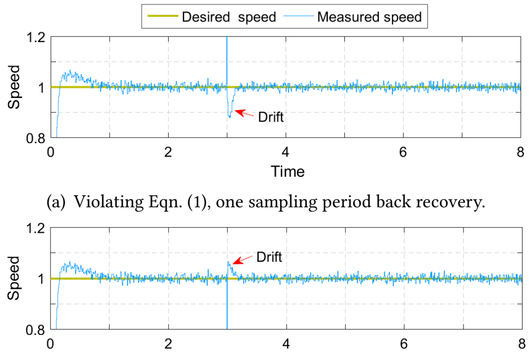
where the current i and the angular velocity w are considered as the two individual states of the system. The applied voltage v is the control input, and the angular velocity w is the output of the system. The resistance R and the self-inductance L are set as 1 and 0.5, respectively. Both the armature constant K_m and the EMF constant K_b are set as 0.01. The viscous friction constant K_f is set as 0.1. The inertial load J is 0.01. The reference velocity or desired speed is set as 1. the sampling period is set as 0.01. Sensor (of speed) noise obeys Gaussian distribution with the variance of $\Theta = 0.0001$.

We use a PID controller to supervise and control the motor's speed. The scenario considered for this experiment is that the operator specifies the desired motor (or vehicle) speed, and the controller needs to ensure this speed even if the system performs roll-forward recovery. However, this goal may be compromised if the recovery is carried out based on inconsistent cyber-physical states.

4.1 Results for Cyber-Physical Consistency

We consider the recovery occurs at time 3. The recovery occurs once, and during the recovery period, the system uses recovered state for control. After recovery, i.e., from time 3.01, the system comes back to utilize sensor measurements for control.

Fig. 5 demonstrates results for cyber-physical logic-consistency. The recovery uses the cyber-physical state of one sampling period back. Fig. 5(a) shows control performance if violating Eqn. (1). To produce this violation, we add one to the original value of the measured speed. Base on this new value, the recovered state (i.e., motor speed here), shown by the spike in Fig. 5(a), is large and far from the actual value. That is, it is incorrect, i.e., it incorrectly reflects the actual motor's speed. Thus, the controller decelerates the motor, as shown by the drift-off in this figure. Fig. 5(b) depicts control performance for the case if violating Eqn. (2) . To produce this violation, we change the control input to a much smaller value, e.g., −800 here. Based on this new value, the recovered state (i.e., motor speed here), shown by the sharp decrease in Fig. 5(b), also becomes smaller and is far from the actual value. Thus, the controller accelerates the motor and make it drift off from the desired speed, as shown in this figure. Based on these observations, we conclude that



(a) Violating Eqn. (1), one sampling period back recovery.

(b) Violating Eqn. (2), one sampling period back recovery.

Figure 5: Cyber-Physical Logic-Consistency.

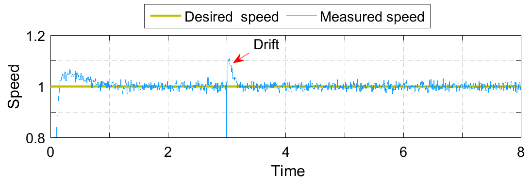
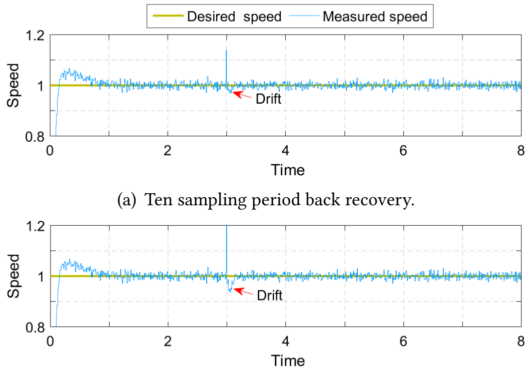


Figure 6: Cyber-Physical Syn-Timing-Consistency.



(a) Ten sampling period back recovery.

(b) One hundred sampling period back recovery.

Figure 7: Cyber-Physical Exp-Timing-Consistency.

roll-forward recovery based on logic-inconsistent cyber-physical states can significantly compromise control performance.

Fig. 6 shows control performance when violating cyber-physical syn-timing consistency. The recovery uses the cyber-physical state of the preceding sampling period. To make it violate syn-timing consistency, we change the speed value of the cyber-physical state to zero. This is the same as using the speed at time 0, which is also zero. Doing this makes the two state variables i.e., speed and current, have different time stamps. Based on this newly made state, the recovered state, as shown by the sharp decrease in this figure, becomes far smaller than the desired speed. Thus, the controller accelerates the motor and make it drift off from the desired speed, as shown in the figure. Hence, we can conclude that roll-forward recovery based on syn-timing-inconsistent cyber-physical states can also cause considerable control performance degradation.

Fig. 7 plots results for cyber-physical exp-timing-consistency. We consider control performance degradation for carrying out one time of recovery using cyber-physical states back to different sampling periods. Fig. 7(a) shows control performance for the case if the recovery uses the cyber-physical state of 10 sampling periods back. We can see that for this case, the motor's speed quite well tracks the desired speed because the drift-off is very small. In other words, using logic-consistent cyber-physical state that is fresh enough, to carry out recovery results in little performance degradation. By contrast, control performance compromises more when using the cyber-physical state of 100 sampling periods before, as shown by Fig. 7(b). From this figure, we can see that the recovered state/motor speed is far from the realistic value. This causes the controller to decelerate the motor considerably and thus much drifts off. These observations demonstrate the key point as follows. Not all historical cyber-physical states is appropriate to be used for recovery, and it needs to utilize fresh enough states in order to guarantee control performance. This further demonstrates the necessity to define and attribute expiration time for cyber-physical states.

5 CONCLUDING REMARKS

We study consistencies for CPS roll-forward recovery. Speically, we define 1) cyber-physical logic-consistency based on whether cyber values can accurately reflect the corresponding physical state, 2) cyber-physical syn-timing-consistency based on time stamp difference of individual elements of estimated states and control inputs, and 3) cyber-physical exp-timing-consistency based on expiration times of cyber-physical states. We build a simulator of PID controlled DC motor to evaluate how control performance is affected by these consistencies when conducting recovery.

REFERENCES

[1] Houssam Abbas, Hans Mittelmann, and Georgios Fainekos. 2014. Formal property verification in a conformance testing framework. In *ACM/IEEE Conference on Formal Methods and Models for Codesign (MEMOCODE)*. IEEE.

[2] Alvaro A Cardenas, Saurabh Amin, and Shankar Sastry. 2008. Secure control: Towards survivable cyber-physical systems. In *International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE.

[3] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, et al. 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces.. In *USENIX Security Symposium*. San Francisco.

[4] Kyong-Tak Cho and Kang G Shin. 2016. Error Handling of In-vehicle Networks Makes Them Vulnerable. In *ACM Conference on Computer and Communications Security (CCS)*. ACM.

[5] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. 2014. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Trans. Automat. Control* (2014).

[6] Radoslav Ivanov, Miroslav Pajic, and Insup Lee. 2016. Attack-resilient sensor fusion for safety-critical cyber-physical systems. *ACM Transactions on Embedded Computing Systems* (2016).

[7] Fanxin Kong, Meng Xu, James Weimer, Oleg Sokolsky, and Insup Lee. 2018. Cyber-Physical System Checkpointing and Recovery. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. ACM.

[8] Chiheb Kossentini and Paul Caspi. 2006. Approximation, sampling and voting in hybrid computing systems. In *International Workshop on Hybrid Systems: Computation and Control (HSCC)*. Springer.

[9] Miroslav Pajic, James Weimer, Nicola Bezzo, Paulo Tabuada, Oleg Sokolsky, Insup Lee, and George J Pappas. 2014. Robustness of attack-resilient state estimators. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. ACM.

[10] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. 2015. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe* (2015).

[11] Aviva Hope Rutkin. August 14, 2013. "Spoofers" Use Fake GPS Signals to Knock a Yacht Off Course.

[12] André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik Johansson. 2012. Attack models and scenarios for networked control systems. In *Proceedings of the International Conference on High Confidence Networked Systems (HiCoNS)*. ACM.