Security-Aware Functional Modeling of Cyber-Physical Systems

Jiang Wan[1], Arquimedes Canedo[2], and Mohammad Abdullah Al Faruque[1]

[1]{*jiangwan, alfaruqu*}*@uci.edu*
[1]Department of Electrical Engineering & Computer Science, University of California, Irvine, USA
[2]*arquimedes.canedo@siemens.com*
[2]Siemens Corporation, Corporate Technology, Princeton, USA

***Abstract*—Security is one of the major challenges for Cyber-Physical Systems (CPS) design. Identifying flaws as early as possible in the CPS design saves time and money [6]; between 5× to 10× less expensive [7] than finding them during the detailed design stages. This paper makes a case for finding *cybersecurity flaws* as early as possible. Not only for the temporal and cost benefits, but more importantly, for the integrity of the system once in operation. We introduce a security-aware functional modeling methodology, supported by simulation to validate the robustness of the system in the presence of attacks and countermeasures. Our ideas are implemented in a design automation tool in Amesim and Matlab/Simulink. We use an automotive use-case as an example to validate the methodology and the tool.**

I. Introduction and Related Work

The increasing amount of software and communication networks in Cyber-Physical Systems (CPS) deployed in many industries is also making them more vulnerable to the cyber-attacks [8], [14]. One easy-to-relate example is automobile design; over the years it has shifted from being mechanical-centric to software-centric [4]. It has been shown that different types of attackers and attack scenarios such as through wireless access to the car, through physical access to a car, and through access to the in-vehicle network (tampering Electronic Controls Units (ECU)) may happen in the real world scenarios [5], [9], [15], [17]. An important observation is that even though systems are becoming software-centric, attacks can be physics-based. In [16], the authors have implemented an ABS spoofer that could physically attack on the wheel speed sensor so that the functionality of the Anti-lock Braking System (ABS) is completely distorted. These principles can be applied to other complex systems that control critical infrastructure.

Unfortunately, there is currently a gap between the design automation tools and cybersecurity. This paper advocates the development of new methodologies and tools to support designers, as early as possible during the design cycle, to identify and solve cybersecurity problems in complex systems [10]. In [19], a Model-Based Design (MBD) method to assess the security of CPS is proposed with four architecture-level attack models. Authors in [13] have discussed a MBD technique to quantify the security metrics at the early design stage. To achieve the proposed metrics, a security modeling method is proposed in this paper [13]. Furthermore, graph-based modeling methods are used in many security problems. Authors in [20] have proposed a systematic method for analyzing cyber-attacks on CPS based on an extended Data Flow Diagram (xDFD) approach. Existing work in [12] has proposed an attack tree based approach for the system-level security design. However, most of the modeling methods used in the existing works are limited on modeling the cyber domain software for security analysis.

At the early design stage, MBD is widely used for CPS design. Our work proposes the formulation of the security problem or attack patterns *before* the system is built. We exploit the same observation than in concept design where identifying and fixing problems at the early stages is economically beneficial for the costs, performance, and reliability of the system. Therefore, we have developed a design automation tool that uses simulation to validate cybersecurity vulnerabilities at the system-level. We model cybersecurity attacks and countermeasure functionalities using a novel *security-aware functional modeling* language (see Section III) implemented in the commercial design and simulation tools (see Section IV). Currently, our library of attacks consists of six attack models. We validate the methodology and demonstrate that our idea can be integrated to current design workflows (see Section V). Our experimental results show the benefit of our methodology using an automotive use-case.

II. Our Novel Contributions

Building upon our previous work on functional modeling [2], [3], [4], [18], this paper proposes a *security-aware* functional modeling methodology. Moreover, since both the cyber domain (control blocks and signal flows) and the physical domain (physical process and energy flows) may be modeled using functions, we extend this notion to represent *cybersecurity functions* that represent attacks and countermeasures. The benefit of such methodology, is that we characterize the cyber-physical security problem in an implementation independent manner.

The novel contributions of this paper are:

1) A security-aware functional modeling methodology.
2) Cybersecurity functions and their associated attack and countermeasure models.
3) System-level modeling and simulation of attacks beneficial for the concept design stage of CPS.

III. Security-Aware Functional Model

Traditional functional models describe only two types of functions: physical and cyber. Functions interact with each

other through flows that reflect the base functions using energy, material, and signal flows. These flows carry real physical and cyber properties such as mechanical, electrical, thermal energy, and data. Thus, existing functional models naturally *leak* information that can be used to attack the system via the signal flows in the cyber domain or energy/material flows in the physical domain. We extend the functional modeling concept and include cybersecurity functions. Our proposed *security-aware functional models* provide the means to both analyze the effect of cybersecurity attack functions in the system, and refine the design using cybersecurity countermeasure functions.

An example of a security aware functional model for a car is shown in Figure 1. In this example, both a cyber and a physical attacks are modeled. The blue arrows in the flows represent the cyber attack vectors, and the red arrows represent the physical attack vectors. The purpose of the analysis is to quantify the effects of these attacks in the `Export TME` function that maps to the velocity of the car. In other words, "is the speed control of the car vulnerable to a cyber-physical attack?".

In order to evaluate the security level of the Export TME function, we simulate the model and analyze the impact of the attack at the system-level. This step is important to identify functions with low security levels. These low security functions can be protected by refining the functional model and adding cybersecurity countermeasure functions. This iterative approach facilitates the analysis of different scenarios. An important benefit of the iterative approach is that more complex scenarios can be composed.

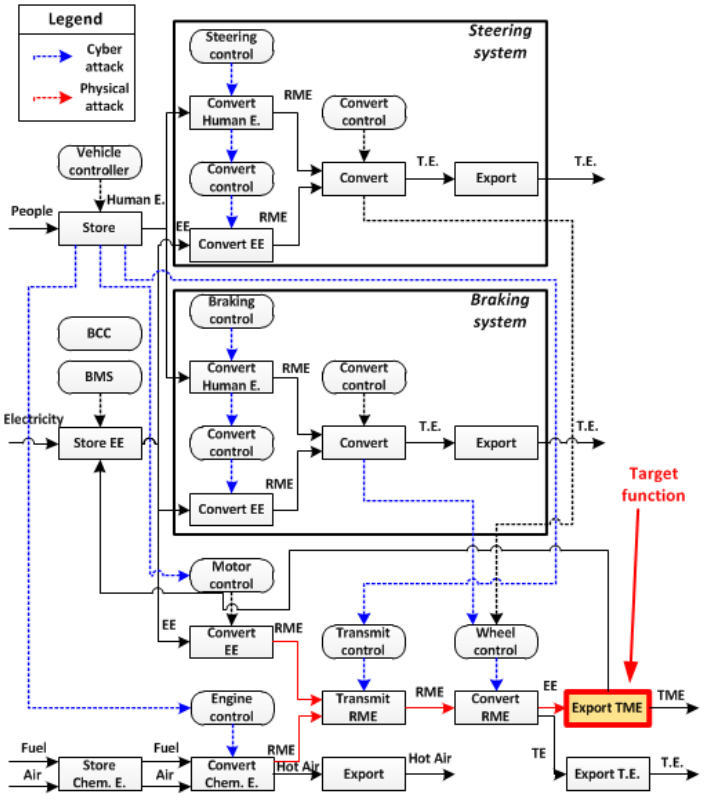

Fig. 1. Security aware functional level model of automotive.

From the analysis of this security-aware functional model, a system-level simulation model can be directly generated using the synthesis technology provided in the existing works [2], [3], [4], [18].

IV. Design Automation Tool

To validate the methodology, we have implemented a design automation tool using commercial of-the-shelf software as shown in Figure 2. Modeling of the system functions (multi-physics), cybersecurity functions, and scenarios (e.g., environmental conditions) is done in Amesim in order to take advantage of the libraries of subsystems. And the modeling of attack models is done in Matlab/Simulink because we take advantage of the advanced mathematical capabilities of the software.

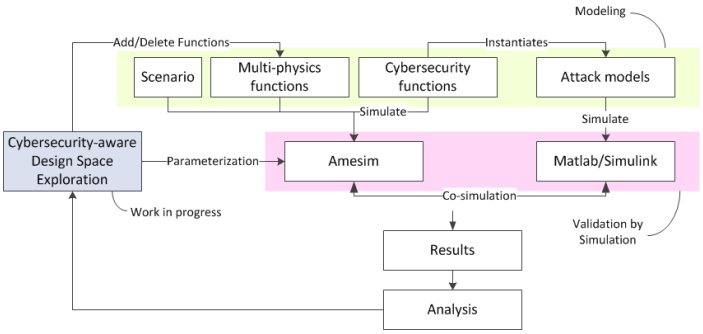

Fig. 2. Our proposed design automation tool.

Notice that cybersecurity functions in Amesim are proxies for the attack models modeled in Matlab/Simulink; this is shown in the screenshot of our tool in Figure 3. Cybersecurity functions are instantiated into attack models during simulation. The composed system is validated by simulation using a co-simulation between Amesim and Matlab/Simulink. All the functions are instantiated to components in Amesim libraries, and the cybersecurity functions are instantiated in the Matlab/Simulink attack models. The results are analyzed by the user using standard time series and plotting capabilities of the simulation software. Currently, we are working on automating the analysis process and performing a cybersecurity-aware design space exploration. Our goal is to allow the tuning of parameters in the system to make it more resilient to attacks in different scenarios.

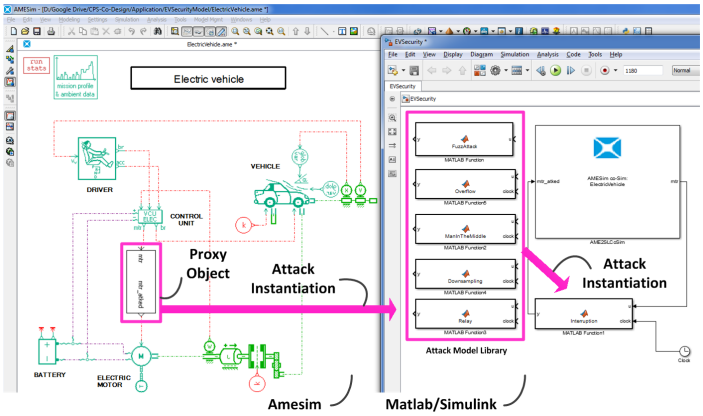

Fig. 3. Implementation of the proposed design automation tool.

We have developed and integrated six attack models to our design automation tool. The developed attack models are mathematically represented as follows. u_s is the original signal/flow and v_s is the attacked signal/flow. Most of the attack happens during the attack period from τ_{start} to τ_{end}.

Fuzzy attack model [19] adds a random distribution (e.g., uniform distribution $unif(atk_{upper}, atk_{lower})$ triggered by the poisson clock $fclock_{pois}$) to the selected functional signals/flow to destabilize the system.

$$\begin{aligned} &Event = fclock_{pois}(\lambda) \\ &atk_s = \begin{cases} unif(atk_{upper}, atk_{lower}) & Event \neq 0 \\ atk_s & Event = 0 \end{cases} \\ &v_s = u_s + atk_s \end{aligned} \tag{1}$$

Interruption attack model [19], also referred to as denial-of-service attack, stops the functional signal/flow during the attack period.

$$v_s = \begin{cases} 0 & \tau_{start} < \tau < \tau_{end} \\ u_s & else \end{cases} \tag{2}$$

Man-in-the-middle attack model [19] mimics a human attack behavior. When the attack occurs, the functional signal/flow is changed to the manipulated signal/flow controlled by the attacker.

$$v_s = \begin{cases} u_m & \tau_{start} < \tau < \tau_{end} \\ u_s & else \end{cases} \tag{3}$$

Replay attack model [19] records the functional/signal flows over a period of time in a vector. When the attack starts, this attack model replaces the actual signal/flow with recorded data of other subsystems.

$$\begin{aligned} &u_r[\tau \bmod \tau_{period}] = u_s \\ &v_s = \begin{cases} u_r[\tau - \tau_{start}] & \tau_{start} < \tau < \tau_{end} \\ u_s & else \end{cases} \end{aligned} \tag{4}$$

Overflow attack model [5], [11] extends the number of bits of the message in a signal in order to overflow the buffers on the receiver's side.

$$v_s = \begin{cases} u_s >> Num_{bits} << Num_{bits} & \tau_{start} < \tau < \tau_{end} \\ u_s & else \end{cases} \tag{5}$$

Down-sampling attack model [1] reduces the sampling rate of functional signal/flow. This attack is known to reduce the Quality of Control (QoC) of a system.

$$\begin{aligned} &u_{low} = \begin{cases} u_s & \tau \bmod rate_{down} = 0 \\ u_low & else \end{cases} \\ &v_s = u_{low} \end{aligned} \tag{6}$$

V. Experimental Results

To demonstrate the proposed design automation tool and the attack models during the early design stage, we use the design of an Electric Vehicle (EV) as a case study shown in Figure 3. First, we run the simulation of the ELECTRIC MOTOR without instantiating any attack model, the results are shown in Figure 4. In order to evaluate our attack models on a realistic scenario, we inserted the attack model interface in between the motor and the ECU controlling the motor.

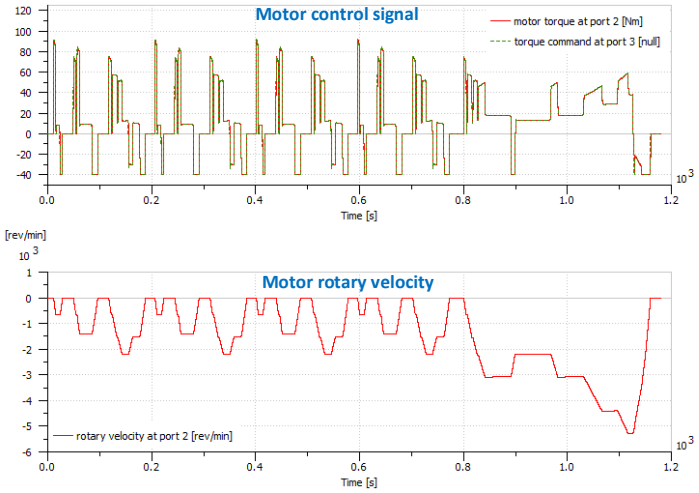

Fig. 4. Simulation results without attack model.

The attack model interface is instantiated by one of the attack models from the attack model library as presented in Figure 3.

Figure 5(a) shows that the fuzzy attack may destabilize the motor by adding noise to the control signals. Figure 5(b) shows that the control signal is interrupted and out of service from 200s to 800s. Figure 5(c) shows that the motor is controlled by a linear signal that is defined manually by the attackers in the attack model. Figure 5(d) shows that, instead of getting control signals based on the current status, the attack model is replacing the signals with recorded signals from 200s to 300s. Figure 5(e) is showing that the overflow attack reduces the accuracy of the control signals by overflowing the buffer, thus the lower-bits of the control signals is lost. As a result, the total Quality of Control (QoC) is reduced. However, in this example, the simulation results demonstrate very few reductions on QoC, which means the overflow attack will not affect this signal. Figure 5(f) is showing that the overflow attack model will reduce accuracy of the signals, and thus reduce the QoC of the motor.

VI. Conclusion and Future Work

In this work, functional model is used to help the analysis of security problem during the early design stage. In future, we will explore the capability of the functional model to not only help the analysis, but also to automatically generate/synthesize system-level simulation models including attack models and system architecture to help the evaluation of security strength. Moreover, for the demonstration purpose, six different attack models are developed in the current attack library. Additional attack models which may capture both cyber domain and physical domain attacks will be developed to enrich the library in our future work. Moreover, to help the system designers to quickly identify the security problem during the early stage. Security cost metrics may be developed and used in the proposed security-aware functional models.

References

[1] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina. Impact of packet sampling on anomaly detection metrics. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 159–164. ACM, 2006.

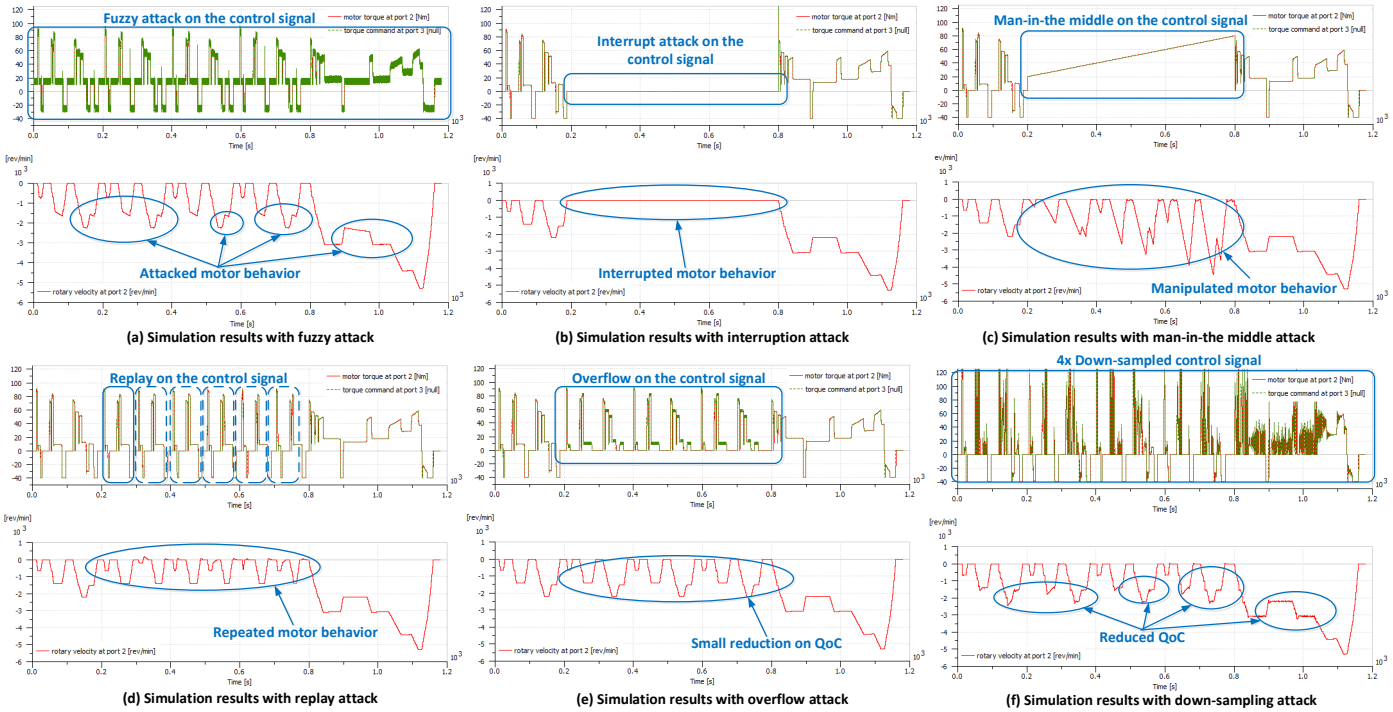

Fig. 5. Simulation results with proposed attack models.

[2] A. Canedo, A. Faruque, M. Abdullah, and J. H. Richter. Multi-disciplinary integrated design automation tool for automotive cyber-physical systems. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 315. European Design and Automation Association, 2014.

[3] A. Canedo, E. Schwarzenbach, and M. A. Al Faruque. "Context-sensitive synthesis of executable functional models of cyber-physical systems". *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 99–108, 2013.

[4] A. Canedo, J. Wan, and M. A. A. Faruque. "Functional Modeling Compiler for System-Level Design of Automotive Cyber-Physical Systems". In *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, 2014.

[5] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*. San Francisco, 2011.

[6] D. Dumbacher and S. R. Davis. Building operations efficiencies into nasas ares i crew launch vehicle design. In *54th Joint JANNAF Propulsion Conference*, 2007.

[7] G. Fortney. "Model based systems engineering using validated executable specifications as an enabler for cost and risk reduction". *Proceedings of the 2014 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2014.

[8] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Moderator-Ravi. Security as a new dimension in embedded system design. In *Proceedings of the 41st annual Design Automation Conference*, pages 753–760. ACM, 2004.

[9] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al. Experimental security analysis of a modern automobile. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 447–462. IEEE, 2010.

[10] F. Koushanfar, A.-R. Sadeghi, and H. Seudie. Eda for secure and dependable cybercars: challenges and opportunities. In *Proceedings of the 49th Annual Design Automation Conference*, pages 220–228. ACM, 2012.

[11] P. Meerwald and S. Pereira. Attacks, applications and evaluation of known watermarking algorithms with checkmark. In *Proc. SPIE*, volume 4675, page 294, 2001.

[12] A. P. Moore, R. J. Ellison, and R. C. Linger. Attack modeling for information security and survivability. Technical report, DTIC Document, 2001.

[13] D. M. Nicol, W. H. Sanders, and K. S. Trivedi. Model-based evaluation: from dependability to security. *Dependable and Secure Computing, IEEE Transactions on*, 1(1):48–65, 2004.

[14] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady. Security in embedded systems: Design challenges. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(3):461–491, 2004.

[15] F. Sagstetter, M. Lukasiewycz, S. Steinhorst, M. Wolf, A. Bouard, W. R. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty. Security challenges in automotive hardware/software architecture design. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 458–463. EDA Consortium, 2013.

[16] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava. Non-invasive spoofing attacks for anti-lock braking systems. In *Cryptographic Hardware and Embedded Systems-CHES 2013*, pages 55–72. Springer, 2013.

[17] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaâniche, and Y. Laarouchi. Survey on security threats and protection mechanisms in embedded automotive networks. In *Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference on*, pages 1–12. IEEE, 2013.

[18] J. Wan, A. Canedo, and M. A. A. Faruque. "Functional Model-based Design Methodology for Automotive Cyber-Physical Systems". *IEEE Systems Journal (ISJ)*, 2014.

[19] A. Wasicek, P. Derler, and E. A. Lee. Aspect-oriented modeling of attacks in automotive cyber-physical systems. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6. IEEE, 2014.

[20] M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits. Systematic analysis of cyber-attacks on cps-evaluating applicability of dfd-based approach. In *Resilient Control Systems (ISRCS), 2012 5th International Symposium on*, pages 55–62. IEEE, 2012.