

Received October 6, 2017, accepted December 16, 2017, date of publication January 1, 2018, date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2789020

# Shared and Distributed X-in-the-Loop Tests for Automotive Systems: Feasibility Study

VIKTOR SCHREIBER<sup>1</sup>, VALENTIN IVANOV<sup>1</sup>, (Senior Member, IEEE),  
KLAUS AUGSBURG<sup>1</sup>, MATTI NOACK<sup>1</sup>, BARYS SHYROKAU<sup>2</sup>,  
CORINA SANDU<sup>3</sup>, AND PIETER SCHALK ELS<sup>4</sup>

<sup>1</sup>Technische Universität Ilmenau, 98693 Ilmenau, Germany

<sup>2</sup>Technische Universitat Delft, 2628 CD Delft, The Netherlands

<sup>3</sup>Virginia Tech, Blacksburg, VA 24061, USA

<sup>4</sup>University of Pretoria, Pretoria 0002, South Africa

Corresponding author: Valentin Ivanov (valentin.ivanov@tu-ilmenau.de)

This work was supported by the European Unions Horizon 2020 Research and Innovation Programme through the Marie Skłodowska-Curie actions under Grant 645736.

**ABSTRACT** X-in-the-loop (XIL) technologies are receiving increased attention in modern automotive development processes. In particular, collaborative experiments, such as XIL tools, have efficient applications in the design of multi-actuated, electric, and automated vehicles. The presented paper introduces results of such a collaborative study for XIL, which focused on the feasibility of coordinated real-time simulations for the control of vehicle dynamics systems. The outcomes are based on extensive co-simulation tests performed using remote connections among different geographical locations; the connections were between Germany, on one side, and USA, South Africa, and The Netherlands, from the other side. The performed study allowed formulating requirements for further shared and distributed XIL-experiments for functional validation of automotive control systems.

**INDEX TERMS** X-in-the-loop, co-simulation, vehicle models, automotive control, remote tests.

## I. INTRODUCTION

The increased complexity of modern vehicle systems causes many challenges in the development and design processes associated with such systems. One such challenge concerns the development of new multi-actuated vehicle concepts for electric mobility and automated driving. For example, to reach sufficient accident-free functionality of an autonomous vehicle, several billion kilometers of virtual proving grounds and road tests are required. Comparable efforts in experiments can also take place for state-of-the-art conventional cars equipped with dozens of electronic control units (ECU), on-board processors, and sensors and actuators to guarantee fail-safe operation during the whole life cycle of the vehicle. These factors are motivating researchers and developers to seek new, powerful and efficient procedures allowing rapid prototyping, design, and validation with reduced time efforts.

Traditionally, the development of vehicles and automotive systems is successively implementing software-in-the-loop (SIL), model-in-the-loop (MIL), and hardware-in-the-loop (HIL) tools, which together can be concisely referred as

“X-in-the-loop” or “XIL”. These established XIL technologies are currently being advanced with the development of new classes of design concepts. For example, the work [1] proposed an extension of the XIL framework through a connection with the Integrated Product Development Model and Knowledge Management Systems, widely used for components of industrial design. Another variant of an XIL tool was introduced in [2], where the so-called concept of “test-rig-in-the-loop” (TRIL) is described. The TRIL technique aims at real-time integration of two or more test rigs from different domains, for example, dynamometers and HIL test setup, and continues the ideas of Internet-based hardware experiments presented in [3] and [4].

Fig. 1 shows a generic example of an XIL architecture, as proposed by Ivanov *et al.* [5]. This example includes different elements and was tailored for the development of on-board vehicle dynamics control systems. In the proposed architecture, MIL tools are used for full system simulation in a virtual environment. The SIL technique is also applied for investigations on functional reliability of embedded software applications. TRIL is represented by three test setup variants:

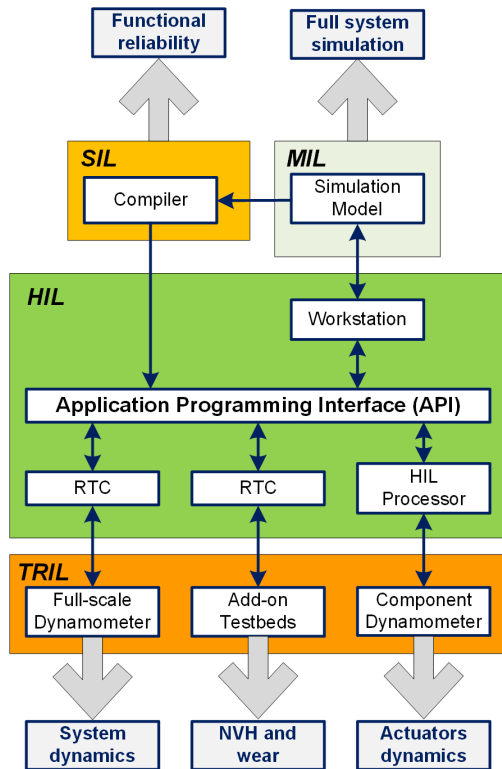


FIGURE 1. Example of XIL architecture.

full scale dynamometer to emulate the vehicle dynamics, component dynamometer to include actuator dynamics (e.g., brake system) into the control circuit, and add-on testbeds allowing the observation of noise-vibration-harshness (NVH) and wear processes in the process of vehicle dynamics control. This XIL architecture can be further extended with additional virtual software environments and test rigs depending on research and development tasks. However, it is difficult to collect all components of such sophisticated architecture, as shown in Fig. 1, within one host. Thus, the next logical step can be to consider sharing and distributing the development tasks and the corresponding design and testing components between different hosts or partners. Hence, this step leads to the demand on collaborative experimental environments.

The idea of collaborative, geographically distributed design processes, has arisen about two decades ago [6] and is mainly implemented in applications requiring co-simulation procedures. The co-simulation can be considered, in general, as a connecting link between software and system engineering [7] and is being used in different domains for the development of complex systems. For instance, some application cases demonstrating the efficiency of co-simulation in the design processes are: Engine-in-the-loop environment [8], traffic network control [9], renewable energy production and smart grids [10]–[12]. However, as it was demonstrated in [13], a careful selection of the co-simulation method is required to solve the trade-off between performance and accuracy. This task reaches a higher level of complexity

when the co-simulation is used with real-time integration of HIL or TRIL components, and in the case of geographical distribution. Various studies proposed different methods, such as the virtual synchronization technique [14], Functional Mock-up Interface [15], [16], but there are no universal approaches in this regard. One of the many reasons for this situation is that the efficiency of the different methods depends strongly on the communication resources used in co-simulation or the XIL architecture.

Different communication versions are being used in XIL tasks, but for remote or geographically distributed design processes the connection utilizing User Datagram Protocol (UDP) remains a more acceptable solution, as demonstrated, for example, in [17]. As for real-time UDP-based applications, known problems related to the data losses and communication dynamics are still subjected to intensive investigations [18]–[20]. Nevertheless, UDP solutions are being considered as the basis for further Internet protocols, especially for the Internet of Things [21] (IoT).

The current study, which will be further discussed in this paper, addresses the problems mentioned in the previous paragraphs. The main target of this study is to perform a feasibility check for sufficient real-time connectivity in the case of remote and shared experiments, where various parts of the tests have to be performed at different geographical hosts. Referring to Fig. 1, these tests relate to the real-time MIL technique, as well as to the emulation of the TRIL approach, where hardware testbeds are replaced with software simulators. Section II describes relevant targets of real-time capability and details of communication protocols and procedures of plausibility check. Section III introduces the test setup organized between the hosts in Germany, the Netherlands, South Africa, and USA. Section IV presents the results of remote and shared tests with the corresponding analysis.

## II. SPECIFICATION OF REAL-TIME DISTRIBUTED TESTS

### A. CHARACTERIZATION OF REAL-TIME CAPABILITY

Accordingly to DIN 44300 specification, a Real-Time Operating System is an operating system, which includes programs (processes, tasks) ready and intended to serve data within a fixed time period. These tasks can be processed either in a scheduled time or randomly (event triggered). However, a Real-Time Operating System should be considered as a part of any real-time control system having its own tasks and resources. From the viewpoint of the control architecture, User Interfaces and RT-system are usually strictly separated.

The typical requirements for RT systems are characterized as follows:

#### 1) DETERMINISM

In contrast to sequences, simultaneous or quasi-simultaneous operations (see Table 2) are not predictable. A Real-Time System is deterministic if all possible states and any amount of input information have an unambiguously amount of

TABLE 1. Real-time level.

level	description
hard RT	The RT system has to ensure that a task is done until a deadline. Missing the deadline causes a system failure, e.g., ABS.
firm RT	Firm RT system tolerates infrequent missing of deadlines while processing. However, exceeding the deadline criteria reduces the quality of service. The outcome after missing a deadline can't be used, e.g., multimedia.
soft RT	Soft RT systems desires deadlines. For a certain threshold it is allowed to miss a deadline. The outcome after missing a deadline is usable, but reduces the quality of service, e.g., air conditioner.
near RT	The term near RT is not well defined in publications, literature, or standards. The authors propose to define near RT as a deterministic delay of process outcomes, e.g., telecommunication. This delay does not influence the quality of service significant.
non RT	Non RT processes do not require an immediately execution, e.g., simulation of Finite Elements.

TABLE 2. Types of simultaneous processing.

processing	description
simultaneous	Each tasks owns a processor.
quasi-simultaneous	One processor or multiple processors handle multiple tasks.

output informations and an unambiguously amount of next states.

An essential requirement for a deterministic operation is a finite amount of system states. Hence, a deterministic system ensures an appropriate response within a scheduled time slot for any conditions and any time. The response time for all output informations must be known.

2) PROMPTNESS

Real-time systems requires processing of tasks just in time. Therefore, the real-time hardware shall have appropriate resources due to performance reasons. However, deadlines for task processing are strongly dependent on the application scenario. Table 1 shows the classification by RT level.

3) SIMULTANEITY

Real-time systems must respond rapidly in technical processes, which are running at the same time (e.g., controlling applications, measuring). Hence, all tasks should be handled simultaneously. Due to cost reasons, common real-time systems run quasi-simultaneously to reduce the effort. The difference between simultaneously and quasi-simultaneously processing is shown in Table 2.

4) JUST-IN-TIME RESPONSE TO SPONTANEOUS EVENTS

Basically, there are two procedures for handling spontaneous events: polling and interrupts.

Polling is a cyclic request of all input changes. Typically, polling is using “while” loops until a response on polling occurs or the “while” loop is terminated by a condition (limited loop number). Polling is not robust. A limitation of loop numbers is required to avoid infinite loops.

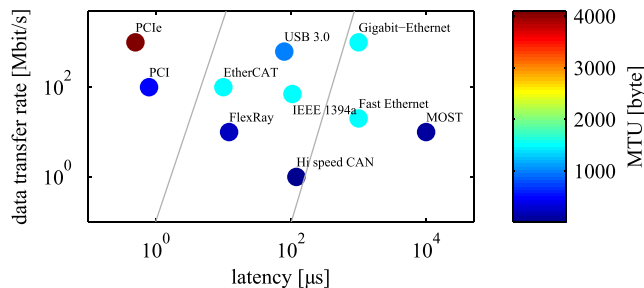


FIGURE 2. Generalized benchmarking of communication protocols.

An interrupt is an unpredictable temporary interruption of the program without a context to the executed operations. They occur when handling external events or significant condition changes. Important tasks for handling the event are combined in one task. Interrupts occur asynchronously with the task execution and require additional hardware and priorities for handling.

5) RELIABILITY

To enhance the reliability of real-time systems, critical hardware components are configured with redundancy. This ensures safe operation even if the hardware fails or is damaged. Therefore, the Real-Time Operating System should support failure detection, troubleshooting and it should be able to return to normal operation after failure.

B. MOTIVATION FOR UDP

In the field of automation technologies and automotive applications many communication protocols have been established, as it can be seen in Fig. 2. However, they are all designed for specific applications (e.g., TTEthernet [22]) and cannot offer the flexibility like the User Datagram Protocol (UDP). By default, the UDP protocol is not reliable or capable to real-time and it requires an adequate implementation by the user. On the other side, the UDP protocol has other advantages, which are discussed below.

1) HIGH FLEXIBILITY

UDP is based on the IP/Ethernet standard, which is commonly used for communication. Hence, this technology can be found in any facility, computer devices, or software tools. Moreover, the API for UDP communication are non-proprietary and are available for all existing operating systems. This enables a high flexibility for shared or distributed XIL tests without high effort.

2) BIG MAXIMUM TRANSMISSION UNIT (MTU)

According to the standard RFC 768 the UDP protocol provides a maximum MTU size of 1472 bytes without segmentation of the datagram. The usual protocols on the automotive field, such as CAN, MOST, or FlexRay, have only an MTU size of 8, 80 or 256 bytes. Thus, the UDP payload size is suitable for a wider range of applications

**TABLE 3.** UDP protocol in OSI layers representation.

Layer	PDU	Protocol	Function
7	Application	Data	SDU interface for high level application (e.g. API for RT systems)
6	Presentation	Data	SDU translation of data for the application (encoding, data compression, encryption)
5	Session	Data	SDU managing the communication session and exchange of information
4	Transport	Datagram	UDP transmission (segmentation, acknowledgement, multiplexing) of Service Data Units (SDU)
3	Network	Packet	IP managing a network (addressing, routing, control) and relaying datagrams
2	Data link	Frame	Ethernet transmission of data frames from physical device
1	Physical	Bit stream	Ethernet transmission and reception of raw bit stream over physical device

and does not require unnecessary segmentation of messages which exceed the specific MTU size. In addition, the MTU size enables an implementation of high level services or protocols.

### 3) HIGH PERFORMANCE COMMUNICATION

The performance of the UDP protocol will be discussed by means of the data transfer rate and latency. The typical Ethernet devices supports data transfer rates of 10, 100 and 1000 Mbit/s. This allows a cyclic point-to-point communication time of significantly less than 1 ms. Of course, the latency increases due to the size of a communication network and the physical distance between hosts. Nevertheless, a cyclic communication time of 1 ms ensures reliable XIL testing of mechatronic or automotive systems.

### 4) LOW PROTOCOL OVERHEAD

UDP is a message oriented communication protocol. It offers a suitable real-time characteristic in comparison to the Transmission Control Protocol (TCP). Because of its connectionless mechanism, UDP is not using the Three-Way-Handshake like TCP for managing retransmissions, error correction, or checking an appropriate packet sequence. Due to the demand for the latest information, mechanisms such as the Three-Way-Handshake are not required for real-time communication. To enhance the reliability of UDP, it is desirable to enhance the real-time system with mechanisms for detecting and compensating the loss of messages.

### 5) NETWORKING OVER LONG DISTANCES

The interconnection of computers by the global internet system is state-of-the-art. It makes possible to communicate worldwide via TCP or UDP protocols. Therefore the UDP/IP protocol provides services like routing and addressing. This makes it feasible to share distributed RT systems via UDP communication.

## C. METHOD FOR RT PLAUSIBILITY CHECK

According to the described characterization of RT systems, the capability for RT has been investigated in the present study. In particular, criteria such as determinism, promptness, and reliability were checked. The following indicators have been used:

### 1) TIME DELAY

The time delay represents the promptness of the UDP communication between hosts. In order to consider the time delay, a time stamp with a high accuracy of  $\mu s$  was exchanged. Therefore, the host transmits the time stamp  $t_{stamp}$  at a certain time  $t_{Tx}$  to a receiver. The receiver host sends this time stamp back to the sender. The passing time between sending at  $t_{Tx}$  and receiving at  $t_{Rx}$  is defined as time delay, according to Eq. 1. By using the time delay, the actuality of data can be determined.

$$t_{delay} = t_{stamp}(t_{Rx}) - t_{stamp}(t_{Tx}) \quad (1)$$

### 2) JITTER

Jitter is the deviation from a deterministic periodicity. For clock-based applications, it is called timing jitter. Jitter is a significant factor for communication and RT systems. Commonly, jitter is computed as the root-mean-square (RMS), peak-to-peak displacement, or spectral density. In this paper, the jitter effect will be indicated by the standard deviation  $\sigma$  according to Eq. 2 and Eq. 3.

$$\mu = \sqrt{\frac{1}{N} \sum_{k=1}^N t} \quad (2)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{k=1}^N |t - \mu|^2} \quad (3)$$

### 3) KERNEL DENSITY

To identify the determinism of the communication, the authors recommend using statistical methods. Thus, the kernel density estimator (KDE) was used to calculate the probability density function (PDF) of the time delay. This non-parametric representation of the PDF is able to consider uncertainties better than methods based on normal distribution. Specifically, an estimator of the form in Eq. 4 is selected, where the bandwidth  $H$  controls the weighting window size around a certain value  $t$ . The function  $KDF$  is called kernel, and it controls the weight of the neighborhood around each value  $t$  based on their proximity. An typical example of a kernel function is the Gaussian density function according to Eq. 5.

$$PDF(t|N, H) = \frac{1}{NH} \sum_{n=0}^N KDF(t|\mu, \sigma) \cdot \left(\frac{t - t_n}{H}\right) \quad (4)$$

$$KDF(t|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \quad (5)$$

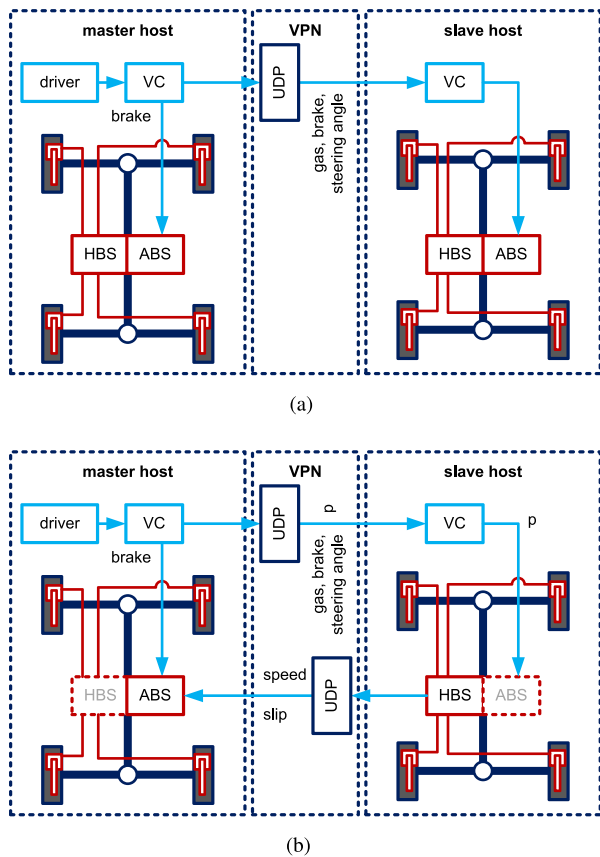


FIGURE 3. Setup types. (a) Open-loop. (b) Closed-loop.

4) MESSAGE LOSS

To execute vehicle testing with shared and distributed XIL technologies, a reliable communication is required. In this regard, the loss of messages indicates the reliability. The loss in receiving and transmitting messages is defined in Eq. 6. A loss of 0% represents a reliable communication. The bigger the loss the less the reliability of the communication.

$$RxLoss = \frac{\# \text{ received messages}}{\# \text{ requested messages}} \cdot 100 \quad (6)$$

III. TEST SETUP

To investigate the XIL technologies for shared and distributed testing, the setup in Fig. 3a and 3b was considered. In accordance with these schemes, two types of hosts and two types of loops have taken place. These setups will demonstrate the impact of the remote coupled sub-systems for an example of braking systems. The hosts are represented by the RT-capable full-vehicle simulations running on the computers with network devices, in which the simulation and the communication have been cyclically executed with a fixed-step size of 1 ms.

For the open-loop testing setup, the master host controls the slave host remotely. In this context, the slave host does not hold a virtual driver. Hence, the master and slave host is using the same inputs. However, all the hosts have their own anti-lock

braking system (ABS), hydraulic braking system (HBS), and vehicle controller (VC); respectively, the vehicle parametrization is identical.

For the closed-loop testing setup, the master host is controlling the slave host, too. In contrast with the open-loop testing, the master and the slave are sharing the hydraulic braking system (HBS) and the anti-lock braking system (ABS). More precisely, the master vehicle is using the responses of the slave HBS to perform the master ABS control. This exchange of states and data is executed in a closed loop. For this reason, the delay of the communication or RT system will influence the response of the system.

A. DESCRIPTION OF TEST CASES

The impact of shared and distributed XIL testing is demonstrated in this study by examples of ABS scenarios. Typically, ABS operates between 3 Hz, and 4 Hz, and a wide range of physical phenomena can influence the ABS performance. To properly consider the vehicle dynamics parameters, e.g., the relaxation length of tyres, it is recommended to use a cut-off frequency of minimal 20 Hz. To take such phenomena into account, low  $\mu$  ( $\mu = 0.3$ ) and  $\mu$ -split ( $\mu = 0.6$  on left road side,  $\mu = 0.3$  on right road side) conditions on a straight road were selected. Furthermore, all test cases have been executed several times to check the reproducibility.

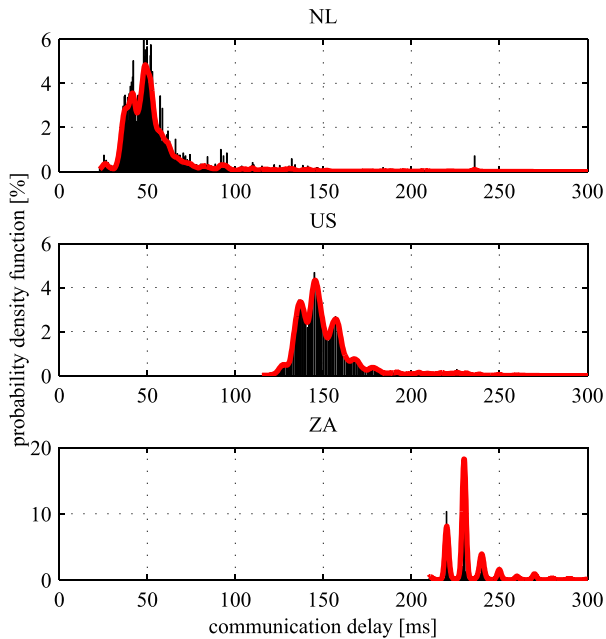
The principle of the implemented ABS algorithm is described in Eq. 7. Here one can see a rule-based four-phase ABS method to control the brake slip according to eq. 8. The friction coefficient between the tyre and the road has a nonlinear characteristic and varies for different road surfaces (dry, wet, icy road). The ABS algorithm controls the wheel slip to achieve a maximum friction coefficient  $\mu$ . To avoid the locking of the wheels, this method releases the brake pressure faster if the slip exceeds a value of 22 %. This ensures a stable contact with the road surface (road holding). In contrast, the application of the brake pressure is smoother, in order to prevent the wheels from locking. In the range of 10 % and 20 % wheel slip the friction coefficient is maximum. Therefore, the pressure is held within this range to ensure maximum deceleration. The proposed ABS logic can be described as follows:

$$\frac{\partial p}{\partial t} = \begin{cases} -2 \frac{\text{bar}}{s} & \text{if } \lambda \in (0.22 \quad 1.00] \\ -1 \frac{\text{bar}}{s} & \text{if } \lambda \in (0.2 \quad 0.22] \\ \pm 0 \frac{\text{bar}}{s} & \text{if } \lambda \in [0.1 \quad 0.20] \\ +1 \frac{\text{bar}}{s} & \text{if } \lambda \in [0.0 \quad 0.10] \end{cases} \quad (7)$$

$$\lambda = 1 - \frac{\omega r}{v_x}, \quad (8)$$

where  $\lambda$  is the wheel slip,  $\omega$  is the rotational wheel velocity,  $v_x$  is the vehicle velocity,  $p$  is the brake pressure.





**FIGURE 4.** Probability density function (PDF) of the communication time delay based on kernel density function (KDF).

#### IV. TEST RESULTS AND ANALYSIS

According to the concept of shared testing, the hosts were distributed on different continents. For all testing cases the master was located at the Technische Universität Ilmenau (Germany). The slave hosts were located at the University of Pretoria (South Africa, ZA), Virginia Tech (United States of America, US), and Delft University of Technology (The Netherlands, NL). This distribution considers the internet routing for different distant hosts. The results obtained for the RT-capability, open-loop, and closed-loop testing are discussed next.

##### A. REAL-TIME

This section discusses the influence of the communication over long distances. To investigate this characteristic, the kernel distribution according to Eq. 4 and Eq. 5 is shown in Fig. 4. It is obvious that the communication delay is not constant. This effect is commonly known as jitter, which describes the deviation from a periodic behaviour. Also, it can be noticed that the probability density function (PDF) of the delay is not normally distributed. It can be also said that the distributions are asymmetric to the left or to the negative side. This phenomena indicates values significantly below the expected range (mean < median).

There is also an influence of the host distance, respectively routing, on the shape of distribution. In this regard, the distributions in Fig. 4 are multimodal; this effect is more characteristic for longer communication distances. The most frequent mode is called the major mode. The less frequent modes are known as minor modes, which represent the multiples of the half-standard deviation.

**TABLE 4.** Benchmarking of time delay, jitter and message loss.

	NL	US	ZA
$t_{delay}$	49 ms	147 ms	230 ms
jitter	$\pm 24$ ms	$\pm 20$ ms	$\pm 17$ ms
RxLoss	5.5 %	58.8 %	8.9 %

On the other hand, for a smaller distance between the master and the slave host, the response resembles a normal distribution. Considering a deterministic behaviour, a normal distribution is more likely to be estimated.

As shown in Table 4, the UDP protocol has a limited reliability. However, in the case of the Netherlands (NL) and South Africa (ZA), the loss of messages is reasonable. In the case of the UDP communication between Germany and USA the loss of messages (58.8 %) was extraordinary; that can be caused by the use of a hotspot with a limited access in USA. Nevertheless, despite the message loss, the exchange of information was considered sufficient.

The most significant impact has the delay in communication. According to the Nyquist-Shannon sampling theorem, a discrete system can be sufficiently assumed as (quasi-) continuous, if the sampling rate is higher or equal to the double of the smallest natural frequency of the observed system [23]. All frequencies beyond that are considered as noise or aliasing. In this regard, the time delays for the communication with US and ZA correspond to 6.8 Hz and 4.3 Hz respectively. Outgoing from the required sampling rate of 20 Hz for anti-lock braking systems, these sampling rates are not sufficient for ABS testing. The time delay for the communication with NL, though, is corresponding to a sampling rate of 20.4 Hz that satisfies the sample rate requirements.

##### B. OPEN-LOOP TESTING

ABS has a diverse set of objectives, such as steerability, vehicle stability, and brake performance. Therefore, the response of the system plays an essential role in ensuring the corresponding indicators of vehicle dynamics at braking. Thus, for some testing cases a loop back is required. However, open-loop tests are performed without any response of the system. In this regards, the feasibility of low  $\mu$  and  $\mu$ -split testing cases are further discussed based on wheel velocities, deceleration, and yaw rate.

Fig. 5a shows that the ABS algorithm is working correctly without any impairment of the functionality. The yaw rate is almost zero and the deceleration and pitch are smooth, Fig. 6a. Also, the average ABS frequency of 3 Hz can be observed. In contrast, the  $\mu$ -split test case in Fig. 5b shows no characteristic ABS phases. This is caused by the yaw motion of the vehicle: an effect of the different friction coefficients under the left and the right tyres produces the yaw moment on the vehicle that also leads to the side slip. In these tests, a high value (100 km/h) of initial vehicle velocity at braking was selected to achieve an unstable vehicle motion due to yaw dynamics and side slip, when ABS functionality is not deteriorated.

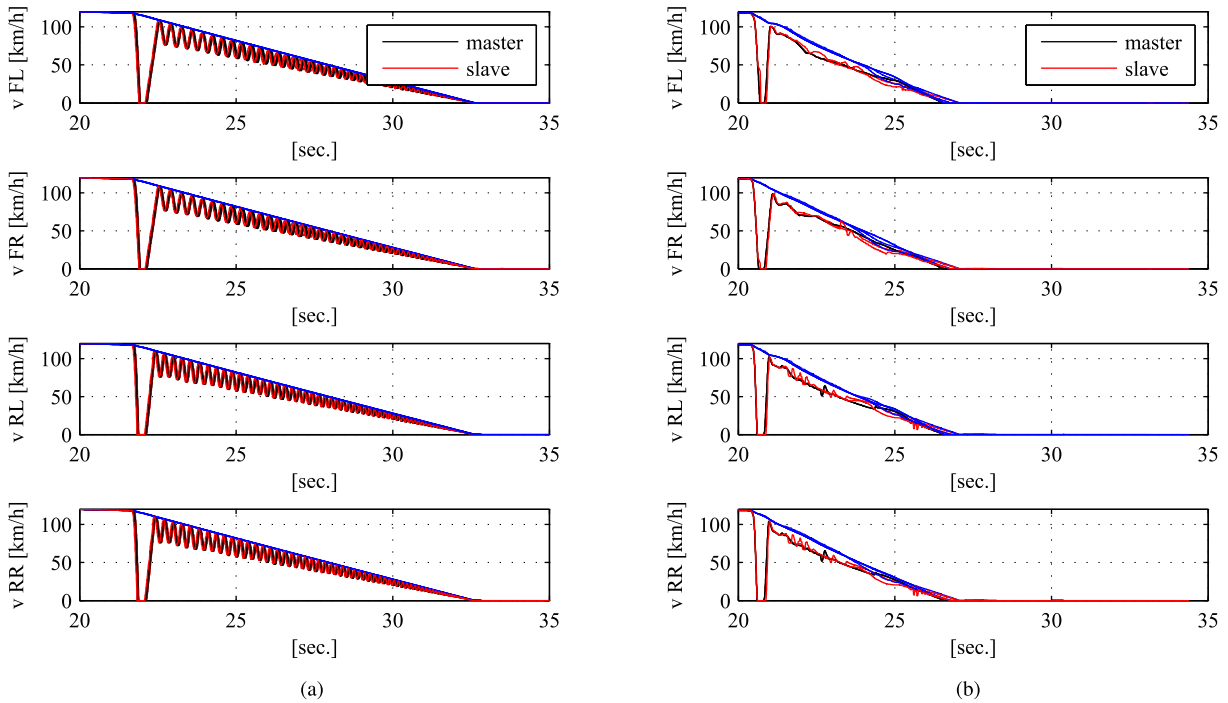


FIGURE 5. Velocity profile while ABS braking for the open-loop testing case. (a) low  $\mu$ . (b)  $\mu$ -split.

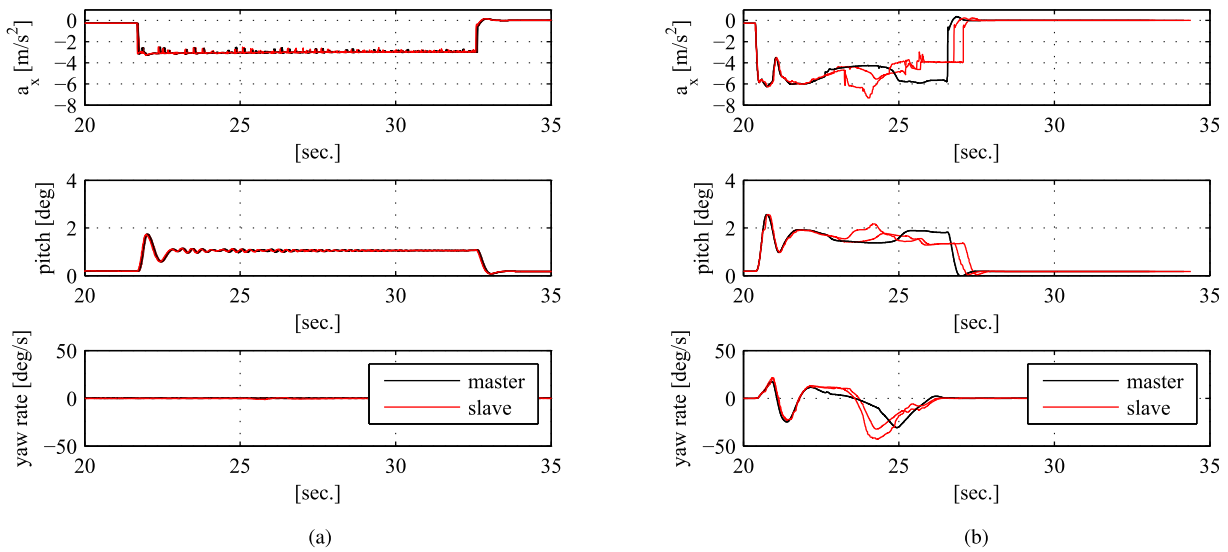


FIGURE 6. Deceleration, pitch and yaw rate while ABS braking for the open-loop testing case. (a) low  $\mu$ . (b)  $\mu$ -split.

It can be seen that in these test cases the master and the slave show the same reproducible behaviour of the ABS system, but the vehicle dynamics in terms of deceleration, and pitch and yaw rate differs for the open-loop split- $\mu$  tests significantly Fig. 6b.

**C. CLOSED-LOOP TESTING**

The closed-loop testing technique considers the response of coupled systems. Thus, the master uses the output of the slave to perform the ABS control.

It was observed that for the low- $\mu$  test case the jittering of the communication causes oscillations in the system, as seen in Fig. 7a and Fig. 8a. Since the jittering communication delay time is significantly exceeding 50 ms (by a sample rate of 20 Hz), the feedback becomes under-sampled. The ABS controller is not able anymore to react appropriately and the brake distance is increased. Moreover, a slight shift of the ABS phases can be observed on Fig. 7a.

In comparison to the open-loop setup, the closed-loop setup ensures a stability of the vehicle. The characteristic

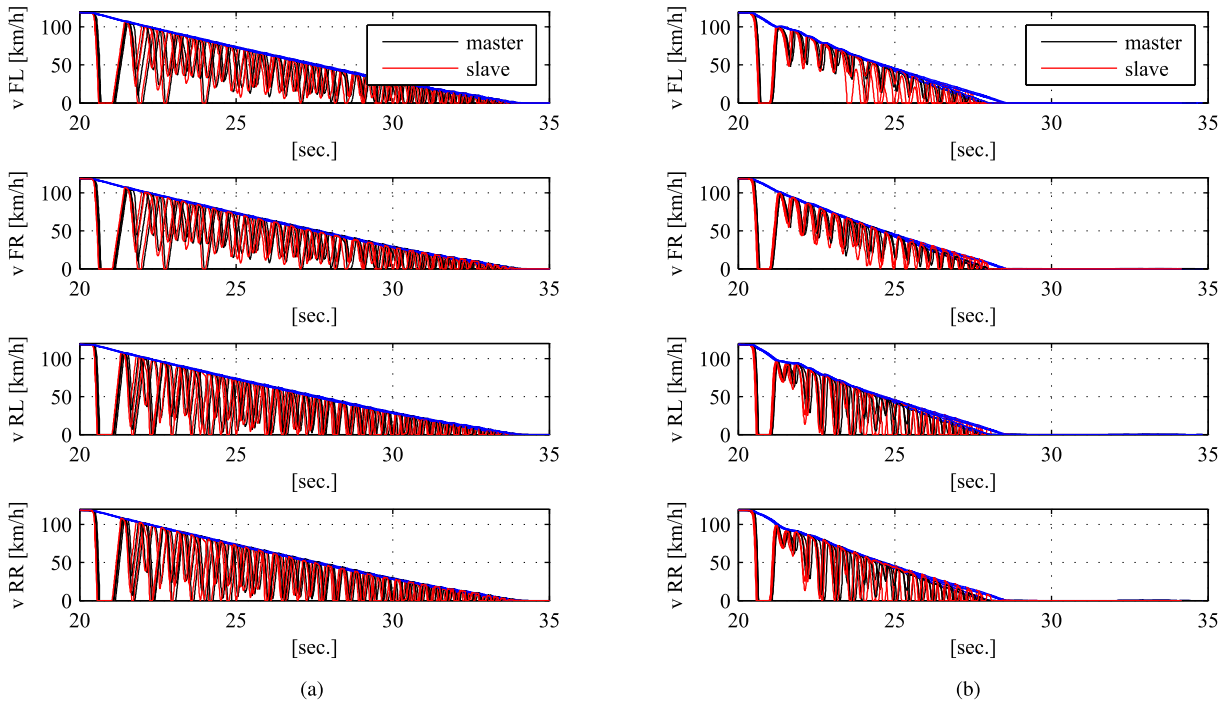


FIGURE 7. Velocity profile while ABS braking for the closed-loop testing case. (a) low  $\mu$ . (b)  $\mu$ -split.

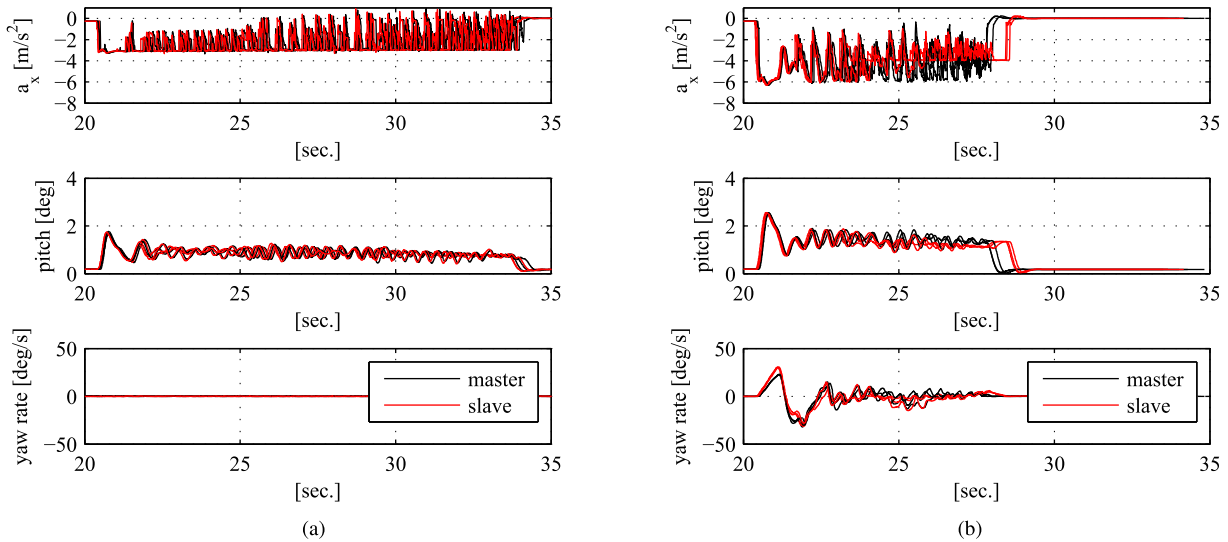


FIGURE 8. Deceleration, pitch and yaw rate while ABS braking for the closed-loop testing case. (a) low  $\mu$ . (b)  $\mu$ -split.

ABS phases are formed and the vehicle remains steerable. In Fig. 8b it can be seen that the yaw rate is settle near zero after initial excitation.

**V. CONCLUSION**

This paper investigates the feasibility of shared and distributed XIL testing techniques for automotive systems. Therefore, RT-simulations using test setups located in different countries (South Africa, United States of America, The Netherlands, and Germany) have been performed via

UDP communication. As a case study, ABS scenarios for a low  $\mu$  and split- $\mu$  road have been carried out. The study outcomes show that the test setups can be classified as soft RT systems. This fact induces some limitations on the use of the developed approach for designing automotive control systems. In particular, RT simulation of the control processes for complex vehicle dynamics (e.g., as closed-loop ABS control on split- $\mu$  road in the presented study) can require essential modifications of described test procedures.



To enhance the quality of the tests service, the following approaches can be considered for further studies:

- (i) high level services or algorithms for compensation of delay, e.g., Kalman-Filter estimator
- (ii) a higher priority for UDP messages to reduce the delay and increase the reliability
- (iii) approach for redundant routing to enable a reliable UDP communication
- (iv) algorithms compensating the loss of messages

## ACKNOWLEDGMENT

This work was partially performed as part of the research project Advanced Co-Simulation Open System Architecture (ACOSAR) within the European EUREKA cluster program ITEA3. The authors thanks the German Research Foundation and the Open Access Publication Fund of the Technische Universität Ilmenau for the support of the Article Processing Charge.

## REFERENCES

- [1] A. Albers, M. Behrendt, J. Schroeter, S. Ott, and S. Klingler, "X-in-the-loop: A framework for supporting central engineering activities and contracting complexity in product engineering processes," in *Proc. 19th Int. Conf. Eng. Design (ICED)*, vol. 6, Seoul, South Korea, 2013, pp. 391–400.
- [2] K. Augsburg, S. Gramstat, R. Horn, V. Ivanov, H. Sachse, and B. Shyrokau, "Investigation of brake control using test rig-in-the-loop technique," SAE Tech. Paper 2011-01-2372, 2011.
- [3] H. K. Fathy, Z. S. Filipi, J. Hagen, and J. L. Stein, "Review of hardware-in-the-loop simulation and its prospects in the automotive area," *Proc. SPIE*, vol. 6228, p. 62280E, May 2006.
- [4] T. Ersal, M. Brudnak, A. Salvi, J. L. Stein, Z. Filipi, and H. K. Fathy, "Development of an Internet-distributed hardware-in-the-loop simulation platform for an automotive application," in *Proc. ASME Dyn. Syst. Control Conf.*, 2009, pp. 73–80.
- [5] V. Ivanov, K. Augsburg, D. Savitski, V. Schreiber, S. Els, and M. Dhaens, "Collaborative engineering of integrated chassis control for ground vehicle: Case study of lifelong learning technologies in automotive mechatronics," in *Proc. IEEE Int. Conf. Mechatronics (ICM)*, Feb. 2017, pp. 449–454.
- [6] H. Lavana, A. Khetawat, F. Brglez, and K. Kozminski, "Executable workflows: A paradigm for collaborative design on the Internet," in *Proc. ACM 34th Annu. Design Autom. Conf.*, 1997, pp. 553–558.
- [7] H. Shokry and M. Hinchey, "Model-based verification of embedded software," *Computer*, vol. 42, no. 4, pp. 53–59, 2009.
- [8] S. Klein et al., "Engine in the loop: Closed loop test bench control with real-time simulation," *SAE Int. J. Commercial Veh.*, vol. 10, no. 1, pp. 95–105, 2017.
- [9] S. Baldi, I. Michailidis, E. B. Kosmatopoulos, and P. A. Ioannou, "A 'plug and play' computationally efficient approach for control design of large-scale nonlinear systems using cosimulation: A combination of two 'ingredients,'" *IEEE Control Syst.*, vol. 34, no. 5, pp. 56–71, Oct. 2014.
- [10] B. Palmintier, B. Lundstrom, S. Chakraborty, T. Williams, K. Schneider, and D. Chassin, "A power hardware-in-the-loop platform with remote distribution circuit cosimulation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2236–2245, Apr. 2015.
- [11] S. Chatzivasilieadis et al., "Cyber-physical modeling of distributed resources for distribution system operations," *Proc. IEEE*, vol. 104, no. 4, pp. 789–806, Apr. 2016.
- [12] P. Palensky, A. A. Van Der Meer, C. D. Lopez, A. Joseph, and K. Pan, "Cosimulation of intelligent power systems: Fundamentals, software architecture, numerics, and coupling," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 34–50, Mar. 2017.
- [13] S. Yoo, K. Choi, and D. S. Ha, "Performance improvement of geographically distributed cosimulation by hierarchically grouped messages," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 5, pp. 492–502, Oct. 2000.
- [14] D. Kim, C.-E. Rhee, and S. Ha, "Combined data-driven and event-driven scheduling technique for fast distributed cosimulation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 5, pp. 672–678, Oct. 2002.
- [15] E. Widl, F. Judex, K. Eder, and P. Palensky, "FMI-based co-simulation of hybrid closed-loop control system models," in *Proc. Int. Conf. Complex Syst. Eng. (ICCSE)*, Nov. 2015, pp. 1–6.
- [16] J. P. Tavella et al., "Toward an accurate and fast hybrid multi-simulation with the FMI-CS standard," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–5.
- [17] C.-H. Yang, G. Zhabelova, C.-W. Yang, and V. Vyatkin, "Cosimulation environment for event-driven distributed controls of smart grid," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1423–1435, Aug. 2013.
- [18] C. Breitwieser, I. Daly, C. Neuper, and G. R. Müller-Putz, "Proposing a standardized protocol for raw biosignal transmission," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 3, pp. 852–859, Mar. 2012.
- [19] G. Carofoglio and L. Muscariello, "On the impact of TCP and per-flow scheduling on Internet performance," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 620–633, Apr. 2012.
- [20] Y. Gao, C. W. Tan, Y. Huang, Z. Zeng, and P. R. Kumar, "Characterization and optimization of delay guarantees for real-time multimedia traffic flows in IEEE 802.11 WLANs," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1090–1104, May 2016.
- [21] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An application protocol for billions of tiny Internet nodes," *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62–67, Mar./Apr. 2012.
- [22] T. Steinbach, F. Korf, and T. C. Schmidt, "Real-time ethernet for automotive applications: A solution for future in-car networks," in *Proc. IEEE Int. Conf. Consum. Electron.-Berlin (ICCE-Berlin)*, Sep. 2011, pp. 216–220.
- [23] A. I. Zayed, *Advances in Shannon's Sampling Theory*. Boca Raton, FL, USA: CRC Press, 1993.



packages within this project.

**VIKTOR SCHREIBER** received the M.Sc. degree in automotive engineering from the Technische Universität Ilmenau, Germany, in 2012. Since 2012, he has been a Research Fellow with the Automotive Engineering Group, Technische Universität Ilmenau. His research is focusing on vehicle dynamics, brake systems, distributed testing methodologies (e.g., XIL), and rapid (control) prototyping. He is a Core Member of the European funded project ACOSAR and coordinating work



**VALENTIN IVANOV** (M'13–SM'15) received the Ph.D. and D.Sc. degrees from Belarusian National Technical University, Minsk, Belarus, in 1997 and 2006, respectively, and the Dr.-Ing.habil. degree from Technische Universität (TU) Ilmenau, Germany, in 2017. From 1995 to 2007, he was an Assistant Professor, an Associated Professor, and a Full Professor with the Department of Automotive Engineering, Belarusian National Technical University. In 2007, he became an Alexander von Humboldt Fellow, and, in 2008, he became a Marie Curie Fellow with TU Ilmenau, Germany. He is currently an EU Project Coordinator with the Automotive Engineering Group, TU Ilmenau. His research fields are vehicle dynamics, electric vehicles, automotive control systems, chassis design, and fuzzy logic. He is a member of SAE International, the Society of Automotive Engineers of Japan, the Association of German Engineers, IFAC (Technical Committee Automotive Control), and the International Society for Terrain-Vehicle Systems.



**KLAUS AUGSBURG** received the Dr.-Ing. degree from Technische Universität Dresden, Germany, in 1985. From 1984 to 1993, he was with industry on leading engineer positions and then from 1993 to 1999 as a Senior Assistant with TU Dresden. In 1999, he was a Full Professor and the Chair of Automotive Engineering Group, Technische Universität Ilmenau, Germany. In 2011, he founded Thuringian Centre of Innovation in Mobility, where he is coordinating many industrial and public research projects. He has authored over 100 research papers, guest lectures, and patents. He is a member of the Association of German Engineers, the Chairman of Workgroup Automotive Engineering VDI Thüringen, and the CEO of Steinbeis-Transferzentrum Fahrzeugtechnik.



**MATTI NOACK** received the B.Sc. degree from Technische Universität (TU) Ilmenau, Germany, in 2015, the M.Sc. degree in technical cybernetics and systems theory from the TU Ilmenau, and the M.Sc. degree in control engineering and automation from the Pontificia Universidad Católica del Perú, Lima, in 2017. He is currently pursuing the Ph.D. degree with the Control Engineering Group, TU Ilmenau. His research interests include observer design for parameter and state estimation as well as nonlinear systems theory and optimal control.



**BARYS SHYROKAU** received the Dipl.-Ing. degree (*summa cum laude*) in automotive engineering from the Belarusian National Technical University, Belarus, in 2004. From 2010 to 2014, he was a Research Ph.D. Student in the framework of the Joint Ph.D. Program between the Division of Control and Instrumentation, Nanyang Technological University, Singapore, and the Institute of Automotive Technology, Technical University Munich. The Ph.D. research was oriented to develop a new approach for integrated vehicle motion control, coordinating multiple vehicle subsystems of a passenger electric car, using optimization-based control allocation and hardware-in-the-loop investigation. Since 2014, he has been a Post-Doctoral Researcher with the Intelligent Automotive Systems, Department of Precision and Microsystems Engineering, Delft University of Technology. His main research interests include vehicle dynamics and control, and current activities focus on the driving simulators and advanced driver assistant systems.



**CORINA SANDU** received the Dipl.-Ing. degree in precision mechanics from the University Politehnica of Bucharest, Romania in 1991, and the M.S. and Ph.D. degrees in mechanical engineering from The University of Iowa, USA, in 1995 and 2000, respectively. She is currently a Professor with the Mechanical Engineering Department, Virginia Tech, USA, and the Director of the Advanced Vehicle Dynamics Laboratory. She published 64 journal and 102 proceedings papers, six book chapters, and over 250 other papers. Her research expertise is in multibody dynamics (modeling, simulation, uncertainty quantification, parameter estimation, sensitivity analysis, and design optimization), vehicle dynamics (suspension, handling, ride, and performance), and terramechanics (vehicle-terrain interaction, tire/track modeling, and soil and terrain characteristics modeling). She is also an ASME Fellow and an SAE Fellow. She is the Editor-in-Chief of the *SAE International Journal of Commercial Vehicles* and an Associate Editor of the *ASME Journal of Computational and Nonlinear Dynamics* and of the *Journal of Mechanics Based Design of Structures and Machines*. She is the Chair of the ASME Design Engineering Division and the Vice-President of the International Society for Terrain-Vehicle Systems.



**PIETER SCHALK ELS** received the Ph.D. degree from the University of Pretoria, South Africa. He was with industry from 1994 to 1999, where he was involved in developing and testing wheeled military vehicles. This included semi-active dampers and the world's first semi-active hydraulic rotary damper for heavy vehicles. In 1999, he joined the University of Pretoria as a Permanent Staff Member. He is currently Research Leader of the Vehicle Dynamics Group. His research is focused on the use of semi-active spring-dampers to improve ride, handling, rollover propensity and life of off-road, and heavy vehicles. This includes tyre, terrain, and suspension characterisation and modeling as well as vehicle dynamics control. He is a member of ISTVS, SAE International, ASME, and SAIMechE and an Editor for *Journal of Terramechanics*.

...