A Federated Approach for Simulations in Cyber-Physical Systems

Hoang Van Tran[1](✉), Tuyen Phong Truong[1], Khoa Thanh Nguyen[1],
Hiep Xuan Huynh[2], and Bernard Pottier[1]

[1] Université de Bretagne Occidentale, Brest, France
`tvhoang75@gmail.com`, `tptuyen@gmail.com`, `ngthanhkhoa@gmail.com`,
`pottier@univ-brest.fr`
[2] Can Tho University, Can Tho, Vietnam
`hxhiep@ctu.edu.vn`

Abstract. In this paper, we propose a new approach to federate simulations for cyber-physical systems. A federation is a combination of simulations being able to interact with each other upon a standard of communication. In addition, mixed simulations are defined as several parallel simulations that are taken place in a common time progress. These simulations run on the models of physical systems, which are built based on cellular automata. The experimental results are performed on a typical federation of three simulations for forest fire spread, river pollution diffusion and wireless sensor network. The obtained results can be used to predict as well as observe the behavior of physical systems in their interactions.

Keywords: Cyber-Physical System (CPS) · Cellular Automata (CA) · Federation · High Level Architecture (HLA) · Mixed simulations

1 Introduction

In recent years, more and more researches focus on the cyber-physical system (CPS) [1], which is defined as an integration of computation with the physical systems. Taking advantage of Wireless sensor network (WSN) [12], sensing ability of CPS is significantly considered over the last years. This ability allows CPSs to be able to sense data from the physical world by the facilitation of sensor networks.

Simulating the sensing ability before real implementations will highly reduce cost and effort of development of CPSs. However, due to uncertain interactions of complex physical systems, simulating this type of system is much more complicated compared to the traditional computing systems. One of critical challenges is the involvement of the interoperability in the models.

Several solutions were suggested to confront with that issue over the last years [5,6,20]. But, they do not consider on federating physical systems instead of integrating exiting tools and languages. Furthermore, those approaches are

P.C. Vinh and V. Alagar (Eds.): ICCASA 2015, LNICST 165, pp. 165–176, 2016.
DOI: 10.1007/978-3-319-29236-6_17

almost targeting embedded systems. This leads to challenges due to the needs of modeling various kinds of physical systems in the sensing process, natural systems as examples.

Cellular automata (CA) [2,3] model has emerged as a very promising technique for dealing with complex physical systems [4]. A typical CA consists of two components: cellular space and transition rule. The cellular space component presents a lattice of cells, each with an identical pattern of local connection to other cells (neighborhood pattern). In addition, each cell consist of a set of states. Meanwhile, the other component indicates how one cell achieves the new states (at time t+1) according to the current local states and states of its neighborhood (at time t).

In this paper, we propose a new approach for federating simulations. This enables to coordinate several parallel simulations as a distributed simulation system, so-called mixed simulation. To achieve it, we at first present a method to model physical systems in accordance with the CA [2,3] model. Then, the parallelism is used to accelerate large scale simulations. A federation of several parallel simulations is conducted according to high level architecture (HLA) [19] standard.

The remainder of this paper is organized as follows. In Sect. 2, we introduce related work. Section 3 describes mixed simulations in the context of CA. A federation of mixed simulations and related discussions are presented in Sect. 4. Section 5 gives some experiments of running a federation. The conclusion of this work will be expressed in Sect. 6.

2 Related Work

In the last decade, several researches have focused on handling challenges of interoperability of various components in CPSs [1].

An approach about coordinating data communication and time synchronization between simulation frameworks is considered. Some practical works following that track are presented in [5,23]. These solutions allow to federate several simulations, however, the issues related to synchronization time are not taken into account.

Likewise, [22] presents a framework for exchanging data and time synchronization by integrating two available tools. This work aims to facilitate design and evaluation of networked control and cyber-physical system (NCCPS) [22] in CPSs.

There are no work replying on applying the CA model for representing physical systems in the context of CPSs. Modeling phenomena and natural systems as well as their interactions in space and time are not taken into account in almost situations.

3 Mixed Simulations

3.1 Modeling Physical Systems Based on Cellular Automata

A physical system is considered as a region which may be monitored by sensor networks such as river, ocean, forest, or road system. However, since their complex behavior and large size, modeling this type of system confronts with many issues. Cellular automata (CA) [2,3] is feasible to handle those issues [4].

In order to model physical systems, a definition of cell system is considered. It consists of a collection of cells. Each cell holds its local states (namely pollution density, insect population), and connections to surrounding cells (neighboring cells), as presented in Fig. 1. We also suppose to use two common types of neighborhood patterns Von neumann 1 (4 neighbor) and Moore 1 (8 neighbor) [2] in this study.

Figure 2 depicts a process of developing physical simulations in terms of the cell systems. Initially, geographic data are processed to generate a cell system, which is associated with definitions of states and transition rules make up a complete model. Physical simulations are achieved by the model executions.

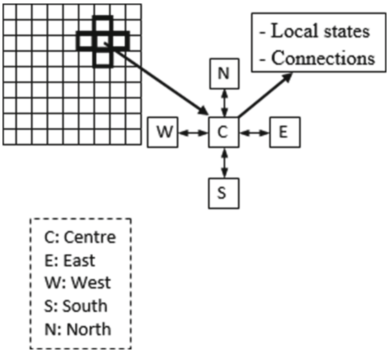

Fig. 1. Illustrating a cell system structure with the Von neumann 1 pattern.

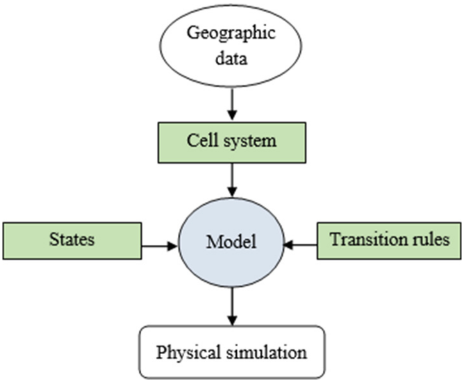

Fig. 2. A process of developing physical simulations from cell systems.

Two models of physical systems are suggested in the next sections: forest fire spread model and river pollution diffusion model. We also suppose that a WSN [12] is deployed to monitor fire in the forest, and its model is thus presented.

3.2 River Pollution Diffusion Model

River pollution diffusion model is built based on the method in the previous section. In the context of pollution, it is possible to think of various potential situations such as chemical, oil, or contaminant. Generally, the diffusion significantly depends on the pollution density. Therefore, pollution density is chosen as cell states

- *state:* The cell state holds a value of the pollution density.
- *transition rules:* Updating the pollution density at $cell_i$ at time t+1, termed as $S(i)_{t+1}$.

```
$S(i)_{t+1} \leftarrow S(i)_t/2$
for (j in neighbor of cell_i)
    $S(i)_{t+1} \leftarrow S(i)_{t+1} + S(j)_t$ / (2 * number of neighbor of cell_j)
end for
```

3.3 Forest Fire Spread Model

The fire spread model is defined for simulating the fire spread in the forest. For the sake of simplicity, cell state is represented by one of the four values: tree, fire, ash, and empty.

- *state:* tree, fire, ash, and empty.
- *transition rules:* Updating the new state of *cell*$_i$ at time t+1, termed as $S(i)_{t+1}$.

```
if ($S(i)_t$ is TREE and at least one of its neighbor is FIRE)
    $S(i)_{t+1} \leftarrow FIRE$
end if
if ($S(i)_t$ is FIRE)
    $S(i)_{t+1} \leftarrow ASH$
end if
if ($S(i)_t$ is ASH)
    $S(i)_{t+1} \leftarrow EMPTY$
end if
```

3.4 Wireless Sensor Network Model

To represent a sensing component of CPSs, wireless sensor network (WSN) is also taken into account. It regularly collect raw data from the forest, processes those data, and raises emergency alerts in the case of the fire detected. To carry it out, a collection of sensors will first be deployed to monitor the forest. In this case, each sensor node is considered as a cell in cell system. The connectivity of the model is formed by radio links among sensors. And sensing data can be viewed as cell states of the model.

- *state:* The cell state holds a value of sensed data collected from the forest. Thus, its value should be fire or normal.
- *transition rule:* Every simulation cycle, *cell*$_i$ checks the sensed data. Signals will be emitted as the fire is detected.

3.5 Parallel Simulations

The simplicity of the CA models actually brings about several benefits in computations. However, since huge sizes and complicated behavior of the physical systems, simulating them poses a challenge to existing simulator, especially in the case of several simulations working together. The parallelism mechanism is thus employed to deal with that in our work.

To implement it, we propose to use the Graphic Processing Unit (GPU) [7,8]. This architecture comes with a set of threads running in parallel. As a result, the computations of cells are easy to be executed simultaneously.

3.6 Mixed Simulations

A mixed simulation is considered as a collection of parallel simulations that are capable of communicating with each other while running concurrently as a distributed system. It is expected that the mixed simulations are able to imitate not only behavior of real systems, but also interactions between them. A central component is responsible for controlling that interoperability, as shown in Fig. 3.

For instance, considering on interactions between the river and the forest system, as shown in Fig. 4. Ashes produced by fire in the forest can pollute the river at the frontier between them. Otherwise, evaporation will also affects fired spread in the forest. These physical interactions are regarded in this type of simulation.

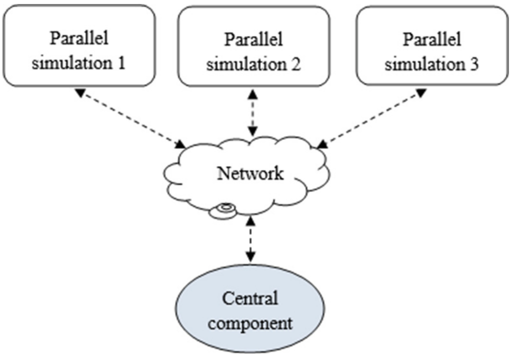

Fig. 3. A general architecture of mixed simulations.

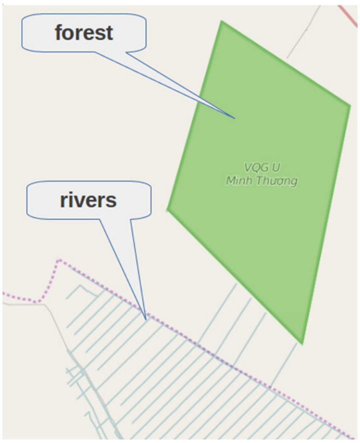

Fig. 4. An example for describing interactions can be happened between river and forest. (data source: OpenStreetMap [16]).

4 Federation of Simulations

As mentioned earlier, a mixed simulation consists of several parallel simulations that concurrently run on different hosts and are connected via a network system. Thus, a federation approach is required to make those parallel simulations working together as a synchronous system. To do that, a model designed in accordance with the High Level Architecture (HLA) [13,14,19] is applied.

In HLA terminology, the entire system is considered as a *federation* which contains several *federates* connected via the central component *Run-Time Infrastructure (RTI)* [19]. The HLA is formally defined by three components.

1. A set of rules describes the responsibilities of federates and their relationship with RTI. An example is that *all exchange of data among federates should occur via the RTI during a federation execution.*
2. An interface specification provides services for managing federates and interactions. For example, it indicates how a federate join or leave a federation.
3. An Object Model Template [15] defines how information is communicated between federates, and how the federates and federation have to be documented (using Federation Object Model FOM) [15]. FOM defines the shared objects, attributes, and interactions for a whole federation.

We assume that some interactions occur between the three systems. Firstly, as fires of the forest approach to the river, ashes produced by the fires will pollute the river. Secondly, the evolution of the physical environment results in the changing of WSN's behavior, emitting signals as soon as the fire is detected.

The overall architecture of the proposed federation is presented in Fig. 5. It is noted that *Observer*, a passive federate, used to visualize the federation.

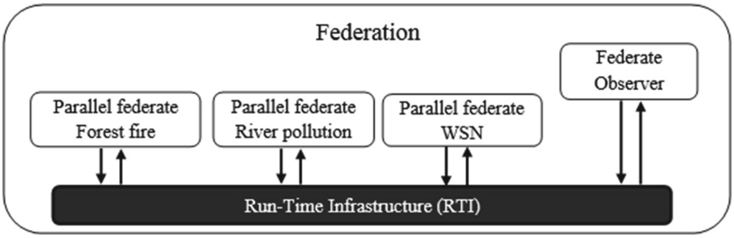

Fig. 5. A federation of four federates: River, forest, WSN, and observer.

4.1 Exchanging Data

Exchanging data between federates is an obligation to the proposed model. To deal with that, a publish-subscribe mechanism is used. This enables subscribers may automatically receive updates from publishers. Thus, federates have to declare what information they publish and subscribe to the central component before the execution of the federation, as shown in Table 1.

The communications are taken place at the end of simulation cycles. This means that the new states are simultaneously computed on the GPUs of the federates which are independently run on their host. Then, data are copied to the CPUs to serve interactions. Thus, the states of publishers at time t may be involved in the evolution of other subscribers at time t+1, as depicted in Fig. 6.

Table 1. Describing publishers and subscribers of shared data between the four federates: forest, river, WSN, and observer.

Object Class	Attributes	Published by	Subscribed by
ForestNode	State, Position	ForestNode	River, WSN, Observer
RiverNode	Pollution density, Position	RiverNode	Observer
WSNNode	State, Position	WSNNode	Obsever

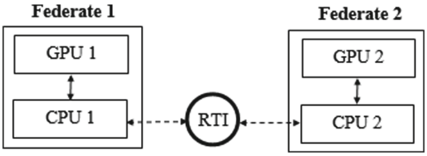

Fig. 6. The model of exchanging data in the federation with two federates.

Clearly, this approach enables to make of an environment of several parallel simulations running at the same time. Although, it costs a little of time for copying data, it significant benefits in the case of several complex and large systems.

4.2 Time Management

Time management is a mechanism for controlling the advancement of each federate in simulation time. This enables federates in the federation to be able to synchronize their local logical time together to ensure the causality. In this case, each federate owns its logical time that will be associated with the sending data, so-called time-stamp. Upon that information, the central component is capable of synchronizing federates as a synchronous system.

This mechanism comes up with two properties, constrained and regulating. The former ensures the federates to be able to send updates. Meanwhile, the latter allows the federates to receive updates from the central component. Therefore, both of them are often enabled for all federates, and only the constrained property is assigned to the observer federate since it is designed without any sending. Table 2 shows time policies proposed for the proposed federation.

Time stepped federates will calculate values based on a point in time and process all data being sent up to the next point in time (current time + time step). Thus, to advance logical time for the time-stepped simulation, each federate has to send its request to the central component. Then, all receive data, which have been sent from federates, with the time-stamp less than or equal to the time requested will be released from the central component. After those data have been received by federates, a time grant is returned to the requesting federate. And then, the federate is able to advance its logical time. In addition, Fig. 7 illustrates how to initialize the synchronizing point for all federates.

Table 2. Time management of the four federates.

Federate	Time constrained	Time regulating	Time advance
Forest	Yes	Yes	Time stepped
River	Yes	Yes	Time stepped
WSN	Yes	Yes	Time stepped
Observer	Yes	Yes/No	Time stepped

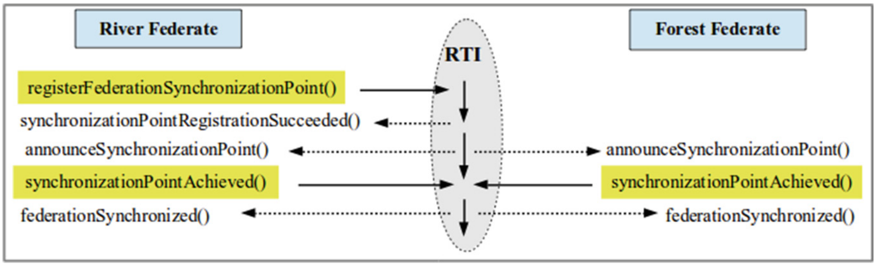

Fig. 7. Federate synchronization for the river and the forest federate.

Therefore, for the proposed federation, the time synchronization of the four federates can be obtained via the following steps.

1. All federates connect to the federation initialized earlier.
2. The federates are put into a common time progress by requesting synchronization point services, as demonstrated in Fig. 7.
3. The federates update the new states, and then send the updates to the central component if they are publishers.
4. Each federate sends an advance time request to the central component.
5. For each federate, it needs to wait until all data on the central component with the time stamps are less than or equal to its requested time are received by subscribers, then it is able to advance its logical time.

5 Experiment

5.1 Data Used

Data used in this study was taken from OpenStreetMap [16]. In which, we considered on a small area located in the Mekong Delta of Vietnam as a study region 4. Those data were used to generate cell system of river, forest, and WSN. The pattern neighborhood were 8, 4, and 4 neighbor, respectively. Since our current focus is the federation of mixed simulation, input data were randomly created for simulations.

5.2 FEMIS Tool

We have developed the FEMIS (FEderation of MIxed Simulation) tool by using C/C++ language. It enables to develop parallel simulations and federate them as distributed simulations from outputs of the PickCell tool [11]. The framework CERTI [17,18] was used in this project as role of RTI. Meanwhile, the X Window System [24] was employed to provide a GUI (Graphical User Interface) environment for displaying simulation results and interacting with users.

5.3 Scenario 1 - The Parallel Simulation of Pollution Diffusion in the River

We used the river model for pollution diffusion that presented in Sect. 3.2. Initially, two polluted points were randomly created in the river. After 4 steps the diffusion of pollution is shown in Fig. 8. The darker regions implied the larger density of pollution, and vice versa.

Fig. 8. The parallel simulation of diffusing pollution in a river. This is initialized with two polluted points (dark points).

5.4 Scenario 2 - The Interactions Between the Forest and the River Federate

Regarding to simulate the interactions of physical systems in CPSs. We lunched a federation of a mixed simulation with three federates: river, forest, and observer. In which, the two first ones run on the GPUs.

In this case, the river federate created and joined the federation on the RTI-CERTI. It waited for the forest federates to enter. The river federate sent a request to others to achieve a synchronization point in the federation. And then the synchronization point was achieved.

Figure 9 showed that the ashes (brown points) formed by the fire (red points) polluted the river from the step 4 as they spread close to the river. This also shows that models based on the CA can work together in the common time progress.

Fig. 9. The screen shoot was taken from the observer federate. It shows the data exchange between the river and forest federate in the federation. Two regions marked the small red circles represent the new polluted points created by the ashes, which are formed from the forest fire after 4 steps (Color figure online).

Fig. 10. The interoperability of four federates under the context of the mixed simulation: river, forest, WSN, and observer. The sensors changed to red color since the fire was detected close to them (Color figure online).

5.5 Scenario 3 - The Federation of Mixed Simulation with the Four Federates: River, Forest, WSN, and Observer

The models were presented in Sect. 3.6. The sensors nodes were represented by black points in the forest. They appeared with the sensing ranges (small circles) and communication ranges (large circles).

As the previous case, the four federates first need to achieved a synchronous point. At each step, these federates exchange data together via the RTI.

Figure 10 presents the results captured from the observer federate. In this scenarios, due to no fire close to the river, until step 4, there were no new polluted points created in the river. Meanwhile, since a sensor recognizes that the fire appeared within its sensing range (smaller circle), it will change its color, sensing and communication range (larger circle) to red color.

6 Conclusion

In context of modeling and simulating for cyber-physical systems, we have described a new approach on the federation for simulations. The models of physical systems are based on cellular automata. In this method, they must have at least two components: cell system and transition rules. The FEMIS tool has developed in order to simulate those models in parallel and perform the federations of the those simulations. The parallel computations on the GPU aim to reduce the simulating time for large and complex models. The experimental results were obtained by federating the three parallel simulations for forest fire spread, river pollution diffusion and wireless sensor network. By using federated simulations, the behavior of physical systems with their interactions could be observed in simulation progress.

References

1. Lee, E.A.: CPS foundations. In: Proceedings of the 47th Design Automation Conference (DAC), pp. 737–742. ACM (2010)
2. Wolfram, S.: Computation theory of cellular automata. Commun. Math. Phys. **96**(1), 15–57 (1984)
3. Hoekstra, A.G., Kroc, J., Sloot, P.M.A.: Simulating Complex Systems by Cellular Automata, chap. 1. Springer-Verlag, Berlin and Heidelberg (2010)
4. Wolfram, S.: Cellular automata: a model of complexity. Nature **31**, 419–424 (1984)
5. Riley, D., Eyisi, E., Bai, J., Koutsoukos, X., Xue, Y., Sztipanovits, J.: Networked control system wind tunnel (NCSWT)- an evaluation tool for networked multi-agent systems. In: The Fourth International Conference on Simulation Tools and Techniques (SIMUTools), Barcelona, Spain, p. 918 (2011)
6. Branicky, M.S., Liberatore, V., Al-Hammouri, A.T.: Co-simulation tools for networked control systems. In: Egerstedt, M., Mishra, B. (eds.) HSCC 2008. LNCS, vol. 4981, pp. 16–29. Springer, Heidelberg (2008)
7. Li, D., Li, X., Liu, X., Chen, Y.M., Li, S.Y., Liu, K., Qiao, J.G., Zheng, Y.Z., Zhang, Y.H., Lao, C.H.: GPU-CA model for large-scale land-use change simulation. Chin. Sci. Bull. **57**(19), 2442–2452 (2012). SP Science China Press

8. Liang, Q., Xia, Y., Du, J.: Parallel simulation based on GPU-acceleration. In: Xiao, T., Zhang, L., Fei, M. (eds.) AsiaSim 2012, Part II. CCIS, vol. 324, pp. 355–362. Springer, Heidelberg (2012)
9. Blecic, I., Cecchini, A., Trunfio, G.A.: Cellular automata simulation of urban dynamics through GPGPU. J. Supercomput. **6**(2), 614–629 (2013). Springer, US
10. Gulati, K., Khatri, S.P.: GPU Architecture and the CUDA Programming Model. Hardware Acceleration of EDA Algorithms, chap. 3. Springer, US (2010)
11. Pottier, B., Lucas, P.-Y.: Dynamic networks NetGen: objectives, installation, use, and programming. Université de Bretagne Occidentale (2014)
12. Oliveira, L.M.L., Rodrigues, J.J.P.C.: Wireless sensor networks: a survey on environmental monitoring. J. Commun. **6**(2), 143–151 (2011)
13. IEEE: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. IEEE Std 1516TM-2010, pp. 1–38 (2010)
14. IEEE: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification. IEEE Std 1516TM-2010, pp. 1–378 (2010)
15. IEEE: IEEE standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (omt) Specification. IEEE Std 1516.2TM-2010, pp. 1–110 (2010)
16. Mekong Delta Region, South of Vietnam. https://www.openstreetmap.org
17. Noulard, E., Rousselot, J.-Y., Siron, P.: CERTI, an open source RTI, why and how. Spring Simulation Interoperability Workshop (2009)
18. d'Ausbourg, B., Siron, P., Noulard, E.: Running real time distributed simulations under linux and CERTI. In: 2008 Euro Simulation Interoperability Workshop Proceedings, 08E-SIW-061 (2008)
19. Alvarado, J.R., Osuna, R.V., Tuokko, R.: Distributed simulation in manufacturing using high level architecture. In: Ratchev, S., Koelemeijer, S. (eds.) Micro-Assembly Technologies and Applications. International Federation for Information Processing, vol. 260, pp. 121–156. Springer, US (2008)
20. Lasnier, G., Cardoso, J., Siron, P., Pagetti, C., Derler, P.: Distributed simulation of heterogeneous and real-time systems. In: Proceedings of IEEE International Symposium on Distributed Simulation and Real-Time Applications, pp. 55–62 (2003)
21. Kyoung-Soo, W., Jong-Chan, K., Chang-Gun, L.: A novel simulation framework for supporting real-time cyber-physical interactions. In: The Fifth IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–3 (2011)
22. Wu, G., Wu, Y., Xu, J., Lin, J.: NCCPIS: a co-simulation tool for networked control and cyber-physical system evaluation. In: Park, J.J., Zomaya, A., Yeo, S.-S., Sahni, S. (eds.) NPC 2012. LNCS, vol. 7513, pp. 85–93. Springer, Heidelberg (2012)
23. ForwardSim Inc., Simulation and Technologies: HLA Toolbox for MATLAB - HLA Blockset for Simulink (2013). http://www.forwardsim.com
24. Scheifler, R.W., Gettys, J.: The X window system. ACM Trans. Graph. **5**(2), 79–109 (1986)
25. Zhou, F., Li, S., Hou, X.: Development method of simulation and test system for vehicle body CAN bus based on CANoe. In: The 7th World Congress on Intelligent Control and Automation (WCICA), pp. 7515–7519. IEEE (2008)