

COMPUTATIONAL MODEL OF THE AVIATION DIESEL ENGINE FOR HIL TESTING

Pavel Kučera¹, Václav Pištěk², David Svída³, Martin Beran⁴

Summary: This article deals with the development of a computational diesel engine model for the aerospace industry and HIL testing. The model is assembled from the blocks of own libraries in the Simulink software. To verify functionality, it was implemented on hardware for HIL testing, and responses to inputs generated by a simple real-time control algorithm were monitored. The aim was to verify and optimize the computational model for subsequent HIL testing with the prototype ECU.

Key words: computational model, diesel engine, control algorithm, (HIL) Hardware in the Loop, Simulink, NI VeriStand

INTRODUCTION

The development of piston aircraft engines advances to the use of more modern technologies which in most cases are taken over from the automotive industry. The effort of all engine developers is to accelerate this development and reduce its costs. An integral part of each engine is the use of mechatronic systems to control such an engine. Therefore, it is an attempt by development companies to use tools and methods for such development that they themselves develop or purchase. Such tools or methods may include several phases of development and associated methods.

First method is Model in the Loop - MIL testing. This is a combination of a computational model of device with a developed control algorithm. This is followed by Software in the Loop - SIL testing, where the control algorithm is already in the form of generated code, but it is tested in a computer environment. The next step is Processor in the Loop - PIL testing, when the control algorithm is already implemented on hardware as opposed to SIL testing. There is tested whether the hardware is sufficiently conceived for the control algorithm and there is no delays in the calculation loops. The last phase is Hardware in the Loop - HIL testing and the hardware serves as an ECU replacement. Then the mechatronic system developed can be tested on the vehicle without an ECU prototype. Of course, different hardware, software and

¹ Ing. Pavel Kučera, Ph.D., Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automotive Engineering, Technická 2896/2 , 616 69 Brno, Tel.: +420 541 142 274, 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

³ Ing. David Svída, Ph.D., Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automotive Engineering, Technická 2896/2 , 616 69 Brno, Tel.: +420 541 142 248, Fax: +420 541 143 354, E-mail: svida@fme.vutbr.cz

⁴ Ing. Martin Beran., 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: beran.m@fme.vutbr.cz

computational models are needed for each phase. If computational models are not available, the company is forced to develop its own computational model.

Therefore, this article dealt with the development of a diesel engine computing model for Hardware in the Loop testing. This engine is designed for the aerospace industry and is a six-cylinder boxer. The article will describe the computational model created with respect to Real-Time testing. The model is assembled of blocks contained in own library. Each block mathematically describes the behaviour of engine parts and the user is able to assemble different types of engine computational models. Subsequently, the user can test their dynamics, analyse vibrations or test mechatronic systems. The last point is the main purpose of this article. The aim was to create a computational model describing the behaviour of a diesel engine with the possibility of using it for HIL testing. This means that this computational model could be used by the firm developing the electronic control unit - ECU for the diesel engine for HIL testing.

Further, the article describes a simple control algorithm designed to verify the function of a computational model of an engine. Consequently, the engine model and control algorithm were implemented on hardware for HIL testing. So, we can say that this was PIL testing because no other devices were connected to the hardware. The development of the computational model, was preceded by MIL and SIL testing. But this article describes the results of PIL testing and thus verifies whether the computational model is ready for HIL testing in connection with the ECU prototype. At the end of the article, other development procedures are described.

1. COMPUTATIONAL MODEL

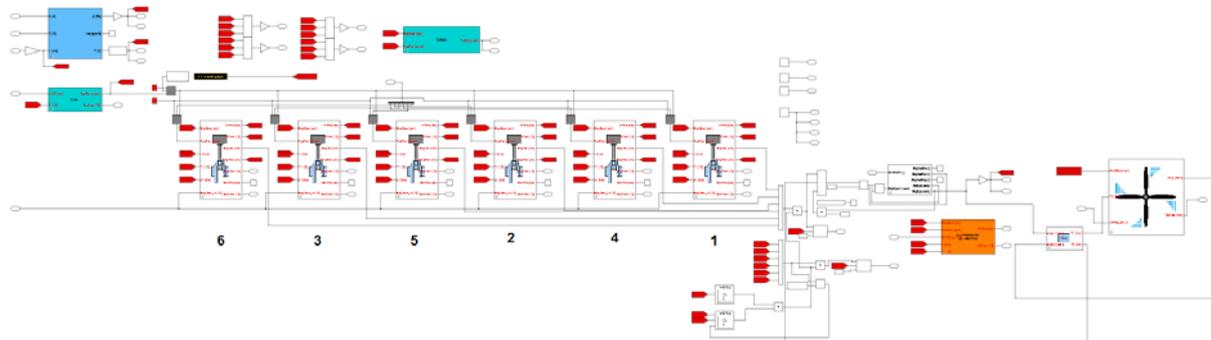
The computational model of the diesel engine is assembled from blocks of own library, which is generally aimed at different parts of vehicles. Using them, the user can assemble whole vehicle computational models and simulate their dynamics, analyse vibrations or test mechatronic systems. The library is still expanding and a new library block for the engine computational model for the aerospace industry will be described here. The creation of library blocks is realized in the Simulink software and the basic prerequisites for computational models are described in the literature (1), (2) and (3). The library is also described in articles where the description is focused on the creation of vehicle computational models (4) and (5).

The diesel engine computational model block is built from other subsystems representing a mathematical description of the crank mechanism for each cylinder, the ISA model block (6), the fuel batch block, the turbocharger block, the starter block, the oil behaviour description block, the shaft block between the engine and propeller and propeller block.

The block describing the International Standard Atmosphere - ISA model determines the basic quantities needed for the computational model of the engine. It depends on the altitude, the basic set temperature and velocity. From the basic equations defined in (6), the temperature and pressure are calculated. They are input to the block of the crank mechanism.

The aim is to use the overall computational model of the engine for HIL testing. It is important to assemble the computational model with respect to hardware performance and to the sufficient description of the engine behaviour. Therefore, in a number of blocks, tables with data representing a given quantity are used and the output value is then interpolated or extrapolated by input variables. This greatly reduces the hardware demands to run the HIL

testing. The fuel dose block uses data that is evaluated from the table. The input is the position of the row pump rod and the output is the fuel batch. Part of the block is also a defined delay of pressure rise or descent when changing the position of the rod. The second block output is the fuel temperature resulting from the air flow temperature around the fuel system.



Source: Authors

Fig. 1 – Engine computational model

The crank mechanism block is divided into four subsystems that describe the characteristics of air inlet, exhaust, cylinder head heating, and torque and power output. The first subsystem searches the values of the pressure and temperature of the air inlet in the tables using input of the engine speed and the fuel batch. The search for output variables is done through interpolation and extrapolation. The output variables are limited by dynamic saturation, so that in some states the evaluation of the milestone information is not possible. The second subsystem evaluates the exhaust temperature and pressure. Again, these values are evaluated from table data, depending on the fuel batch and engine speed. The third subsystem describes mathematically the transfer of heat generated by the engine. From the data, the heat output value is determined by the cycle of the crank mechanism, depending on the fuel batch and engine speed. Further, heat conduction equations include heating the cylinder head. There is implemented the influence of the flowing air, which on the other hand cools the head. The fourth subsystem determines from table data the value of the engine torque and the engine power for one cylinder. There are used six cylinder blocks. This is also shown in Fig. 1 where the internal configuration of the engine model is shown.

The engine torques from the individual cylinders are summed up and the engine speed is evaluated using the following equation.

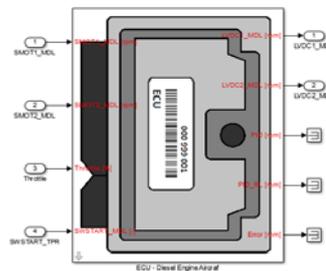
$$\omega = \int \frac{(T_1 - T_2)}{J} \quad (1)$$

Where T_1 - engine torque, T_2 - torque which is taken by, for example, mechanical losses, J - engine inertia moment and ω - engine angular speed which is recalculated at engine speed. Included in this calculation are mechanical losses, torque and power correction for different ambient temperatures and pressures. Also included is a block with a description of starter behaviour. This means that if the engine speed is zero and the user sends a start signal, the

engine speed is calculated with this block in the first stage. The speed is gradually increased to the maximum speed that is given by the starter speed. During this state, the ECU should start responding and set such a fuel dose to the engine start and to control idle speed. Above the speed given by the starter, the engine speed is calculated according to the previous equation.

A shaft block is connected to the main part of the engine computational model and the propeller block. In the shaft block, the user can set torsional stiffness, initial deformation or damping. Propeller block is assembled from the above equation too. The propeller block includes the regulation of the propeller speed. This speed is controlled at the specified engine speed.

The oil block is designed to evaluate the engine lubrication pressure and oil temperature. The pressure is evaluated from the table in relation to the temperature and engine speed. The oil charge temperature is calculated using equations for heat management including cooling.



Source: Authors

Fig. 2 – Control algorithm block

The last block of the turbocharger determines its speed depending on the fuel batch and the engine speed. Block includes delay representing the turbo-effect. Of course, there are detailed computational models available for this phenomenon but with emphasis on the computational demands of the whole model this simplified approach has been taken. Generally assembled engine model requires a series of input data. These were provided by the manufacturer from a measurement or simulation.

2. CONTROL ALGORITHM

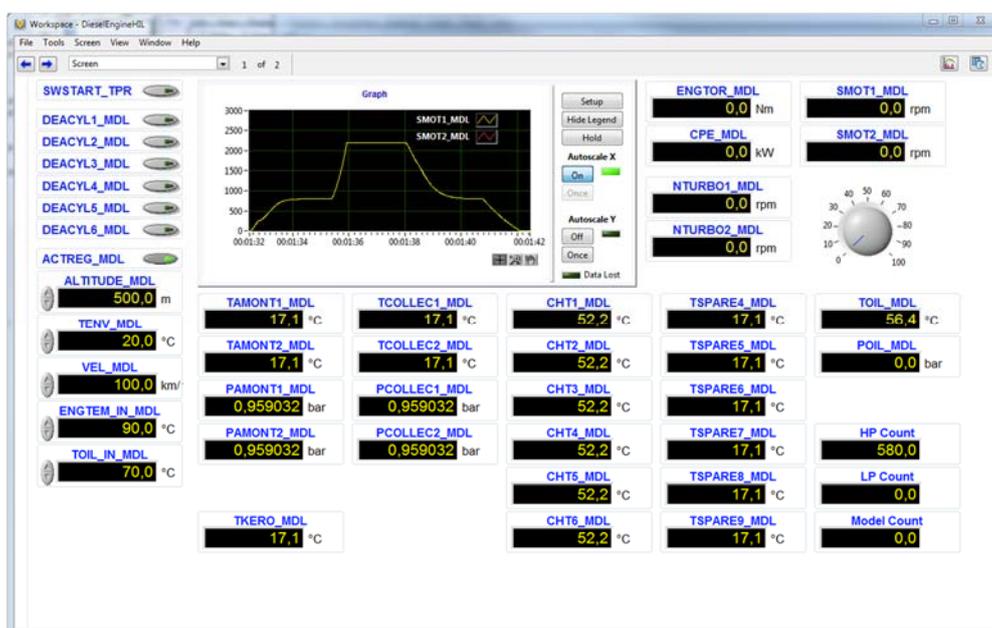
A basic control algorithm was programed to test the engine computational model. The control algorithm is described in the block shown in Fig. 2. Its function is such that the control algorithm monitors the engine speed and when the set engine speed is exceeded the PID control is triggered. It regulates the position of the row of the in-line injector pump so that the engine starts and the fuel dose stabilizes the engine speed at its idle speed. Subsequently, the value of the fuel set by the pilot is monitored. If this value is higher than the fuel value at idle speed, the engine speed starts to increase. Conversely, if the fuel batch is lowered, the PID controller is again regulated. The PID controller is discrete and its step is 0.01 s. The PID controller also includes saturation which is set to the minimum and maximum position of the row pump rod. Additionally, Anti-windup and Tracking mode are used in the controller for transitions between control via a PID controller or pilot.

3. MODEL VERIFICATION

Verification of the engine computational model for HIL testing was performed in NI VeriStand software. This software is designed to implement computational model to hardware for HIL testing. It allows users to set up various tests, record data, or display them using the Workspace. For PIL testing, hardware was used by National Instruments NI 3110 RT where a processor is located.

There was implemented an engine computational model and a control algorithm. Loop speeds have been set for the engine model and control algorithm of 1 kHz. Some parts of the control algorithm are performed at a frequency of 100 Hz. A sample of PIL testing is shown in Fig. 3. The correct function and response of the engine computational model to the fuel batch setting or engine start values were monitored. It should be mentioned that the correctness of the operation of the computational engine has already been verified in MIL testing. Therefore, here it is mainly checked the computational demands of the computational model. First, test was on the Windows target. It could be said that it was SIL testing. There has been occasionally to omit some computational loops. But in PIL testing on hardware for HIL testing, the delays did not occur.

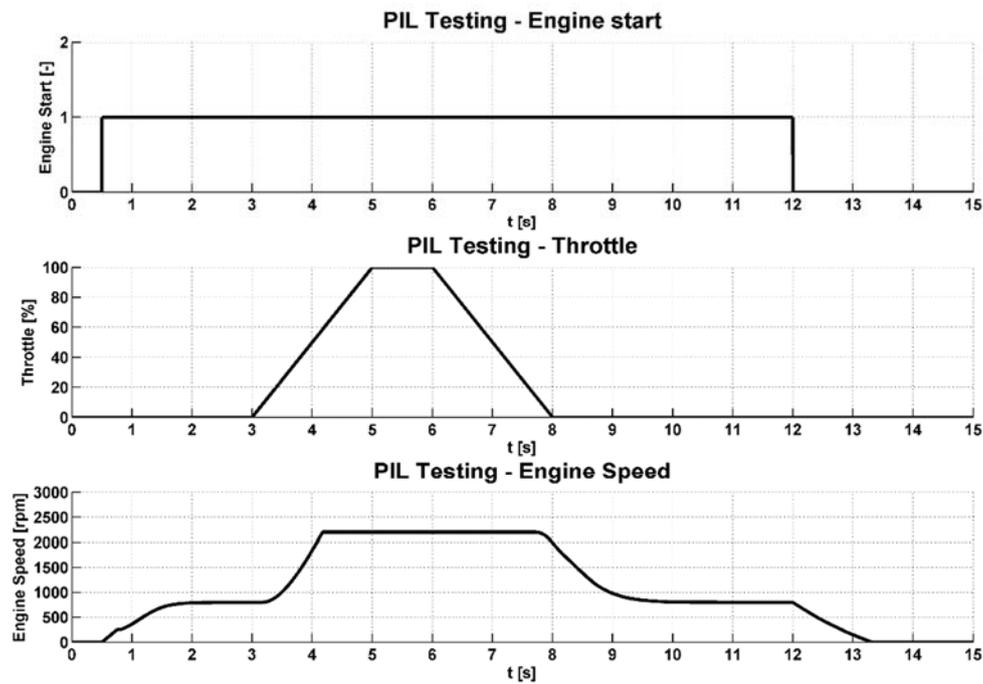
The results from the PIL testing are shown in Fig. 4. The first graph shows the signal for the engine start during testing. So the start occurred at 0.5 s and the engine speed gradually increase to 250 rpm. The control algorithm started working when the 150 rpm was exceeded. It continuously adjusts the position of the injection pump rod. Third graph shows the course of the engine speed. Speed was regulated at idle of 800 rpm. Subsequently at time 3 s, the pilot started setting the fuel dose to the maximum value. At that stage, the idle speed control no longer acted, and the engine speed began to rise.



Source: Authors

Fig. 3 – SIL testing in the NI VeriStand software

During increasing engine speed, propeller control began to react and the engine speed was set at 2200 rpm. The pilot still added fuel. This increases the thrust of the propeller and the aircraft will gradually begin to run along the starting runway.



Source: Authors

Fig. 4 – Results of the PIL testing

As the pilot began to take the fuel dose to zero at 6 s, the PID control began to work again and the engine speed was stabilized to 800 rpm. Subsequently, at 12 s, the pilot switched off the engine and the engine speed dropped to 0 rpm. Of course, many test scenarios were run on PIL testing, but here is the illustration. From the results of PIL testing it can be concluded that the engine response and computational complexity is correct. Therefore, let's say that the computational engine model can be used by the developer ECU with the control algorithm for HIL testing. This was the goal of this article.

CONCLUSION

To accelerate the development of mechatronic systems, it is necessary to use different tools. These include various hardware, software and methods. It is also important to have different computational models for testing mechatronic systems. If these models are not commercially available, they must be created for different types of testing.

Therefore, this article dealt with the creation and PIL testing of the engine computational model. It was a diesel engine designed for the aerospace industry. The goal was to adequately describe its functionality with regard to the computational difficulty for HIL testing. The results show that the engine responds correctly to the input quantities and there is no delay in implementing hardware for HIL testing. Testing was performed on hardware for HIL testing, but no peripherals for digital and analog inputs/outputs were still used. Therefore, It was PIL

testing. The computational model has been verified in connection with the control algorithm and it can be replaced by the prototype ECU.

Further development of the computational model of the engine will be verified with the measured prototype data of the engine. It will be used for correct the input data of the model.

ACKNOWLEDGEMENT

This project has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 754869



REFERENCES

- (1) DABNEY, James B. a Thomas L. HARMAN. *Mastering Simulink*. Upper Saddle River: Pearson Prentice Hall, 2004. 376 s. ISBN 0-13-142477-7.
- (2) HEISLER H. *Advanced Engine Technology*. Arnold, Oxford, 2002.
- (3) GREPL, Robert. *Modelování mechatronických systémů v Matlab SimMechanics*. 1. vyd. Praha: BEN, 2007. 151 s. ISBN 978-80-7300-226-8.
- (4) KUČERA, P. *Mechatronický přístup v dynamice vozidel*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 116 s. Vedoucí dizertační práce Prof. Ing. Václav Píštěk, DrSc.
- (5) KUČERA, Pavel a Václav PÍŠTĚK. Virtual prototype of a heavy duty off-road truck driveline in Simulink software. In: *Transport means 2014: Proceedings of 18th International Conference*. Kaunas: Technologija, 2014, s. 5-8. ISBN 9955-09-935-6.
- (6) ISO 2533:1975. *Standard Atmosphere*. 1. Geneva Switzerland: International Organization for Standardization, 1975.