

PAPER • OPEN ACCESS

A dynamic model of a pneumobile vehicle

To cite this article: Tamás Szakács 2021 *J. Phys.: Conf. Ser.* **1935** 012016

View the [article online](#) for updates and enhancements.

You may also like

- [Validity of an automated algorithm to identify waking and in-bed wear time in hip-worn accelerometer data collected with a 24 h wear protocol in young adults](#)
Joanne A McVeigh, Elisabeth A H Winkler, Genevieve N Healy et al.
- [Signal quality indices and data fusion for determining clinical acceptability of electrocardiograms](#)
G D Clifford, J Behar, Q Li et al.
- [Automated algorithms for detecting sleep period time using a multi-sensor pattern-recognition activity monitor from 24 h free-living data in older adults](#)
Verónica Cabanas-Sánchez, Sara Higuera-Fresnillo, Miguel Ángel De la Cámara et al.



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

A dynamic model of a pneumobile vehicle

Tamás Szakács

Óbuda Universití

Bánki Donát Faculty of Mechanical and Safety engineering

Institute of mechatronics and vehicle engineering.

szakacs.tamas@uni-obuda.hu

Abstract: Vehicle dynamics modelling is a tool in the hands of design engineers, that saves time and cost during the development phase. By using the model for simulation, a real vehicle and the measurements made on it can be generated; optimizations can be done with automated algorithms, sensors and measurements can be replaced, and the development process is simplified. However, for this to be feasible, it is necessary to develop a suitable model, that describes reality, with sufficient accuracy. The paper presents a model, suitable for running dynamic simulations of a pneumatically driven vehicle [1], and [2].

1. Introduction

The legitimacy of computer simulations has long been unquestioned. Without them, the current level of vehicle technology development would be inconceivable. Computer modeling and the use of models for simulation purposes is a natural, self-evident, tool in all areas of engineering [3]. A vehicle stability program based on a self-developed, validated, modular vehicle combination model [6] was presented. It originally served up driving stability for vehicle combinations, consisting of agricultural tractors and trailers, but also other vehicles such as tactical vehicles, off-road vehicles, pneumatically driven vehicles, Intelligent Driver Model [4], in addition to vehicle and environmental analysis [5] etc.).

During the development of the model, special attention was paid to its flexibility. Easily changeable parameters and a modular structure was used; making it easy to modify and transform from one model to another. A very important aspect is the use of the model in university education, for example in the teaching of vehicle informatics, vehicle dynamics, modeling and simulation subjects. The model does not use any special toolbox, only basic Matlab, and Simulink. The model was created in Matlab/Simulink R2014b.

The aim of this paper is to show the structure and the modular concept of the vehicle model, which in this case has been parametrized to be a pneumobil. Chapter 2 introduces the model as an overall structure, and its details. Chapter 3 shows simulation results, including tire model parameter identification. Chapter 4 draws conclusions and takes a position.

2. Model structure

Figure 1 shows the top layer of the pneumobile model made in a Matlab/Simulink® environment. After loading the model, this interface will appear on the software.

The following set of figures will explain the most important block of this parametric model.



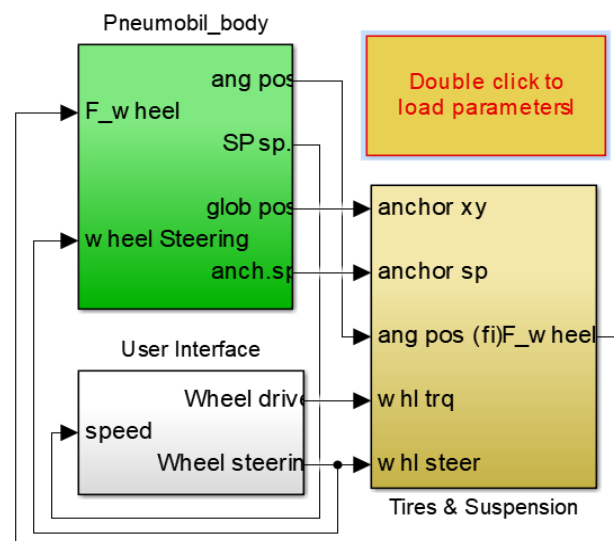


Figure 1. Top layer of the pneumobile model in Matlab/Simulink.

Figure 2 shows the vehicle body part of the vehicle model. This is the model block called Pneumobil body.

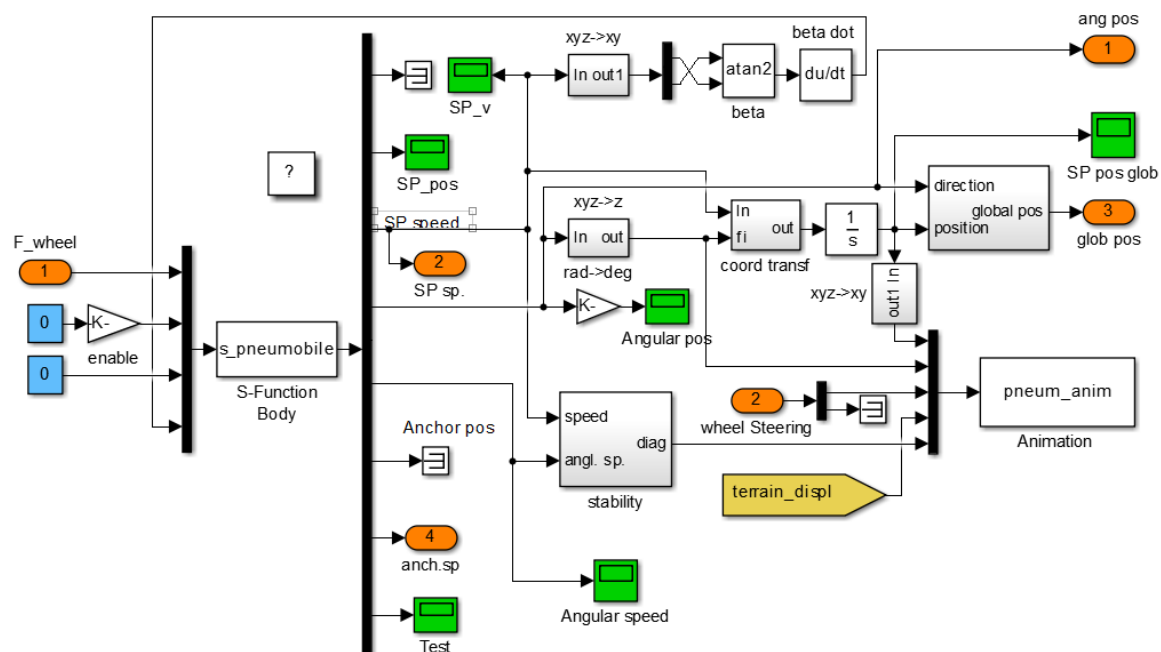


Figure 2. The top layer of the Vehicle Body submodel.

In this submodel, the motion of the vehicle body is determined. The linear and axial accelerations due to the forces acting on the vehicle body (wheel forces, suspension, wheel drive, etc.) are determined by the different submodels to be presented later and are calculated by the block called S-Function. The function calculates the derivatives and outputs, according to state-space theory and is programmed in the function, in Matlab language. The outputs of the block are the mass center and wheel anchor positions, their linear and angular velocity values. This interface includes the handling of motion animation as well as a stability program, that can differentiate between neutral, under-, and oversteer, as described in previous publications.

Figure 3 shows a submodel of the wheels and the suspension model, which becomes visible by opening the lower left block of the first figure called “Tires & suspension”.

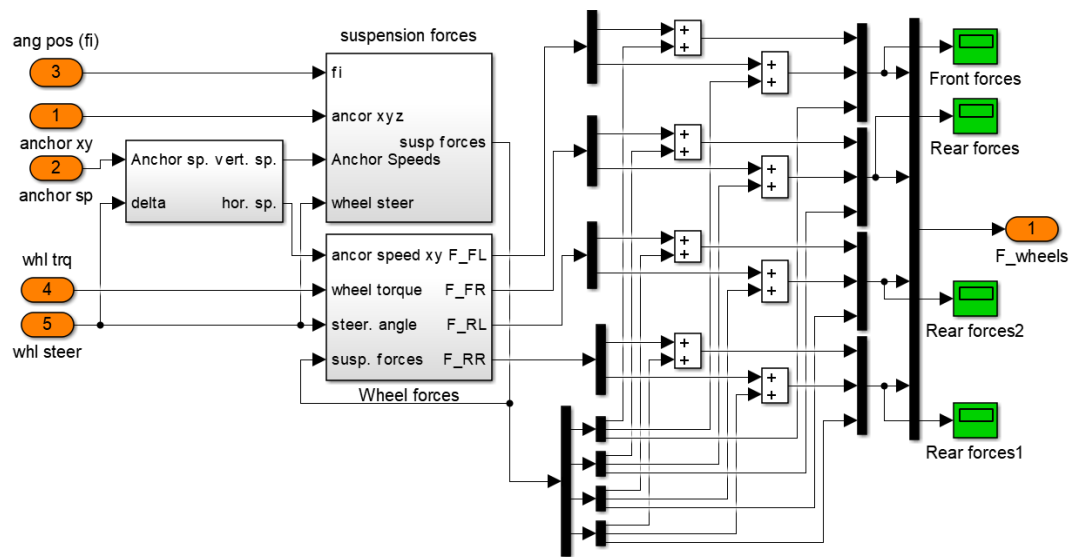


Figure 3. The top layer of the wheel suspension, wheel force submodel.

Interestingly, I also show the figure of the same block, for an eight-wheeled vehicle (Figure 4). The figure shown is a wheel and sub-model of the suspension of a BTR-80 tactical troop transport vehicle.

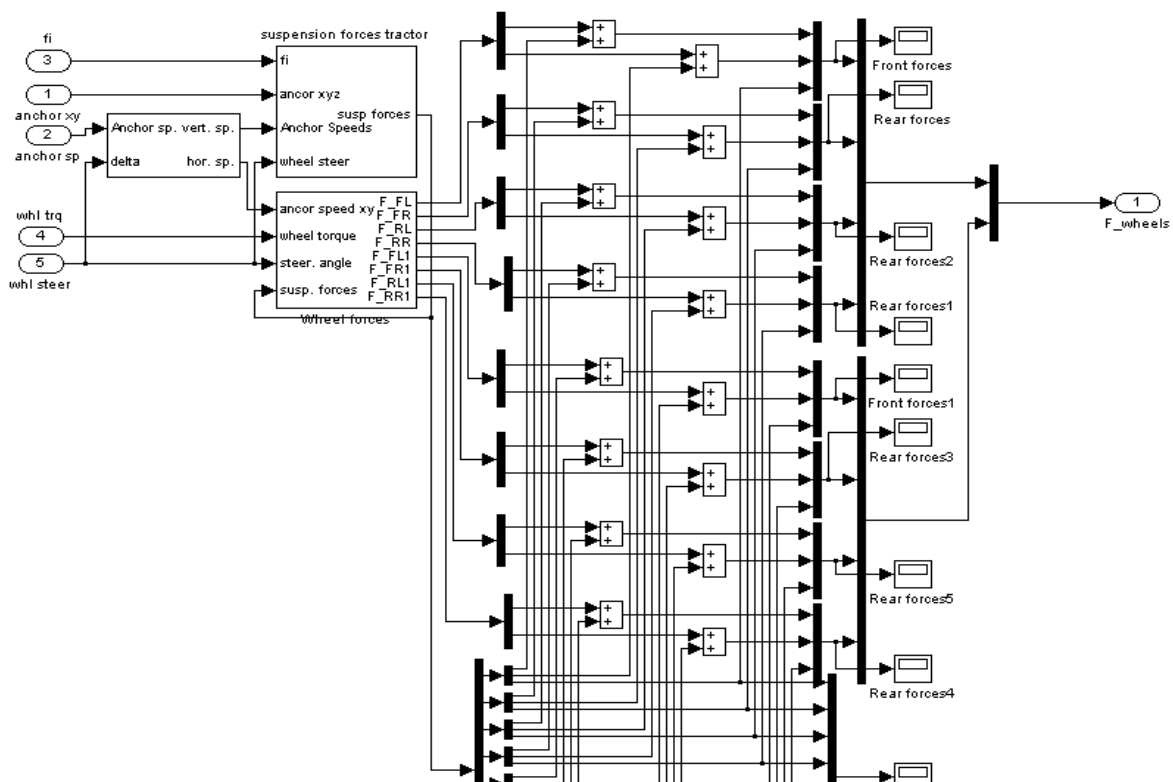


Figure 4. The upper layer of the wheel submodel of the BTR-80, tactical troop transport vehicle.

Figure 5 is part of the wheel suspension, wheel force submodel suspension shown in Figure 3. So, this is the 3rd layer of the model depth. The structure of the model is otherwise 4 layers deep.

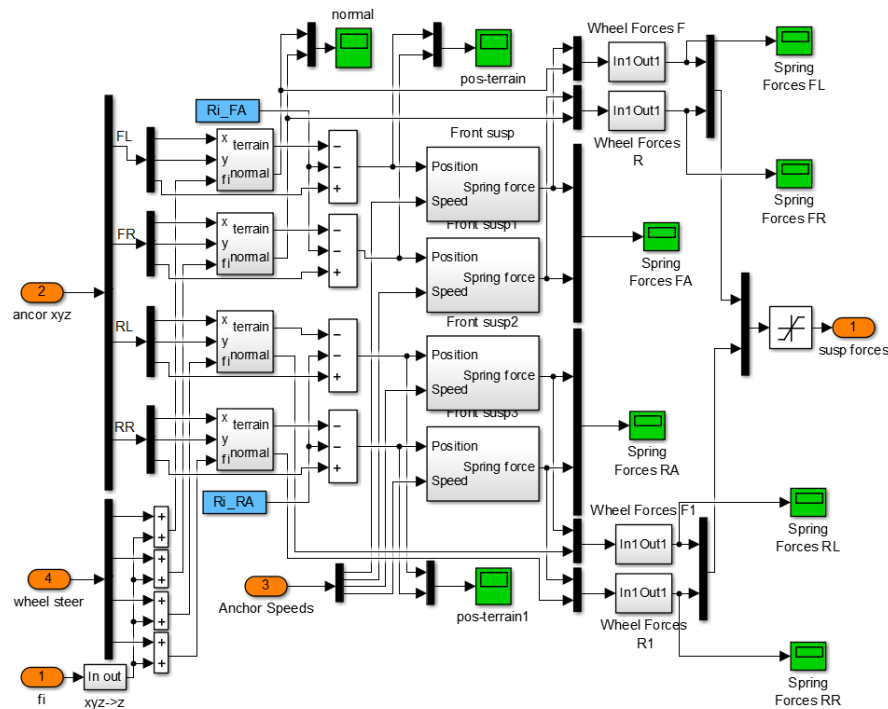


Figure 5. Suspension model.

Figure 6 is a part of the wheel suspension, and wheel force submodel shown in Figure 3. It can be observed that the determination of wheel forces, in the case of driven wheels, is also suitable for performing the ASR function. At this point rear wheels can have ASR.

Wheel forces are calculated per wheel by a 4-parameter parametric wheel model, coded also in an S-Function block from wheel load, wheel slip, and normalized side skidding angle values.

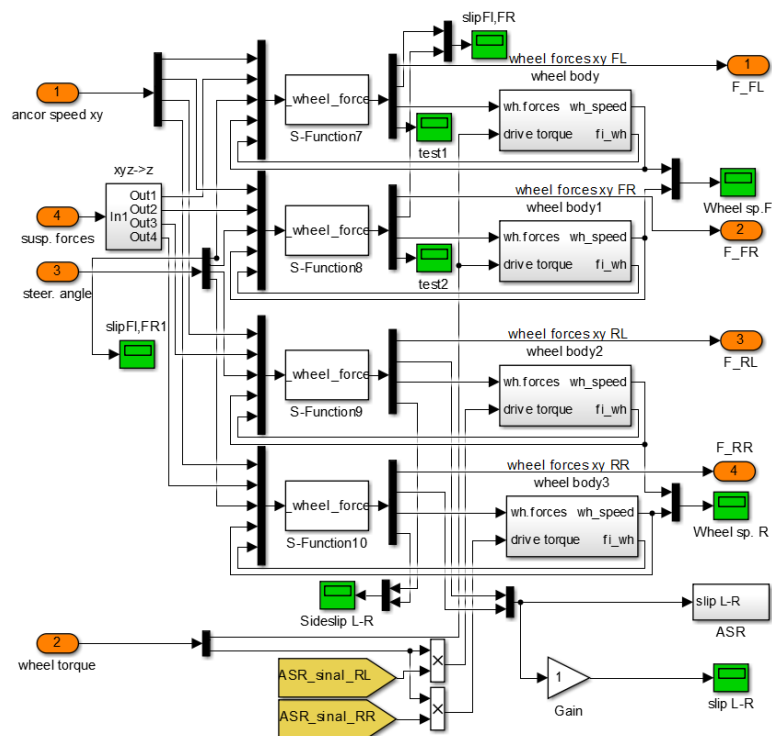


Figure 6. Wheel forces model

The mechanical movement of the wheels (wheel rotation angles, wheel speeds) is determined by the block called “wheel body” shown in Figure 7.

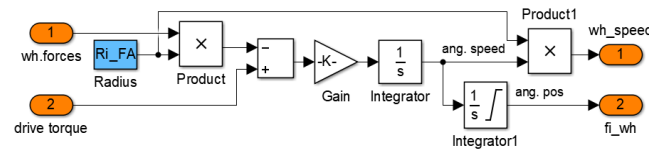


Figure 7. Wheel body model.

Figure 8 shows the ASR function of the model.

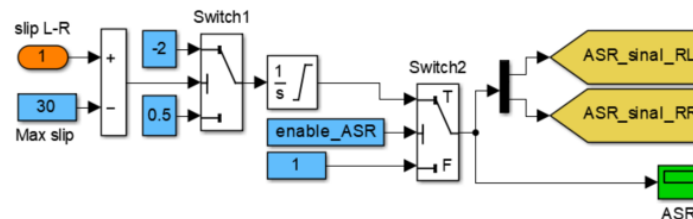


Figure 8. The ASR model

In Figure 9, opening the white block in the top layer, called “User Interface,” reveals the details of the driver model. That model includes the sub-models of wheel drive and steering.

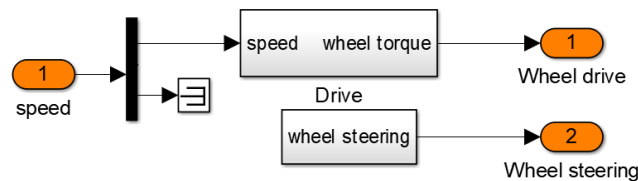


Figure 9. Driver interface

By opening the drive, i.e. the wheel drive block, the ways of driving the driven wheels are shown (Figure 10.) This block has pre-programmed driving scenarios, like constant speed, ramp function, and alternatively, left and right step inputs. A PI controller is included, in order to determine the required driving torque to maintain desired speed.

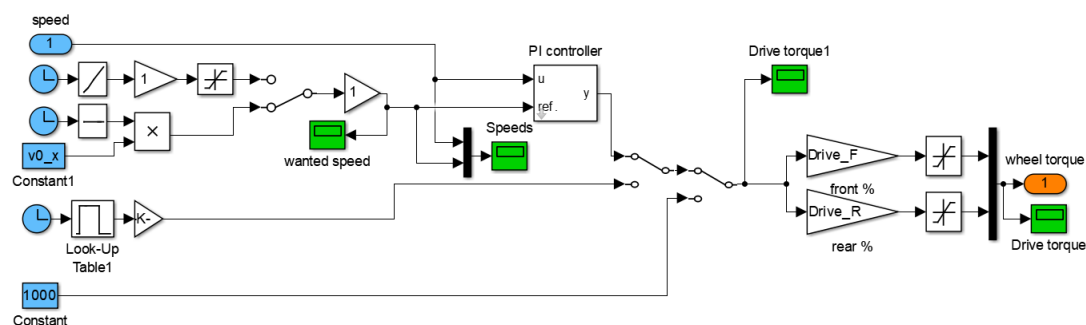


Figure 10. Wheel protrusion model

By opening the lower block of the driver interface shown in Figure 9, the wheel steering block, the submodel in Fig. 11 can be seen. The block contains the calculation of the Ackermann wheel angles for the steered wheels. The Ackermann angular position of the wheels, ensures non-slip rolling of the vehicle's wheels during cornering.

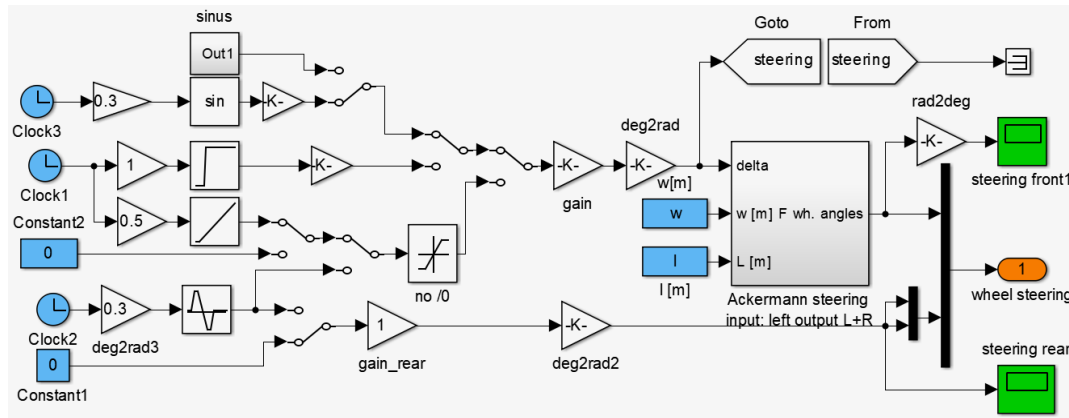


Figure 11. Steering programs

At the top level of the model, double-clicking on the “Double click to load parameters” block, loads the parameters of the vehicle model and then opens the graphical interface (GUI), for managing the parameters; two states of which are shown in Figure 12. The values of the model, simulation, and other parameters are displayed here and can be modified.

The model can also be run from this interface. It is also possible to record the simulation animation from here.

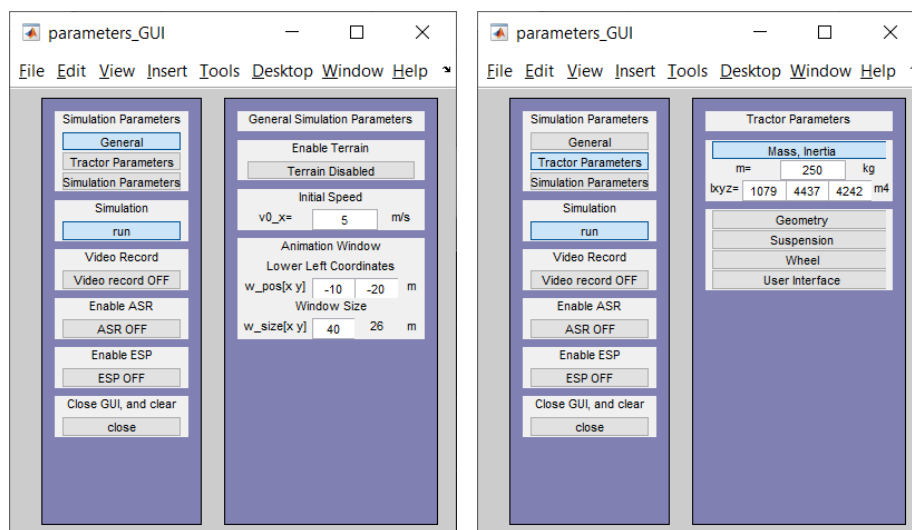


Figure 12. The Graphical User Interface (GUI)

A function appears between the wheel parameters, which can also be used to display and change the parameters of the wheel model. (Figure 13).

Equation of the modified parametric wheel function:

$$F_k = \frac{A \cdot F_z \cdot s}{B + C \cdot s \cdot D} \quad (1)$$

where:

F_k the longitudinal, and lateral wheel forces

F_z a wheel load,

s slip (wheel slip, or in case of lateral forces, normalized wheel side skidding angle)

Parameters A-D: their value determined by the quality of the wheel-road surface condition, in both longitudinal and lateral directions.

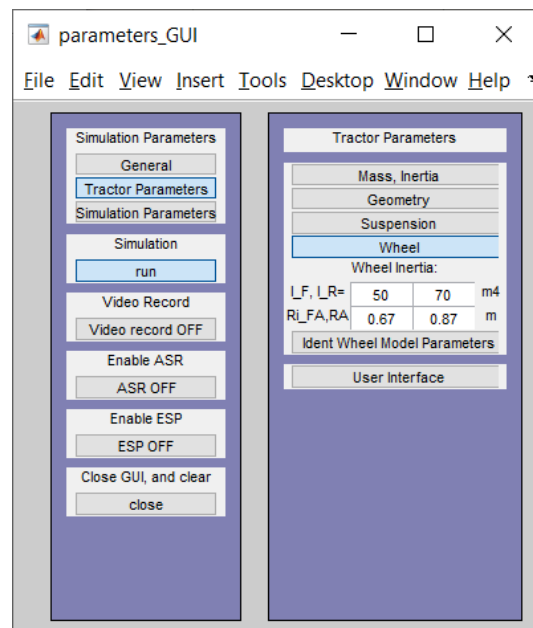


Figure 13. GUI for wheel parameters, parameter identification

3. Results

3.1 Wheel parameters identification

Clicking on “Ident Wheel Model Parameters” on Figure 13, the user interface for displaying and modifying wheel model parameters is displayed. (Figure 14)

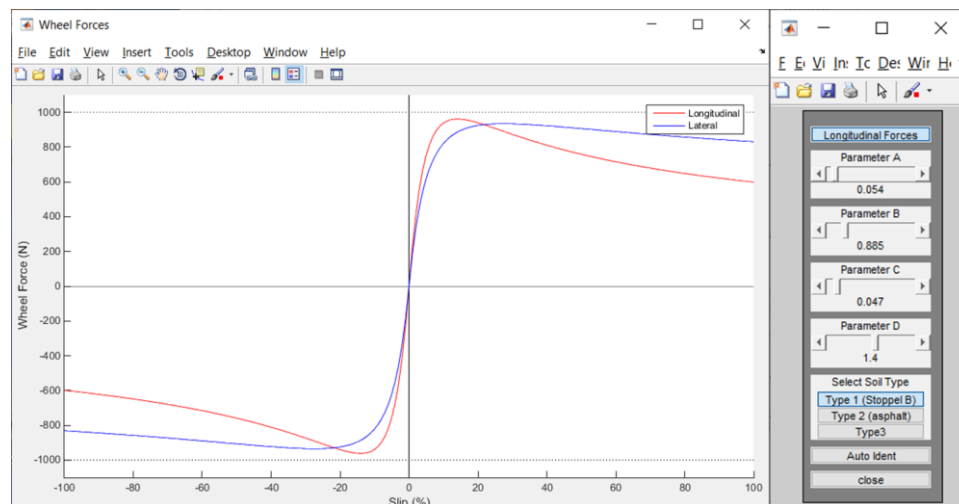


Figure 14. Wheel forces model parameters

The parameters of the wheel model can be adjusted with sliders. Here it is possible to load measurement data and to fit the wheel parameters to the measurement points. (Figure 14.)

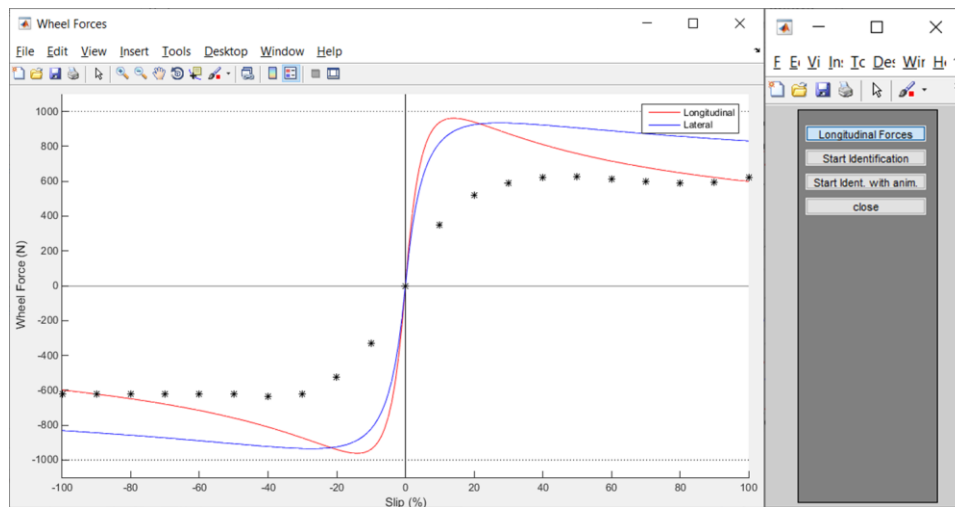


Figure 15. Wheel forces parameters identification

After loading the measurement results, there are two options for identifying the parameters. One is faster and the other is slower, because it also graphically depicts the steps of identification (mainly for educational purposes). This identification means the identification of the parameters of a four-parameter function.

Figure 16 shows the completed identification. Then, by clicking on close, we get back the interface of the parameters of the wheel models (Fig. 14); with the value of the newly identified wheel parameters and the slip-normalized tire force, function curve.

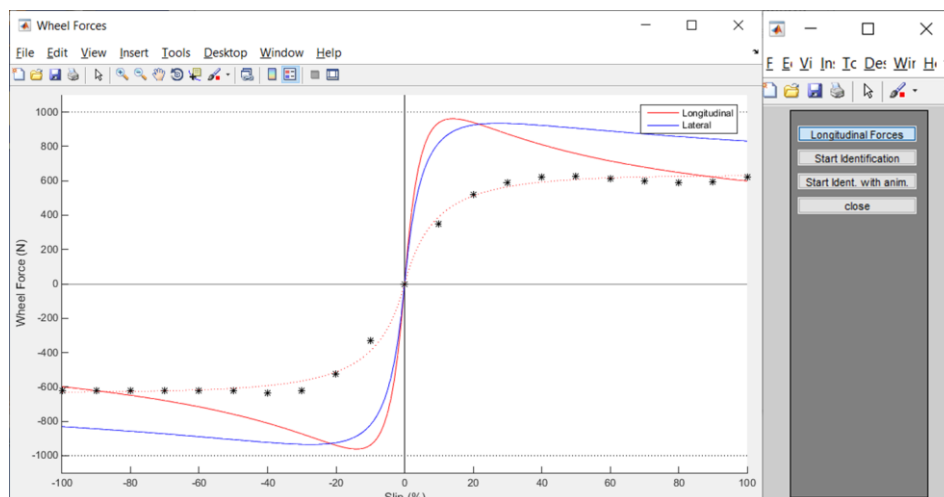


Figure 16. End of wheel forces parameters identification

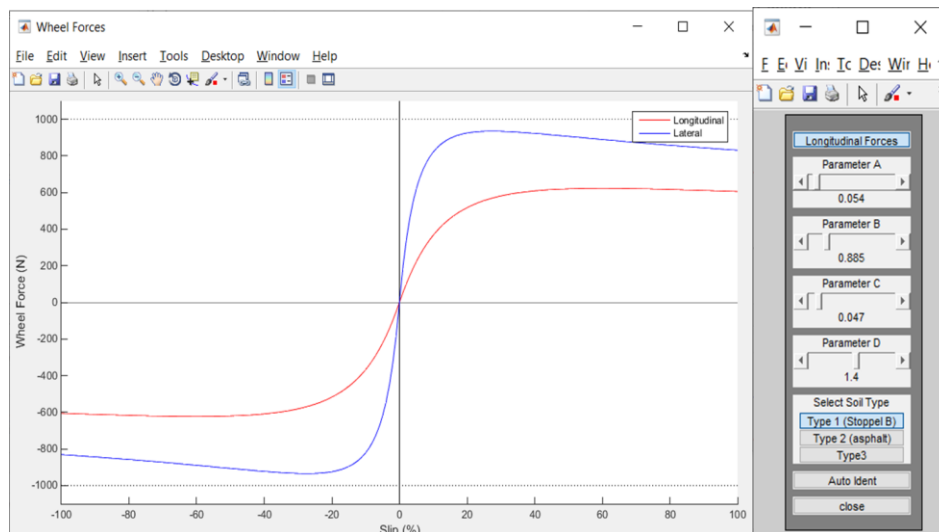


Figure 17. The freshly identified parameters

3.2 Vehicle motion simulation

When having the model run, which is possible from the GUI interface, or traditionally from the Matlab menu, an animation window appears with a vehicle symbol as an animation of the vehicle movement (Figure 18).

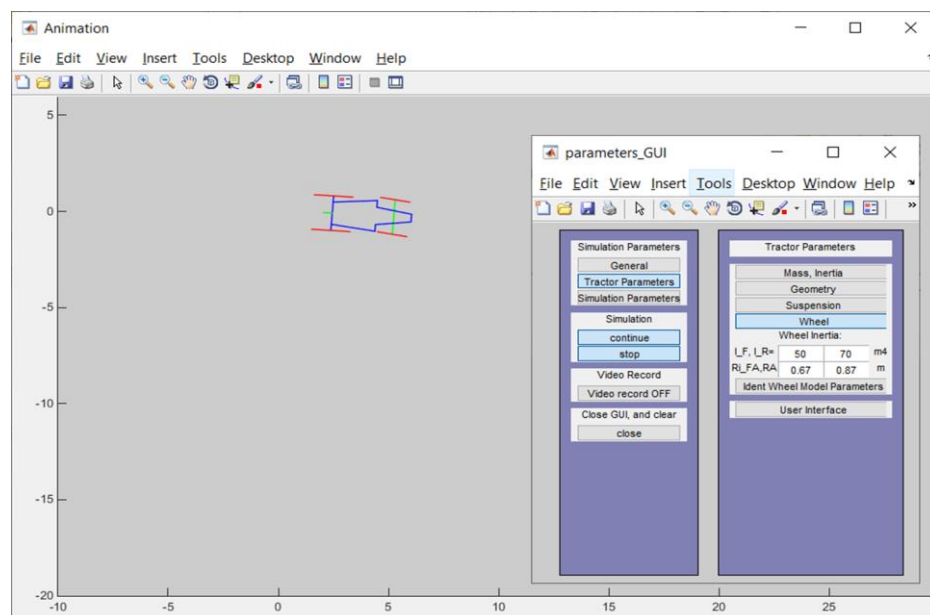


Figure 18. Simulation

After clicking *run* in the GUI, the „run” button will be replaced by a “continue” and an additional “stop” button; in order to continue or stop the model.

The following set of figures show simulation data, which would otherwise be measured on a prototype vehicle. Using computer simulation, this data is gained without time and cost consuming measurements. For simulation input in this example, the wheel steering is set to a sinusoid wave with 63 seconds of period time, as it is shown in the Figure 19. The left wheel angle is controlled by this input signal, the right wheel angle is calculated according to the Ackermann condition, as was shown earlier in Figure 11.

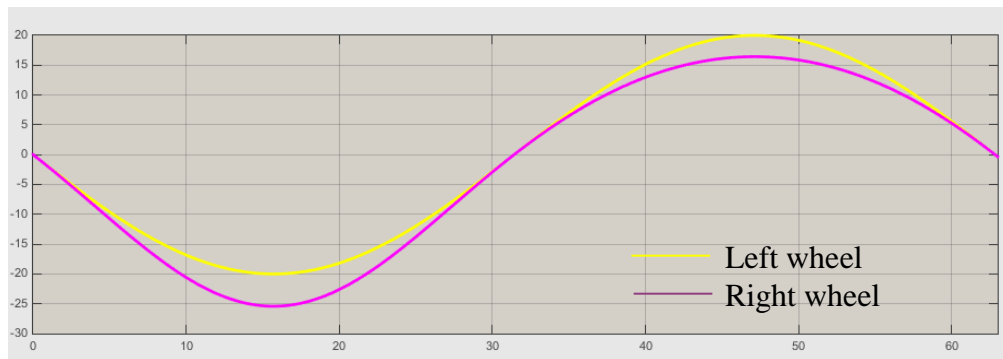


Figure 19. Wheel steering angles left and right (angles)

The vehicle speed is set to a desired 5 m/s. The PI controller on the drive block (Figure 10), controls the speed by affecting wheel drive torque, as it is shown in Figure 20, and later on Figure 26.

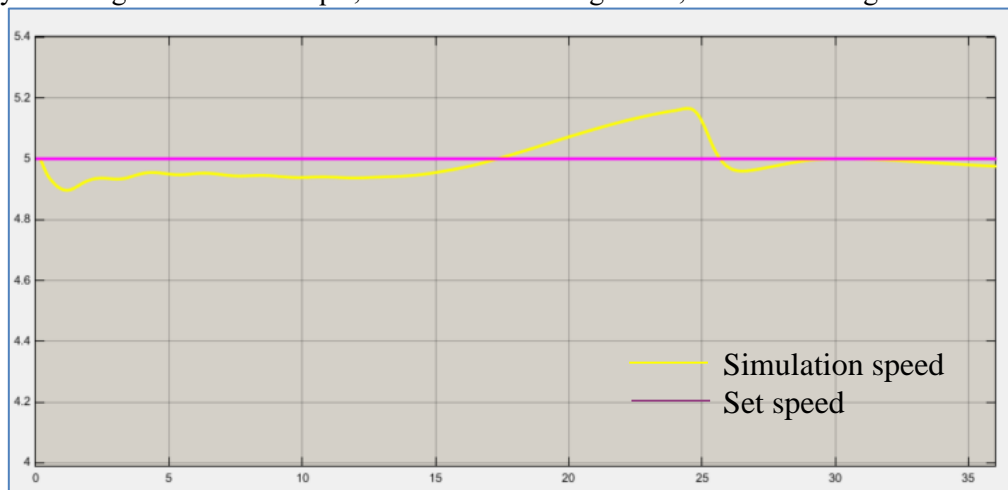


Figure 20. Vehicle speed set, and actual (m/s)

In these conditions let's see the slip curves of the driven wheels: First with ASR off (Figure 21)

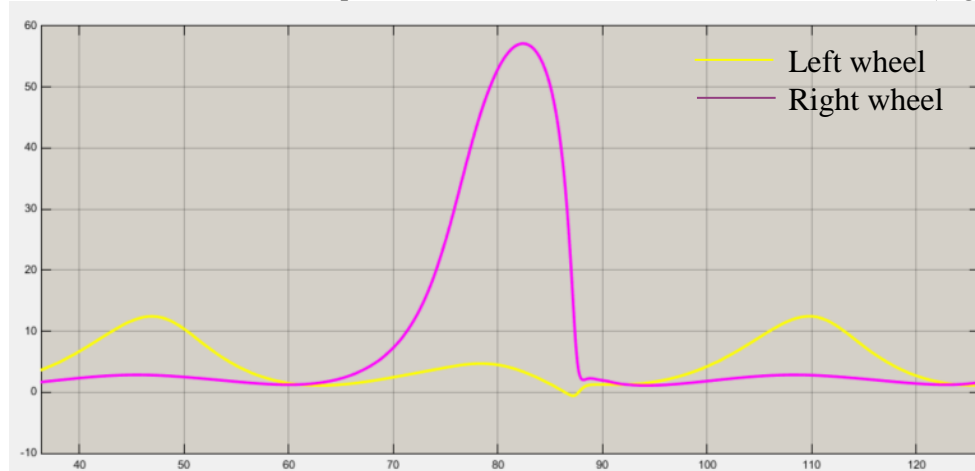


Figure 21. Driven wheel slip left and right. ASR off (%)

This curve has been made to demonstrate extreme slip values. Figure 22 shows the same simulation with the ASR control on. The ASR control is based on wheel drive torque control. In this example ASR is set to limit the wheel slip to 30%. The actual configuration can be seen on Figure 6, and Figure 8.

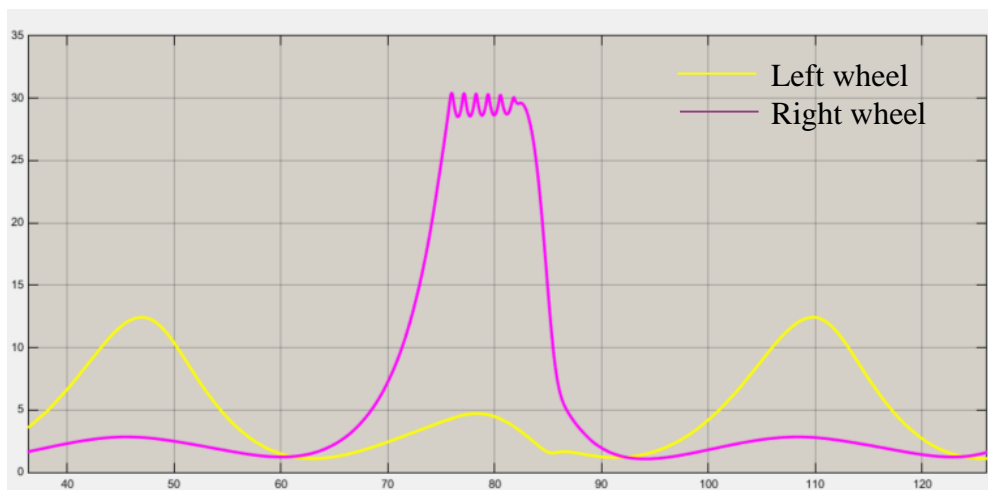


Figure 22. Driven wheel slips left and right. ASR on (%)

The former 60% peak has been reduced to 30%. The wheel speeds during the run are shown in the next two figures: (Figure 23-24)



Figure 23. Front wheel tangential speeds (m/s)

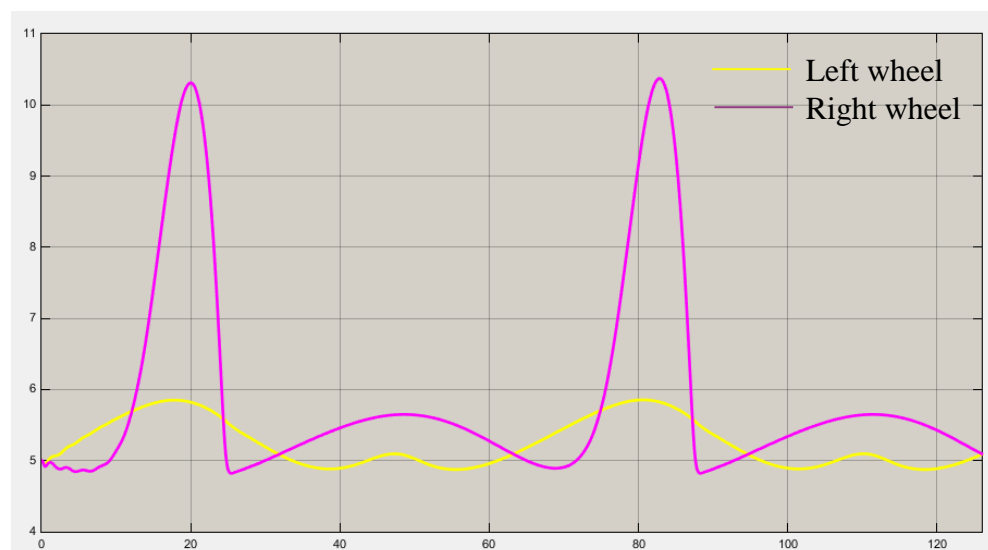


Figure 24. Rear (driven) wheel speed left and right (m/s)

Figure 26 shows the slip values of the front wheels, Negative values are seen, because front wheels are non-driven (pushed) wheels. This negative slip produces tire forces to overcome resistances, in this case mainly roll resistances.

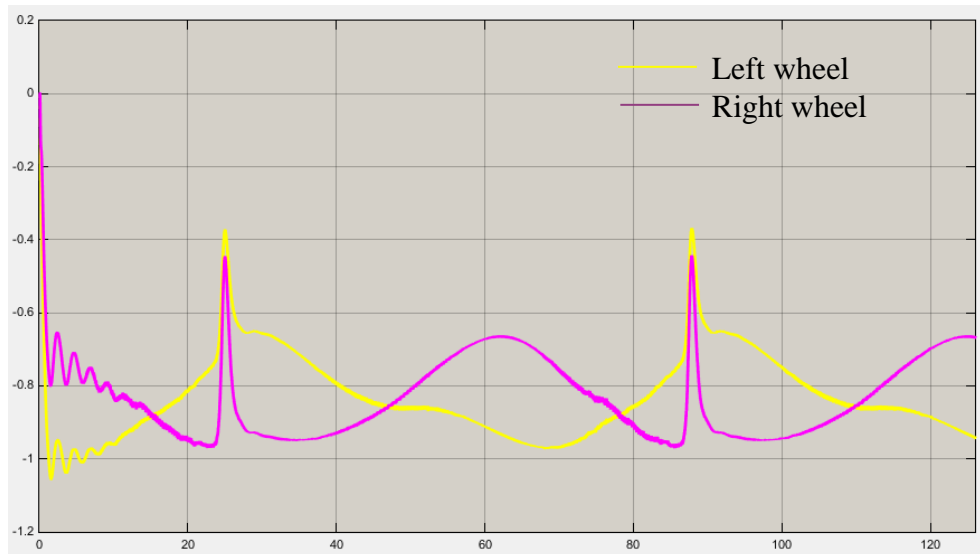


Figure 26. Front (pushed) wheel slips left and right (%)

The last figure shows the driving torque, which maintains the constant set 5m/s speed.

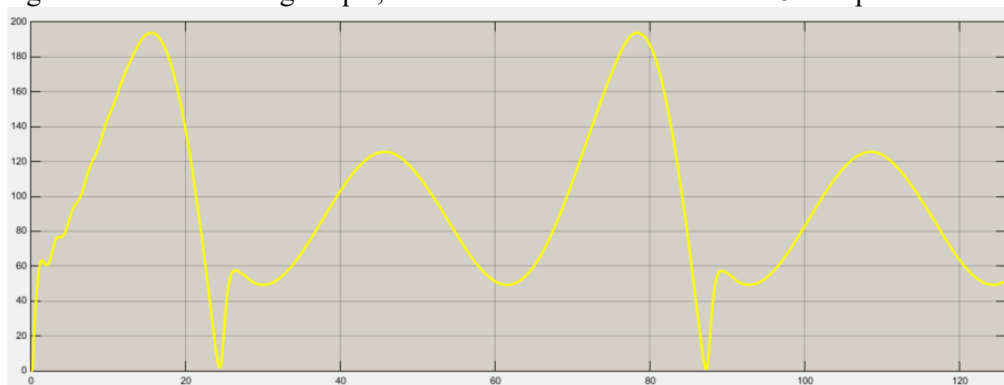


Figure 27. Drive torque.

The set of figures demonstrate how the model can be used for replacing sensor data for research, development, and education. The recorded animations will be presented at the conference, and can be requested via email, from the author.

4. Conclusion

The article presents a complete parameterized vehicle model, which in this case has been defined to model and simulate the vehicle dynamics of a pneumatically driven race car, a so called pneumobile. The aim of modeling, is to perfect the sensor network of an autonomous, Pneumobile, vehicle, which is currently under development; in order to avoid the need for costly and time-consuming measurements at this stage of the research. This model is producing the motion characteristics of the vehicle, i.e. virtual sensor values of wheel speeds, vehicle speeds, yaw rate, driving torque, as well as longitudinal and lateral wheel slip values.

Besides that, the model can be used to combine the pneumatic system of a previously developed pneumatic engine model, giving it a realistic vehicle model-load.

In addition, the model can be used to develop the 2021 Pneumobile racing vehicle, as well as for university education.

The obvious feature of the Matlab/Simulink® program is that after running the simulation, it is possible to display all of the model parameters and save them in the usual way, (not presented in the article). It is also possible to develop and control the autonomous motion control of a vehicle.

5. Acknowledgment

The research was supported by the project EFOP-3.6.2-16-2017-00016: “Dynamics and control of autonomous vehicles in the synergy of the requirements of automated transport systems” with the support of the Hungarian State and the European Union, co-financed by the European Social Fund.

References

- [1] Szakács T.:Modelling and Validation of a Pneumobil Proceedings of the 3th Agria Conference on Innovative Pneumatic Vehicles – ACIPV 2019
- [2] Attila Szántó, Sándor Hajdu, Gusztáv Áron Sziki: Dynamic Simulation of a Prototype Race Car Driven by Series Wound DC Motor in Matlab-Simulink. Acta Polytechnica Hungarica, Vol. 17, No. 4, 2020
- [3] Pokorádi, László: Models in Safety Management, MACHINE DESIGN 11: 3 pp. 85-94. , 10 p. (2019)
- [4] Péter, T.; Lakatos, I: Vehicle Dynamic-based Approach for the Optimization of Traffic Parameters of the Intelligent Driver Model (IDM) and for the Support of Autonomous Vehicles’ Driving Ability. Acta Polytechnica Hungarica Vol. 16, No. 3, 2019 DOI: 10.12700/APH.16.3.2019.3.7 ISSN:1785-8860
- [5] Péter, T.; Lakatos, I.; Szauter, F.; Pup, D.: Complex analysis of vehicle and environment dynamics MESA 2016 - 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications - Conference Proceedings7 October 2016, Article number 758711212th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, MESA 2016; The University of AucklandAuckland; New Zealand; 29 August 2016 through 31 August 2016; Category numberCFP16MES-ART; Code 124211
- [6] Szakács T.: Vehicle Train Simulation Model for Developing Stability Control System. Proceeding ISTVS: New Developments in Off-Road Vehicles and Machinery Budapest 2006.10.3-6 ISBN 963 06 08324.