EV2019
ELECTRIC VEHICLES INTERNATIONAL CONFERENCE & SHOW
www.evshow.ro
Icpe Solar Park
București, România
3-4 OCTOBER 2019

Using Statistical Methods and Co-Simulation to Evaluate ADS-Equipped Vehicle Trustworthiness

Khalid HALBA
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
khalid.halba@nist.gov

Edward GRIFFOR
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
edward.griffor@nist.gov

Patrick KAMONGI
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
patrick.kamongi@nist.gov

Thomas ROTH
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
thomas.roth@nist.gov

Abstract — With the increasing interest in studying Automated Driving System (ADS)-equipped vehicles through simulation, there is a growing need for comprehensive and agile middleware to provide novel Virtual Analysis (VA) functions of ADS-equipped vehicles towards enabling a reliable representation for pre-deployment test. The National Institute of Standards and Technology (NIST) Universal Cyber-physical systems Environment for Federation (UCEF) is such a VA environment. It provides Application Programming Interfaces (APIs) capable of ensuring synchronized interactions across multiple simulation platforms such as LabVIEW, OMNeT++, Ricardo IGNITE, and Internet of Things (IoT) platforms. UCEF can aid engineers and researchers in understanding the impact of different constraints associated with complex cyber-physical systems (CPS). In this work UCEF is used to produce a simulated Operational Domain Design (ODD) for ADS-equipped vehicles where control (drive cycle/speed pattern), sensing (obstacle detection, traffic signs and lights), and threats (unusual signals, hacked sources) are represented as UCEF federates to simulate a drive cycle and to feed it to vehicle dynamics simulators (e.g. OpenModelica or Ricardo IGNITE) through the Functional Mock-up Interface (FMI). In this way we can subject the vehicle to a wide range of scenarios, collect data on the resulting interactions, and analyze those interactions using metrics to understand trustworthiness impact. Trustworthiness is defined here as in the NIST Framework for Cyber-Physical Systems, and is comprised of system reliability, resiliency, safety, security, and privacy. The goal of this work is to provide an example of an experimental design strategy using Fractional Factorial Design for statistically assessing the most important safety metrics in ADS-equipped vehicles.

Keywords — ADS-equipped vehicles, cyber-physical systems, trustworthiness, co-simulation

I. INTRODUCTION AND RELATED WORK

The current state of modeling and simulation for vehicles includes proprietary tools, such as IGNITE [1], CANOE [2], and CANALYZER [3], and open-source tools such as OpenModelica [4]. [1]

These tools include libraries to simulate vehicle dynamics such as steering, braking, energy conversion and transmission, and power management. Some of these tools allow for communication with external simulators through the FMI [5].

The ability to interface with external simulators widens the range of simulation scenarios that may include interactions with other components through control and sensing functions [6], [7].

In this work we use UCEF [8] to demonstrate that co-simulation can enable innovation in Automated Driving System (ADS)-equipped vehicle research while allowing specialized simulation platforms to independently develop and improve traditional vehicle dynamics simulation.

In Section II we review the testbed functional components, their roles in our co-simulation, and their implementation. In Section III we demonstrate the potential of UCEF to implement the functional components in a co-simulation. In Section IV we propose a strategy for assessing safety metrics by defining an ODD and defining input and output parameters of interest using the Fractional Factorial Experiment Design [9]. The conclusion summarizes our effort and highlights future work.

[1] CERTAIN COMMERCIAL PRODUCTS ARE IDENTIFIED IN THIS PAPER TO FOSTER UNDERSTANDING. SUCH IDENTIFICATION DOES NOT IMPLY RECOMMENDATION OR ENDORSEMENT BY THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, NOR DOES IT IMPLY THAT THE MATERIALS OR EQUIPMENT IDENTIFIED ARE NECESSARILY THE BEST AVAILABLE FOR THE PURPOSE.

978-1-7281-0791-2/19/$31.00 ©2019 IEEE

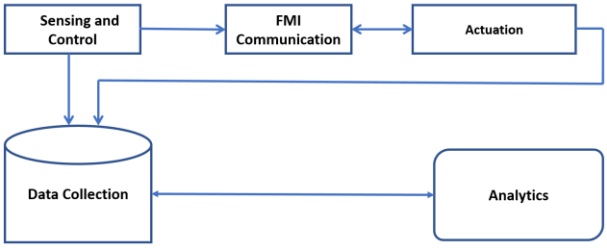

Fig. 1. ADS-Equipped Vehicles Testbed Functional Components

II. ADS-EQUIPPED VEHICLES TESTBED FUNCTIONAL COMPONENTS AND IMPLEMENTATION

In this section we describe the functional components of our ADS-equipped vehicle testbed and its implementation.

A. *Functional Components*

The ADS-equipped vehicle testbed is composed of the functional components described in Figure 1:

a) the Sensing and Control component responsible for receiving input data from the environment such as weather, road condition, traffic infrastructure, and events such as obstacle detection data. The control feature of this component reacts to the data received by the sensing feature and generates a corresponding drive cycle, i.e., an acceleration, and braking pattern. Decision support algorithms determine this behavior based on the given ODD;

b) the FMI Communication component plays a bridge role by transmitting the drive cycle produced from the sensing and control component to the actuation component;

c) the Actuation component receives the drive cycle produced by the sensing and control component then performs either acceleration or deceleration functions;

d) the Data Collection component collects data about the interactions between the sensing and control component and the actuation component for further processing and trustworthiness metrics assessment;

e) the Analytics component analyzes the data collected by the data collection component and runs statistical techniques and machine learning algorithms to assess trustworthiness metrics of the ADS-equipped vehicles testbed experiments.

B. *Implementation*

Now we describe the implemented features of the functional components. The implementation uses a National Institute of Standards and Technology (NIST) co-simulation platform called UCEF that is built using an Ubuntu virtual machine. UCEF relies on the concept of multiple federates that interact using a publish-subscribe message pattern to simulate different functions of a CPS. In the present work, UCEF is used to simulate ADS-equipped vehicle autonomy or decision-making functions focused on acceleration and deceleration.

a) the sensing and control functional component is implemented as two federates: i) a Sensing Federate that simulates a binary obstacle detection notification (vehicle or other obstacle) at a specific simulation time and sends that notification over the HLA bus to the Control Federate.
ii) a Control Federate that processes the notification provided by the Sensing Federate and generates corresponding acceleration, and deceleration requests. The Control Federate also implements a User Datagram Protocol (UDP) server that sends drive cycle data to the actuation module over the communication component. Code for the Sensing and Control Federates is available in GitHub [10];

b) the communication component is responsible for communication between the UCEF federates and the Actuation component, built as Functional Mock-Up Unit (FMU). The FMU implements a UDP client that listens on the incoming drive cycle data from the UCEF UDP server implemented in the Control Federate and feeds that information to the Actuation ccomponent. We have built this FMU based on the Q.Tronic FMU Software Development Kit (SDK) and shared its source code in GitHub [11];

c) the Actuation component implementation can be realized using different simulators, including IGNITE, Modelica, and MATLAB that provide both libraries for running traditional vehicle functions and an FMI master algorithm that enables them to interact with other simulators. In this work we use Ricardo IGNITE, a vehicle simulation platform that can be installed on any Microsoft Windows computer, for modeling and simulation of electric and fuel-based vehicle models. IGNITE provides a built-in FMI Master algorithm, and models legacy vehicle functions based on the UCEF generated drive cycle;

d) the Data Collection component is implemented using a MySQL database UCEF federate. Each interaction, or message exchange, is time-stamped and represented in this database as a record that comprises data encoding, the sender, the receiver, the time of transmission, the time of reception, a description of the carried signal, and the payload of the message;

e) the Analytics component will be implemented in a future work as a federate to perform two functions: i) an assessment function using machine learning algorithms and Fractional Factorial Experiment Design scripts. In this work the Fractional Factorial statistical analysis is done using formulas implemented in a standalone excel data form; ii) a Vert.X-backend [12] / Vue.js-frontend [13] microservice for data visualization was implemented and tested for visualizing post-experiment data collected by the Data Collection component.

III. CO-SIMULATION OF THE FUNCTIONAL COMPONENTS

UCEF enables the co-simulation of a wide variety of CPS using the IEEE High Level Architecture (HLA), including systems at scale such as power grids [14]. Figure 2 shows a federation that will simulate message exchange for the sensing, control, communication, and analytics components using simulated J1939 CAN frames.

These federates were modeled in UCEF using the Web-based Generic Modeling Environment (WebGME), which includes JavaScript extensions to convert the models into Java code.

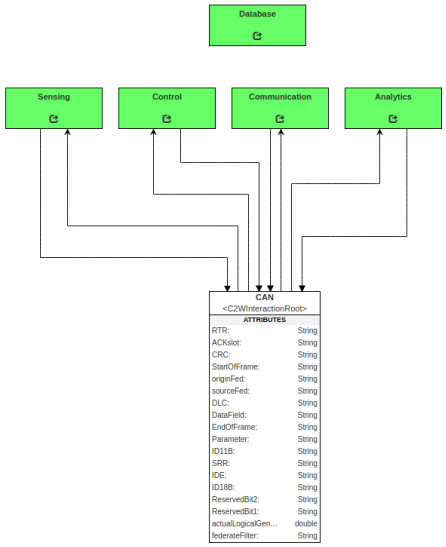

Fig. 2. ADS-equipped vehicles federation where the functional components exchange simulated CAN messages that are logged in a database

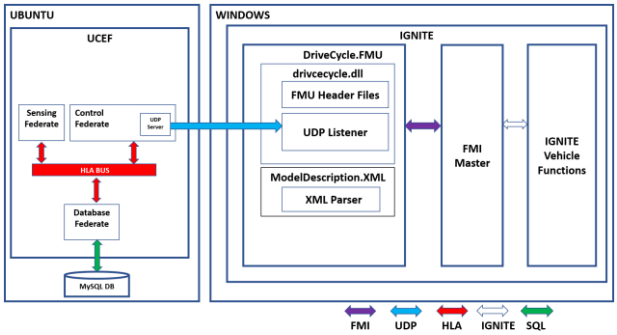

Fig. 3. UCEF implementation of ADS specific functions for an ADS-equipped vehicle testbed

WebGME was used to generate Java code for the model, and that code was implemented with the desired behavior for each federate. Java federates were modeled to represent the features of the functional components. The sensing federate sends obstacle detection data to the control federate over the HLA bus. The control federate adjusts the drive cycle based on the sensing federate input and generates a new drive cycle that will be fed to the actuation module through the FMU. The database federate collects and stores the simulated message exchange between the different federates.

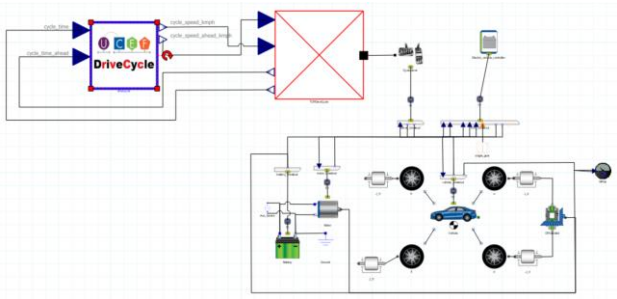

Fig. 4. Custom UCEF drive cycle loaded into IGNITE as an FMU that provides acceleration and braking signals to a simulated Electric SUV

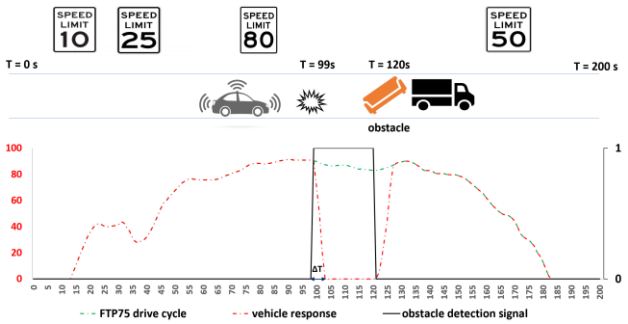

Fig. 5. ODD Overview with the simulated 200s of the FTP 75 drive cycle portion [150-350] fed to the Actuation Module. Stopping Time ΔT is also illustrated

Other legacy vehicle simulators, such as IGNITE, have a mature implementation of traditional vehicle functions. As such it makes more sense for an ADS-equipped vehicle testbed to focus on extending the capabilities of these simulators by enabling interactions between them and environments like UCEF designed to simulate ADS or non-traditional vehicle functions. Figure 3 shows the integration of IGNITE with the sensing and control federation to produce a co-simulation of the of ADS-equipped vehicle testbed.

We have simulated an exchange of drive cycle data between UCEF and IGNITE. UCEF generates a portion of FTP75 [15] drive cycle data stream. The FMU implements a UDP listener. Figure 4 shows the integration of this external drive cycle data into the IGNITE simulation environment.

IV. Safety Study

In this section we present a scenario that demonstrates the potential of our ADS-equipped vehicles testbed in assessing trustworthiness of ADS-equipped vehicles. According to the National Highway Transportation Safety Administration (NHTSA) [16] and Waymo's Safety Report [17], an ODD refers to the conditions under which a self-driving system can safely operate. The domain includes geographies, roadway types, speed range, weather, time of day, and state and local traffic laws and regulations. We describe the scenario and the ODD in which our simulated ADS-equipped vehicles safety will be assessed.

A. ODD Description

The vehicle in this ODD is moving along a single lane road and performing a subset of the Federal Test Procedure 75 (FTP75) drive cycle generated by UCEF as shown in Figure 5.

We focus on a subset of the FTP75 drive cycle [150 s – 350 s] where vehicle speed falls from 90 km/h to 0 km/h when an obstacle is detected.

We represent this subset on a [0s- 200s] time scale.

At t = T_control, the sensing component in UCEF triggers an alert that indicates an obstacle detection. The vehicle completely stops at t = T_actuation. For a given ODD there is an interval of time ΔT in which the vehicle performs full braking and completely stops when an obstacle is detected. ΔT can be expressed as follows:

$$\Delta T = T_{actuation} - T\_control \quad (1)$$

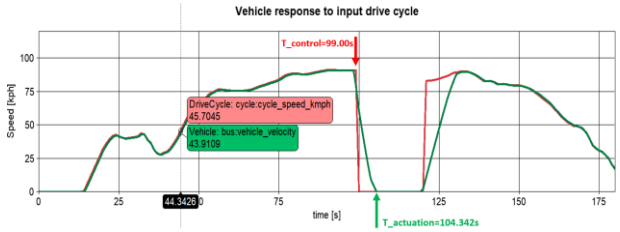

Fig. 6. IGNITE RPOST : Vehicle response to the input drive cycle: the vehicle completely stops 5.342s after the intended theoretical drive cycle control input. T_control and T_actuation are illustrated

The control component in UCEF generates a drive cycle based on the obstacle detection information it receives.

The Actuation Component (IGNITE-based) receives this drive cycle and simulates the vehicle response. We can compare both input drive cycle and output vehicle response in order to judge whether the vehicle has successfully performed the desired braking time. We assess ΔT once an obstacle is detected. Figure 6 shows vehicle velocity in response to input drive cycle generated by UCEF.

We have calculated for the default IGNITE parameters a Stopping Time equals to ΔT=5.342 s between the drive cycle braking control message (T_control) and vehicle response (T_actuation). ΔT is assessed within the described ODD to determine whether it falls within safe boundaries, i.e., stopping before hitting the detected obstacle. A safe scenario verifies the following property:

$$0 < \Delta T < \Delta T\_Limit \quad (2)$$

ΔT_Limit is the time the vehicle is predicted to collide, and the Stopping Time will be beyond safe boundaries. ΔT depends not only on the drive cycle, which is the result of decision support algorithms implemented in UCEF's control component, but it is also influenced by information collected by UCEF's sensing component and IGNITE's vehicle model parameters (vehicle mass for example); changing a single parameter may have a significant impact. Figure 7 is a list of vehicle parameters that could influence ΔT.

	Name	Type	Units	Group	ommer	Case 1	Case 2
Case Title						Case 1	Case 2
Enabled						☑	☑
soc init	soc_init	Real			Initial ...	0.3	0.3
A	A	Real	m^2			2	2
Vnom	Vnom	Real	V			375	375
Coeff RR	Coeff_RR	Real				.08	.08
T radius	T_radius	Real	m			.3814	.3814
Pm	Pm	Real	W			235000	235000
J wheel	J_wheel	Real	kg.m^2			2	2
Mass	Mass	Real	kg			2000	5000
C bat	C_bat	Real	A.hr			136	136
Cd	Cd	Real				0.24	0.24
Paux	Paux	Real	W			200	200
FDR	FDR	Real				9.73	9.73
Tm	Tm	Real	N.m			440	440
Trac Eff	Trac_Eff	Real				.85	.85

Fig. 7. A subset of IGNITE's vehicle model parameters. Different ODDs can be defined to assess a trustworthiness metric such as safety. We can define multiple cases each representing a run sequence, where a single or multiple parameters are altered. This figure shows two cases where vehicle mass was the altered parameter

B. *Trustworthiness metric assessment using Fractional Factorial Experiement Design*

One of the goals of the UCEF-based ADS-equipped vehicle testbed is to study ADS-equipped vehicles trustworthiness metrics. As in the NIST CPS Framework [18], trustworthiness comprises safety, security, privacy, resilience and reliability. This work has focused on illustrating the potential of co-simulation for assessing ADS-equipped vehicle safety measurement strategies. We have used the Fractional Factorial Experiment Design methodology to run multiple experiments while varying parameters of interests.

Many factors can impact the results of the experiments. Going forward we aim to determine the relative importance of these factors. The sheer number of these factors can present challenges. In this section we use the characteristics of the UCEF-based ADS-equipped vehicle testbed to study these parameters and to design our experiments, using the statistical Design of Experiments (DEX) methodology. Four factors were identified for assessing their influence on the output parameter ΔT (the time required for vehicle speed to go from 90 km/h to 0 km/h). The identified parameters are vehicle mass, tire rolling resistance coefficient, vehicle aerodynamics resistance surface, and wind speed.

The experiment design we adopt is a 2^k design that takes into consideration 2 levels per factor. This approach is best suited for exploratory experimentation purposes. The outcome of the 2^k factorial experiment helps in identifying the relative importance of factors and offers rapid insight into the interaction effects (Table I). The Fractional design is expressed as follows:

Design expression: $L^{(K-P)}$, L: number of levels of each factor investigated, K: number of factors investigated, P: size of the fraction of the full factorial to be eliminated, L^P: fraction of the full design L^K, M: number of experiments

TABLE I. LIST OF INPUT PARAMETERS CONSIDERED IN THE TWO-LEVEL FRACTIONAL FACTORIAL DESIGN, AND THE VALUES CHOSEN FOR THE TWO LEVELS FOR EACH VARIABLE.

Variable	Description	Low (-1)	High (+1)
X1	Vehicle Mass (Mass)	2000 kg	5000 kg
X2	Tire Rolling Resistance Coefficient	0.01	0.08
X3	Vehicle Aerodynamics Resistance Surface (Area)	2 m^2	5 m^2
X4	Wind Speed	0 m/s	20 m/s

TABLE II. ΔT IS THE OUTPUT PARAMETER (RESPONSE) MEASURED BASED ON VARIATIONS OF VEHICLE PROPERTIES AND WEATHER CONDITIONS.

Output parameter Y1 = ΔT = F (X1, X2, X3, X4)
Y1: Stopping Time

For a system with K=4 factors, L=2 levels of each factor, and P=1, the number of experiments in a fractional 2^{K-P} will be $M = 2^{4-1} = 8$ experiments. The half factorial design would reduce M, the number of experiments by half. Table III describes the different sets of experiments with the fractional 2^{K-P} experiment design without replication. We have explained in this section how the experiment design using statistical fractional factorial techniques can be used to discern which factors yields the most significant response on the output parameters of interest and as a result assess safety related to the specified ODD parameters.

By assessing which vehicle characteristics and road conditions have the most influence on ΔT outcomes, we can understand which factors have the most significant impact on the stopping time.

TABLE III. FRACTIONAL FACTORIAL 2K-P (K=4, P=1, M= 8)

X1	X2	X3	X4	RunSeq
-1	-1	-1	-1	1
1	-1	-1	1	2
-1	1	-1	1	3
1	1	-1	-1	4
-1	-1	1	1	5
1	-1	1	-1	6
-1	1	1	-1	7
1	1	1	1	8

The DEX mean plot [19] is appropriate for analyzing data from a designated experiment, with respect to important factors, where the factors are at two or more levels. The plot shows mean values for two or more levels of each factor plotted by factor. The mean values of a single factor are connected by a straight line. For the given factor levels results shows that "Vehicle Mass" is by far the most important factor in influencing ΔT. "Tire Rolling Resistance" plays the next most critical role. The experimental parameters tested for "Wind speed" and "Vehicle Aerodynamics Resistance Surface" did not indicate a statistically significant effect on stopping time. The average Stopping Time ΔTavg across all experiments is equal to 5.44s. If ΔT_Limit = ΔTavg, we can evaluate safe braking boundaries just by reviewing the DEX mean plot (Figure 8). Raw results can be found in [20].

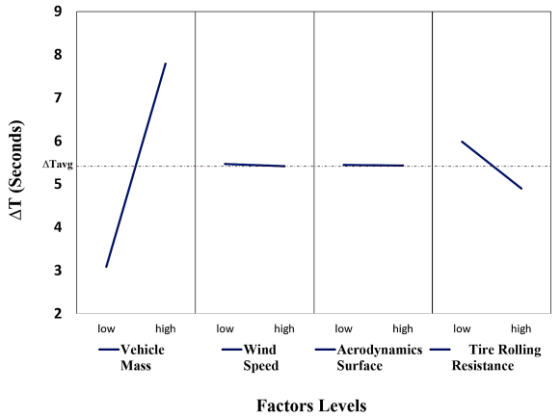

Fig. 8. The DEX mean plot: ΔT respose to 4 factors of interest

V. CONCLUSIONS

In this work we have described the NIST UCEF-based-ADS-equipped vehicles testbed and its potential for co-simulation of ADS-equipped vehicle applications. We have described implementation elements of the testbed and considered an ODD and a metric strategy for assessing safety, one component of trustworthiness. Finally, we have described an experiment design methodology that can be used to assess trustworthiness metrics.

Going forward, we intend to perform additional experiments to further study the safety and other trustworthiness metrics for ADS-equipped vehicles.

We will also design deep learning architectures to explore trustworthiness assessment techniques. For instance, we plan to leverage UCEF to study the safety of ADS functions and synthesize ground truths for training deep learning models. These deep learning models will be used for two purposes: to approximate the safety metric and forecast safety violations. Since many ADS functions are safety critical, we will design learning architectures that are interpretable and explainable.

REFERENCES

[1] Ricardo IGNITE Vehicle Simulator, https://software.ricardo.com/products/ignite.

[2] Vector CANoe, https://www.vector.com/int/en/products/products-a-z/software/canoe/.

[3] Vector CANalyzer, https://www.vector.com/int/en/products/products-a-z/software/canalyzer/.

[4] Feng, S., He, J. and Zhang, L., 2013. Modeling vehicle dynamics based on modelica. International Journal of Multimedia and Ubiquitous Engineering, 8(3), pp.307-318.

[5] Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Elmqvist, H., Junghanns, A., Mauß, J., Monteiro, M., Neidhold, T., Neumerkel, D. and Olsson, H., 2011, June. The functional mockup interface for tool independent exchange of simulation models. In Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany (No. 063, pp. 105-114). Linköping University Electronic Press.

[6] Palmieri, M., Bernardeschi, C. and Masci, P., 2017, September. Co-simulation of semi-autonomous systems: the line follower robot case study. In International Conference on Software Engineering and Formal Methods (pp. 423-437). Springer, Cham.

[7] Bünte, T., Ho, L.M., Satzger, C. and Brembeck, J., 2014. Central vehicle dynamics control of the robotic research platform robomobil. ATZelektronik worldwide, 9(3), pp.58-64.

[8] Burns, M., Roth, T., Griffor, E., Boynton, P., Sztipanovits, J. and Neema, H., 2018, January. Universal CPS Environment for Federation (UCEF). In 2018 Winter Simulation Innovation Workshop.

[9] Anderson, V.L. and McLean, R.A., 2018. Design of experiments: a realistic approach.

[10] Khalid HALBA. Sensing and Control component : Sensing, Control, and Communication Federates: https://github.com/KhalidHALBA-GR-NIST/UCEF-IGNITE.

[11] Khalid HALBA. Communication Module implementation: DriveCycle.FMU & an FTP75 DrveCycle Generator: https://github.com/KhalidHALBA-GR-NIST/FMU-IGNITE.

[12] Vertx Microservices Toolkit. https://vertx.io/.

[13] VueJS, https://vuejs.org/.

[14] Roth, T., Song, E., Burns, M., Neema, H., Emfinger, W. and Sztipanovits, J., 2017, June. Cyber-physical system development environment for energy applications. In ASME 2017 11th International Conference on Energy Sustainability.

[15] FTP75 Drive Cycle, https://www.dieselnet.com/standards/cycles/ftp75.php.

[16] Automated Driving Systems 2.0, A Vision for Safety, NHTSA, https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf.

[17] Waymo Safety Report, On the Road to Fully Self-Driving, Waymo. https://storage.googleapis.com/sdc-prod/v1/safety-report/Safety%20Report%202018.pdf.

[18] Griffor, E.R., Greer, C., Wollman, D.A. and Burns, M.J., 2017. Framework for cyber-physical systems: Volume 1, overview (Special Publication (NIST SP)-1500-201).

[19] The DOE Mean Plot. Exploratory Data Analysis. http://www.itl.nist.gov/div898/handbook/eda/section3/dexmeanp.htm

[20] Raw experiment results: https://github.com/KhalidHALBA-GR-NIST/FMU-IGNITE/blob/master/dex%20mean%20plot.xlsx