www.czasopisma.pan.pl PAN www.journals.pan.pl
POLSKA AKADEMIA NAUK

Metrol. Meas. Syst., Vol. 29 (2022) No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

PAN POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS

Index 330930, ISSN 0860-8229
www.metrology.wat.edu.pl

M&S

VERIFICATION OF HARDWARE-IN-THE-LOOP TEST BENCH FOR EVALUATING STEERING WHEEL ANGLE SENSOR PERFORMANCE FOR STEER-BY-WIRE SYSTEM

Michał Pietruch[1], Andrzej Wetula[2], Andrzej Młyniec[1]

1) *AGH University of Science and Technology, Faculty of Mechanical Engineering and Robotics, al. A. Mickiewicza 30, 30-059 Kraków, Poland* (✉ *mpiet@agh.edu.pl, +48 792 803 144, mlyniec@agh.edu.pl*)
2) *AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, al. A. Mickiewicza 30, 30-059 Kraków, Poland* (*wetula@agh.edu.pl*)

Abstract

In recent years, the Steer-by-Wire (SBW) technology has been gaining popularity and replacing classical steering systems. It plays the most crucial role in autonomous cars where the vehicle must perform maneuvers on its own without driver's intervention. One of the key components of this system is the steering wheel angle sensor (SAS). Its reliability and performance may affect driver's life and health. The purpose of this paper is to show a test system to comprehensively evaluate the performance of the steering wheel angle sensor in the SBW system during real-world maneuvers and show how SAS parameters such as accuracy of angle, angular speed etc. affect car trajectory resulting in hit cones.
For this purpose, a test system was built, with the use of virtual test drives based on CarMaker software, CANoe and VTSystem hardware. In order to evaluate its performance, the errors introduced by the system were determined. Additionally, using the realised test system, three commercial steering wheel angle sensors were tested and compared during a virtual test drive. Their errors were determined, as well as their performance in the SBW technology and the consistency of the obtained results with the parameters declared by the manufacturer were verified as well.

Keywords: steering wheel angle sensor, SAS, steer-by-wire, hardware-in-the-loop, virtual test drives.

1. Introduction

The *Steer-by-Wire* (SBW) technology has increasingly been used in the latest cars. First of such cars were the Infiniti Q50 and Q60 [1]. In SBW, there is no mechanical connection between the steering wheel and the steering rack, as shown in Fig. 1. Wheel turning is performed by an electric motor based on the measured steering wheel angle. The main advantage of such a system is that the steering ratio can be changed at will, thanks to which one can freely define how strongly the car should react to steering wheel turns, *e.g.* depending on the vehicle speed or the

Article history: received April 11, 2022; revised June 21, 2022; accepted August 5, 2022; available online September 22, 2022.

vehicle driving mode. In addition, vibrations from the wheels are not transmitted to the steering wheel [1, 2], which improves driving comfort, although this feature can also be considered as a disadvantage. SBW technology can also be applied in autonomous cars, allowing the cars to be controlled without the steering wheel movement. As for the disadvantages, first of all, such system is much more complicated because of its many *Hand-Wheel Sensors* (HWS), *Road-Wheel Sensors* (RWS), *Hand-Wheel Motor* (HWM), *Road-Wheel Motor* (RWM) drives, and complicated control methods [1].

Testing such types of systems as well as other devices affecting safety needs to be performed on many levels. Starting from *Model-in-the-Loop* (MIL) through *Hardware-in-the-Loop* (HIL) to road traffic testing [4]. Road testing of immature systems is very dangerous to both the driver and other road users, so there is always the dilemma of whether the manufactured device is mature enough to perform such tests safely.

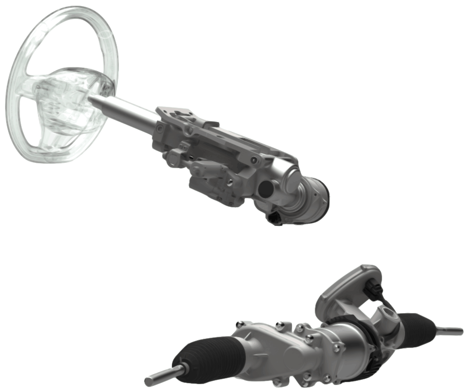

Fig. 1. Steer-By-Wire steering system with visible separation of the steering column from the actuators [3].

One of the components of the SBW system is the *steering wheel angle sensor* (SAS) which is responsible for measuring the angle and angular velocity of the steering wheel and sending this data via a bus to the cooperating devices. An example of such sensor is shown in Fig. 2. This sensor is also used in the operation of many systems such as: *electric power steering* (EPS), *electronic stability program* (ESP), active steering, *lane keep assist* (LKA), four-wheel steering, active suspension [5], adaptive headlights [6] or driver drowsiness detection [7, 8]. The most commonly used protocols are *Controller Area Network* (CAN), *Time Triggered CAN* (TTCAN), FlexRay [1] and *Time Triggered Protocol* (TTP/C) [9]. Based on this data, the RWM actuator performs wheel turns.

Often, SAS testing is limited to basic use cases that have little to do with actual maneuvers in real driving, *e.g.*

1. Turn clockwise/anti-clockwise x angle at low/high angular speed and verify that the SAS shows the correct x angle.
2. Turn right/left with angular velocity x and verify that the SAS shows the correct angular velocity x.

Of course, such tests verify that the SAS correctly measures angle and angular velocity. Unfortunately, the measurements are taken only to a very small extent compared to the extent of its usage in the real world. In real driving, SAS will be responsible for measuring angle and angular velocity in various situations, including much more critical, life and health threatening situations.

Metrol. Meas. Syst.,Vol. 29 (2022), No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

Such maneuvers may include overtaking, avoiding obstacles (moose test) or recovering from a skid. In such cases, the correct operation of the SAS can be of great importance to our health and life. This is the reason which forces us to test such critical cases.

Obviously, some of the data could be collected from real drives, but these would be limited to drives which are safe for the driver. No one would expose the driver to life-threatening maneuvers. In such a solution the coverage of test cases would be higher, but still the number of maneuvers would be limited. Additionally, we are still deprived of test data from critical maneuvers, such as car skidding. In such cases our lives would depend on these tests.

It is also common to see implementations where the test input data was generated using various tools, including MATLAB/Simulink [9, 10] and visualized using the V-Real Builder 2.0 and 3D animation toolbox. Nevertheless, these data do not fully reflect the complexity of the real world and are limited to some maneuvers encountered on the road like obstacle avoidance, lane change [11] or slalom [12].

Fig. 2. Example of a steering wheel angle sensor (SAS).

As an intermediate step between classical HIL tests and road tests, the extension of HIL tests with virtual test drives can be used. During a virtual test drive, the relevant data can be collected in real time and then sent as input vectors to the tested device.

There are several companies in the market that provide such solutions. One solution is the CARLA simulator based on the Unreal Engine 4 graphics engine which generates high quality graphics. For this reason, the images displayed are ideal for testing cameras used for autonomous driving [13]. Another software is one from IPG called CarMaker. It is an advanced software in which you can make virtual test drives represented in a graphical environment. The software allows you to create roads (or to load real roads), specify the number of lanes, slope level, insert trees and buildings and other traffic participants such as cars or pedestrians [4]. You can define specific car maneuvers both common in real driving and less likely to happen. The software has built-in car models, and their kinematics and other characteristics corresponding to real cars. These models can be an alternative to the models defined in such programs as CarSim [14], ADAMS/Car [15] or MATLAB/Simulink [16]. Thanks to this, the virtual car behaves very similarly to its physical counterpart. Moreover, the software allows us to interfere with and modify the default settings of the car: its kinematics, suspension settings, engine characteristics, gearbox, brakes, tires, and many others. It also has models of various types of sensors such as radars, lidars, cameras, *etc*., which can be sources of data for stimulators. Thanks to its enormous configurability, virtual test drives can be a very plausible counterpart of real test drives and complement or even replace them. For this reason, it was decided to build a test system based on this software to test the steering wheel angle sensor very accurately in various driving situations.

2. Test Bench Description

The test system consists of the following components:

- IPG CarMaker software for virtual test drives and their visualization.
- Vector CANoe software to simulate other CAN bus devices working with the SAS.
- Vector VTSystem as CAN interface between CANoe and SAS and as a pulse generator to control stepper motor driver.
- Stepper motor driver.
- Stepper motor with a built-in encoder.
- The tested steering wheel angle sensor.

A schematic of the testbench with interfaces included is shown in Fig. 3, while its actual implementation is shown in Fig. 6.

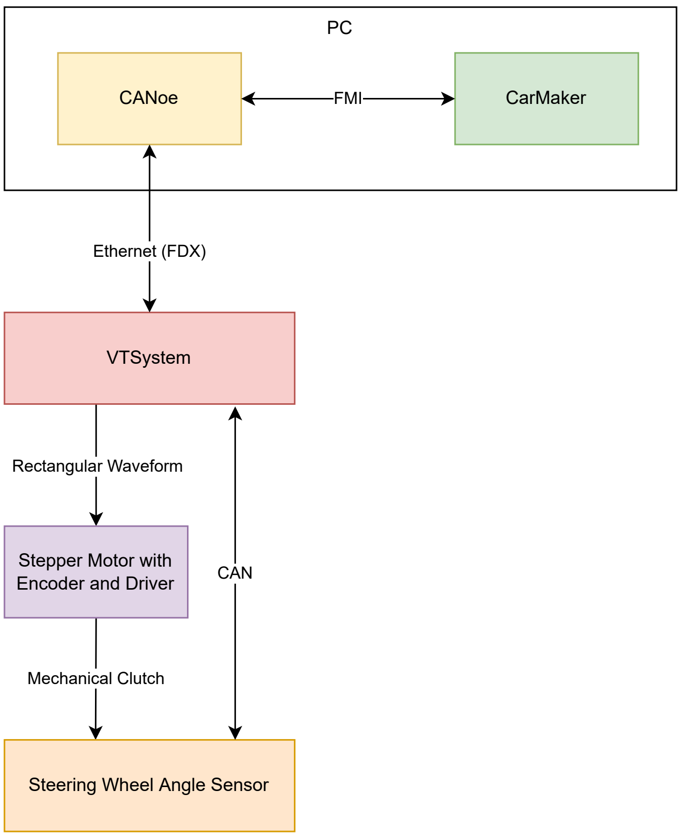

Fig. 3. Testbench diagram including the communication protocols used.

Metrol. Meas. Syst., Vol. 29 (2022), No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

2.1. CarMaker and CANoe

A virtual test drive was prepared in CarMaker consisting in performing a slalom between 10 cones spaced every 18 meters, for a total distance of 230 meters as shown in Fig. 4.

Fig. 4. Visualization of test car during slalom performance.

Of the various test maneuvers available, the slalom was chosen because, while performing it, the steering wheel is turned over a wide range of angles, in both directions, and at different angular velocities. This accurately verifies the performance of the steering wheel angle sensor. The car that performed the run was a Volkswagen Beatle with default parameters set in CarMaker. The car started the slalom with an initial speed of 55 km/h and tried to maintain this speed throughout the slalom execution. The speed of 55 km/h was set experimentally because at this speed the car was able to complete the slalom in the shortest time without knocking down any cones.

In CANoe, a CAN network simulation has been prepared to simulate the other devices that the SAS communicates with. This makes the SAS work properly by measuring both angle and angular velocity and sending and receiving all the data important to it.

2.1.1. Functional Mock-Up Interface

The *Functional Mock-Up Interface* (FMI) was used to communicate between CANoe and CarMaker. It is an open standard that is used for exchanging models between different engineering applications and for co-simulation, which is the execution of simultaneous simulations by two different applications exchanging data in real time [17]. Such a model called *Functional Mock-Up Unit* (FMU) comes in the form of a zipped package with several files in it that define parameters such as interface version, interface type, data exchange period, FMI port number, and most importantly, variables which will be sent and which will be received [18]. The general principle of communication between CarMaker and CANoe using FMU is presented in Fig. 5.

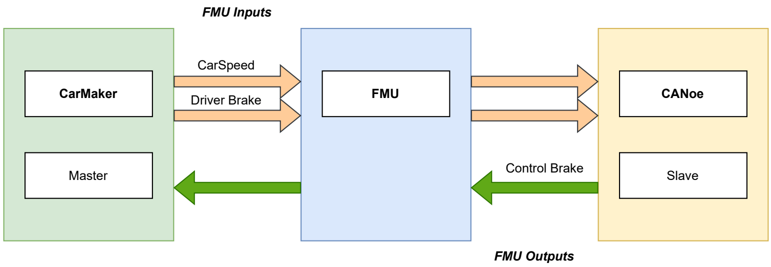

Fig. 5. Scheme of communication between CarMaker and CANoe using FMU in the master-slave architecture.

2.2. VTSystem

VTSystem was used as the interface between CANoe and the SAS under test. VTSystem is a test device that consists of freely exchangeable measurement cards. This choice was made for several reasons. Firstly, VTSystem ensures real-time test execution by using a VT6011 card which provides its own processor with a real-time operating system. Second, it provides a CAN interface for communication with the SAS, for which the card VT6104A was used. Third, by using the VT2848 input/output card, it is able to generate a stable rectangular signal, which is necessary to control the stepper motor driver. In addition to the components above, the VTSystem consists of mating components: VT System Desktop Case 42 HP, VT8006A motherboard, VTC8920B power supply and CANpiggy 1057Gcap transceiver.

The VTSystem is connected to a PC via Ethernet and uses the *Fast Data Exchange* (FDX) protocol to communicate with. Thanks to the FDX protocol, it is possible to exchange data such as system and environment variables, CAN signals or start and stop measurements and simulations [19, 20]. You can also define the variables to be exchanged yourself by using an XML file or by using the FDX Editor graphical environment built into the CANoe software.

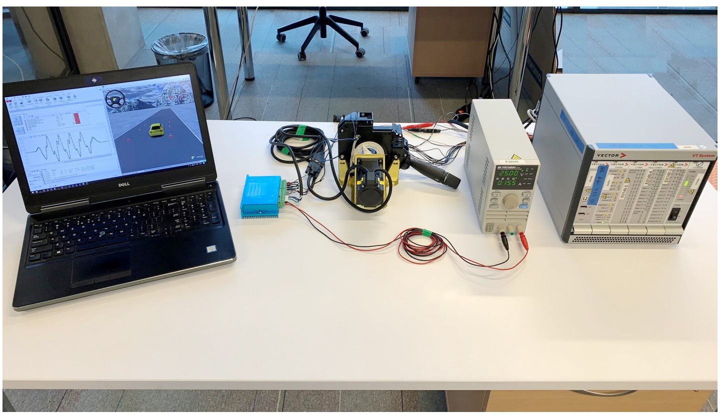

Fig. 6. Steering wheel angle sensor test bench.

2.3. Stepper motor controller

The hybrid controller, HBS57AJ, was used in the design of the workstation, although another controller can be used operating with similar control signals. The most important feature of using

Metrol. Meas. Syst., Vol. 29 (2022), No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

the driver is the possibility of work in feedback with the use of an encoder built in the stepper motor. The controller operates within a voltage range of 24 to 80 V, while it accepts control signals within a range of 5 to 24 V. The controller accepts 2 digital signals: one which determines the number and speed of the performed steps, and the second one defining the direction of rotation. The number of steps performed by the motor is equal to the number of pulses that we have fed to the controller, while the angular velocity of the motor is proportional to the frequency of feeding these pulses.

The controller also has four DIP switches that can be used to enable microstep operation. Depending on the switch combination, a resolution of up to 200 microsteps can be achieved.

2.4. Stepper motor

A stepper motor with a built-in encoder was chosen because stepper motors controlled in an open loop can often lose steps and do not reach the required angle of rotation. The encoder provides feedback, based on which the controller compensates for any errors, causing the motor to reliably rotate by the set angle.

The motor used is a two-phase bipolar stepper motor with a maximum torque of 2.5 Nm with a minimum angle of rotation of 1.8° with a tolerance of ±5%. The result is that 200 pulses need to be applied evenly for the motor to rotate 360°.

2.5. Principle of operation and variable mapping

The FMU was defined by creating a model using a graphical application included in CANoe – *Functional Mock-Up Interface* (FMI). Since CANoe is used only for visualization while working with the VTSystem, with the calculations performed directly on the VTSystem, after generating the FMU file it was still necessary to change the IP address in it to that of the VTSystem.

The finished FMU was loaded into CarMaker and then the variables from CANoe were mapped to internal CarMaker variables. After loading the FMU, the control of the car was transferred from CarMaker to the model FMU. From that moment, the car was controlled not from CarMaker but based on the variables received from CANoe.

A virtual test run is started in CarMaker, which simultaneously starts CANoe. From CarMaker, the angle and angular velocity values with which the driver would like to turn the steering wheel are sent to CANoe via FMI. Then, based on the received values in CANoe using *Communication Access Programming Language* (CAPL) and using using equations (1) and (2), the required parameters were calculated and the stepper motor was controlled.

$$\text{Pulse number} = \frac{\text{Requested Angle}}{\text{Angle of single motor step}}, \tag{1}$$

$$\text{Pulse frequency} = \frac{\text{Requested angular velocity}}{\text{Angle of single motor step}}. \tag{2}$$

At the same time, SAS measures the angle and angular velocity value, which it then sends to CANoe via the CAN bus. Canoe upon receiving the data frame immediately sends it back to CarMaker. Finally, CarMaker applies the received data to the steering rack and the car performs the maneuver reflecting this in the graphical environment.

It is worth adding here that the exchanged variables are of double 64-bit floating point type. This allows for a very accurate data exchange and to avoid rounding or truncation errors. The flow and mapping of the variables is shown in Fig. 7.

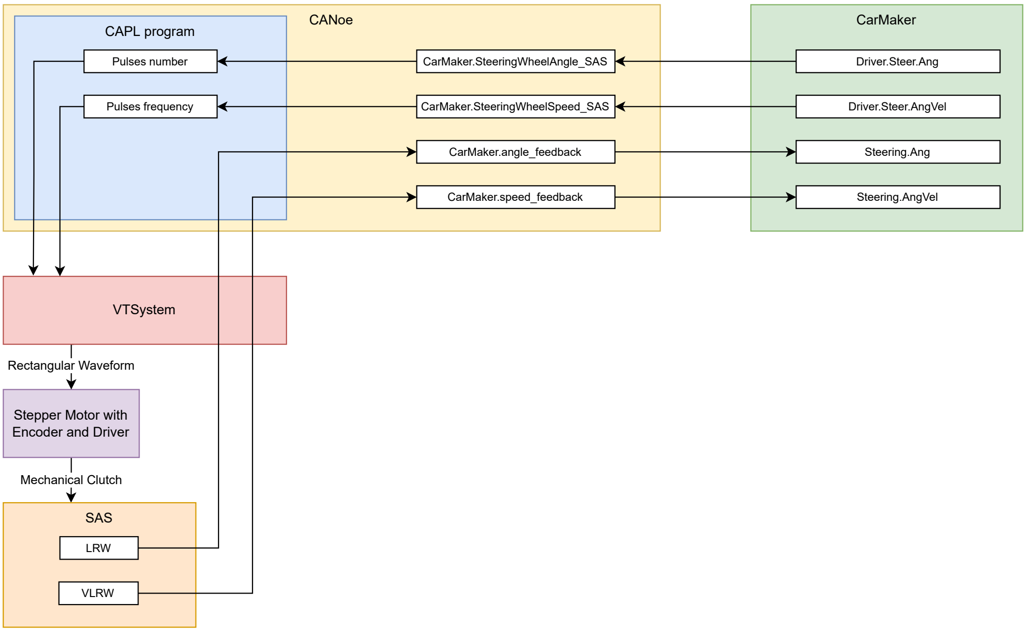

Fig. 7. Flow and mapping of variables between the test system components.

2.6. System performance tests

Since the exchange of information between FMUs occurs cyclically at discrete moments in time, and the communication between the VTSystem and the PC running CANoe and CarMaker introduces delays, the bench introduces an unknown delay between the output and input of the model running in CarMaker. Since the values of this delay define constraints on the model, it became necessary to determine their effect on the performance of the model. For this purpose, a connection was set up and shown in Fig. 8. The data received in CANoe from CarMaker after passing through the VTSystem was immediately sent back to CarMaker. To determine the quality of the system, it was decided to determine 3 parameters: the *maximum absolute angle error* (MAE), the *root mean squared error* (RMSE) and the number of hit cones.

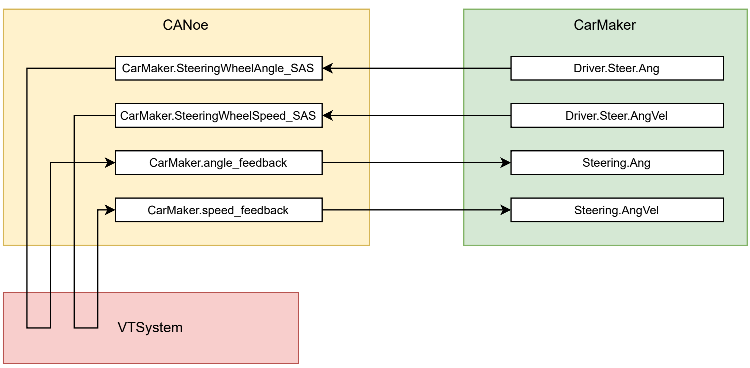

Fig. 8. Devices and variable mapping during determination of errors introduced by the test system.

Metrol. Meas. Syst., Vol. 29 (2022), No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

Note here that delay measurements do not include delays introduced by the stepper motor driver and the stepper motor itself. Determining these delays would require additional hardware and disconnection of these devices. However, these delays can be ignored, because the motor gets commands every 1 ms, which means that the angle difference it has to make during the 1 ms is very small, on the order of the motor's accuracy, *i.e.* about 0.01°.

The data exchange rate between CarMaker and CANoe is configurable, and in this case, it was set to a period of 1 ms, which means that data is exchanged between these applications every 1 ms. Such a period of time allows, on the one hand, for a reliable transfer of the data between applications without overloading the processor, and on the other, provides sufficient accuracy required for testing. This value at the same time constitutes the lower limit of the data transmission period from SAS.

2.6.1. Test procedure

The following procedure was used both to determine the errors introduced by the test system and to test the steering angle sensors discussed in later sections of this publication.

1. Start the virtual test drive.
2. Collect the waveforms:
   a) the steering angle requested by the driver and
   b) the steering angle measured by the car
3. Note down how many cones the car knocked down
4. Process the data and determine the RMSE (3) and MAE (4)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i,\,j=1}^{n}\left(X_i - X_j\right)^2}\,, \tag{3}$$

$$\mathrm{MAE} = \max\left(\Delta x_1,\ \Delta x_2,\ \ldots,\ \Delta x_n\right), \tag{4}$$

where:

$$\Delta x_n = |X_{in} - X_{jn}|. \tag{5}$$

n is the number of samples, X_i is requested value, X_j is measured value.

2.6.2. Results

After performing a virtual test run, the angle values sent from CarMaker and received from CANoe at the given time points were collected and the difference calculated, as shown in Fig. 9. Additionally, the number of cones knocked over was noted.

The MAE was 0.96° and the RMSE was only 0.06°. The low RMSE indicates that the sent and received waveforms are very close over the entire range, as can be seen in Fig. 9. The MAE is an order of magnitude larger at 0.96°, and this is due to the very rapid change of the steering wheel angle when avoiding the cones. This results in larger errors, whereas, since it is a very short maneuver, despite the large value it has negligible effect on the average error of the whole run. Moreover, the car drove the whole slalom without knocking down any cone. These values show that the errors introduced by CANoe and VTSystem are negligible and such a system can be used to test the steering wheel angle sensor.

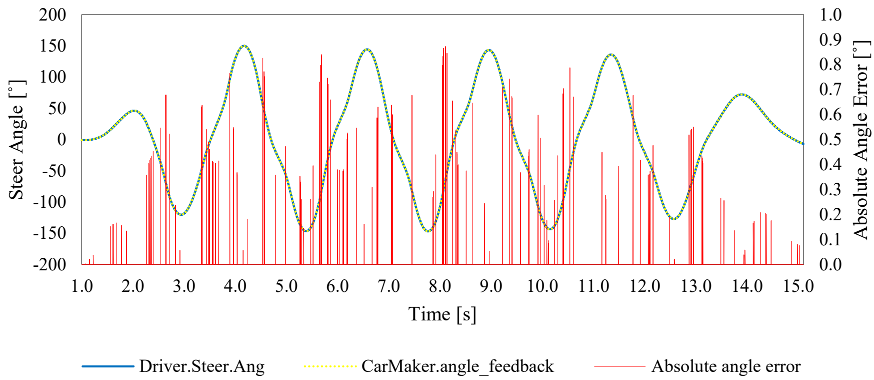

Fig. 9. Superimposed CarMaker transmitted and received angle waveforms and absolute angle error.

3. Steering wheel angle sensors testing

Two generations of steering wheel angle sensors from the same manufacturer SAS I and SAS II and one competitor's sensor SAS III were tested based on the same virtual test drive. These are commercial sensors used in passenger cars. According to the manufacturer, they have an angle measurement accuracy of 0.5° to 0.1° and a frame sending period of 10 ms, which is 10 times the data exchange period of the system. The sensors use a 500 kbit/s High Speed CAN bus.

The most accurate sensor which was tested measures with an accuracy of 0.1° therefore the accuracy of the stepper motor must be better. The motor is controlled in the microstep mode and a single step of 1.8° was divided 200 more times. In the end, each single pulse rotates the motor by 0.009° which results in sufficient accuracy. In this case, a complete rotation of the motor by 360° requires applying 40 thousand impulses to it.

3.1. SAS I

The value of the reference angle and the value of the angle measured by SAS I during the entire run were measured. As can be seen in Fig. 10, the angle waveform is delayed with respect

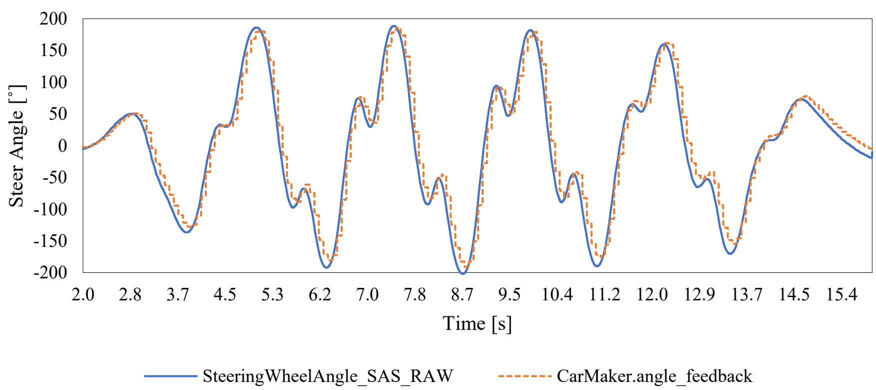

Fig. 10. The steering wheel angle requested by the driver and the angle measured by SAS I.

Metrol. Meas. Syst., Vol. 29 (2022), No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

to the commanded waveform and heavily stepped. The main reason of the delay is the sending of the frame with a period of 10 ms and, to a lesser extent, the delay introduced by the test system. The stepped waveform is caused by low angle measurement accuracy, much lower than the one declared by the manufacturer, which is 0.5°. Analyzing the collected data, it was found that although the claimed measurement resolution is 0.5°, in reality SAS I very rarely returns half values, and most of the time it returns integer values. This may indicate that the accuracy is more than the claimed 0.5°. The RMSE was 23.4°, while the MAE was as high as 77.6°, resulting in 1 cone falling over during the run. The error graph is presented in Fig. 11.

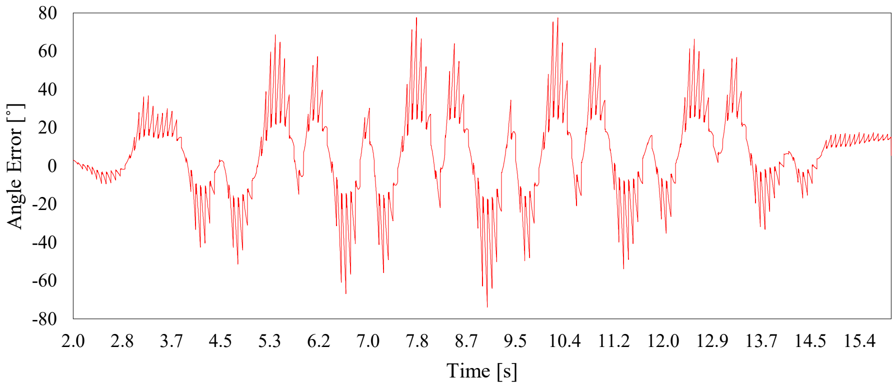

Fig. 11. Angle error of SAS I.

3.2. SAS II

In SAS II, the mechanical and electronic components and the algorithm for calculating the angle have been modified, but the angle is still measured with an accuracy of 0.5° and sent to the bus at a period of 10 ms. As can be seen in Fig. 12, the measured waveform is also delayed, as the angle is still reported to the bus every 10 ms and there is also a delay introduced by the test system. On the other hand, the waveform is much smoother and there are no visible steps on it. This is because SAS II reports half angles much more often than SAS I, according to the claimed accuracy of 0.5°. This may indicate that the sensor meets the parameters declared by the

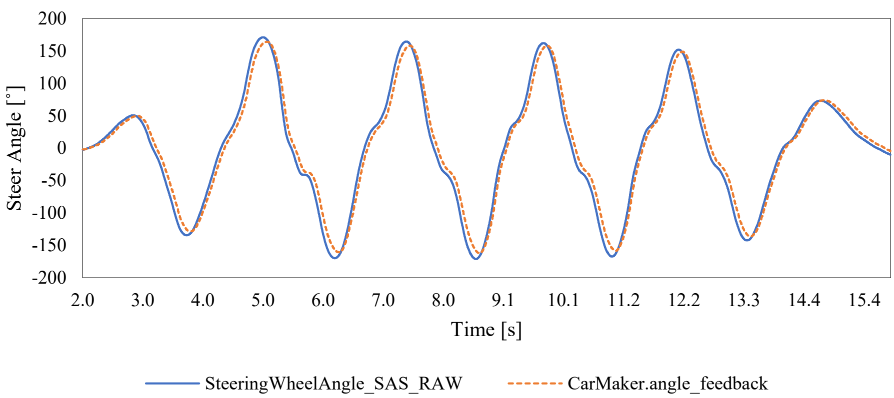

Fig. 12. The steering wheel angle requested by the driver and the angle measured by SAS II.

manufacturer. The RMSE was 14.4°, almost 2 times less, and the MAE was 35.4°, more than 2 times less than for SAS I, which made it possible to complete the slalom without knocking down any cones. The graph of the error is presented in Fig. 13.

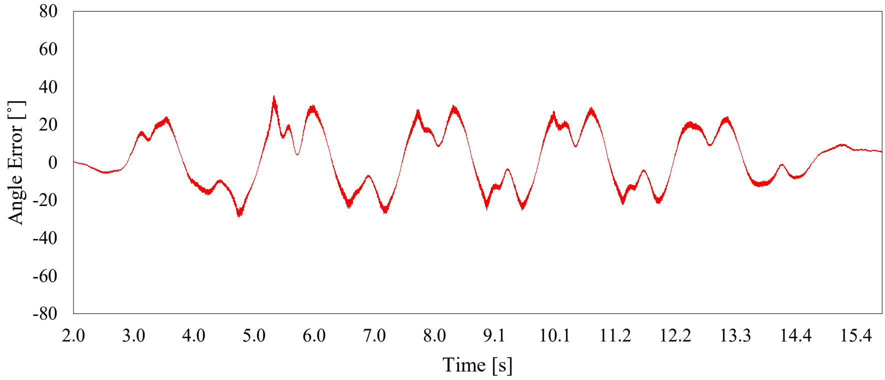

Fig. 13. Angle error of SAS II.

3.3. SAS III

The sensor has the best claimed accuracy of 0.1°, which leads us to assume that the results obtained will be the best. As can be seen in Fig. 14 below, the waveform is minimally delayed relative to the commanded, as the angle is still reported to the bus every 10 ms and there is also a delay introduced by the test system. The waveform itself is smooth throughout and covers the set waveform more accurately than its predecessors, a result of the sensor's high resolution of 0.1°. This is confirmed by the RMSE, which is just 7.95°, almost 2 times less than for SAS II. The biggest differences, compared to the two previous sensors, can be observed at the point where the steering wheel direction changes (peak values of the waveform). It can be seen that in these places, the courses of the set angle and the received angle are more similar, which is confirmed by the MAE equal to 17.1°, *i.e.* more than 2 times less than for SAS II. The absolute error waveform is presented in Fig. 15. The high performance of the sensor was finally confirmed during the slalom performance, as the car crossed the track flawlessly without knocking down any cones.

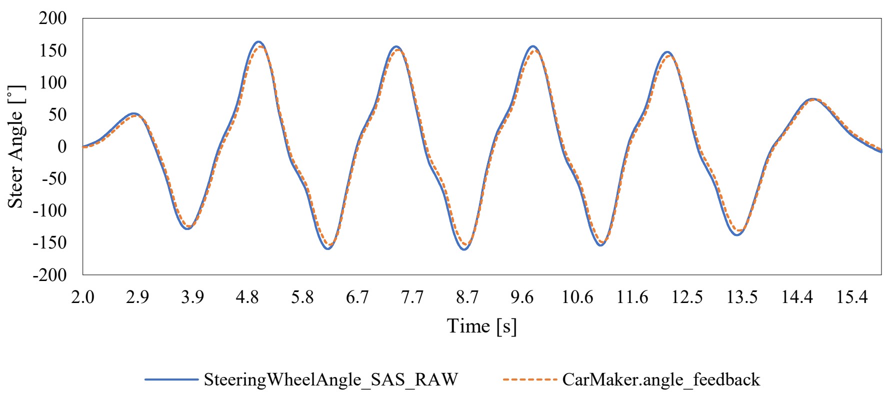

Fig. 14. The steering wheel angle requested by the driver and the angle measured by SAS III.

Metrol. Meas. Syst., Vol. 29 (2022), No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

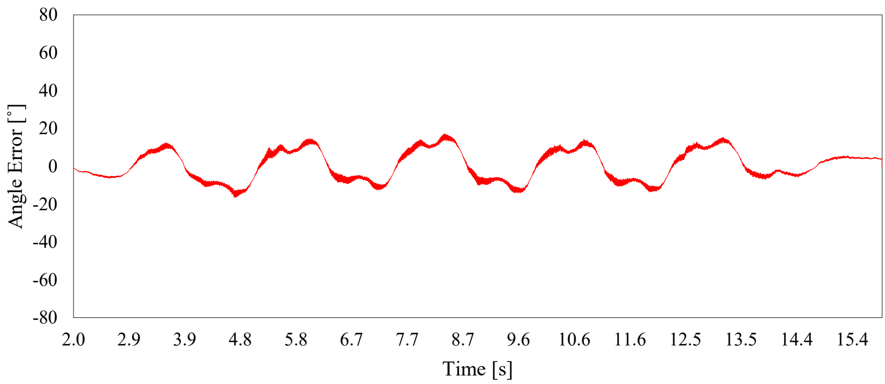

Fig. 15. Angle error of SAS III.

3.4. Comparison of results

The values summarized in Table 1 show that the performance of SAS II is twice the value of SAS I, despite having the same accuracy of 0.5° catalogued. As written earlier, it is likely that SAS I does not meet the parameters claimed by the manufacturer and its accuracy is closer to 1.0° than 0.5°. In contrast, SAS III proved to be twice as efficient as SAS II. This result was to be expected since the claimed angle accuracy was 0.1°. However, the results show that an increase in accuracy does not at all yield a proportional increase in sensor performance during actual driving. For example, 5 times better accuracy of SAS III compared to SAS II resulted in only a 2× improvement in the slalom performance. The cars equipped with both SAS II and SAS III did not knock down any cones, while the car with SAS I knocked down one cone. For this reason, SAS I is not suitable for use in the SBW system because it can cause danger to passengers and other road users.

Table 1. Comparison of errors obtained during measurements of SAS I, SAS II and SAS III.

	SAS I	**SAS II**	**SAS III**
MAE	77.6°	35.4°	17.1°
RMSE	23.4°	14.4°	7.95°
Number of cones hit	1	0	0

4. Conclusions

The steering wheel angle sensor is widely used in many systems and plays the key role in them. The reliability and performance of the sensor can affect driver's health and life. This is why the sensor should be tested comprehensively in a wide range of use cases. The test system was built exactly for that purpose using virtual test drives and based on CarMaker, CANoe and VTSystem. Initial tests of the developed system gave satisfactory results allowing it to be used in practice. Tests based on this system revealed that the parameters of commercial sensors may differ from those declared by the manufacturer. In addition, it has been shown how steering wheel angle sensor's parameters affect an SBW equipped car during a slalom performance resulting in hit cones. These results prove that the built test system can be a great tool to verify the parameters declared by the manufacturer, to compare the performance of different sensors during real-life maneuvers and, most importantly, to comprehensively test the sensor under various critical road conditions before it would go under testing in road traffic.

References

[1] Mortazavizadeh, S. A., Ghaderi, A., Ebrahimi, M., & Hajian, M. (2020). Recent Developments in the Vehicle Steer-by-Wire System. *IEEE Transactions on Transportation Electrification*, *6*(3), 1226–1235. https://doi.org/10.1109/TTE.2020.3004694

[2] Mohamed, Eid S., & Albatlan, Saeed A. (2014). Modeling and Experimental Design Approach for Integration of Conventional Power Steering and a Steer-By-Wire System Based on Active Steering Angle Control. *American Journal of Vehicle Design*, *2*(1), 32–42. https://doi.org/10.12691/ajvd-2-1-5

[3] Nexteer Automotive. (n.d). *ADAS & Automated Driving STEER-BY-WIRE*. Retrieved January 10, 2022, from https://www.nexteer.com/a-d-a-s-automated-driving/steer-by-wire/

[4] Pietruch, M., Młyniec, A., & Wetula, A. (2020). An overview and review of testing methods for the verification and validation of ADAS, active safety systems, and autonomous driving. *Mining – Informatics, Automation and Electrical Engineering*, *1*(1), 19–27. https://doi.org/10.7494/miag.2020.1.541.19

[5] Zacharia, S., George, T., Rufus, E., & Alex, Z. C. (2017, July). Implementation of steering wheel angle sensor system with Controlled Area Network. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)* (pp. 54–60). IEEE. https://doi.org/10.1109/ICICICT1.2017.8342534

[6] Youn, J., An, J., Di Yin, M., Cho, J., & Park, D. (2016). Safe Adaptive Headlight Controller with Symmetric Angle Sensor Compensator Using Steering-swivel Angle Lookup Table. *Transactions of the Korean Society of Automotive Engineers*, *24*(1), 112–121. https://doi.org/10.7467/ksae.2016.24.1.112

[7] Moon, B. J., Yeon, K. B., Lee, S. G., Hong, S. P., Nam, S. Y., & Kim, D. H. (2012). Drowsy Driving Detection Algorithm Using a Steering Angle Sensor and State of the Vehicle. *The Institute of Electronics and Information Engineers*, *49*(2), 30–39. https://www.koreascience.or.kr/article/JAKO201223052005905.pdf

[8] Li, L., Werber, K., Calvillo, C. F., Dinh, K. D., Guarde, A., & König, A. (2014). Multi-sensor soft-computing system for driver drowsiness detection. In *V. Snášel, P. Krömer, M. Köppen, & G. Schaefer* (Eds.), *Soft Computing in Industrial Applications. Advances in Intelligent Systems and Computing* (pp. 129–140). Springer. https://doi.org/10.1007/978-3-319-00930-8_12

[9] Ataide, F., Santos, M. M., & Vasques, F. (2004, December). A Comparison of the Communication Impact in CAN and TTP/C networks when supporting Steer-by-Wire Systems. In *2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT'04*. (Vol. 2, pp. 1078–1083). IEEE. https://doi.org/10.1109/icit.2004.1490227

[10] Shah, M. B. N., Husain, A. R., & Dahalan, A. S. A. (2013, November). An analysis of CAN-based steer-by-wire system performance in vehicle. In *2013 IEEE International Conference on Control System, Computing and Engineering* (pp. 350–355). IEEE. https://doi.org/10.1109/ICCSCE.2013.6719988

[11] Wang, C., Li, F., Wang, Y., & Wagner, J. R. (2021). Haptic Assistive Control with Learning-Based Driver Intent Recognition for Semi-Autonomous Vehicles. *IEEE Transactions on Intelligent Vehicles*, 1–13. https://doi.org/10.1109/TIV.2021.3137805

[12] Wang, C., Wang, Y., & Wagner, J. R. (2019). Evaluation of a Robust Haptic Interface for Semi-Autonomous Vehicles. *SAE International Journal of Connected and Automated Vehicles*, *2*(2), 1–16. https://doi.org/10.4271/12-02-02-0007

Metrol. Meas. Syst., Vol. 29 (2022), No. 4, pp. 639–653
DOI: 10.24425/mms.2022.143065

[13] Jabłoński, P., Iwaniec, J., & Jabłoński, M. (2021). Multisensory Testing Framework for Advanced Driver Assistant Systems Supported by High-Quality 3D Simulation. *Sensors*, *21*(24), 1–20. https://doi.org/10.3390/s21248458

[14] Lee, S. J., Park, K., Hwang, T. H., Hwang, J. H., Jung, Y. C., & Kim, Y. J. (2007). Development of Hardware-in-the-Loop Simulation System as a Testbench for ESP Unit. *International Journal of Automotive Technology*, *8*(2), 203–209. https://www.koreascience.or.kr/article/JAKO200734514893963.pdf

[15] Yeh, C. J., Ho, S. R., Lin, M. C., Hu, T. H., & Hsu, T. H. (2007, September). Development of a test bench for tuning and validating electric power steering control method. In *2007 IEEE Vehicle Power and Propulsion Conference* (pp. 618–622). IEEE. https://doi.org/10.1109/VPPC.2007.4544197

[16] Anwar, S. (2010). Fault Detection, Isolation, and Control of Drive by Wire Systems. In W. Zhang (Eds.), *Fault Detection* (pp. 231–254). IntechOpen. https://doi.org/10.5772/9079

[17] Modelica Association. (2020). *Functional Mock-up Interface for Model Exchange and Co-Simulation. Document version: 2.0.2*. Retrieved January 10, 2022, from https://fmi-standard.org/downloads/

[18] IPG Automotive GmbH. (2019). *CarMaker Tips & Tricks No. 6-005. Connecting CarMaker with CANoe through FMU*. Retrieved January 10, 2022. https://ipg-automotive.com/uploads/tx_pbfaqtickets/files/86/CarMaker_CANoe_FMU_connect.pdf

[19] Vector Informatik GmbH. (2018). *Manual CANoe FDX Protocol*. *Version 2.0*

[20] Vector Informatik GmbH. (2015). *Fast Data Exchange with CANoe*. *Version 1.0*. [Application note]. https://cdn.vector.com/cms/content/know-how/_application-notes/AN-AND-1-119_Fast_Data_Exchange_with_CANoe.pdf

Michał Pietruch obtained the B.Sc. degree in mechatronics in 2014 and the M.Sc. in electrical engineering in 2016, both from the AGH University of Science and Technology in Cracow, Poland. Since 2018 he has been pursuing the Ph.D. degree in Automatics and Robotics at the same university. Since 2015 he has worked in the automotive industry and currently holds a position of Software & System Tests Engineering Group Manager. His research activity focuses on testing and verification of advanced driver assistant systems, drive-by-wire systems, and autonomous driving.

Andrzej Młyniec received his PhD in 2011 from the AGH University of Science and Technology, Cracow, Poland and the D.Sc. (habilitation) in 2016 from the Warsaw University of Technology. He has been involved in research and development in the automotive industry for over 10 years. He is currently employed as Professor at the AGH University of Science and Technology and his research interests are application of modern modeling methods, including machine learning in scientific research and industrial applications.

Andrzej Wetula obtained the M.Sc. and the Ph.D. degrees in electrical engineering from the AGH University of Science and Technology in Cracow, Poland. He is currently employed as Assistant Professor at the Department of Measurement and Electronics, Faculty of Electrical Engineering, Automatics, Computer Science, and Biomedical Engineering of the same university. Author and co-author of 73 papers and patents.