Simulation Complex for Debugging of Control Algorithms of the Test Bench

I A Shmidt¹, P V Ivanov¹, I N Gribkov²

¹Department of Microprocessor Automation, Perm National Research Polytechnic University, Perm, Russia
²JCS “UEC-Aviadvigatel”, Perm, Russia

E-mail: shmidt@msa.pstu.ac.ru

Abstract. The object of research and development is an air heating and delivery automatic control system of the test bench for testing of gas turbine engines. It is based on a test bench equipped with a gas turbine engine, a long air duct with dampers regulating the airflow, and the tested mechanism. It must maintain the required airflow characteristics and the air charge motor’s operating mode line. To debug the algorithms of the object of research, a simulation complex was developed, which can reproduce all the features of the test bench's behavior. The combination of the automatic control system and a simulation complex is a semi-natural model. Purpose is to develop a simulation complex for the debugging of the control algorithms and a mathematical model of the stand and the engine under test. As result, a simulation model based on the implementation of VeriStand technology was created. The program provides a uniform way to access heterogeneous models executed in Simulink. These models display different ways of describing the processes and features of the behavior of the modelled object. The simulation results can be compared to the behavior of a real system, which allows to debug the algorithms of the automatic control system. The developed simulation complex is currently used by specialists to make the mathematical models of the test bench components and the tested product more precise, as well as to debug the control algorithms.

1. Introduction. Air Heating and Boost Air Charging Automatic Control System Research Objective

The object of the research is the air heating and boost air-charging automatic control system (ACS) for the test bench of the gas generator of a gas turbine engine (GTE). The main idea of the stand consists of the gas generator, the most stress-bearing part of the GTE, being tested separately from the rest of the engine. A simplified layout of the stand is shown in figure 1.
Figure 1. Simplified layout of the air heating and boost air charging ACS test stand.

The air heating and boost air charging mechanism is a breadboard gas turbine engine, which boost charges air into the air duct, where dampers regulate the airflow, the tested gas generator is located at the air duct outlet [1].

The bench air heating and boost air charging ACS must control crucial characteristics of the airflow at the gas generator inlet: temperature, pressure, airflow at the inlet to the tested mechanism. The priority, in relation to the adjustable parameters, is the regulation of the position of the breadboard motor operating point relative to its operating mode line (Figure 2). Data is presented in dimensionless units.

Figure 2. Breadboard engine operating mode line.

This position characterizes the state of the system at the current time; the ACS must in any case ensure that the operating point is in the zone of the operating mode line [2, 3].

To maintain the required parameters, the air heating and boost air-charging ACS must regulate the airflow using dampers, as well as control the changes of the breadboard engine's mode by setting its fuel consumption.

Since the control criteria for the air heating and boost air charging ACS are initially nonlinear, as well as the controlled process itself, the ACS algorithms are based on fuzzy sets. The debugging of such regulators for all possible operating modes of the gas generator and the breadboard engine, taking into account the “non-standard” operation process of stand's individual subsystems, is a non-trivial task.

To debug the algorithms of the automatic control system, it was decided to implement a semi-natural stand, in which the ACS is a realistic element, and the components of the air heating and boost air charging system and the tested gas generator are presented in the form of mathematical models.

1.1. Literature review

The issues covering the construction semi-natural models for debugging ACS algorithms in relation to GTE tests have been repeatedly considered in the literature.

The article [4] gives an overview of the formation of semi-natural modeling stands intended for the ACS-GTE testing.
Various aspects of using semi-natural gas turbine engine stands for the debugging of the control algorithms are covered in publications of 2017-2020. Both the electronic part of the ACS [5-7] and the hydro mechanical unit [8, 9] are thought of as the object of the bench tests. Summarizing the structures of semi-natural stands covered in these publications, a universal generalized layout of such a stand can be represented graphically (Figure 3).

Figure 3. Generalized block-diagram of a semi-natural GTE stand.

GTE ACS (electronic, hydro mechanical) is present on the stand in full scale, and GTE is reproduced by the means of a mathematical model. The ACS supplies control actions to the model input. The output parameters calculated by the model are supplied back to the GTE ACS through the sensor simulators. For the hydro mechanical part of the ACS, the GTE shaft rotation is additionally reproduced by an electric drive.

In articles [10-13] various aspects of the construction of the mathematical engine models for semi-natural stands are covered. Either element-by-element GTE models or simplified models based on dynamic characteristics are proposed. Of particular interest is the technique proposed in [13] for the construction of the mathematical gas turbine engine models in the form of recurrent neural networks. The analysis of the accuracy and adequacy of the developed models is carried out. A detailed classification of dynamic modeling methods for the modelling of a gas turbine with an accentuation on levels of detail is presented in [14].

Much attention in the available sources is paid to the display and interpretation of the semi-natural bench test process [15].

In relation to our problem, the proposed approaches can be used in the development of models of stand components such as breadboard engine and gas generator.

2. Semi-natural Stand Structure
Summarizing all the above-mentioned publications, we can conclude that there are no semi-natural stand prototypes that directly correspond to our goals. All of the available stand variants are made, one way or another, with the purpose of performing GTE ACS tests and are reduced to the structure shown in Figure 3.

Below is a block diagram of a semi-natural stand designed specifically for the debugging of the ACS algorithms of the test stand designed for GTE gas generators testing (Figure 4).
Figure 4. Block diagram of a semi-natural stand designed for the debugging of the air heating and boost air charging ACS algorithms.

Air heating and boost air charging ACS is the realistic tested object, executed in the form of a separate controller, and the components of the stand (breadboard engine, an air duct with dampers, a gas generator with its own regulator) are represented by their mathematical models.

2.1. Simulation complex task description, its structure and principle of operation
The simulation complex must provide a complete simulation of the test bench and the automatic control system work. The following functionality should be available to the tester:

- Control elements for gas generator fuel consumption setting.
- Control element imitating a joystick of the breadboard engine for fuel consumption control.
- Control elements for damper position regulation.
- Control elements for atmosphere parameter setting during testing.
- Information window with the operating mode lines and the position of the operating point.
- Information indicators of the system parameters: gas generator fuel consumption, gas generator air consumption, engine air consumption, etc.

Air heating and boost air charging ACS work process must be checked both in the test bench full stimulation mode, and in the manual setting mode of certain parameters.

Examples of such simulation stand implementation are given in [16, 17]. The first publication covers one of the mathematical modeling methods for the control system and its units. Mathematical modeling was carried out using the MATLAB.Simulink package; shell programming was performed with the use of NI LabVIEW graphical programming package. The second work considers an approach based on the use of simulation models, which receive information from the object in real time with the help of a SCADA-system, i.e. the focus is placed on the organization of the user interface.

Our complex is structurally composed of two blocks: a block that provides user-direction interface, and a block that particularly simulates the test bench components.

To implement the block, which directly performs the modeling, it was decided to utilize the models created with the use of MATLAB.Simulink and compiled into dll format. NI Veristand software and NI Real Time libraries were used to upload these models into the LabVIEW environment. Examples of such use of the Veristand environment for the creation of real-time applications were considered in [18]. The study shows that MATLAB.Simulink to NI VeriStand model conversion can significantly speed up the modeling process.
2.2. Implementation of a shell for the mathematical model

The basis of everything is a shell for the mathematical model. The shell is an intermediate layer between the models of the stand components, represented as dlls, which were imported from MATLAB.Simulink, and the system user interface. In addition to the above-mentioned basic functionality the shell should have one more feature – a uniform reference to various mathematical models.

Since the air heating and boost air charging stand practically consists of 3 elements: an engine, an air duct with dampers and a gas generator, the mathematical model can be presented in many different ways. Three particular mathematical models are possible, with different or, on the contrary, equal calculation steps, they can also combine several elements of the stand, etc. To implement this feature, it was decided to structure the shell as follows, as shown in Figure 5.

![Figure 5. Shell structure of the mathematical model.](image)

The shell contains one special part – an adapter for the mathematical model. It consists of two elements: a model initializer and an object, which sends control actions and receives output data. Both of these elements are designed as NI LabVIEW virtual instruments (VIs). The adapter for each type of model is made differently, however, uniformity is still maintained in the following way. The connections between the initializer and the data exchange object, as well as between the exchange object and indicators, are implemented with the use of unified clusters, i.e. they are always the same for all adapter versions, which makes it possible not to touch the entire project, but to apply changes to only two files. In addition, for the ease of implementation it was decided that the parameters, which are equal in different versions of the model, should have the same name, for the possibility of performing an easier search by name and for the further time reduction for changing the adapter for each separate version of the mathematical model of the stand.

As it was mentioned earlier, a problem of the difference in different model calculation steps also poses a difficulty, i.e. each version of the model can have different calculation steps, for example, 0.001 or 0.0001, in addition to this, if the model consists of several dll combinations, then these dlls can also have different calculation steps.

This raises two problems. Firstly, at sufficiently small steps of the calculation, performance drops significantly and the model starts to perform much slower than it should be. Secondly, if the model is represented by several dlls, then a discrepancy arises during operation. These problems were solved by the introduction of an internal cycle into the data exchange element, which performs the required number of iterations for each dll, so that the parts of the model work correctly. Thus, for example, if we have two parts: the first with a calculation step of 0.001 seconds, and the second part with a step of 0.0001, then for the second part the element will execute a cycle in which there are 10 times more iterations than in the premiere cycle for the first part. Such an introduction of cycles into the exchange significantly increases productivity, since the number of iterations for the main file is reduced, and most of the calculation is carried out inside the adapter part, which is much less resource-intensive for the performing of cyclic operations. Despite the fact that the models are calculated with different steps, the step outputs the calculated parameters to the outside always takes 0.1 seconds, which is more than enough for the testing of our regulator.

The adapter looks like this on the NI LabVIEW block diagram (Figure 6).
Figure 6. Implementation of the two adapter parts on the block diagram.

Here, the model initializer and the VI that provides input / output of parameters to the model are called InitialDLL and In/OutDLL. The cluster shows the output parameters of the model.

The adapter takes up very little space, but it is of very big importance. As it was described above, there are two objects with unified clusters emplaced between them. On the right is an element that receives all the necessary parameters from the unified cluster, which are the same for all combinations of models, which ensures uniform treatment. This feature makes the use of this complex for various combinations of mathematical models very convenient.

The models themselves are executed by a modeling specialist in the MATLAB.Simulink program and compiled into dll-libraries, which then are placed in a specific directory. The shell's, and more specifically the adapter's objective is to load these models and supply them with the necessary parameters: effects on the damper, fuel consumption of the breadboard engine and gas generator, as well as the atmospheric parameters: air pressure and temperature. Such separation from the application of the models in the form of dlls allows to quickly change the models depending on the need to study in detail one or another feature of the stand systems behavior.

In addition, the simulation complex implements the ability to debug the model itself, i.e. it is possible to compare simulation results with the results obtained on a real bench. When checking the model, control actions can be received both from the ACS and from the shell itself in manual mode. This allow to check the model's certain parameters at different operating points without starting up the control system, which significantly speeds up the verification process. When the ACS and the automatic mode are turned on, this feature disables itself.

When working as a part of a simulation complex, the shell (adapter) receives the model output parameters and displays them for the user in a convenient form – it organizes the user interface.

Front panel of the shell, i.e. the user interface is shown in Figure 7.
Figure 7. The front panel of the shell for the mathematical model of the air heating and boost air charging ACS.

On the left the control elements are located:

- The block “Atmosphere params” allows to set the parameters of the atmosphere.
- “GG Control” sets the parameters of the gas generator working process.
- “D30 Joystick” signifies the panel for the control of the breadboard engine fuel delivery.
- Below are the blocks for the control of the air duct dampers.

To the right of the control elements, all required information parameters are displayed: a graph with the operating mode lines and the position of the operating point, indicators displaying the most informative parameters: breadboard engine and gas generator air flow rate, temperature parameters, pressure parameters, etc.

2.3. The results
Engineers have already used the following simulation complex for a long time. It helps to make adjustments to the mathematical models of the test bench components and aids with the debugging of the control algorithms.

This project makes it possible to debug the air heating and boost air-charging ACS both in the manual mode and in the automatic mode, which provides an opportunity to study the controller's response to various control actions narrowly. At the same time, we check the regulator's reaction to the behavior of the test bench, which fully simulates the real gas generator tests.

References
[1] Inozemtsev A A, Gallyamov M D, Dvinskikh A V, Gribkov I A and Polulyach AI 2017 Abstract of invention «Testing bench for gas generators of turbojet bypass engines» (Federal service for intellectual property) pp 1-7
[2] Akimov V M, Bakulev V I, Kurziner R I, Polyakov V V, Sosunov V A and Shlyakhteiko S M
1987 *Theory and calculation of air-jet engines*, ed. Shlyakhtenko S M (Moscow: Mechanical Engineering), p 568

[3] Kulagin V V 2003 *Theory, calculation and design of aircraft engines and power plants* (Moscow: Mechanical engineering) p 616

[4] Denisova E V and Chernikova M A 2019 *Modern high technologies* 7 122

[5] Pleshivich A S, Zaborskikh A A and Fatykov A I 2017 *PNRPU bulletin*. 22 90

[6] Volkov D I, Komarov V P and Nerubaskiy V V 2015 *Bulletin of engine building* 92

[7] Kulikov G G, Arkov V Yu, Fatikov V S, Abdulnagimov A I and Pogorelov G I 2009 *Bulletin of the Samara State Aerospace University* 3 392

[8] Shendaleva E V 2017 *The Russian automobile and highway industry journal* 3 111

[9] Shendaleva E V 2016 *Omsk scientific bulletin* 3 39

[10] Kuznetsov A V and Makaryants G M 2017 *Samara University Bulletin. Aerospace equipment and mechanical engineering* no. 2 16 65

[11] Kavalerov B V, Petrochenkov A B, Odin K A and Tarasov V A 2015 *Russian Electrical Engineering* 86 no 6 pp 331-338

[12] Shendaleva E V 2012 *The Russian automobile and highway industry journal* no. 5 106

[13] Abdulnagimov A I and Ageec G K 2019 *USATU bulletin* no. 4 23 115

[14] Schobeiri M T 2017 *Gas Turbine design, components and system design integration* (Springer) part 18 pp 477-509

[15] Andre Kaufmann 2014 *Forschung im Ingenieurwesen* 1-2 78 45

[16] Blyumin K V 2012 *Bulletin of the Samara State Aerospace University* no. 3 75

[17] Krivocheev I A and Sukhanov A V 2014 *USATU bulletin* no. 2 18 134

[18] Alikas D V, Matveeva A S and Mayzel A V 2017 *SPBU Science Week. Materials of a scientific conference with international participation. Institute of Physics, Nanotechnology and Telecommunications* 159