

VIRTUAL DIESEL ENGINE IN SIMULINK

Pavel Kučera¹, Václav Pištěk²

Summary: The article describes the modelling of a virtual diesel engine. The engine is created using basic blocks from the Simulink library. Its input values were obtained using measurements and modified according to the actual engine configuration. The details of these measurements are not presented in this article. The description of a virtual engine simulation is given and, in the end, a comparison with the values of a real engine is discussed.

Key words: Diesel Engine, Virtual Engine, Simulink, Torsion Model

INTRODUCTION

Nowadays, the demand for the development of virtual applications is continuously rising in order to avoid as many problems associated with the first prototype testing as possible. This also helps to accelerate the development and subsequently reduces the costs of running the actual prototype. The support for the development of these applications varies between different fields of use, aided significantly by pre-prepared default models for numerous applications. In some cases, only modifications of the input parameters are necessary for their successful use. And if a need for broader adjustments arises, an integrated library with a wide range of basic elements, from which our desired computational model can be built, is available.

This article presents a virtual diesel engine create using the basic elements contained in the library of the Simulink software package. The model, based on a real eight-cylinder diesel engine, will be used to drive virtual powertrains in further simulations. The aim was to create a model of the engine capable of representing the real engine in a virtual environment with sufficient accuracy. Therefore, flexible elements for the simulation of torsional vibrations were also incorporated.

The Simulink software environment was used to create the model. This environment is able to serve for developing custom models and running various simulations, such as real-time simulations if appropriate hardware equipment is available. In that situation, it is even possible to run an assembly of multiple models representing the whole vehicle.

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 Pištěk, 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

1. THE INPUT DATA FOR A VIRTUAL ENGINE

1.1 Engine torque calculation

One of the inputs for the model is a curve of pressure indicated in the engine cylinder for speeds ranging from 800 rpm to 2000 rpm, from which the engine torque curve was calculated in accordance with the literature (4). Fundamental engine parameters were used for the calculation, i.e. the cylinder diameter of 120mm, the engine stroke of 140 mm and the length of the connecting rod of 260 mm. From these input values, forces acting on the crank mechanism as shown in Fig. 1 could be calculated.

The force resulting from the pressure in the cylinder:

$$F_{p_i} = S_p (p_{ind_i} - p_{atm}) \quad (1)$$

where S_p is the area of the piston, p_{ind} the indicated cylinder pressure, p_{atm} the atmospheric pressure and i the i -th computational step.

As the inertial forces of moving parts are also included in the calculation, it was necessary to know the reduced overall mass of relevant components. This mass is included in the following equation for the inertial forces of movable parts:

$$F_{sp_i} = -m_p R_{crank} \omega_m^2 (\cos(\alpha_i) + \lambda \cos(2\alpha_i)) \quad (2)$$

where m_p is the reduced overall mass of movable engine components, R_{crank} the radius of the crankshaft, λ the crankshaft ratio and α the crank angle.

The force acting in the axial direction of the cylinder:

$$F_{cp_i} = F_{p_i} + F_{sp_i} \quad (3)$$

The force acting on the connecting rod axis:

$$F_{o_i} = \frac{F_{cp_i}}{\cos(\beta_i)} \quad (4)$$

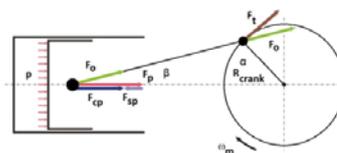
where β is the connecting rod angle.

The force acting tangentially on the crank throw:

$$F_{t_i} = F_{o_i} \sin(\alpha_i + \beta_i) \quad (5)$$

The engine torque:

$$M_{R_i} = R_{crank} F_{t_i} \quad (6)$$



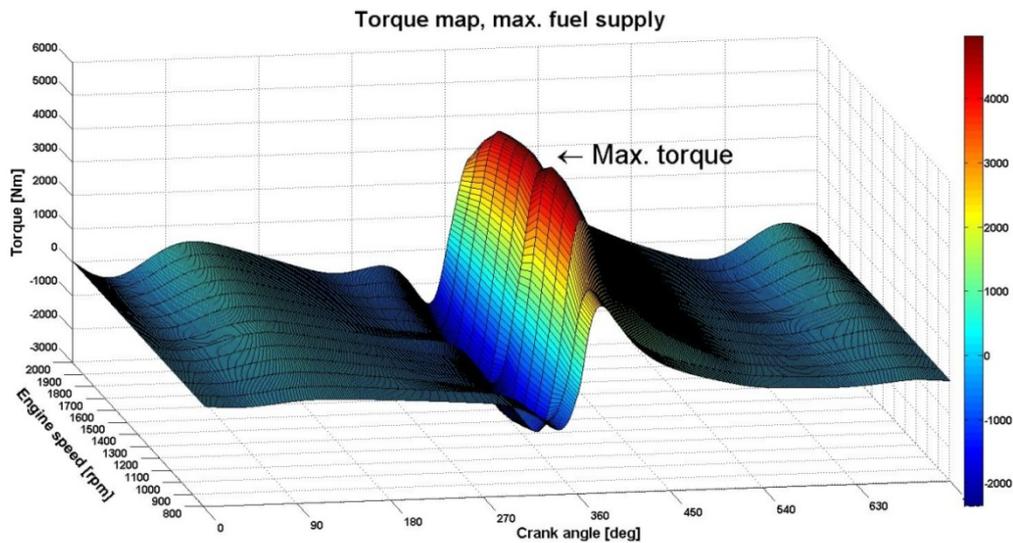
Source: Author

Fig. 1 – The forces in the crank mechanism

1.2 Torque maps

To obtain the engine torque maps, a script which solves the torque values for a given engine speed from the indicated cylinder pressure according to the equations in chapter 1.1 was written using the Matlab software package. Individual values are sorted in the appropriate elements of three-dimensional matrices with the dimensions of 13x720x2 for each cylinder. A three-dimensional matrix is composed of two two-dimensional arrays which contain the

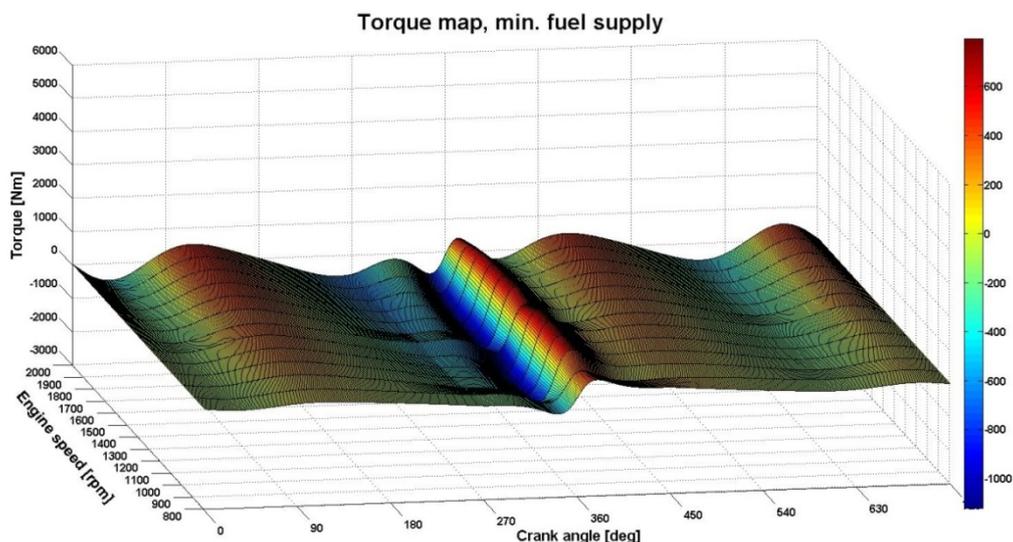
values of torque for maximum and minimum fuel supply respectively. The 13 matrix rows represent the engine speed range from 800 rpm to 2000 rpm with a step of 100 rpm.



Source: Author

Fig. 2 – The torque map for maximum engine load and maximum fuel supply

The 720 matrix columns stand for the change of the torque after a 1 degree step in the crank angle. It therefore makes up for a one whole four-stroke engine cycle (two rotations of the crank) which is divided into 720 steps.

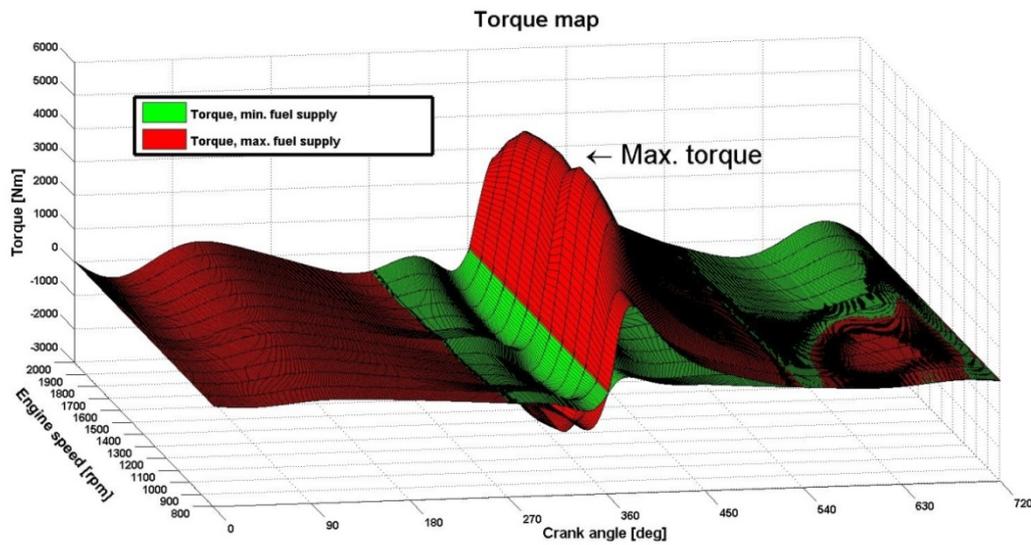


Source: Author

Fig. 3 – The torque map for engine braking and minimum fuel supply

These two two-dimensional matrices are concatenated, which results in a three-dimensional matrix. The third dimension has a value of 2 which corresponds to maximum and minimum fuel supply. The matrices are then individually shifted so that they are in phase with the corresponding cylinder ignitions. Fig. 2 shows the torque map for a maximum load and a maximum fuel supply with the torque maximum located between engine speeds from 1100

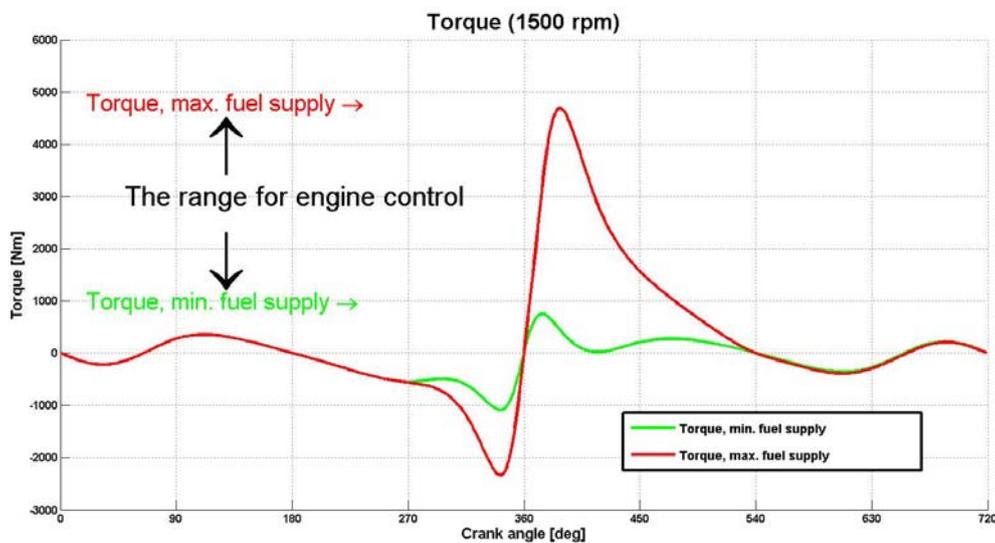
rpm to 1200 rpm highlighted. The torque map for a minimum fuel supply and engine braking is shown in Fig. 3. A comparison of the two maps is then given in Fig. 4.



Source: Author

Fig. 4 – The comparison of two torque maps

Between the torque curves' extremes, a range for the model's torque output control emerges for certain crank angles. This range is illustrated in Fig. 5



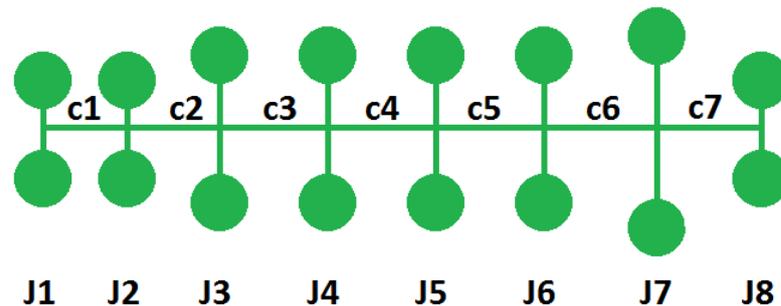
Source: Author

Fig. 5 – The range for engine control

The above mentioned matrices are used as an input to the engine model and the torque values for a given engine speed, the crankshaft position and the accelerator pedal position are loaded from them in each simulation step.

1.3 The torsional vibrations of the engine

The computational model includes flexible elements in order to simulate torsional vibrations. This vibration simulation was based on the torsional system shown in Fig. 6.



Source: Author

Fig. 6 – Dynamic crank mechanism model

The torsional model is composed of seven reduced engine inertia values and six stiffness blocks between discs. A disc with the J8 inertia represents a dynamometer, while c7 is the stiffness of a shaft connecting the engine to the dynamometer. To compare the torsional model with the virtual engine, general Lagrange's equations in their simplified form which assumes undamped vibrations without external forces are used in accordance with the following equation:

$$M\ddot{\varphi} + C\dot{\varphi} = 0 \quad (7)$$

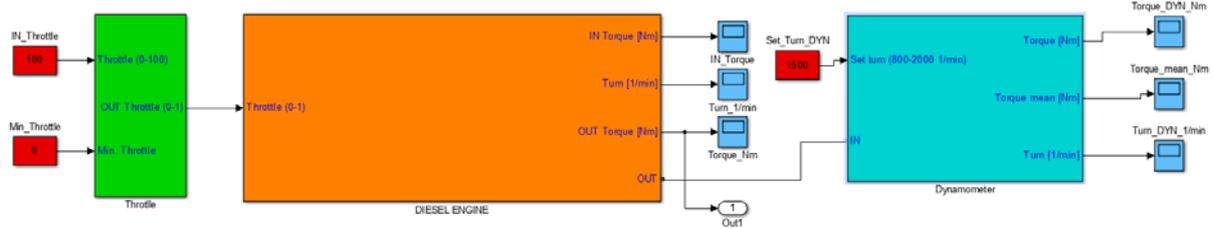
where φ is the vector of generalized coordinates. The equation is then transformed to the eigenvalues solution and angular natural frequencies are thus obtained by a square root of the eigenvalues of the $M^{-1}C$ matrix (5).

The resulting natural frequencies are compared with the natural frequencies acquired by linearization of the computational model in chapter 3.1, Tab. 1.

2. VIRTUAL ENGINE MODEL

2.1 Subsystems

The virtual engine model in Fig. 7 consists of the basic elements from the Simulink library and is formed by three subsystems. The first subsystem performs the function of a throttle pedal signal generator, where the input value is chosen from the range of 0-100, with an additional input for the engine's idle speed settings. Signals are then transformed by the subsystem to a range of 0-1. The second subsystem is the model of a diesel engine, which loads the input data including those of torsional stiffness, moments of inertia, torque maps or the value of engine speed limit. Finally, the third subsystem is used for running the engine in various modes – it is basically a virtual dynamometer. The instructions for modelling in Simulink were gathered from the book (1).

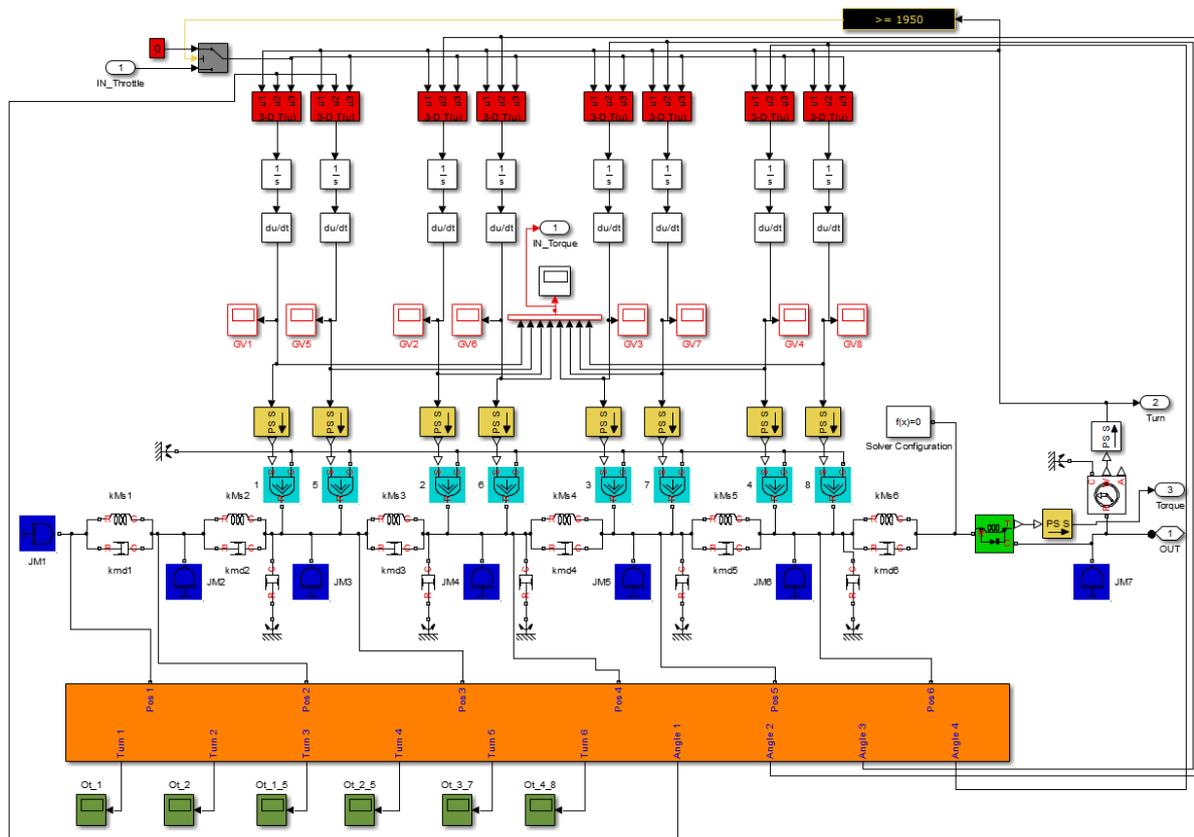


Source: Author

Fig. 7 – A virtual engine composed of subsystems

2.2 The engine model

The main subsystem of the model is the engine model as shown in Fig. 8



Source: Author

Fig. 8 – The engine subsystem

The engine model is based on a torsional model which includes rotating masses - represented in Fig. 8 by the blue blocks. Therefore, their moments of inertia serve as an input. Individual rotating masses are interconnected by torsion springs defined by their stiffness and damping properties. These elements represent only the rotational movement while further blocks are connected to each of the rotating masses JM3-JM6, serving as a torque output from individual cylinders. To each of these blocks, a three-dimensional table that loads the generated three-dimensional matrix with the torque input data is connected. As already mentioned, the matrix is shifted for each cylinder separately according to the engine's ignition

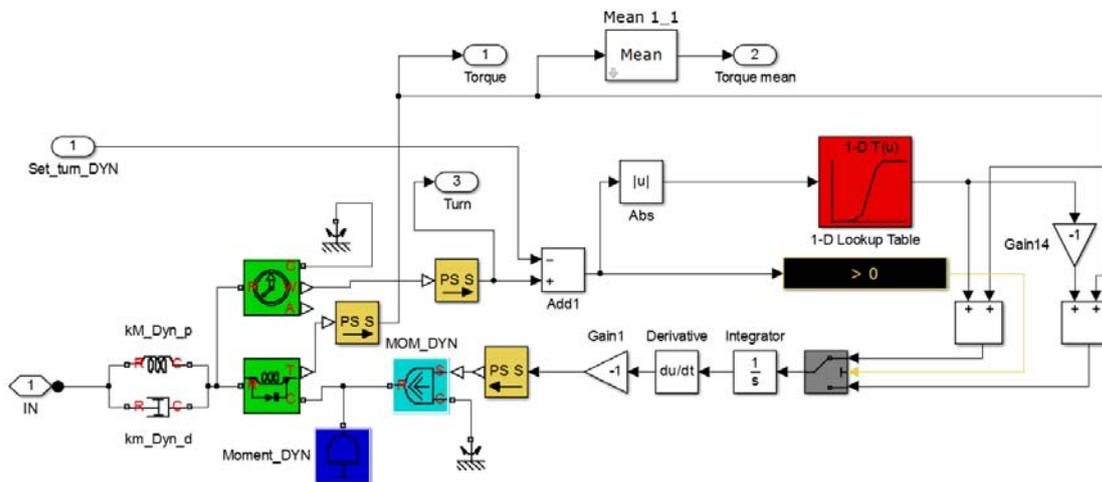
sequence. Three values must be entered into these tables so that the appropriate value can be selected from the 3D map.

If this selected value lies between the values specified in the three-dimensional matrix, it is automatically calculated by the means of extrapolation and interpolation methods used between the nearest values. These methods are set as linear. The first input is the actual engine speed in the range from 800 rpm to 2000 rpm. The second input is the crank angle which is monitored by the orange block depicted in Fig. 8. There are other subsystems for each crank throw in this block, which synchronize the crank angle with the torque map.

The third input is the throttle position which indicates its position in regards to the maximum (maximum load) and minimum (engine braking) fuel supply. Another important feature of the engine model is the engine speed limiter which sets the throttle position to zero after exceeding the pre-set limit. The final parts of the model are the elements substituting mechanical losses. For this purpose, rotary dampers which increase the torque with increasing speed are used. This torque counteracts the torque generated by the indicated pressure in the engine cylinders and thus simulates the engine mechanical losses. The output torque of the model should therefore be consistent with the torque measured on a real engine.

2.3 Dynamometer model

The dynamometer model is formed by a rotational mass with the dynamometer's moment of inertia assigned to it and by a block representing the stiffness of the shaft linking the dynamometer to the engine. This subsystem generates a torque which counteracts the engine torque to achieve the desired engine speed. It is possible to replace this dynamometer subsystem with a virtual powertrain in order to simulate the vehicle propulsion.



Source: Author

Fig. 9 – The dynamometer subsystem

3. VIRTUAL ENGINE SIMULATION

3.1 Linearization of the model

The calculation was set to a fixed step and the ode14x solver that combines the Newton's and extrapolation methods was used. This solver is recommended when running program calculations for models with flexible elements. The same settings were used for all subsequent simulations. Virtual models created in Simulink can be linearized using Matlab commands. As stated in the book (2), *linmod* is a suitable command for this purpose. This command finds matrices A, B, C, D, where the matrix A is a matrix system. From this matrix, it is possible to obtain the eigenvalues through the *eig* command. Imaginary components of these numbers contain the natural angular frequencies of the system that are looked for. The results of natural frequency calculations are recorded in Tab. 1. These values are then compared with the values obtained from the equations of motion introduced in chapter 1.3 and simulations presented in chapter 3.2.

Tab. 1 – Evaluation of natural frequencies of torsional models

The natural frequency of the torsional model with the dynamometer [Hz]	The natural frequency of the linearized virtual engine [Hz]	The natural frequency of virtual engine simulation [Hz]
3336,2	3336,2	Not evaluated
1355,0	1355,0	Not evaluated
1138,7	1138,7	Not evaluated
1031,4	1031,4	Not evaluated
731,1	731,1	Not evaluated
274,3	274,2	274,3
139,7	139,7	139,3

Source: Author

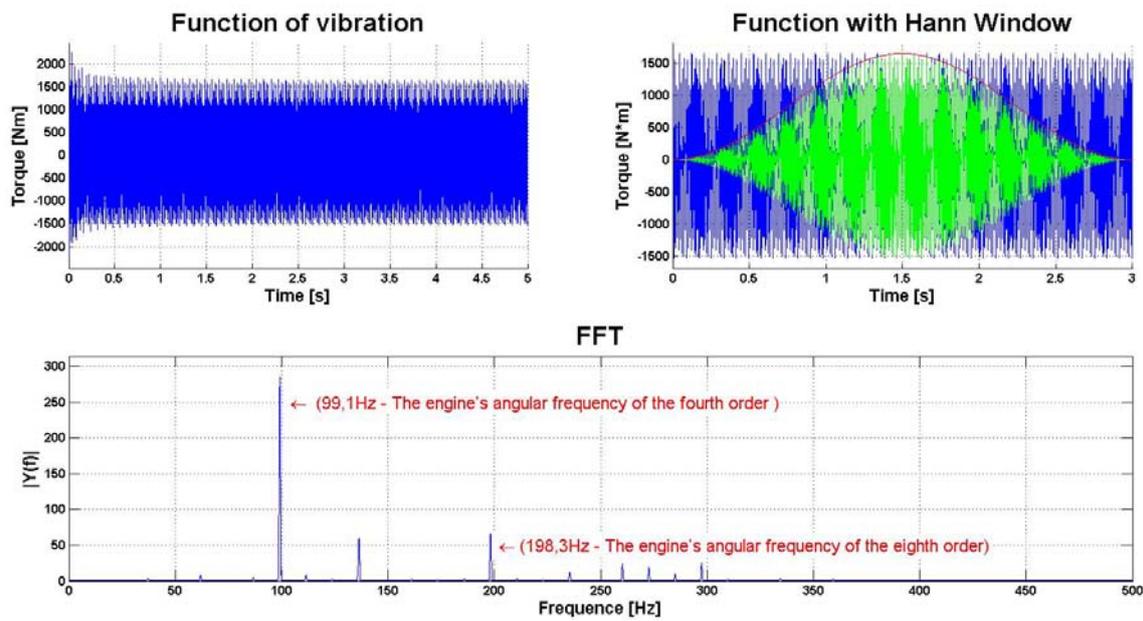
3.2 Engine Simulation

The simulation was running for 5 seconds in total, but already after 2 seconds a stabilized state at the required speed was reached, which in this case was 1487 rpm. The torque curve is recorded from the simulation and the Hann Window is used for its enhanced accuracy. Subsequently, frequencies from this function are evaluated using Fourier Transform as shown in Fig. 10.

FFT analysis program was developed in Matlab on the programming basis described in the book (3). Evaluated frequencies of the torsional model represent the natural frequencies of the system and their highest peaks are the engine's angular frequencies of the i-th order. To compare the calculated frequencies with simulations, an engine's angular frequency of the fourth order is calculated in the following equation:

$$\frac{n_m \cdot 4}{60} = \frac{1487 \cdot 4}{60} = 99,13 \text{ Hz} \quad (8)$$

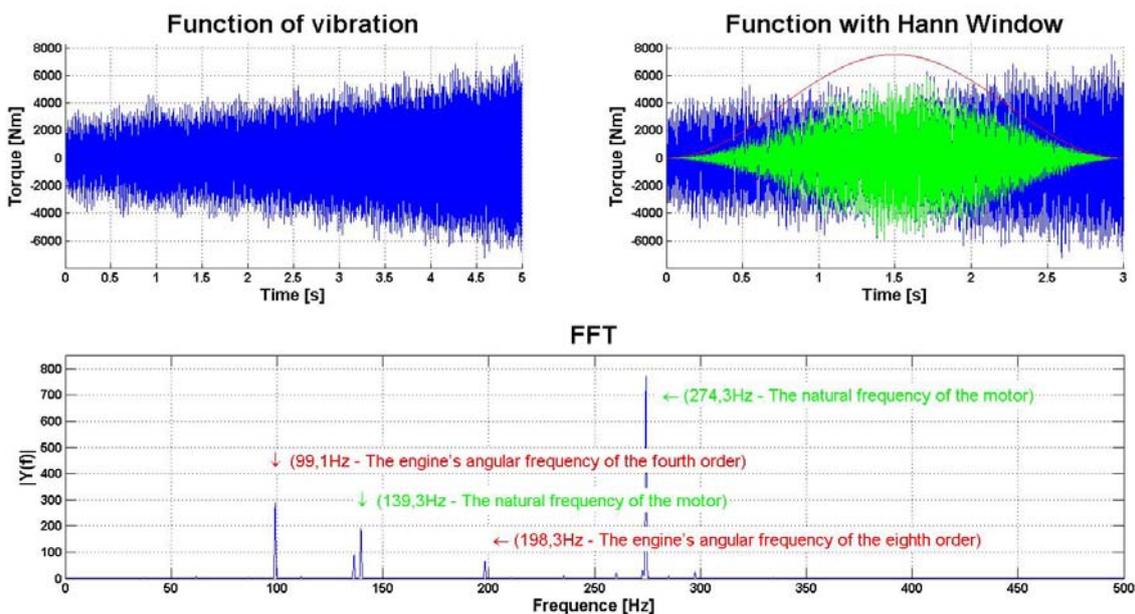
where n_m is the engine speed in the simulation.



Source: Author

Fig. 10 – The engine simulation at 1487 rpm with damping

The fourth order angular frequency is the most prominent one. Other evaluated frequencies are multiplications of the engine’s angular frequencies. When damping is not included in the torsional system, then the peaks of the torsional model natural frequencies are also displayed in the evaluation. These values are presented in Fig. 11 and compared in Tab. 1.

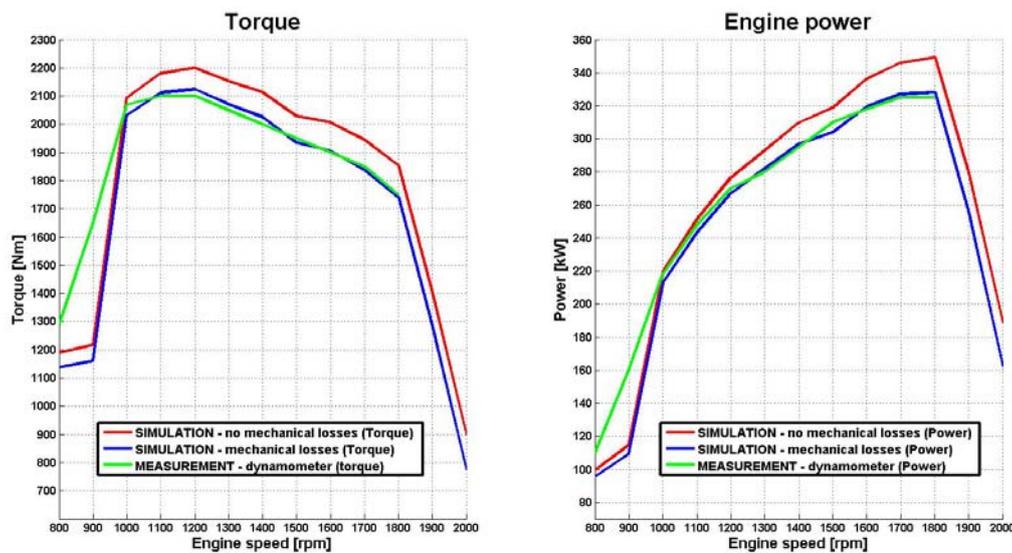


Source: Author

Fig. 11 – The engine simulation at 1487 rpm without damping

3.3 Speed characteristics of the engine model

The next simulation was carried out with the engine being gradually slowed down from a speed of 2000 rpm to a speed of 800 rpm with 100 rpm step. The engine was stabilized at these speeds and a mean torque was obtained. The first simulation was performed without considering the mechanical losses and the engine torque and power were compared with the speed characteristics of the real engine. Then the mechanical losses were included and adjusted so that the torque and power curves of the real and the virtual engine overlapped. The resulting curves from both the virtual and the real engine are shown in Fig. 12.



Source: Author

Fig. 12 – The engine speed characteristic

CONCLUSION

The virtual engine was assembled from subsystems which were created by using blocks from the Simulink libraries. The aim was to get the virtual engine to the real diesel engine as close as possible. In the first simulation in chapter 3.2, the natural frequency of the torsional model is not very clear as the virtual engine includes torsional damping. The fourth order engine angular frequency is distinctive in this simulation which was carried out at an engine speed of 1487 rpm. When torsional damping is not included in the torsion model, the torsional model's natural frequency is shown in the FFT analysis. The evaluation of natural frequencies is presented in Tab. 1. From the results of the simulations and calculations it can be concluded that the values of natural frequencies are identical with those obtained in the situation where torsion elements are included in the model.

The primary aim was to enable the engine model to perform torsional vibrations and to make the torque from the virtual engine comparable to that of the real engine. This can be established by performing simulations at various engine speeds and comparing them with the outputs of a real engine. The results proved that the virtual engine can be successfully used in

a virtual environment for driving virtual vehicles with forced oscillations. The engine can also be easily customized for different applications and different ignition sequences in engine cylinders.

These complex virtual assemblies can then be used for various analyses or for real-time simulations in the Hardware in the Loop systems.

ACKNOWLEDGEMENT

The presented work has been supported by European Regional Development Fund in the framework of the research project NETME Centre – New Technologies for Mechanical Engineering, project reg. No. CZ.1.05/2.1.00/01.0002, under the Operational Programme Research and Development for Innovation and with the help of the project FSI-J-13-2008 Vehicle Dynamics Modelling II granted by specific university research of Brno University of Technology. This support is gratefully acknowledged.

REFERENCES

- (1) DABNEY, J. B., HARMAN, T. L. *Mastering Simulink*. Upper Saddle River: Pearson Prentice Hall, 2004, 376 s. ISBN 0-13-142477-7.
- (2) GREPL, R. *Modelování mechatronických systémů v Matlab SimMechanics*. 1. vyd. Praha: BEN - technická literatura, 2007, 151 s. ISBN 978-80-7300-226-8.
- (3) ZAPLATÍLEK, K., DOŇAR, B. *MATLAB: začínáme se signály*. 1. vyd. Praha: BEN - technická literatura, 2006, 271 s. ISBN 80-7300-200-0.
- (4) KOŽOUŠEK, J. *Výpočet a konstrukce spalovacích motorů II*. 1. vyd. Praha: SNTL, 1983, 483 s.
- (5) PÍŠTĚK, V., ŠTĚTINA, J. *Pevnost a životnost*. 1. vyd. Brno: VUT Brno, 1993, 205 s. ISBN 80-214-0474-4.