DESIGN OF A MATHEMATICAL MODEL OF A SMALL TURBOJET ENGINE USING THE S-FUNCTIONS IN MATLAB

Erika SZEPESIOVÁ, Ladislav FŐZŐ
Letecká fakulta, Technická univerzita v Košiciach
Rampová 7, 041 21 Košice
Slovenská republika
erika.szepesiova@student.tuke.sk, ladislav.fozo@tuke.sk

Summary. The article is focused on the possibilities of transformation of mathematical models of a small jet engine MPM-20 created in Matlab as a file with the extension .m to S-functions in Simulink (Matlab subroutine). It deals with the comparison of different types of S-functions and their advantages and disadvantages.

Keywords: mathematical model, Matlab, Matlab/Simulink, s-functions

1. INTRODUCTION

Mathematical models provide a description of the situation and to reveal significant relationships between elements of the system under study. Modeling and simulation are focused on exploring the native systems by created model. With this created model we can study the behavior of the system in the future [1]. To creating mathematical models can be used Matlab program. This work is based on an analytical model of equilibrium and disequilibrium work of small turbojet engine MPM-20, which is taken from a supervisor [2] and designed in the program Matlab in file with the extension .m. In the future, should in the laboratory of intelligent control systems for aircraft engines in the area LF TUKE this analytical model run Real Time (along with the motor). Engine control and engine diagnostics are run in Simulink, so partial task for the next experiment, is transforming this analytical model to Simulink. One option are S-functions, so it was necessary to analyze and clarify the possible ways of creating mathematical models using S-functions in Matlab- Simulink. From this analytical model is transformed into S-function thermal calculation of this engine.

2. MATHEMATICAL MODEL OF MPM-20 ENGINE

Mathematical model of the engine can be create in one of two ways, analytically and experimentally. In works [3, 4] were models designed with experimental identification. In the first case, neuro-fuzzy non-linear model of a small turbojet engine MPM-20in all operating modes, and the second experimental models have been proposed several with methods of identification, the best results was achieved by the gradual integration method (MPI). To the resulting model entering the fuel supply and the output was speed. In case [2] was created mathematical model with analytical identification because it was necessary to know the permissible area of regulation and identify more measurable parameters like with an experimental identification. Because when we have created model with experimental identification, we do not know what is in the structure. We know only the inputs and outputs. So another reason was the need to know even in the event of necessary data to manage and have under control motor monitoring. This analytical method requires knowledge of physical processes and their mathematical description of ongoing processes in the system. The mathematical model is based on embodiments of the physical laws that characterize the properties and activities of the various nodes and components of the engine, the properties of the working fluid and the thermodynamic and aerodynamic happens, it is necessary to consider the change in the properties of the working fluid. The application of this model in detail MPM20 was necessary to process mathematical models and characteristics of individual parts of the engine. The analytical model of engine was created by the supervisor, and in this work will not present a detailed description. Analytical model of equilibrium and disequilibrium engine running MPM20 with fixed and variable-geometry (two-argument) is based on experience and knowledge of experts in the field, calculating the heat circulation, resp. the results of measurements on real objects [2]. From knowledge of computing mode and the use of diagrams with relative magnitudes are determined by characteristics of the engine. The curve is a graphical representation of a driving position points. For the implementation in situational control, needs the model to be modified so as to be able to read the current value of the position of one point. Understand taken m- file and then the modification of this file would be time consuming and extensive. Since the objective of this work is to understand and analyze possible ways and forms of description S-functions, is used from this model thermodynamic calculation.

3. USING S-FUNCTIONS TO CREATE MODELS IN SIMULINK

At the time when was only indirect management in the laboratory, was created the analytical model of motor MPM20 [2], which was created by using Matlab as m-file. In the laboratory, we want that this engine model will run in Simulink and in real time (model simultaneously with the motor). For the implementation of mathematical models to real engine management is necessary to convert the mathematical model in Simulink. The aim of the work is to find a possible solution for the transformation of analytical models in Simulink. In the title of the chapter you may be observed that to transform will be used the S-functions.

3.1. S-functions

The importance of the S-functions lies for this issue in the possibility of creating custom blocks in Simulink, to which we can inscribe own functions, mathematical descriptions of various systems Fig.1.

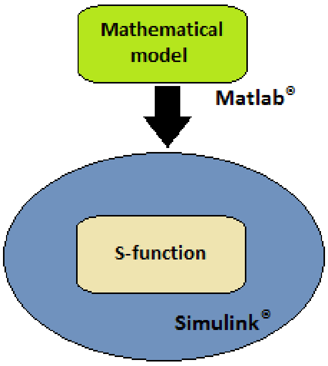

Figure 1 Schematic transfer from Matlab into S-function in Simulink

S-function (system-function) constitute a mechanism for enhancing the possibilities of work in Simulink. S-function is dynamic block in Simulink, which is described and defined in a file that is a file created in Matlab, programming languages C, C ++, or Fortran. S-functions can include continuous, discrete, and hybrid systems. We can create an S-function whit insert the block S-Function from Simulink working environment. The program provides a template S-functions those we can describe with own formula. There are several types of options of describing S-functions. The options will be described in detail in following chapters [5].

S-functions are commonly used to create custom blocks in Simulink. It can be used for a variety of applications such as:

• Adding new universal blocks in Simulink
• Inclusion of existing code in C to simulation
• In describing the system using mathematical equations
• Using the graphical animation.

In our case, the use of S-functions is advantageous in option that we can describe the S-function block in Simulink with own mathematical equations files. The S-function will be described with mathematical equations of engine parts of engine MPM-20.

3.2 How S-function works

To create an S-function, is necessary to understand how S-function works. It's actually a dynamic block, where the outputs are functions of the sampling period, entries and states. In this dynamic block is possible to inscribe various mathematical relationships of various systems.

Figure 2 Dynamic block of S-function

S-function simulation in Simulink takes place in several stages. S-functions contains of Callback methods that performs the necessary tasks in every stage of the simulation. During simulation the model at each simulation stage Simulink calls the appropriate methods for each block of S-functions in the model. For the simulation we need to define the Callback methods. Required Callback methods are: setup- for initialization of basic characteristic of S-function (defined inputs, outputs etc), outputs- calculate outputs, terminate- shall take all actions necessary to terminate the simulation. Optional Callback Methods depends on the requirements that are imposed by the user. Simulink calls them only when they are defined [5] [6].

3.3. Types of S-functions

From Fig. 3 is clear that the S-function can be described using a Matlab or MEX file. The different types of blocks are shown to be used to construct the model in Simulink.

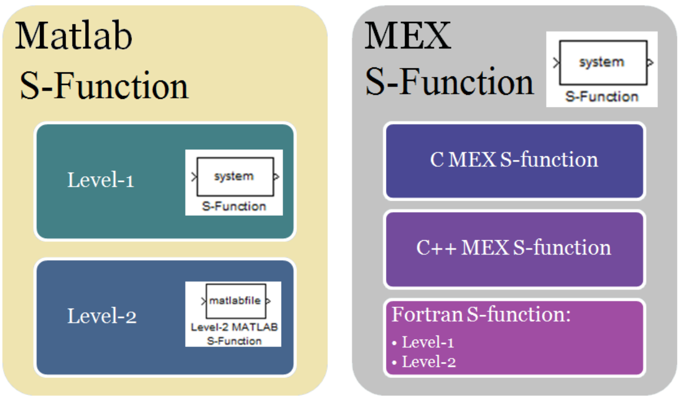

Figure 3 Types of S-functions

ISSN 1339-9853 (online), acta-avionica.tuke.sk ISSN 1335-9479 (print) © 2015 LF TUKE

Advantages of M-file and MEX-file:

The advantage of M- files is the speed of development. M- files prevent timeconsuming compilations-interconnections required by compiler. M-files have easier access to MATLAB and toolbox functions [7].

The main advantage is versatility of MEX- files. A larger count of Callback methods enables to SimStruct access for MEX-files to implement functions that are not accessible to the M-files. For example, the ability to process data types other than doubled complex inputs, matrix inputs etc. [7].

In the following part are created different types of S-functions and they dealt with the pros and cons. S-function will be described with mathematical model of centrifugal compressor of MPM-20 engine. The most preferred type of S-function will be choose and will be transformed into it other parts of MPM20 engine.

4. TRANSFORMATION OF SELECTED PARTS OF MPM20 INTO S-FUNCTIONS

Chapter includes models of selected parts of the engine MPM-20 created using S-functions that calculate the basic parameters of individual output of the motor. Simulation of selected variables will take place in Matlab-Simulink, which meets the requirement of the environment for scientific computing, mathematical modeling, simulation and analysis. Using the simulation model assembled by S-functions could be on the test of intelligent motor control systems to manage, monitor changes required parameters, and then detect possible errors in the system.

4.1. Transformation centrifugal compressor to particular types of S-functions

In this section is transformed the mathematical model of centrifugal compressor to particular type of S-function, then in the most preferred S-function type will be transformed other selected parts of the engine MPM-20.

Transformation options:

Matlab S-function Level-1 provides a simple Matlab interface for interacting with a small amount of callbacks. Matlab S-function Level-2 replaces Level-1 Matlab S-function. Mathematical model centrifugal compressor is characterized by two inputs parameters and four outputs. Level 1 S-functions support one input and one output port. These ports must be vectors, but there is no length limit. The inputs and outputs may therefore be of different lengths. Level-1 support a much smaller subset of the S-function API and its features are limited compared to the Level-2 Matlab S-functions. It is preferable to use a Level-2 [8].

Matlab S-function Level-2 provides access to a broader set of supports code generation. In most cases, is for transformation uses exactly this type of S-function. For users who have programmed in Matlab with little or no programming experience in C it is preferred to use Matlab S-functions. Matlab S-function Level-2 allows you to create blocks with many features and capabilities:

- Multiple input and output ports
- The ability to take the vector or matrix signal
- Support for a variety of signal attributes, including data type, complexity and frame signals
- Ability to work at different sample rates

Matlab S-function Level-2 works by setup command, which set up the basic properties of S-function and callback methods that Simulink calls at the appropriate time during the simulation. This method has the advantage that the block can enter multiple input and output ports, and can easily monitor their values [9].

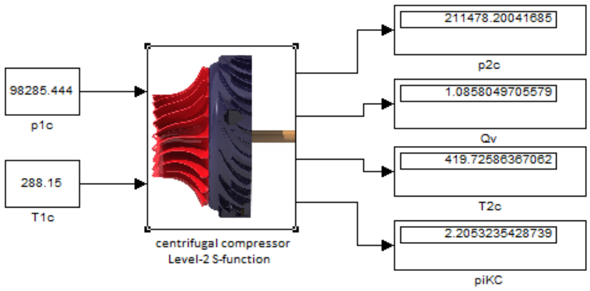

Figure 4 Level-2 Matlab S-function of centrifugal compressor

Hand writing **C MEX S-function** provides the greatest flexibility in programming. It is possible to transform own algorithms in the C MEX S-function, or write wrapper S-function, which means that the S-function is called an existing C, C ++ or Fortran code. C MEX S-function has advantages over the Level-2 S-function in the simulation speed, because the compiler will create a file needed for the simulation. Transform the mathematical model in this type of S-function is different in structure [10].

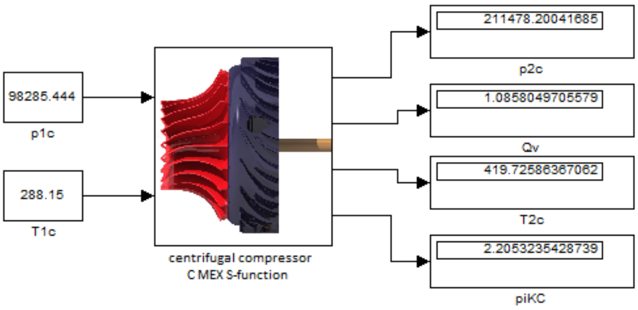

Figure 5 CMEX S-function centrifugal compressor

S-Function Builder is a graphical user interface for programming subgroups of S-functions. This type of S-function is used for those who are new in writing S-functions, it is not necessary to have knowledge about writing S-functions (writing Callback methods, necessary commands etc.). S-Function Builder created during code generation necessary files for simulation capability S-functions [11].

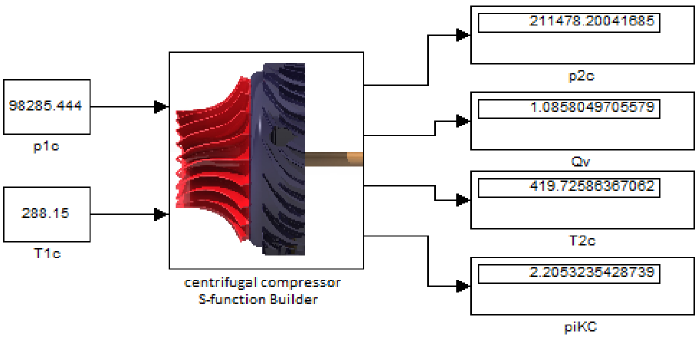

Figure 6 S-Function Builder centrifugal compressor

The process of creating **C ++ MEX S-function** is very similar to the process of creating C MEX S-functions. Format C ++ for the S-function is almost identical to the format for the S-function written in C. The main difference is the need to notify the compiler to use the Callback methods calls for the format C ++ and not format in C language. It is therefore necessary to understand the contents of the Simulink as C ++ code. Therefore, file extension must be CPP, the compiler used the contents of the file as C ++ code. An example can be found in this way: (matlabroot / Simulink / src /

sfun_counter_cpp.cpp) [12]. For our needs, this method is useless. Object-oriented programming is oriented on objects, which are substantially only structures comprising different variables and their associated functions.

Create **Fortran S-function** can be in two ways. One is the Level 1 Fortran-MEX (F-MEX) Level 2 is written in C. Both methods have advantages and may be used in real-time. Level 1 was the first API and development of Simulink is expanded to Level 2 API, which enables the S-function has all the skills as a model in Simulink (except automatic identification solutions and algebraic loops) [13].

Level 1 Fortran S-function template file for Fortran MEX S-function is: matlabroot / Simulink / src / sfuntmpl_fortran.for. To create a new Fortran S-functions using a template, it is necessary:

- Make a copy with a different file name.
- Adjust the copy as needed.
- Compile a customized file with MEX command.
- Include file into the model using the MEX S-function block.

To use the features of a **level 2 S-function** with Fortran code, you must write a skeleton S-function in C that has code for interfacing to Simulink and also calls your Fortran code. Using the C-MEX S-function as a gateway is quite simple if you are writing the Fortran code from scratch. If instead your Fortran code already exists as a stand-alone simulation, there is some work to be done to identify parts of the code that need to be registered with Simulink, such as identifying continuous states if you are using variable-step solvers or getting rid of static variables if you want to have multiple copies of the S-function in a Simulink model [13].

5 CONCLUSION

After the simulated various types of S-functions, we can see that the simulation results are the same, therefore, S-functions are working properly. The calculated values coincide with classic m-file. At the beginning of the chapter shows the advantages and disadvantages when and why to use different types of S-functions. After the application of the mathematical model of centrifugal compressor, the most preferred route to use is in this case Matlab Level-2 S-function, because we don't need knowledge of another programming language. For complex models can be applied for fast calculating C MEX S-functions. There is a possibility convert m-file created by Matlab using the C compiler to a C ++ file [14]. After converting m- file is created a c-file, which we can put into S-function. The disadvantage is the transformation of the c-file into S-function, because is needed knowledge of programming in C language, user needs to know the structure of the c-file.

Mathematical models of selected parts of the engine are transform into S-function as well as a centrifugal compressor. When implementing the various parts of the engine in the S-functions, be vigilant when we set the output of the block, which then entering into the next block. Scheme for the motor using S-functions is shown in Figure 7.

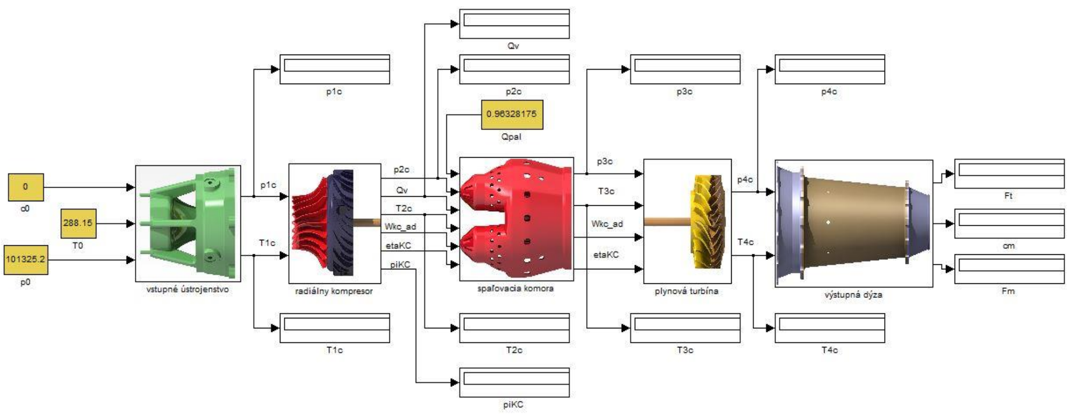

Figure 7 Simulation scheme of engine MPM-20 using S-functions

Transformed model of the engine is working properly, and compared with m-files show the same results. To demonstrate the process of calculating in real time, may be added blocks like Random Source where I enter some values that will vary over time indefinitely, Joystick, which can directly modify the input values with joystick control, and then connect the S-function block to the real engine MPM-20. With problem of implementation of the S-function model in the engine control MPM20 in the laboratory is deals in work [15]. After the implementation of S-function model into laboratory, the engine control and diagnostics of the engine can work via S-functions, and could be used this analytical model of the engine, because as far been used experimental models and also models created using artificial intelligence.

References

[1] Kutiš V.: Základy modelovania a simulácii, FEI STU Bratislava, 20. septembra 2006
[2] Főző L.: Využitie matematického modelu rovnovážneho a nerovnovážneho chodu motora MPM-20 pri návrhu algoritmu riadenia v každom čase., Doktorandská dizertačná práca, Košice 2008
[3] Andoga, R.: Hybridné metódy situačného riadenia zložitých systémov., Doktorandská Dizertačná práca, FEI TUKE, 2006, pp.120
[4] Bašista, Š.: Štatistická analýza a návrh modelu z experimentálne získaných dát malého prúdového motora MPM-20, Bakalárska práca KKUI FEI TU Košice, 2008
[5] S-funkcia. [s.a].[online].[cit 2014-12-12]. Available on the Internet: <https://www.kirp.chtf.stuba.sk/~cirka/vyuka/matlab/kap16.php>
[6] Callback Implementations . [s.a].[online].[cit 2015-1-20]. Available on the Internet:<http://mntek3.ulb.ac.be/Matlab/toolbox/simulink/sfg/sfun.html>
[7] MEX Versus MATLAB S-Functions. [s.a].[online].[cit 2014-12-13]. Available on the Internet:< http://www.mathworks.com/help/pdf_doc/simulink/sfunctions.pdf>
[8] Maintain Level-1 MATLAB S-Functions. [s.a].[online].[cit 2014-12-21]. Available on the Internet:<http://www.mathworks.com/help/simulink/sfg/maintaining-level-1-matlab-s-functions.html?searchHighlight=Level-1%20Matlab>
[9] Write Level-2 MATLAB S-Functions. [s.a].[online].[cit 2014-12-29]. Available on the Internet:<http://www.mathworks.com/help/simulink/sfg/writing-level-2-matlab-s-functions.html>
[10] Creating C MEX S-Functions. [s.a].[online].[cit 2015-1-11]. Available on the Internet:<http://www.mathworks.com/help/simulink/sfg/creating-c-mex-sfunctions.html?searchHighlight=C%20mex>
[11] Building S-Functions Automatically . [s.a].[online].[cit 2015-2-17]. Available on the Internet: <http://mntek3.ulb.ac.be/Matlab/toolbox/simulink/sfg/sfun.html>
[12] C/C++ S-Functions. [s.a].[online].[cit 2015-3-2]. Available on the Internet: <http://www.mathworks.com/help/simulink/c-c-s-functions.html?searchHighlight=C%2B%2B>
[13] Fortran S-Functions. [s.a].[online].[cit 2015-3-2]. Available on the Internet: <http://www.mathworks.com/help/simulink/fortran-s-functions.html>.
[14] Generating C Code from MATLAB Code. [s.a].[online].[cit 2015-4-30]. Available on the Internet: <http://www.mathworks.com/videos/generating-c-code-from- matlab-code-68964.html>
[15] Ríz N.: Implementácia analytického modelu malého prúdového motora do jeho riadiaceho systému, Diplomová práca TUKE LF, 2015
[16] Komjáty, M., Hocko, M.: Application of the Method of Small Alterations to Analyse the Fundamental Parameters of a Small Turbojet Engine MPM-20, In: Acta Avionica Journal. - September 2014. - ISSN 1335-94794. - Volume XVI, 29 – No. 1. Faculty of Aeronautics, Košice, Slovakia.
[17]Korba, P., Piľa, J., Főző, L., Cibereová, J.: The use of visualization in aircraft design nodes by

using CAX systems., In: SGEM 2014 : 14th international multidiscilinary scientific geoconference : GeoConference on Informatics, Geoinformatics and Remote Sensing : conference proceedings : volume 1 : 17-26, June, 2014, Albena, Bulgaria. - Sofia : STEF92 Technology Ltd., 2014 P. 399-406. - ISBN 978-619-7105-10-0