Number 2, Volume IX, July 2014

SIMULINK BLOCK LIBRARY FOR ASSEMBLING A VEHICLE

Pavel Kučera[1], Václav Píštěk[2]

Summary: The article deals with the description of block library creation for compiling computational models of commercial vehicles. These blocks are created using the basic elements of software Simulink and represent the main powertrain parts for the simulation of longitudinal vehicle dynamics. The article describes the design and function of each block, and demonstrates the powertrain assembly of commercial vehicles with 6x6 drive. In conclusion, a simulation of starting and stopping is done to verify the functionality of the assembled model.

Key words: Simulink, Library, Engine, Clutch, Transmission, Shaft, Differential, Tire Model

INTRODUCTION

There are several types of software on the market that enable simulation of vehicle powertrain. However, in most cases these blocks with an internal structure are not accessible to the user. As a result a problem occurs if the user wants to have added functionality or more extensive computational models.

For this reason, this article describes the creation of a library of blocks containing computational models of basic powertrain elements with the internal structure accessible to the user. A library is created from these blocks, which allows the user to build different vehicles driven by one to three axles and simulate their longitudinal dynamics.

Simulink software is used and its basic elements contained in the library are applied to create computational models. The article will describe the principle of function of information flow in the individual created blocks and the assembly. In the second half of the article, the demonstration of the vehicle assembly and simulation of its start is described. Computational models created in Simulink software can be simulated on the hardware for real-time testing, which is important for further use of the library described in this article.

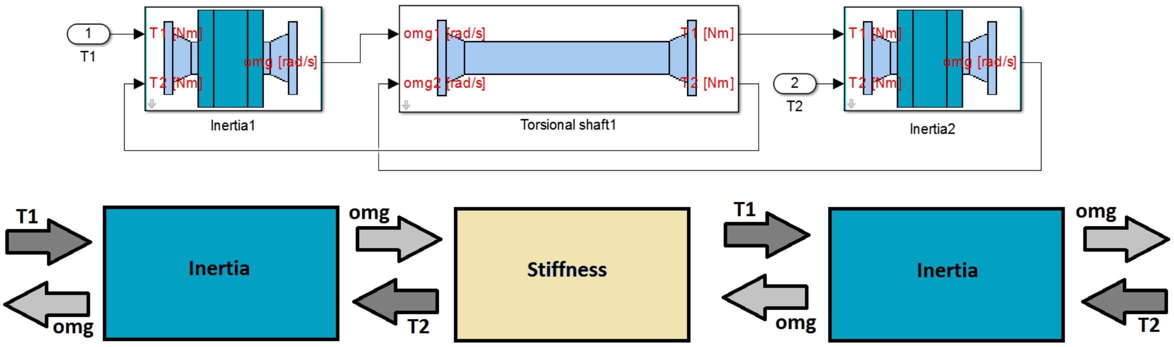
1. PRINCIPLE OF OPERATION

The library contains blocks that can be easily connected to each other and create powertrain according to user requirements. The principle is that the blocks placed behind affect the state at the beginning of the computational model and at the end, too. An example is shown in Fig. 1, where there are two blocks representing the moment of inertia and one block

[1] Ing. Pavel Kučera, Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automotive Engineering, Technická 2896/2 , 616 69 Brno, Tel.: +420 541 142 252, Fax: +420 541 143 354, E-mail: kucera@iae.fme.vutbr.cz

[2] Prof. Ing. Václav Píštěk, DrSc., Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automotive Engineering, Technická 2896/2 , 616 69 Brno, Tel.: +420 541 142 271, Fax: +420 541 143 354, E-mail: pistek.v@fme.vutbr.cz

representing the torsion shaft. Their internal structure is described in the following chapter. If we supply the torque T1 to the first block, the system will simulate the shaft rotation and influence the blocks on the right. When the torque action T2 applies on the right block, the system of blocks located on the left is affected. In this way, any arbitrary torsion assembly can be created, which can be influenced from each block of the assembly because the information is transmitted between blocks in both directions. This principle is adopted also for other blocks described in the next chapter.

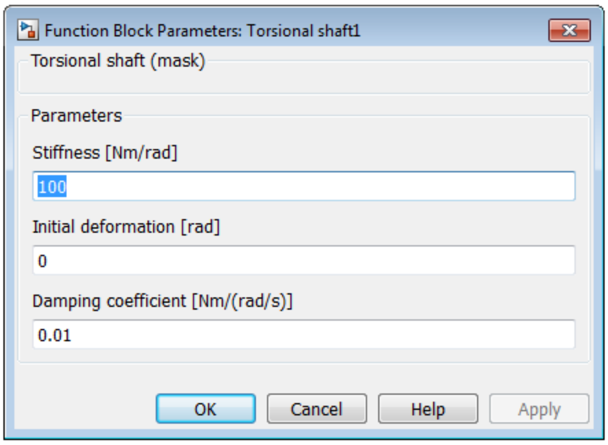


Source: Authors

Fig. 1 – Block diagram of the information flow between the blocks

2. MAIN BLOCKS OF LIBRARY

Using Simulink software and available literature for the creation of computational models (1), (2) blocks were created and have subsequently been adopted to compile the library for creating commercial vehicle powertrain. The library contains 19 blocks with computational models describing the features of engine, clutch, transmission system, differential, drive shaft, rotating discs, tires and longitudinal dynamics of the vehicle. In following chapters, these blocks will be described individually. Each of the blocks has generated a header, where the user enters the input parameters, as shown in Fig. 2.


Function Block Parameters: Torsional shaft1

Torsional shaft (mask)

Parameters

Stiffness [Nm/rad]

100

Initial deformation [rad]

0

Damping coefficient [Nm/(rad/s)]

0.01

Source: Authors

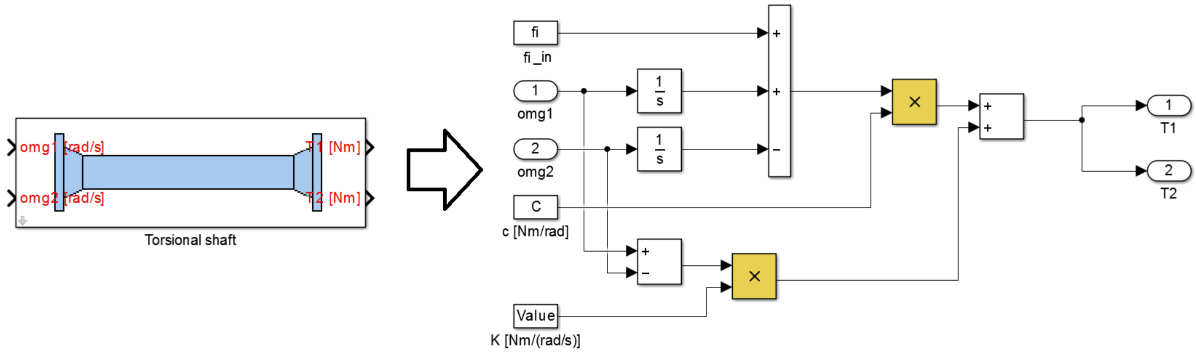
Fig. 2 – The user interface for entering input parameter of the block

2.1 *Shaft block*

The shaft block is a basic block of the whole library. The computational model of this block represents the shaft torsional stiffness and its damping. The internal structure of the block is shown in Fig. 3. Block function is based on the equation

$$T = c(\varphi_i + \varphi_1 - \varphi_2) + k(\omega_1 - \omega_2) \tag{1}$$

where T is the torque, c - shaft torsional stiffness, k - internal shaft damping, φ - rotation, ω - angular velocity. This results in setting the input parameters entered by user which are shaft torsional stiffness and internal shaft damping.



Source: Authors

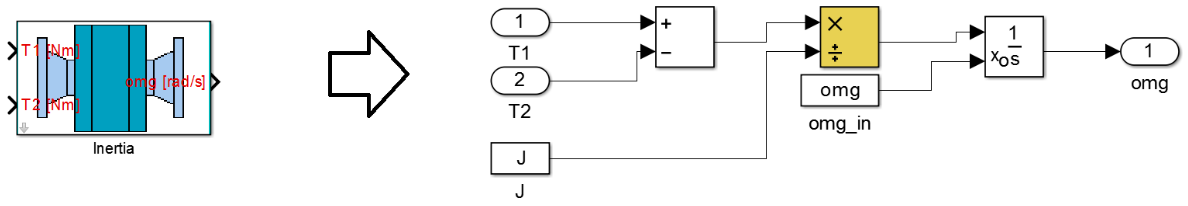
Fig. 3 – Shaft block

2.2 *Inertia block*

Inertia block is the second basic block of the library. The computational model and its structure are shown in Fig. 4. Principle of the block function is based on the equation

$$\omega = \int \frac{(T_1 - T_2)}{J} \tag{2}$$

where ω is the angular velocity, $T_{1,\ 2}$ - torques generated before and behind this block and J - inertia of a rotating mass. These two basic blocks are further extended to more complex computational models of the engine, the transmission system, the tires, etc.

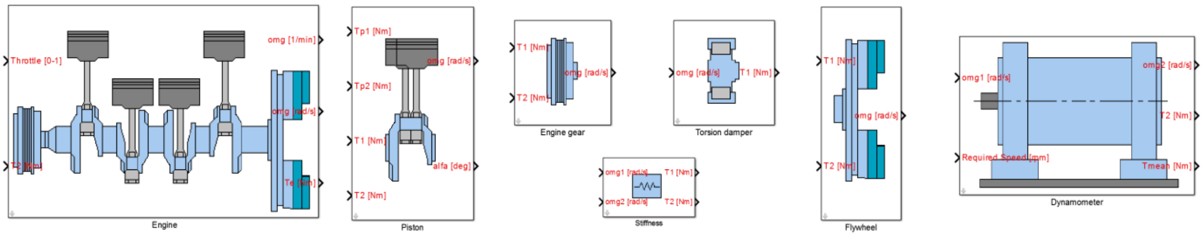


Source: Authors

Fig. 4 – Inertia block

2.3 *Engine block*

The library contains several blocks representing basic motor functions. These blocks are shown in Fig. 5. The first block (Engine) represents the external characteristics of the engine. The user enters the torque curve at the maximum and minimum fuel supply, inertia of the crank mechanism, maximum speed, idle speed and viscous friction. The second block (Piston) in the Figure represents the piston system where the user connects the torque map and enters its input parameters (moments of inertia, idle speed and viscous friction). Compared to the previous engine, there is a simulated waveform torque of each engine working cycle. The principle of the model is described in more detail in the article (6). Any engine can be built using the Piston blocks, Stiffness block and adding blocks that represent the torsional damper and flywheel. The last block represents the computational model of dynamometer, which can be applied to virtually test user's own engine configuration.

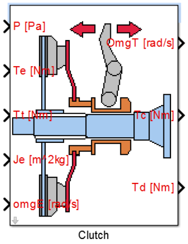


Source: Authors

Fig. 5 – Engine blocks

2.4 Clutch block

Block shown in Fig. 6 contains a clutch computational model. The computational model based on the theory of Karnopp allows simulations without jumps (5) during switching on and off the clutch. The user can select one of two options of calculation method. The first option is the calculation for even clutch wear and the second calculation method is for even pressure distribution on the clutch friction lining. The equations are described in the literature (4).

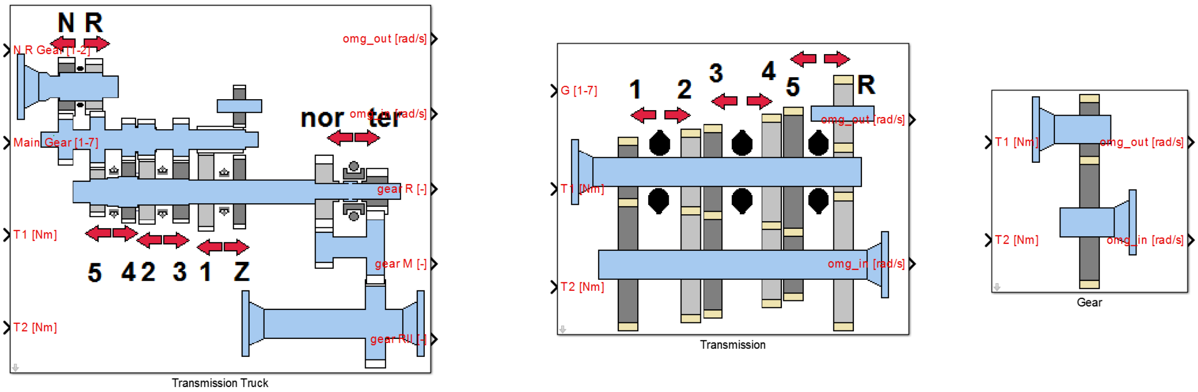


Source: Authors

Fig. 6 – Clutch block

2.5 Transmission block

The library contains blocks transmissions shown in Fig. 7. The first block represents transmission with seven forward gears, one reverse, and with the choice of normal and reduced gear. A more detailed description of the function of the gearbox is in article (7); there is a version without synchrony. The second block represents five forward gears and reverse. The third block contains the simplest version of the gearbox with one gear. User can enter ratios, inertia of each shaft with gear wheels, direction and efficiency of the transmission in these blocks.



Source: Authors

Fig. 7 – Transmission blocks

2.6 *Differential block*

The blocks shown in Fig. 8 represent the differential of the vehicle. Computational model is based on the following equations for the torque acting on the input and output of the differential and the differential angular velocity of the individual shafts.

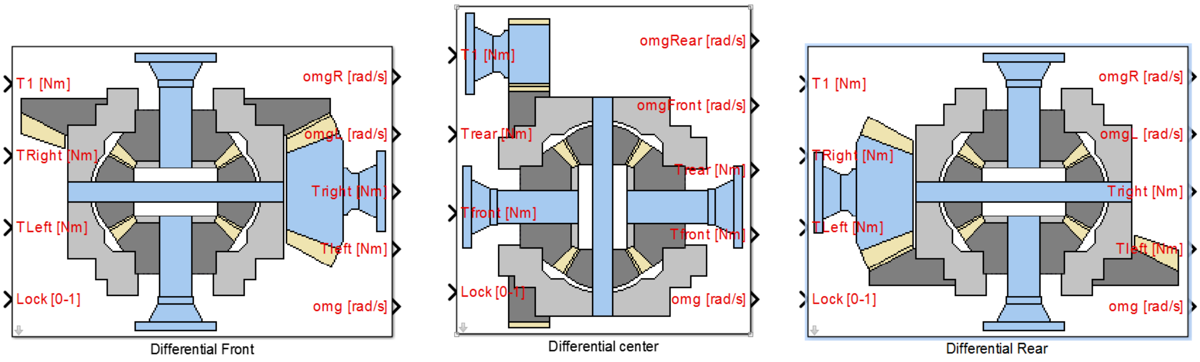
Torque is described by the equation,

$$T_{in} = T_l + T_r \tag{3}$$

where T_{in} is the input torque, T_l - left output torque and T_r - right output torque. The angular velocity is described by the following equation,

$$\omega_{in} = \frac{\omega_l + \omega_r}{2} \tag{4}$$

where ω_{in} is the angular velocity in the input of differential, ω_l - angular velocity of the left output shaft and ω_r - angular velocity of the right output shaft. Input block parameters are the moments of inertia, gear ratio and gear efficiency. The design of the computational model allows the user to control its closure in case of a slip of one of the wheels.

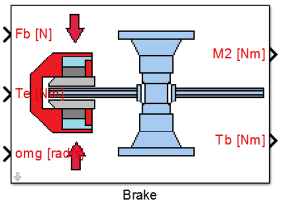


Source: Authors

Fig. 8 – Differential block

2.7 *Brake block*

Brake block is designed so that braking torque acts against the rotation of the attached wheel output shaft. This torque results from contact force on a disc. At the same time, the state when the vehicle stops is controlled, and in this case, the braking torque drops to zero, because the vehicle would move in the opposite direction. The user controls the contact force on the disc and may specify the input parameters for the disc brake.

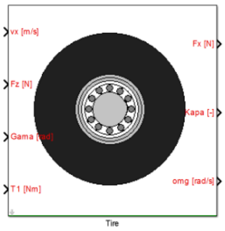


Source: Authors

Fig. 9 – Brake block

2.8 Tire block

The computational model of a tire describes the formation of longitudinal force between the tire and the road. This model is based on Pacejka tire model 'Magic formula' and it uses a single equation of longitudinal forces and tire transient behaviour described in the literature (3), (8). The equations are rather extensive, so they will not be described here. The input parameters are the coefficients for longitudinal force, scale factors, tire radius, wheel inertia and the user can choose whether the tire is on the right or left side. The output is mainly longitudinal force, which is the input to the block of longitudinal dynamics.



Source: Authors

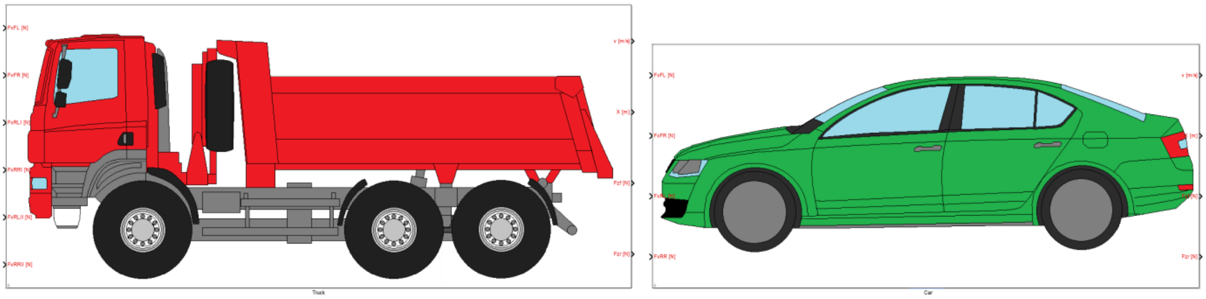
Fig. 10 – Tire block

2.9 Longitudinal dynamics block

There are two blocks for the vehicle longitudinal dynamics and these may be applied for a vehicle with one to three drive axles. The block is assembled using the longitudinal dynamics equation

$$a_x m_v = \sum F_x \qquad (5)$$

where a_x is the vehicle acceleration, m - the vehicle weight and Fx - are the forces acting in the longitudinal direction. These forces include the wheel driving forces and the forces driving resistance. The input parameters are vehicle weight, the centre of gravity, the coefficient for driving resistance, initial velocity, the angle of the slope and the number of wheels on each axle. The output of this block is the vehicle speed and the load of each tire in the vertical direction which is an important input parameter for the block of the tire.

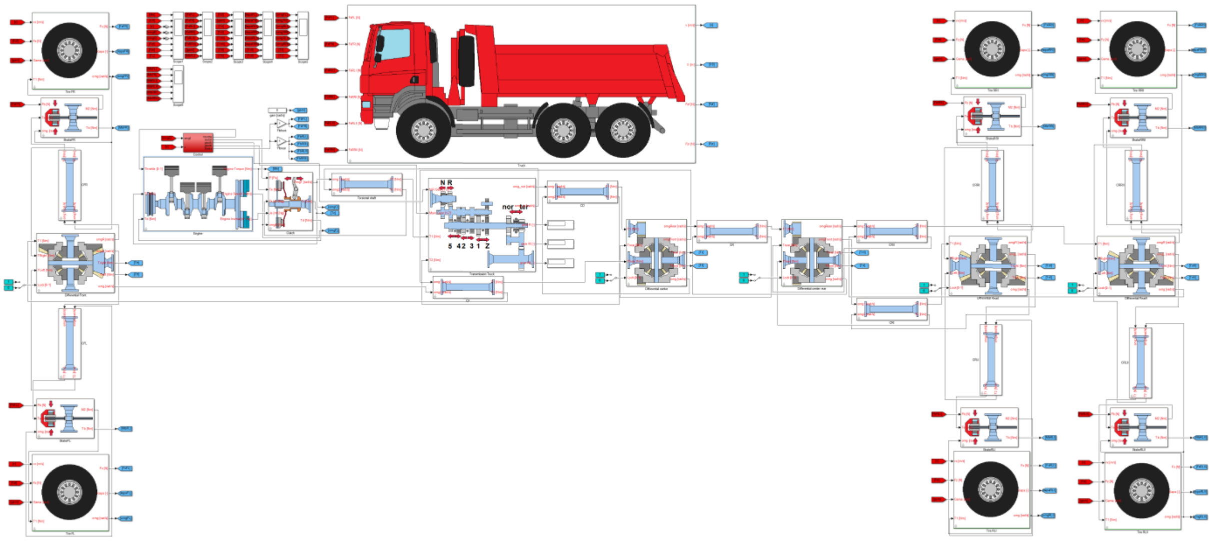


Source: Authors

Fig. 11 - Longitudinal dynamics blocks

3. COMPUTATIONAL MODEL OF COMMERCIAL VEHICLES WITH 6X6 DRIVE

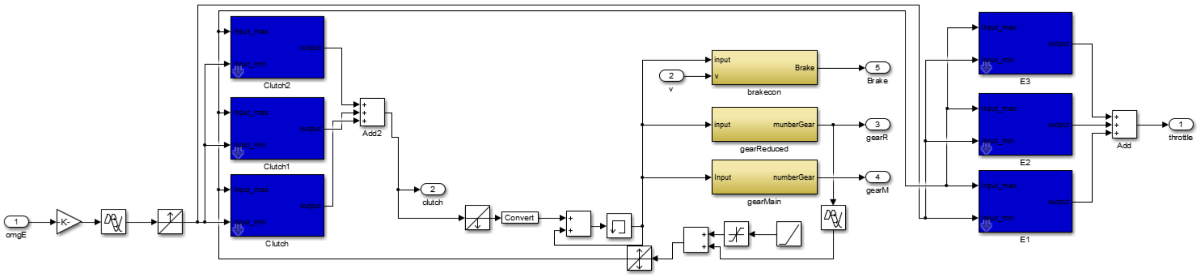
To illustrate the function of each block, an overall computational model for the powertrain vehicle 6x6 was assembled. The computational model consists of engine, clutch, gearbox, five differentials, disc brakes for individual wheels, tire blocks, the longitudinal dynamics block, and a control block of simulation. The weight of the vehicle is 41,000 kg. Basic engine parameters are maximum torque of 2100 Nm at 1100rpm and the engine power is 325kW at 1800rpm. Used tires are marked 315/80R22.5.



Source: Authors

Fig. 12 - Computational model of commercial vehicles

As mentioned above, the model includes a control block with an algorithm for the simulation specified in the following chapter. Internal diagram is shown in Fig. 13. The algorithm controls the position of gas, clutch, each gear shifting and braking.



Source: Authors

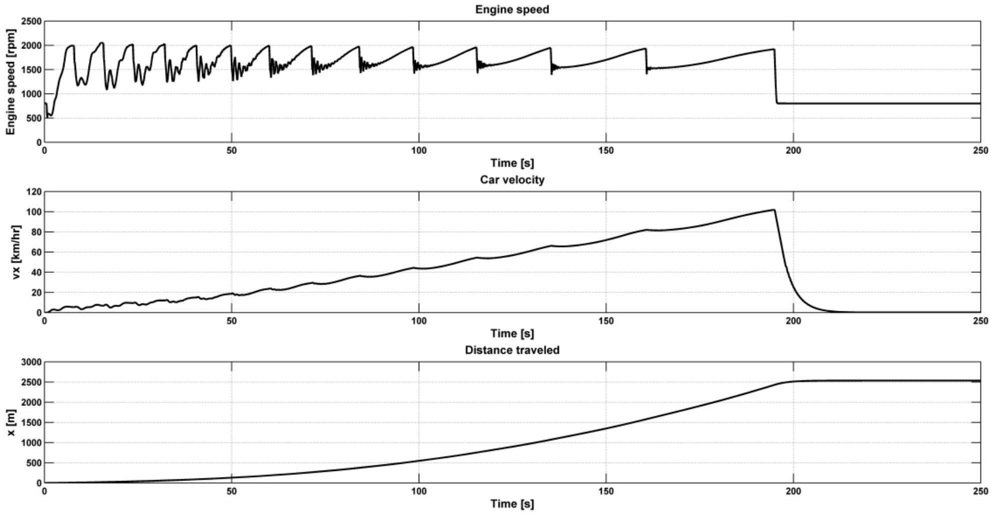
Fig. 13 - Computational model of commercial vehicles

4. SIMULATION BY COMPUTATIONAL MODEL

To verify the correct function of the computational model of the assembled powertrain, simulation of starting and stopping of the vehicle was performed. The simulation was set up with a fixed step and the calculation used ode14x solver that combines Newton and extrapolation methods. The simulation is performed so that the vehicle is stationary in the resting position, the clutch is not transmitting torque and transmission is set to the lowest gear. After a set time, the control algorithm sends a signal to the clutch, which transmits the torque and the vehicle starts to move. Shifting is started after reaching the set engine speed.

After overcoming this situation, the control algorithm sends a signal to the engine, clutch and transmission. This cycle is going on up to run out of possible gears. In a state where the computational model simulates driving in top gear and engine speed reaches the required limits, the control algorithm disconnects the clutch, throttle position sets the minimum and sends a signal to the brakes. At this moment, the vehicle starts to slow down until it stops for a period of time. The course of this simulation is shown in Fig. 13 where drops in the engine speed are clearly visible when shifting each gear.

There is an oscillate shaft, too, as is clearly seen in the engine speed at a lower vehicle speed. Therefore it would be appropriate to adjust the control algorithm for shift and include synchrony into the transmission block in order to continuously change speed in other parts of the drive train. The time dependence of the vehicle speed is shown in the middle of the picture. The distance traveled by the vehicle is shown below in the picture. It can be concluded from these graphs that the computational model can be considered as functional.



Source: Authors

Fig. 14 - Computational model of commercial vehicles

CONCLUSION

The blocks contained in the library for the assembly of powertrain vehicles were created in Simulink software. The aim was to create such blocks that enable the user to assemble any part or the whole drive vehicle with drive of one to three axles.

The first chapter describes the basic principles of the computational model consisting of created blocks. The following extensive part explains individual library blocks, especially their functions, settings and the equations they are assembled from. The third chapter describes the demonstration example assembled from these blocks where the powertrain vehicle with 6x6 drive was assembled. This computational model can simulate the start of the vehicle, shifting, vehicle deceleration and stopping. The fourth chapter describes the simulation conducted for these states. The results were compared with theoretical assumptions and they suggest that the assembled computational model works properly. For accurate verification, the obtained data have to be compared with measurements on a real vehicle. This model also allows to tune the function of powertrain by change of tires, gear ratios, shaft stiffness, moment of inertia and to perform other various simulations, too.

The advantage of this library is that the individual blocks can be further extended and modified unlike special libraries included in Simulink software oriented to powertrain. Furthermore, this model can easily be compiled into the appropriate language and format used in real-time testing and development of mechatronic systems.

ACKNOWLEDGEMENT

This work is an output of research and scientific activities of NETME Centre, regional R&D centre built with the financial support from the Operational Programme Research and Development for Innovations within the project NETME Centre (New Technologies for Mechanical Engineering), Reg. No. CZ.1.05/2.1.00/01.0002 and, in the follow-up sustainability stage, supported through NETME CENTRE PLUS (LO1202) by financial means from the Ministry of Education, Youth and Sports under the ”National Sustainability Programme I“.

REFERENCES

(1) DABNEY, James B. and Thomas L. HARMAN. *Mastering Simulink*. Upper Saddle River: Pearson Prentice Hall, 2004. 376 p. ISBN 0-13-142477-7.

(2) GREPL, Robert. *Modelování mechatronických systémů v Matlab SimMechanics*. 1st ed. Praha: BEN, 2007. 151 p. ISBN 978-80-7300-226-8.

(3) PACEJKA, Hans B. *Tyre and vehicle dynamics*. 2nd ed. Oxford: Butterworth-Heinemann, 2006, 642 p. ISBN 07-506-6918-7.

(4) BUDYNAS, Richard G. and J. Keith NISBETT. *Shigley's Mechanical Engineering Design*. 9st ed. United States of America: The McGraw Hill Companies, 2006. 1059 p. ISBN 0−390−76487−6

(5) KARNOPP, Dean. Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems. *Journal of Dynamic Systems, Measurement, and Control* [online]. 1985, vol. 107, issue 1, p. 100- [cit. 2014-04-28]. DOI: 10.1115/1.3140698. Available from: http://DynamicSystems.asmedigitalcollection.asme.org/article.aspx?articleid=1403660

(6) KUČERA, Pavel and Václav PÍŠTĚK. Virtual Diesel Engine in Simulink. *Perner's Contacts* [online]. 2013, vol. VIII, no. 2, p. 95-105 [cit. 2013-10-14]. Available from: http://pernerscontacts.upce.cz/30_2013/Kucera.pdf

(7) KUČERA, Pavel and Václav PÍŠTĚK. Transmission Computational Model in Simulink. Perner's Contacts [online]. 2013, vol. VIII, no. 4, p. 37-48 [cit. 2014-4-15]. Available from: http://pernerscontacts.upce.cz/33_2013/Kucera.pdf

(8) Mscsoftware: Adams/Tire help - MSC Adams 2011. *Tire_msc2011.pdf* [online]. 2011 [cit. 2014-04-28]. Available from: http://simcompanion.mscsoftware.com/infocenter/index?page=content&id=DOC9837&cat=1VMO50&actp=LIST