Strojnícky časopis – *Journal of* MECHANICAL ENGINEERING,
VOL 75 (2025), NO 3, 13 - 20

VIRTUAL MODEL OF THE KYBURZ VEHICLE USING THE FMI STANDARD

BUCHA Jozef[1*], DANKO Ján[1], MAGDOLEN Ľuboš[1], ČULÍK Ján[1]

[1]Slovak University of Technology in Bratislava, Faculty of Mechanical Engineering, Institute of Automotive Engineering and Design, Nám. Slobody 17,812 31 Bratislava, Slovakia, e – mail: jozef.bucha@stuba.sk

Abstract: The article deals with describing the methodology of connecting Adams/Car and Matlab/Simulink programs using the FMI standard. To demonstrate the connection, a Kyburz vehicle model was created as a virtual vehicle.

KEYWORDS: Adams/Car, Matlab/Simulink, Kyburz eRod, FMI standard

1 Introduction

Simulations play a key role in the modern vehicle development process. They enable virtual testing of functionality, reliability, and safety of components before their physical production. Thanks to simulations, development time can be significantly shortened, prototype costs reduced, and design optimization improved. An additional advantage is the ability to test extreme operating conditions and various configurations without the risk of damaging real components. Simulation tools thus enhance the quality and innovation of the final product. The FMI (Functional Mock-up Interface) standard represents an open format for the exchange and integration of models between different simulation tools. It allows the export of models defined as FMUs (Functional Mock-up Units), which can be used for co-simulation in other software without the need to share source code. FMI supports modularity and reusability of models.

2 Multibody Model of Vehicle in Adams/Car

The vehicle model used to demonstrate the functionality interconnection of the MBD model in the Adams/Car program and the Matlab/Simulink program was the Kyburz eRod. vehicle.

Fig. 1 Kyburz eRod

DOI: 10.2478/scjme-2025-0032, Print ISSN 0039-2472, On-line ISSN 2450-5471

The Adams/Car program uses a three-level vehicle modeling system (Fig. 2).

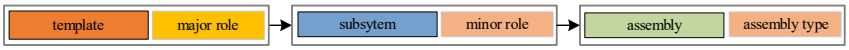

Fig. 2 Modeling system in Adams/Car

1. Template. Templates are parametric models that define topology of models. Basic property of every template is Major Role, which defines the functional part of the vehicle e.g. suspension, steering, …
2. Subsystem. Subsystems are based on templates and allow users to change the parametric data of the template as well as the definition of some components. Basic property of every subsystem is Minor Role, which defines the functional area of the vehicle template.
3. An assembly represents a collection of subsystems, along with optional test rig, which when assembled forms a system that can be analyzed using Adams/Solver.

Figure 3 shows a block diagram of the vehicle assembly (green) with the used templates (red), their respective Major Roles (yellow), subsystems (blue), and their respective Minor Roles (pink).

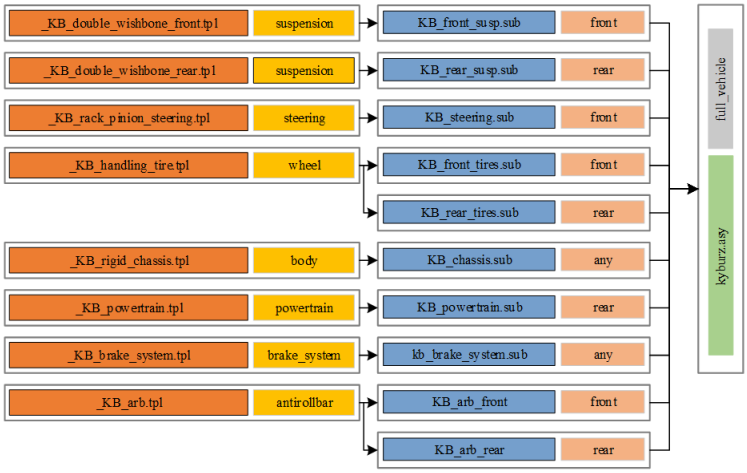

Fig. 3 Block diagram of Kyburz model in Adams/Car

Figure 4 on the shows an example of one template (Major Role: Suspension), which will be used in subsystem as front suspension (Minor Role: Front).

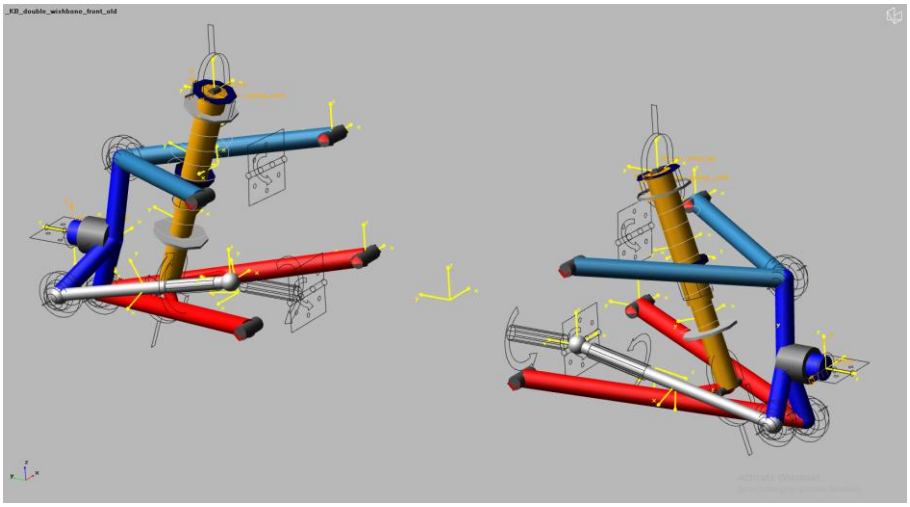

Fig. 4 Template of suspension

Figure 5 shows complete full vehicle assembly of Kyburz in Adams/Car.

Fig. 5 Full Vehicle Assembly of Kyburz

3 Simulation of Full Vehicle Assembly

The Driving Machine in Adams Car is a software module (a virtual driver) that controls the entire vehicle during simulation, ensuring consistent, repeatable, and automated vehicle tests. The Driving Machine steers the vehicle, applies the throttle and brake, and shifts gears (using the clutch) (Fig. 6). The Driving Machine can operate in open-loop mode (without feedback) or in closed-loop mode (with feedback). Adams SmartDriver is an advanced driver simulator that can bring a vehicle to its dynamic limits (Closed Loop). Adams SmartDriver is not used in this article.

An event in Adams Car is a test maneuver that defines what happens to the vehicle during the simulation. An event is represented by an xml file that contains data for defining and transmitting sets of command signals, feedback signals, and parameters for each of the five control inputs: steering, throttle, brake, gear, and clutch (Fig. 7).

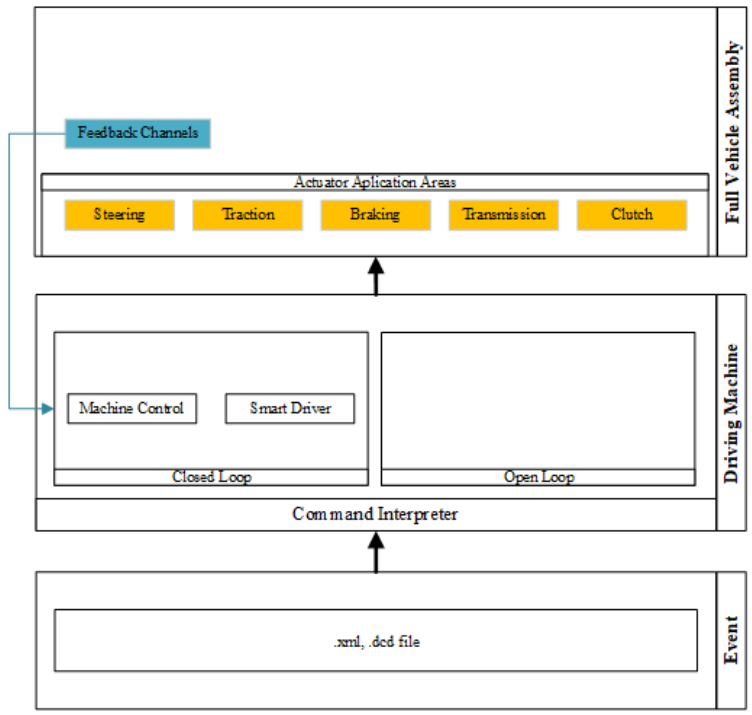

Fig. 6 General Description of Driving Machine

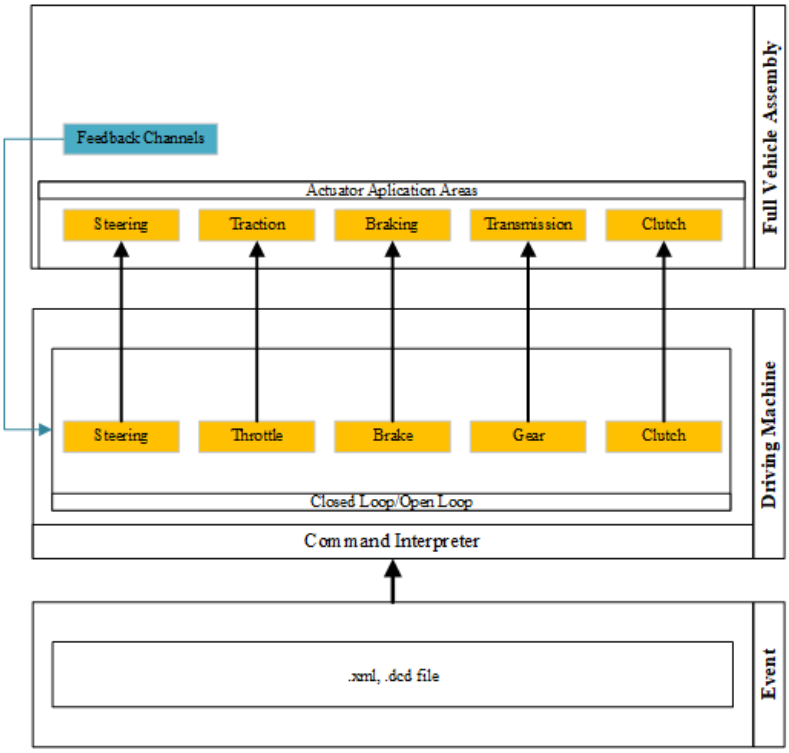

Fig. 7 Actuator areas on Driving Machine

Event is defined for five control signals (steering, throttle, brake, gear, and clutch) separately; therefore, Driving Machine can be defined as Pure Open Loop (Fig. 8), Pure Closed Loop (Fig.9) or Combination of Open and Closed Loop.

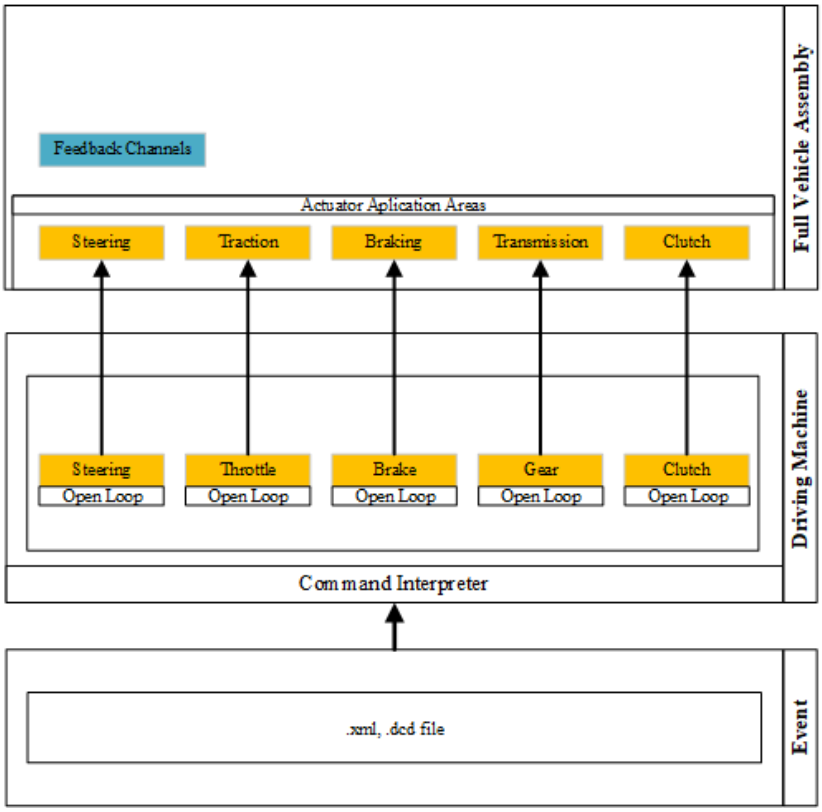

Fig. 8 Pure Open Loop Maneuver

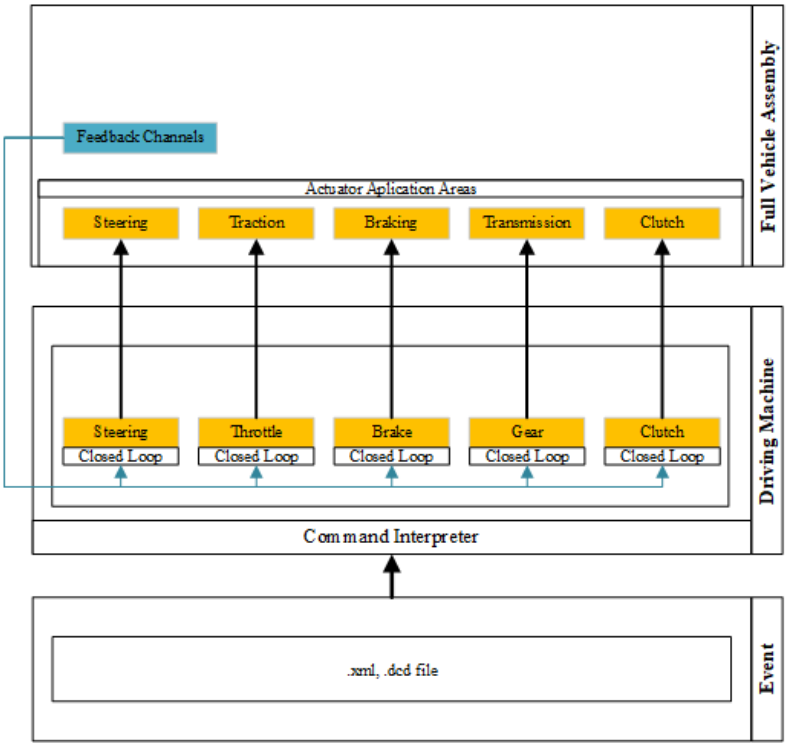

Fig. 9 Pure Closed Loop Maneuver

3 Functional Mock-up Interface

The Functional Mock-up Interface is a free standard that defines a container and an interface to exchange dynamic simulation models using a combination of XML files, binaries and C code, distributed as a ZIP file. It is supported by 250+ tools and maintained as a Modelica Association Project [1].

A Functional Mock-up Unit (FMU) is the executable that implements the interface defined by the Functional Mock-up Interface (FMI) standard.

FMI standard supports two types of co-simulations:

- Model Exchange:
  - The FMU (Functional Mock-up Unit) provides only the model equations (e.g. ODEs, DAEs).
  - The solver is external — it's provided by the master simulator (e.g. Simulink, Dymola, etc.).
  - The master controls Time integration, Step size, State events.
- Co-Simulation:
  - The FMU includes its own solver.
  - Each FMU advances independently in time over given steps.
  - The master only coordinates time steps and data exchange.

MSC Adams can export and use FMU only in Co-Simulation mode, not in Model Exchange and supports FMU v1 and FMU v2.

To incorporate MBD model of vehicle to FMU control subsystem was added to vehicle (Fig. 10).

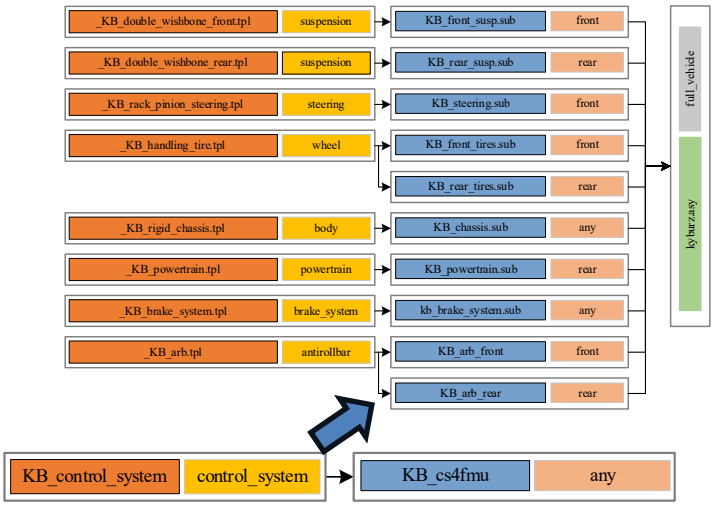

Fig. 10 Control system of vehicle

Control System contains input and output state variables. Control system contains 1 input plant *Vehicle_controls_input*, 2 output control plants *InertFrm_CG*, *BodyFrm_CG*. (Fig.11).

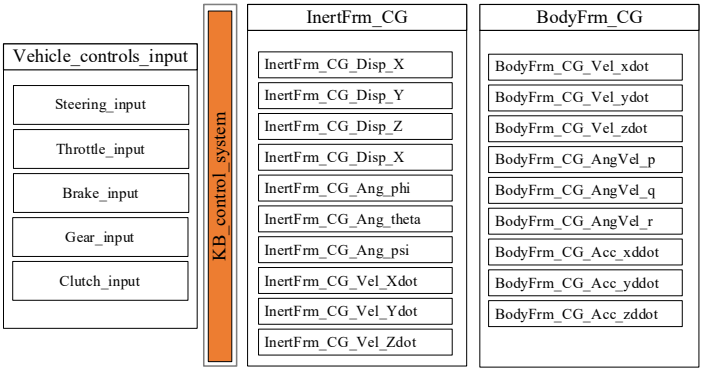

Fig. 11 Input and output state variables

Vehicle_controls_input will be receiving data from Master software (e.g. Simulink), *InertFrm_CG*, *BodyFrm_CG* will be transmitting data to Master software. Control system contains two markers (Fig. 12):

- *EarthFixed_CS_Zdown_ground* (Blue). Attached to ground. Not moving.
- *EarthFixed_CS_Zdown_vehicle* (Green). Attached to vehicle CG. Moving with vehicle.

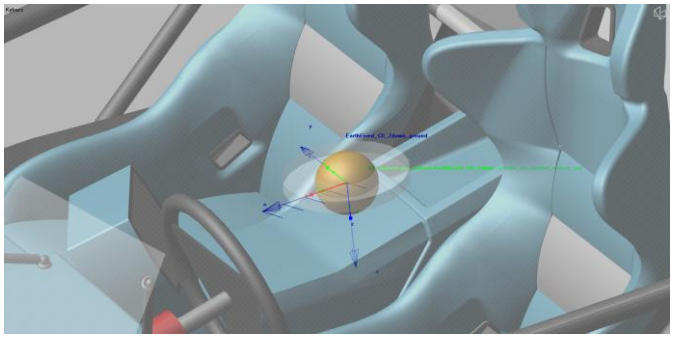

Fig. 12 Markers in control system

For full functionality of the connection and data exchange between the main software (Master) and the FMU vehicle model (Slave), a purely open loop event is used (Fig.7), all five control signals (steering, throttle, brake, gear, and clutch) are defined using Adams function

Varval, with corresponding Adams_id of each state variable used in input control plant (Fig. 10). Block diagram of this process is shown in Fig. 13.

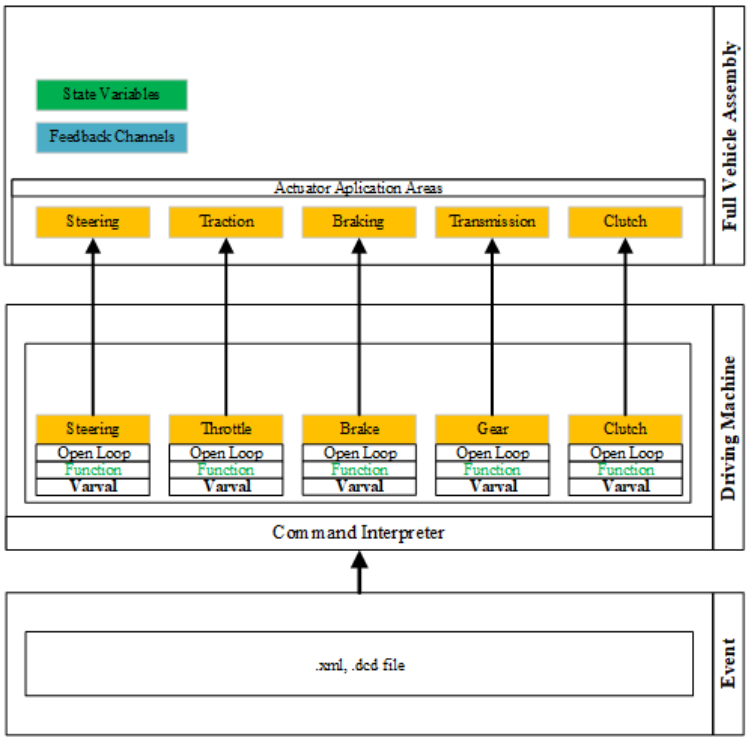

Fig. 13 Block diagram Driving Machine used for FMU

In Adams/Car is possible to create FMU model of vehicle using Adams/Controls plugin. As was mentioned before only valid type of FMI is Co-Simulation, not Model Exchange. Exported FMU from Adam/Car contains full vehicle assembly represented by .adm and .acf file, event file .xml, used road model, tires.

For master software Matlab/Simulink was used. Fig. 14 shows imported FMU of Kyburz into Simulink. Input and output ports of FMU block is treated as Bus. Signals are unitless.

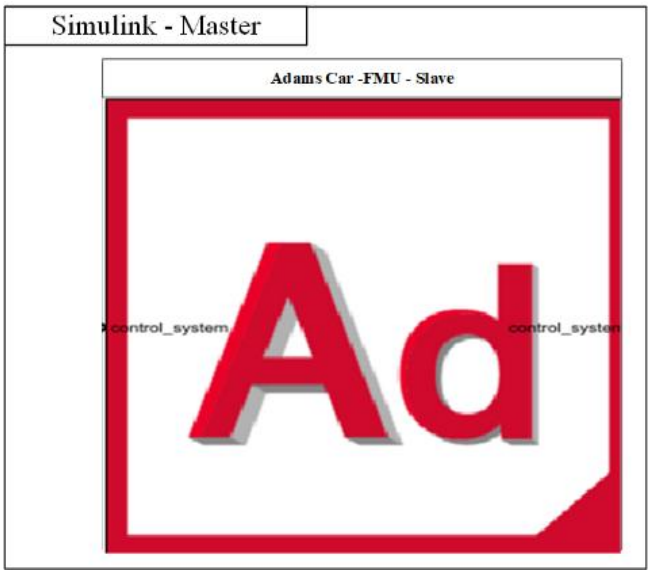

Fig. 14 Imported FMU of vehicle in Simulink

Fig. 15 shows input and output signals of FMU model of Kyburz vehicle presented in article.

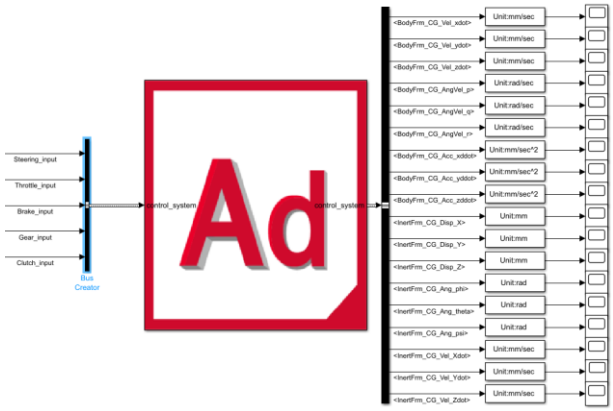

Fig. 15 Input and output signals of Kyburz vehicle model

CONCLUSION

This article successfully demonstrated the methodology for connecting Adams/Car and Matlab/Simulink programs using the Functional Mock-up Interface (FMI) standard, with the Kyburz eRod vehicle serving as a practical case study for virtual vehicle modeling. The integration of a control system with appropriate input and output state variables enabled seamless data exchange between the Adams/Car vehicle model (slave) and Matlab/Simulink environment (master). The FMI standard's tool-independent nature ensures that vehicle models can be shared and utilized across different simulation platforms. This synergistic integration opens new possibilities for advanced automotive applications, particularly in the rapidly evolving field of autonomous vehicles. The Adams/Car model provides highly accurate vehicle dynamics, tire-road interactions, and suspension behaviour, while Simulink offers comprehensive libraries for implementing advanced driver assistance systems (ADAS), autonomous driving algorithms, machine learning models, and sensor fusion techniques. This combination enables researchers and engineers to develop and validate complex vehicle control strategies within a realistic vehicle dynamics environment, bridging the gap between theoretical control algorithms and real-world vehicle behaviour.

ACKNOWLEGEMENT

This research was supported by the Slovak Research and Development Agency under Contracts No. APVV-23-0650, APVV-20-0428. APVV-19-0401, SK-SRB-23-0024, APVV-23-0456, APVV-24-0526.

REFERENCES

[1] Source "Functional Mock-up Interface", [online] Available at: https://fmi-standard.org/ [Accessed: date (22.5.2025)].

[2] Source " eRod - the electric sportscar made in Switzerland | KYBURZ", [online] Available at: https://kyburz-switzerland.ch/en/erod [Accessed: date (22.5.2025)].

[3] Magdolen, Ľ., Danko, J., Milesich, T., Kevický, I., Galinský, M., Bucha, J., Skyrčák, R. "Virtual simulation of overtaking maneuver of autonomous vehicle", Strojnícky časopis – Journal of Mechanical engineering 71 (2), pp. 179 – 188, **2021**.

[4] Adams/Solver reference manual (Version 2025). MSC Software Corporation, Santa Ana, CA, USA, **2025**.

[5] Adams/Car user's guide (Version 2025). MSC Software Corporation, Santa Ana, CA, USA, **2025**.

[6] The MathWorks, Inc., "Simulink Reference," R2025a, The MathWorks, Inc., Natick, MA, USA, **2025**.