

*The paper deals with the development of a motion correction system for an unguided rocket of a certain class. The existing method for calculating the rocket flight range based on the dependence of flight path on the initial rocket pitch angle and average parameters of disturbing effects does not provide the required accuracy of the specified flight range. This is mainly due to uncontrolled range wind deviations in the flight area. At the same time, conducting test rocket launches to identify dispersion characteristics and improve the target accuracy leads to significant material costs. Therefore, computer simulation is the most promising approach for studying rocket dispersion laws and developing a motion correction system for the unguided rocket on this basis. When developing the correction system, classical differential equations were used describing the aerodynamics of a variable-mass rocket, as well as adaptive control methods with a reference model. As a result of the study, a method for recording a program that implements the reference model on the Arduino Due platform was developed. A general Simulink model that simulates the process of rocket flight path correction was built. A half-scale rocket flight model with a correction block was developed. Based on this model, a series of experiments were carried out, which showed a high degree of rocket target accuracy due to rocket path correction. The results make it possible to take the developed system as a basis for developing a practical path correction system for Grad rockets*

*Keywords: flight path correction, reference model of rocket motion, rocket, half-scale simulation*

UDC 62-503.51

DOI: 10.15587/1729-4061.2021.228146

# DEVELOPMENT OF A HARDWARE AND SOFTWARE MODEL OF A ROCKET MOTION CORRECTION SYSTEM

**Talgat Atygayev**

Corresponding author

Postgraduate Student\*

E-mail: atgv95@gmail.com

**Victor Ivel**

Doctor of Technical Sciences, Professor\*

**Yulia Gerasimova**

PhD, Associate Professor\*

\*Department of Energetic and Radioelectronics

M. Kozybayev North Kazakhstan University

Pushkina str., 86,

Petropavlovsk, Kazakhstan, 150000

Received date 01.04.2021

Accepted date 18.05.2021

Published date 30.06.2021

**How to Cite:** Atygayev, T., Ivel, V., Gerasimova, Y. (2021). Development of a hardware and software model of a rocket motion correction system. *Eastern-European Journal of Enterprise Technologies*, 3 (3 (111)), 15–23. doi: <https://doi.org/10.15587/1729-4061.2021.228146>

## 1. Introduction

Multiple launch rocket system (MLRS) is a weapons system that contains a multiple rocket launcher and guided and unguided rockets, as well as auxiliary means such as transport or transport-loading vehicles and other equipment. MLRS refers to rocket weapons, such systems are in service with the ground forces, air forces and navies of many countries [1].

Modern MLRS have a rocket caliber of up to 425 millimeters, a maximum firing range of up to 45 kilometers or more and carry from 4 to 50 rockets. Each rocket has an individual launcher.

The multiple launches of the MLRS determine high fire capabilities and the possibility of simultaneously hitting targets in large areas, which, together with the suddenness achieved by volley fire, provides a high impact on the enemy.

One of the main shortcomings of rocket artillery systems is rather high rocket dispersion. At present, flight path correction systems are installed on rockets to eliminate this drawback [2–5].

Improvement of rocket accuracy can reduce the amount of ammunition, accelerate the pace of war, reduce material consumption and the needs of military logistics, and also minimize damage to civilians and non-military facilities during war [6].

Currently available rocket motion correction systems make it possible to purposefully affect the flight path. This eliminates aiming errors, consequences of disturbing effects

of the atmosphere, initial launch conditions and other factors that deviate the rocket from the target. As a result, the firing accuracy of the rocket with the correction system can be significantly increased, as well as the firing range due to unpowered gliding and, thus, the efficiency of solving various combat tasks [7].

Analysis of scientific publications and inventions on the topic of the present study showed that all the proposed systems, devices and algorithms for improving the flight characteristics of unguided rockets provide rather complex solutions that require significant modifications to the rocket design, such as fuze delays, rocket nose deflections, etc. At the same time, a fairly wide range of issues remains without due attention, for example, analysis of flight characteristics at unsteady aerodynamic parameters or synthesis of optimal correction systems for unguided rockets, requiring minimum design modifications while ensuring maximum firing accuracy. All this indicates the relevance of the problems under consideration and requires additional research involving modern computer modeling systems.

## 2. Literature review and problem statement

The paper [8] considers a correction system using a timer. The built-in timer records the flight time from the moment of firing. Control and correction operations are performed

according to a given algorithm at set time points. Air pressure and temperature in the rocket flight area are continuously measured and recorded as a function of time, and the time point  $t$  at which the rocket engine stops working is recorded. Flight parameters are calculated taking into account air pressure and temperature in the rocket flight area. The actual flight path of the rocket is determined. A computing device generates commands for the rocket flight controller. Thereby, the path is corrected and the rocket dispersion ellipse is reduced.

The disadvantage is that under radio interference, it is not possible to enter data on the actual rocket path using GPS.

In [9], electronic coordinates of the object are entered into the onboard system into the rocket memory before firing. During flight, the rocket receives information about its path through a global navigation system, such as global positioning system (GPS), and, by means of fuze delays, corrects the flight path.

Delays are complex electromechanical devices that seriously complicate the fuze design and have large dimensions, so it is necessary to reduce the size of the rocket warhead, which reduces its efficiency. Such delays generate radio interference arising from the friction of the moving parts, which leads to false actuations of proximity fuzes, and, as a consequence, to rocket path detonations.

The paper [10] considers the development of a deflectable-nose rocket. Rocket nose can be deflected at a certain angle relative to the rocket body axis, including the pressure difference between the windward and leeward sides of the warhead and creating an appropriate aerodynamic control force. Rocket rotation is not taken into account, which leads to deviations from the flight path.

The paper [11] presents the results of a study of path correction based on the stabilization of an infrared image sensor. The system starts working when the rocket passes the peak of the flight path. A magnetometer sensor measures the actual rocket roll angle, then the angle is transmitted to the computer carrying the rocket. To ensure that the infrared image sensor always remains stationary relative to the engine-controlled geodetic coordinate system, the middle and front parts of the fuze rotate at the same speed relative to the rocket. This process will be repeated many times until the rocket enters the final path and the infrared image sensor starts working. Due to rocket micro-rotation, the image on the infrared sensor is not stable, and in dim illumination, the fuze cannot accurately detect the target.

Therefore, the analysis of the above studies shows that the performance of the systems is affected by various external factors, which, in turn, increases deviation from a given path and leads to premature rocket explosion [12].

One way to improve the firing accuracy is to correct the ammunition path to parry disturbing factors and, consequently, to compensate for misses [13].

The creation of various modeling systems has already become a traditional approach to solving many serious applied problems in conducting expensive full-scale experiments [14].

Mathematical modeling and simulation allow carrying out a wide range of studies to analyze the influence of various factors on the firing process [15]. Possible emergency situations, such as critical rocket path deviations, stability loss due to rocket mass asymmetry, critical rocket deviations due to wind are analyzed [16].

Such rockets, especially guided ones, are very expensive and cannot be reused. All this suggests it expedient to conduct a study on the development of a software and hardware complex for conducting experiments without using real rockets.

### 3. The aim and objectives of the study

The aim of the study is to develop an onboard range correction system for multiple launch rocket systems (MLRS) in service with the ground forces.

To achieve this aim, the following objectives were set:

- to analyze the rocket flight without path correction;
- to develop a rocket range correction system using the reference model of the correction object;
- to build a general Simulink model, including rocket flight correction and simulation systems;
- to develop a reference model on the Arduino Due platform (Italy) using the Simulink Support Package for Arduino Hardware and create a program for implementing the model in C++;
- to simulate the rocket launch process in real time using the Simulink Desktop Real-Time packages.

### 4. Materials and methods for studying unguided rocket motion correction

The object of the study is the aerodynamic processes of unguided rockets on the path.

The subject of the study is a mathematical model of unguided rocket motion.

The mathematical model of motion was developed based on an unguided Grad rocket, additionally equipped with a controllable fin assembly. The specifications presented in Table 1 were used in the simulation.

Table 1

Specifications of an unguided Grad rocket

Characteristics	Value
Rocket caliber, mm	122
Rocket length, mm	2,870
Rocket mass, kg	66
Fuel mass, kg	20.5
Fuel combustion time, s	1.67
Engine thrust, N	23,600
Moments of inertia [ $J_{xx}, J_{yy}, J_{zz}$ ], kg·m <sup>2</sup> before and after fuel exhaustion	[0.1499–0.1238; 41.58–33.83; 41.58–33.83]
Initial rotational speed, rad/s	[37; 0; 0]
Initial speed, m/s	26.7

The research methods used in solving the problems are based on the methods of mathematical modeling, computational aerodynamics and simulation.

### 5. Results of the study of the model of the onboard rocket motion correction system

#### 5.1. Analysis of rocket flight without path correction

Rocket motion in the vertical plane is described by the system of differential equations [17]:

- for the speed coordinate system  $X_p O Y_p$ :

$$F_{xp} = m \frac{dV_{xp}}{dt} = T(t) \cos \alpha - D - W \sin \gamma; \tag{1}$$

$$F_{yp} = (L + T \sin \alpha) \cos \mu - W \cos \gamma; \quad (2)$$

$$D = qSC_R = 0.5\rho V_{xp}^2 SC_R = V_{xp}^2 C_D; \quad (3)$$

– for the earth coordinate system  $XOY$ :

$$\dot{X} = V_{xp} \cos \gamma; \quad (4)$$

$$\dot{Y} = V_{xp} \sin \gamma; \quad (5)$$

$$V_{xp} = \int \frac{F_{xp}}{m} dt. \quad (6)$$

Here:

–  $F_{xp}$  – resultant force along the rocket velocity  $V_{xp}$ ;

–  $m$  – rocket mass;

–  $m = m_0 - \int_0^t \dot{m}_s dt$ , where  $m_0$  – mass of loaded rocket,

$\dot{m}_s$  – fuel mass flow per unit of time;

–  $T(t)$  – rocket thrust,  $T(t) = T_0$  for  $t \geq t_c > 0$  and  $T(t) = 0$  for  $t < t_c$ , where  $t_c$  – fuel combustion time;

–  $\alpha$  – angle of attack;

–  $D$  – drag;

–  $q$  – free-stream dynamic pressure;

–  $S$  – rocket reference area;

–  $C_R$  – dimensionless aerodynamic coefficient, depending mainly on rocket shape, Mach number  $M = V/a$ , Reynolds number  $Re = Vd\rho/\mu$ , angles of attack  $\alpha$  and slip  $\beta$ ;

–  $C_D$  – coefficient having a dimension of n/m;

–  $W$  – rocket weight;

–  $\gamma$  – pitch angle;

–  $F_{yp}$  – normal resultant force along the rocket velocity  $V_{yp}$ ,

$$F_{yp} = m\dot{V}_{yp} = m\dot{V}_{xp}\dot{\gamma} \text{ or } \gamma = \int \frac{F_{yp}}{mV_{xp}};$$

–  $L$  – lift,  $L = C_L V_{xp}^2$ , where the coefficient  $C_L$  differs from  $C_D$  by a numerical value, also has a dimension of n/m and at small angles of attack  $\alpha$  becomes proportional to the angle of attack;

–  $\mu$  – roll angle;

–  $[\dot{X}, \dot{Y}]^T$  – translational rocket velocities.

The mathematical model of rocket motion in the vertical plane can be represented as a block diagram in the Matlab Simulink editor (USA) using standard blocks and subsystems of the Aerospace Blockset. The block diagram (hereinafter Simulink model) (Fig. 1) is fully adequate to the differential equations (1)–(6) and provides high accuracy of solutions to the system of equations [18] for different parameter values.

Subsystem 1 implements equations (1) and (2), and subsystem 2 implements equations (4), (5) and (6). The rocket parameters presented in Table 1 were used to create the model. Separately, Fig. 2 shows the «Drag» subsystem implementing equation (3). The «Lift» subsystem is not considered as it is similar to the «Drag» subsystem. All other blocks fulfill the functional relationships shown above in the list of symbols.

When developing the «Drag» subsystem, it was considered that in calculating the wind drag effect, only range wind should be taken into account, i. e.  $VW_{xp} = VW \cos \gamma$ . The subsystem also includes a standard block for calculating air density at the rocket height. The existing technology for field rocket launching assumes the initial setting of the pitch angle  $\gamma_0$  according to the reference book, which contains tables of the angle  $\gamma_0$  depending on meteorological parameters in the flight area  $P$ , primarily average wind velocity. Fig. 3 shows the rocket flight curve obtained on the  $XY$  plot at an average headwind velocity ( $VW$ ) of 10 m/s. According to the reference book, the pitch angle  $\gamma_0 = 0.785$  rad is set [19].

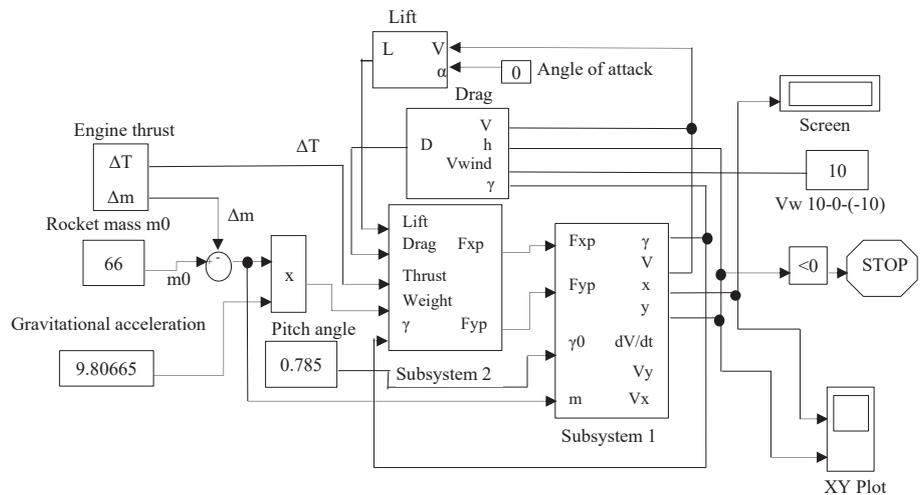


Fig. 1. Simulink model of rocket motion in the vertical plane

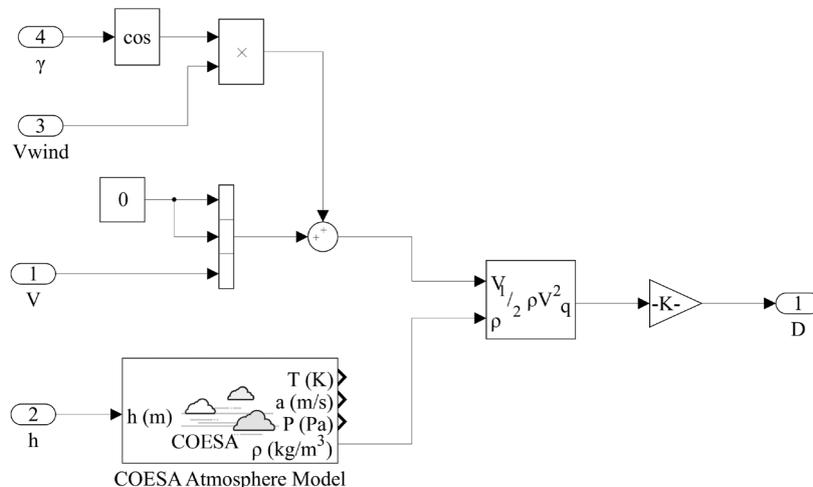


Fig. 2. «Drag» subsystem

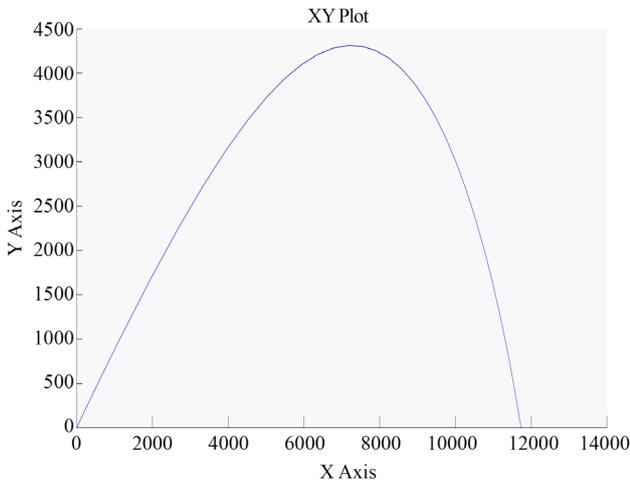


Fig. 3. Rocket flight curve in the range-altitude coordinates

The rocket range according to the screen readings was 11,740 m. If the actual headwind velocity reaches 12 m/s, the rocket range will decrease to 11,640 m, i. e. the error will be 100 m.

**5. 2. Development of the rocket flight model with the range correction system using the reference model**

When constructing the block diagram (Fig. 3) of the rocket flight model with the range correction system, the software implementation of the Simulink model shown in Fig. 1 was used as a reference model.

The block diagram (Fig. 4) includes two main blocks: the onboard correction system and the correction object.

The onboard correction system (OCS) in Fig. 3 includes:

- RM – reference model block implemented on the Arduino Due platform (Italy) programmed using the Simulink Support Package for Arduino Hardware (USA);
- $f = \sqrt{x_m^2 + y_m^2}$  – vector module of the rocket reference model motion in the vertical plane;
- AO – Arduino Due digital-to-analog converter (DAC) (Italy);
- AI1, AI2 – analog-to-digital converters (ADC) of the PCI 1710HQ board (China);
- CB – comparison block of two signals;
- $f = \sqrt{x^2 + y^2}$  – vector module of the rocket model motion in the vertical plane;
- I – integrator;
- SSp – rocket model speed sensor (full speed horizontal projection);
- SAI – rocket model altitude sensor (usually an atmospheric pressure sensor, since atmospheric pressure is proportional to the object altitude);
- Simulink Desktop Real-Time package (USA), which connects the Arduino Due board (Italy) to the general Simulink model.

The Simulink Support Package for Arduino Hardware allows recording (as a program) the reference model built in the Simulink computer environment via a USB interface to the Arduino platform (Italy). After that, every time the Arduino board (Italy) is powered, the reference model program will automatically start. The correction object (OC) or rocket flight model shown in Fig. 3 includes:

- RB – rocket booster;
- L – lift generation block;
- FBI – functional block implementing equations (1)–(5).

- In Fig. 4, the following signal designations are adopted:
- $V_w$  – range wind as an external disturbing effect;
  - $\gamma_0$  – initial pitch angle;
  - $x$  – horizontal coordinate;
  - $y$  – vertical coordinate;
  - $V$  – full rocket velocity;
  - $f_m$  – vector module of the model motion in the vertical plane;
  - $f$  – vertical plane;
  - $T_0$  – rocket launch thrust;
  - $D_w$  – drag;
  - Lift – lift;
  - Thrust – rocket thrust;
  - $\gamma$  – pitch angle (angle between the rocket longitudinal axis and the horizontal plane).

The operation principle of the block diagram of the rocket flight model is as follows.

The rocket booster is started simultaneously with the launch of the program on the Arduino Due platform (Italy). The vector modules of the horizontal and vertical motion of the rocket reference model are removed from the RM block, then the vector module of the rocket reference model motion in the vertical plane  $f_m$  is formed. This signal is fed through the DAC and ADC blocks to the negative input of the CB. The vector module of the rocket model motion in the vertical plane  $f$ , which is formed from the signals  $x$  and  $y$ , enters the positive input of the CB. The signal  $x$  is generated by integrating the signal  $V_x$ , which is measured by the speed sensor SSp. The signal  $y$  is measured by the altitude sensor SAI. As a result, the  $\Delta_{xy} = f - f_m$  difference signal is generated at the CB output, which is fed to the L block through the AI2 ADC. Thus, the fin assembly (FA) turning angle control signal is generated. The angle of attack  $\alpha$  is formed in proportion to the FA turning angle, which changes the rocket flight path.

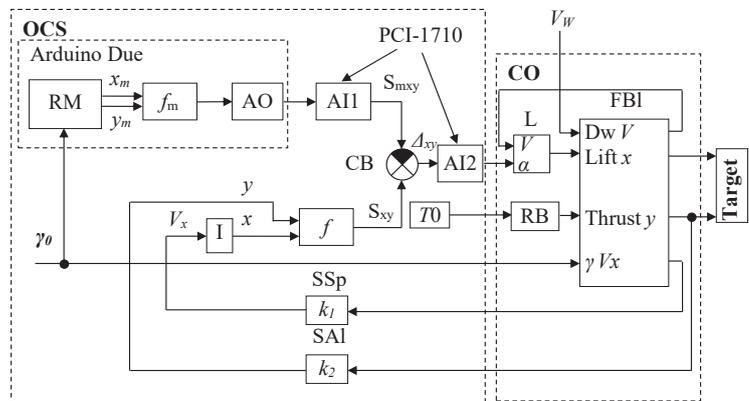


Fig. 4. Block diagram of the rocket flight model with the correction system

As a result, a negative feedback is created, which tends to minimize the  $\Delta_{xy}$  difference signal, i. e. to bring the actual flight path closer to that of the rocket flight reference model.

In a real unguided rocket launch system, the onboard correction system, including the reference model, will be represented by a program recorded on a microcontroller. The prelaunch configuration of the reference model and setting the initial values of the OCS parameters will be carried out via the local WiFi network.

Two options for implementing the OCS are considered: the first is using the STM32F7xx microcontroller (Switzerland)

with a Cortex M7 core (England) together with a WiFi receiver on a separate chip, and the second option is using a dual-core system on an ESP32 chip (China) with built-in WiFi 802.11 support, and Wi-Fi transmitters of this series create a local Wi-Fi network and do not need an external Wi-Fi network.

Preparation for rocket launching and launching itself include setting the parameters of the reference model and the initial pitch angle in accordance with the specified rocket range and weather conditions in the region.

### 5. 3. Building a general Simulink model including rocket flight correction and simulation systems

To determine the parameters of all blocks of the onboard correction system, it is advisable to build a general computer model including the rocket motion model with the rocket range correction block connected to it. The block diagram (Fig. 3) shall be used as the basis.

Fig. 5 shows the diagram of the computer model in Matlab. The diagram includes a Simulink reference model, which is a copy of the Simulink model shown in Fig. 1.

The rest of the diagram contains the same Simulink rocket flight model, supplemented with the elements of the correction block: blocks  $f_m$  and  $f$ , integrator I, comparison block and signal  $\Delta\alpha$  shaper. The signal  $\Delta\alpha$  is generated in the block  $f(\alpha)$ , including an electromechanical amplifier with the controllable fin assembly and representing a low-pass filter  $f(\alpha) = k/(Tp + 1)$  [17]. The error signal from the filter output is fed to the input  $\alpha$  of the «Lift» block. This signal proportionally changes the rocket angle of attack by turning the controllable fin assembly [17] thereby proportionally changing the coefficient  $D_L$  in the formula  $L = D_L V_{sp}^2$  (Section 5. 1). As a result, the difference between the specified and actual rocket flight paths is minimized.

Test runs of the general Simulink model were carried out at various wind velocity deviations set in the «Wind velocity» block of the rocket flight model from the wind velocity set in the «Wind velocity» block of the reference model. Based on the results of these numerical experiments, Table 2 was compiled.

The  $\Delta V_W$  line shows possible wind velocity deviations from that set in the «Reference model» subsystem, i. e. 10 m/s. This velocity corresponds to the rocket range of 11,740 m. The  $X_R$  line shows the rocket range values without the correction system. The  $X_{RC}$  line contains the rocket range values with the correction system.

Table 2

	Rocket range								
$\Delta V_W, \text{ m/s}$	-12	-9	-6	-3	0	3	6	9	12
$X_R, \text{ m}$	12,260	12,100	11,950	11,840	11,740	11,700	11,654	11,540	11,400
$X_{RC}, \text{ m}$	11,780	11,775	11,768	11,750	11,740	11,735	11,710	11,690	11,675

Analysis of the results of model experiments suggests that the correction system significantly increases the rocket flight accuracy, which allows excluding test rocket launches in the field.

### 5. 4. Development of software for the reference model implemented on the Arduino Due platform

Fig. 6 shows the Simulink model of the correction block, which is a visual representation of the program that must be recorded to the microcontroller of the Arduino Due platform (Italy). During implementation, the program provides an output of the rocket flight simulation signal through the Analog Output module from the Simulink Support Package for Arduino Hardware.

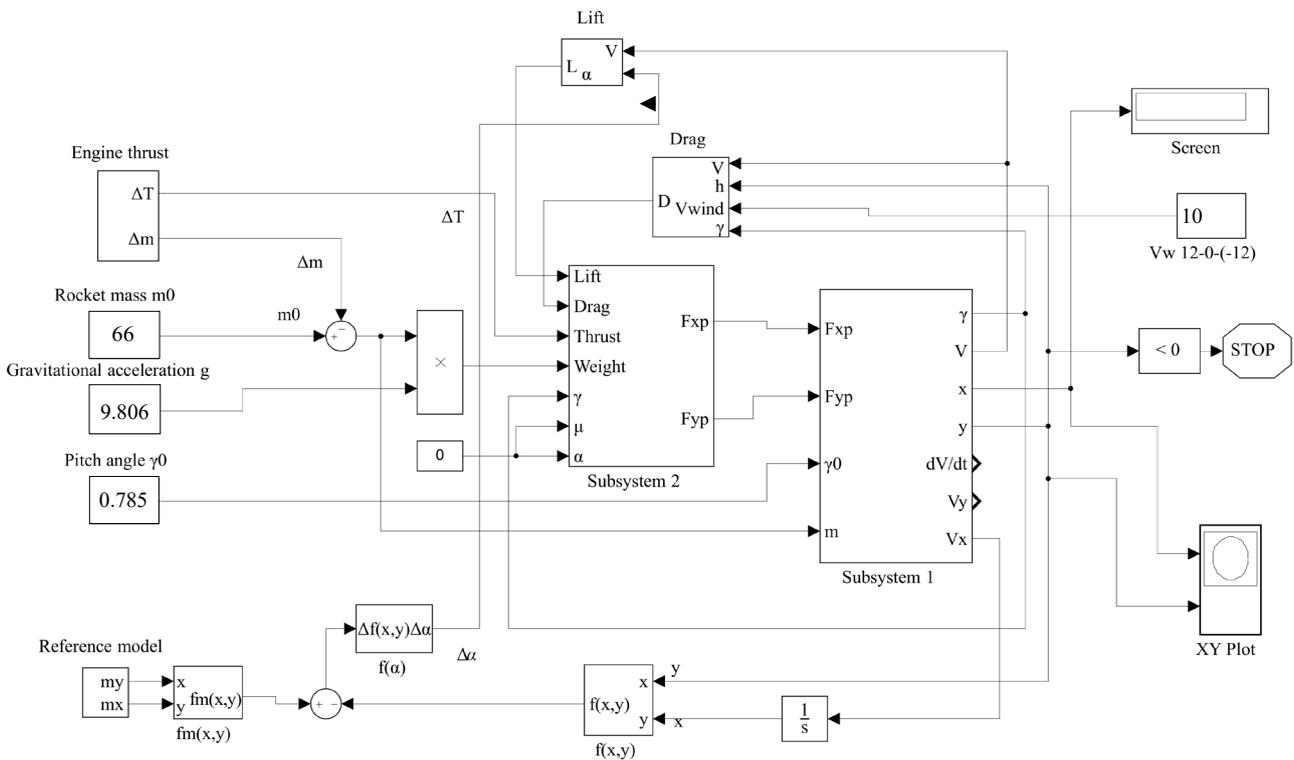


Fig. 5. General Simulink model including rocket flight correction and simulation systems

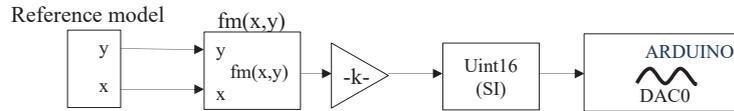


Fig. 6. Simulink model of the correction object reference model

The technology for loading the program executed by the presented Simulink model includes the following operations. On the main panel of the Matlab system, on the Add Ons tab, it is necessary to download the Arduino hardware support package and install drivers for the Arduino Due board. Then, configure the hardware, including opening the Arduino model, installing the COM port, choosing the encoder – Embedded Coder and programming language – C++. Next, connect the Arduino Due and computer platforms with a USB cable. Then the program loading process can be started by selecting the Build, Deploy & Start tab in the model window menu.

As a result, a script (executable program) is written in C++ to the Arduino flash memory and to the C:\Program-Data\...\model\_ert\_rtw folder, including various header and other executive files. Using the resulting script, it is possible to program individual microcontrollers and platforms that are not included in the list of the Simulink Support Package for Arduino Hardware and use them as a reference model in the rocket launch system.

Fig. 7 shows the components of an auxiliary system for preliminary verification of the executable program of the reference model. The general Simulink model of rocket flight is presented on the monitor screen. At the bottom of the screen is the Arduino Due board, connected with a USB cable to the computer. The analog output of the board is connected to an external oscilloscope. Two virtual oscilloscopes on the screen are connected as follows: one (upper oscillogram) – to the output of the rocket flight model, and the other (lower oscillogram) – to the output of the reference model. The plot of the output signal of the reference model implemented on the Arduino Due platform microcontroller is displayed on the screen of the real oscilloscope.

Thus, it is possible to compare the path of the virtual output signal (the upper waveform on the PC screen) and the path of the signal on the screen of the real oscilloscope.

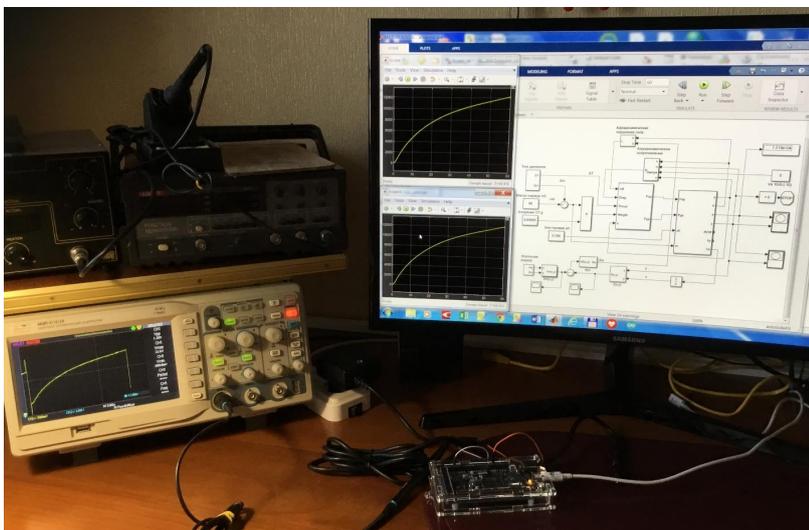


Fig. 7. Computer complex for studying the quality of the loading technology of the reference model program

Visual analysis allows determining the exact match between the diagrams obtained on the real and virtual oscilloscopes.

It follows that the verification of the proposed program loading technology gave positive results and it can be recommended for use in full-scale tests of rocket systems with flight path correction.

### 5. 5. Half-scale real-time rocket launch simulation

Fig. 8 shows a half-scale rocket launch and flight simulation model using the real software block of the reference model connected to the computer model of the control object, i. e., the rocket.

When developing the Simulink model (Fig. 8), the Simulink Desktop Real-Time package was used to ensure the real-time functioning of the model. The assembled system is shown in Fig. 9 in the form of a hardware and software complex, which includes:

- personal computer;
- PCI-1710HG I/O multifunction board (China), installed in the PCI computer bus;
- Arduino Due platform;
- ADAM-3968 SCSI connector (USA) to connect the analog output of Arduino Due to that of the PCI-1710HG board (China).

The hardware and software complex works as follows.

At the moment of starting the rocket flight simulation process, the correction block implemented on the Arduino platform is synchronously activated. The correction block generates an analog signal of the specified rocket flight path, programmed in accordance with measured range wind in the conditional test area. Then the signal goes to the analog-digital input of the PCI-1710HG board. In the Simulink model, this block is referred to as Analog Input. Next, the signal of the specified path is compared in the comparison block (CB) with the path signal obtained from the output of the rocket flight simulation system under conditions of a certain wind velocity deviation from the calculated value. The deviation range is set in the «Wind velocity» block. Further, the  $\Delta xy$  difference signal from the comparison block output passes through the low-pass smoothing filter  $f(\alpha)$  and enters the input of the «Lift» block. Here the signal model  $L = D_L(\alpha)V_{xp}^2$  is formed. A change in the angle of attack (at small angles of attack) leads to a directly proportional change in lift  $L$ . A change in  $L$  leads, in turn, to the correction of the rocket model flight path and minimization of the rocket model flight path deviation from the given one.

The computer monitor shows a virtual oscilloscope with a demonstration of rocket flight in the range-altitude coordinates. Table 3 shows the results of a number of experiments to determine the flight range with the wind velocity deviation from the calculated value.

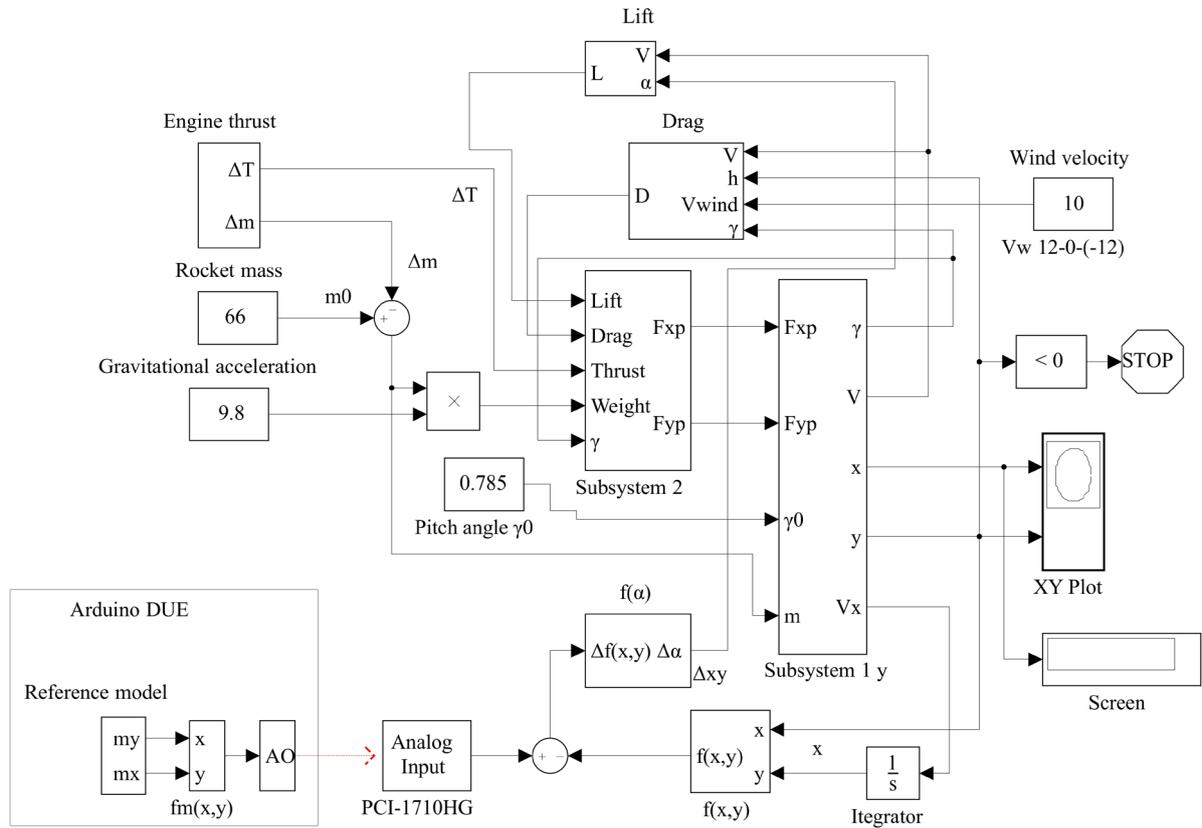


Fig. 8. Half-scale rocket flight model with the correction block

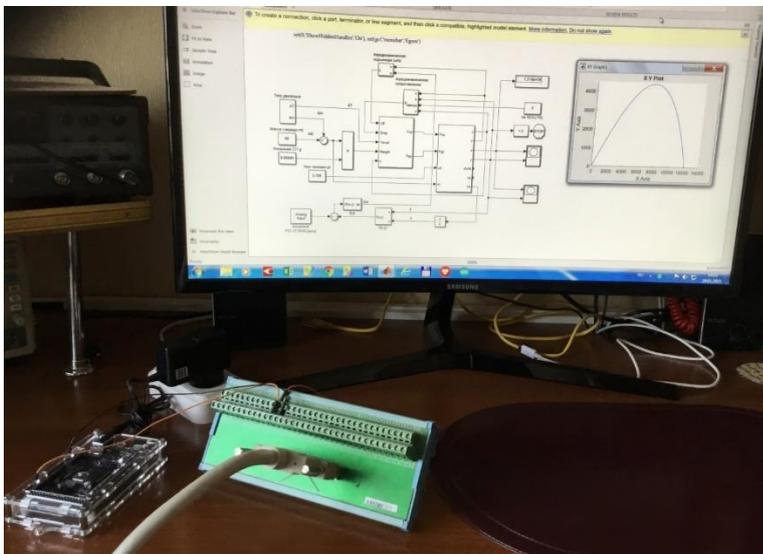


Fig. 9. Hardware and software complex

### 6. Discussion of the results of the software for the onboard correction system

Simulation of rocket flight without path correction in the Simulink system using the classical system of equations (1)–(6) confirmed that the rocket flight range is proportional to the initial pitch angle. The resulting model (Fig. 1) considers rocket motion in the vertical plane. The simulation results showed that the flight range is significantly affected by range wind. A wind change of 2 m/s leads to an up to 100 m error. In such conditions, it is necessary to use the technology of test rocket launches, which increases the cost and time of full-scale tests.

The inclusion of special aerodynamic blocks considering drag and lift effects in the computer model made it possible to use the obtained computer model in developing the rocket flight model using the rocket flight range correction system.

In the proposed block diagram (Fig. 4), the developed computer model was used twice: when constructing the rocket flight model and as a reference model of the correction system. Using the chain: the difference between the vector modules of the RM and rocket model motion (p. 10), the PCI-1710 board ADC, the L block input as a negative feedback made it possible to effectively control the rocket angle of attack and, ultimately, the flight range.

The general Simulink model based on the block diagram (Fig. 5), including the rocket flight correction and

Rocket flight range

Table 3

$\Delta V_W, \text{ m/s}$	-12	-9	-6	-3	0	3	6	9	12
$X_R, \text{ m}$	12,275	12,180	11,980	11,860	11,740	11,700	11,654	11,540	11,400
$X_{RC}, \text{ m}$	11,770	11,765	11,760	11,750	11,740	11,735	11,720	11,700	11,690

As can be seen from Table 3, rocket flight range with the OCS at different wind velocities remains almost unchanged.

simulation systems, allowed carrying out numerical experiments and provided comparative characteristics of errors during rocket launches with and without the flight range correction system (Table 2).

The proposed technology for downloading the RM simulation program to the Arduino platform is one of the ways to implement the RM (Fig. 6). By means of specialized Simulink packages, Arduino Due board, oscilloscope and PC, the quality of the real-time loading and implementation technology for the RM program was studied (Fig. 7). Tests showed the adequacy of the recorded RM program and the program implemented as a Simulink model of the RM.

As a result of the study, a half-scale rocket flight model with the correction block was developed (Fig. 8). The proposed model differs from that in Fig. 5 in using the Arduino Due platform as an OCS, the presence of multifunction PCI-1710HG board and real-time package of the Simulink system. The resulting hardware and software complex (Fig. 9) allowed estimating the operation of a real onboard correction system together with the rocket flight computer model under conditions close to full-scale tests. The experimental results in Table 3 almost do not differ from the results in Table 2.

Analysis of the existing systems for increasing the rocket target accuracy showed that all the proposed rocket correction algorithms involve measuring, as a rule, indirect rocket flight parameters and provide rather complex solutions that require significant modifications to the rocket design, such as fuze delays, rocket nose deflections, etc. The rocket path correction technology developed in this study requires the inclusion of standard sensors in the rocket system and a simple modification in the form of an electromechanical fin assembly turning system. The proposed technology was developed using the classical mathematical description of rocket flight and computer simulation using the well-known Matlab system. The half-scale simulation method using a real correction block makes it possible to take the test results and circuit solutions as a basis for developing a practical path correction system for Grad rockets.

The proposed rocket flight model takes into account rocket motion in the vertical plane, so only the range wind is considered as a disturbing effect. In addition, certain restrictions are imposed by the fact that the simulation was carried out for the flight of a material point, or rather the rocket center of mass.

It is assumed that further research in this direction should include the development of a correction system for a specific modification of Grad rockets. In this case, it is necessary to expand the research considering the overall rocket dimensions and include horizontal motion in the rocket flight model and take into account the moments of force generated by changes in the rocket angle of attack.

---

## 7. Conclusions

---

1. In the Matlab Simulink editor, a computer model of unguided rocket flight in the vertical plane was developed

(Fig. 1). The developed model is based on the classical system of aerodynamic equations for uncontrolled rocket motion. The model allows virtual measurement of all input and output parameters presented in the system of differential equations (1)–(6). Using the computer model, a number of experiments were carried out to determine the rocket flight range depending on external and internal parameters. The flight range is mainly affected by the range wind deviation from the specified one. So, under initial conditions: pitch angle  $\gamma_0=45^\circ$ , angle of attack  $\alpha=0^\circ$ , wind velocity 10 m/s and wind velocity deviation 2 m/s, the error can reach 100 m. Under such conditions, the most effective solution can be to install auxiliary rocket means, eliminating the effect of unpredictable wind deflection by correcting the rocket flight path.

2. The functional arrangement of the computer model, the inclusion of special aerodynamic blocks predetermined the creation of the block diagram of the rocket flight model using the rocket flight range correction system. In the block diagram (Fig. 4), the computer model was used both when constructing the rocket flight model, and as a reference model of the correction system. The negative feedback included a comparison block for the vector modules of the RM and rocket model motion (p. 10), PCI-1710 board ADC, L block, which made it possible to effectively control the rocket angle of attack and, ultimately, the flight range.

3. The Simulink model of unguided rocket flight and the constructed block diagram of the rocket flight model made it possible to proceed to the next stage of the study – construction of a general Simulink model that simulates the process of rocket path correction. The presented general Simulink model (Fig. 5) provided numerical experiments with a variation of input wind effects and made it possible to obtain comparative characteristics of errors during rocket launches with and without the flight range correction system (Table 2).

4. Using specialized Simulink packages, the Simulink model was loaded into the Arduino flash memory. As a result, a script (executable program) in C++ was obtained. The resulting script allows programming individual microcontrollers and platforms and using them as a reference model in the rocket launch system. The proposed technology for loading the RM simulation program, as well as the creation of a general Simulink model, was a preparatory stage in the construction of a half-scale rocket flight model with the correction block (Fig. 8).

5. The half-scale model was implemented in the form of a hardware-software complex (Fig. 9). The resulting hardware and software complex (Fig. 9) includes a real onboard correction system and a computer model of the rocket flight. In conditions close to full-scale tests, a series of experiments were carried out, the results of which are given in Table 3. These results almost coincided with the results in Table 2. Thus, the developed technology of loading the RM simulation program onto the Arduino platform or a separate microcontroller, as well as the proposed hardware and software complex, can be the basis for further development of a real rocket flight path correction system.

---

## References

1. Akhromeev, S. F. (Ed.) (1986). *Reaktivnaya sistema zalpovogo ognya. Voennyi entsiklopedicheskiy slovar.* Moscow: Voenizdat, 625–626.
2. Lahti, J., Salaranta, T., Harju, M., Virtanen, K. (2019). Control of exterior ballistic properties of spin-stabilized bullet by optimizing internal mass distribution. *Defence Technology*, 15 (1), 38–50. doi: <http://doi.org/10.1016/j.dt.2018.10.003>

3. Sun, H., Yu, J., Zhang, S. (2016). The Control of Asymmetric Rolling Missiles Based on Improved Trajectory Linearization Control Method. *Journal of Aerospace Technology and Management*, 8 (3), 319–327. doi: <http://doi.org/10.5028/jatm.v8i3.617>
4. Lei, X., Zhang, Z., Du, Z. (2019). Analysis of an improved trajectory correction scheme based on mass blocks. *Journal of Systems Engineering and Electronics*, 30 (1), 180–190. doi: <http://doi.org/10.21629/jsee.2019.01.17>
5. De Celis, R., Cadarso, L., Sánchez, J. (2017). Guidance and control for high dynamic rotating artillery rockets. *Aerospace Science and Technology*, 64, 204–212. doi: <http://doi.org/10.1016/j.ast.2017.01.026>
6. Yuan, G., Liangxian, G., Lei, P. (2009). Modeling and Simulating Dynamics of Missiles with Deflectable Nose Control. *Chinese Journal of Aeronautics*, 22 (5), 474–479. doi: [http://doi.org/10.1016/s1000-9361\(08\)60128-4](http://doi.org/10.1016/s1000-9361(08)60128-4)
7. vtukh, D. N., Maksimov, S. S. (2012) Analiz vozmozhnykh skhem postroeniya sistem upravleniya RS povyshennoy tochnosti dlya RSZO kalibra 122-mm. *Izvestiya TulGU. Tekhnicheskie nauki*, 11 (1), 167–170.
8. Kuznetsov, N. S. (2014) Nekotorye perspektivnye napravleniya rabot v OAO «NLP «Delta». *Boepripasy*, 2, 9–11.
9. Kuznetsov, N. S. (2019). Pat. No. 0002678922 RU. Sposob korrektsiyi traektorii snaryadov reaktivnykh sistem zalpovogo ognya. published: 04.02.2019. Available at: <https://edrid.ru/rid/219.016.b7e0.html>
10. Xu, Y., Zhijun Wang, Z., Dong, F. (2020). Ballistic Trajectory Modeling for Missile with Deflectable Nose. *Mechanics*, 26 (5), 450–456. doi: <http://doi.org/10.5755/j01.mech.26.5.27874>
11. Zhang, C., Li, D. (2020). Mechanical and Electronic Video Stabilization Strategy of Mortars with Trajectory Correction Fuze Based on Infrared Image Sensor. *Sensors*, 20 (9), 2461. doi: <http://doi.org/10.3390/s20092461>
12. Sun, X., Gao, M., Zhou, X., Lv, J., Tian, F., Qiao, Z. (2021). Guidance Simulation and Experimental Verification of Trajectory Correction Mortar Projectile. *IEEE Access*, 9, 15609–15622. doi: <http://doi.org/10.1109/access.2021.3052883>
13. Walha, A., Wali, A., Alimi, A. M. (2013). Video Stabilization for Aerial Video Surveillance. *AASRI Procedia*, 4, 72–77. doi: <http://doi.org/10.1016/j.aasri.2013.10.012>
14. Li, S., Lu, J., Cheng, L., Zeng, D. (2021). A high precision in-bore velocity measurement system of railgun based on improved Bi-LSTM network. *Measurement*, 169, 108501. doi: <http://doi.org/10.1016/j.measurement.2020.108501>
15. Li, W., Wen, Q., Yang, Y. (2019). Stability analysis of spinning missiles induced by seeker disturbance rejection rate parasitical loop. *Aerospace Science and Technology*, 90, 194–208. doi: <http://doi.org/10.1016/j.ast.2019.04.013>
16. Sun, X., Gao, M., Zhou, X., Lv, J., Tian, F., Qiao, Z. (2021). Guidance Simulation and Experimental Verification of Trajectory Correction Mortar Projectile. *IEEE Access*, 9, 15609–15622. doi: <http://doi.org/10.1109/access.2021.3052883>
17. Dmitrievskiy, A. A., Lysenko, L. N. (2005). *Vneshnyaya ballistika*. Moscow: «Mashinostroenie», 607.
18. Dyakonov, V. (2002). *MATLAB. Obrabotka signalov i izobrazheniy. Spetsialniy spravochnik*. Saint Petersburg: «Piter», 608.
19. Guskov, A. V., Milevskiy K. E., Sotenko A. V. (2010) *Vneshnyaya ballistika*. Novosibirsk: Izd-vo. NGTU, 188.