

# Methods and Technologies of Cosmonaut Rescue Simulation in Virtual Environment Systems

A.V. Maltsev<sup>1</sup>, E.V. Strashnov<sup>2</sup>, M.V. Mikhayluk<sup>3</sup>

Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS)

<sup>1</sup> ORCID: 0000-0003-1776-814X, [avmaltcev@mail.ru](mailto:avmaltcev@mail.ru)

<sup>2</sup> ORCID: 0000-0003-0937-4052, [strashnov\\_evg@mail.ru](mailto:strashnov_evg@mail.ru)

<sup>3</sup> ORCID: 0000-0002-7793-080X, [mix@niisi.ras.ru](mailto:mix@niisi.ras.ru)

## **Abstract**

The paper presents original methods and approaches for modeling the process of cosmonaut rescue using a jetpack within virtual environment system VirSim developed at the SRISA RAS. Proposed solutions are based on modern technologies of virtual reality (VR), including Oculus Rift CV1 VR headset, Oculus Touch controllers and Manus Prime II VR gloves. For human interaction with a computer-synthesized environment, a virtual model of cosmonaut with space jetpack was developed, whose hands copy the movements of operator's real hands. Control of virtual jetpack is performed on the basis of operator impact on the elements of jetpack's three-dimensional control panel by the hands of the cosmonaut model. To introduce this approach into virtual environment system, methods were developed for dynamics simulation and control of cosmonaut model, including implementation of its hand movement, operation of control panel and thrusters. Results obtained in the paper can be used to create training complex for teaching cosmonauts to rescue in case of accidental separation from the International Space Station.

**Keywords:** virtual reality, cosmonaut, jet pack, virtual model, control panel, simulation, control.

## **1. Introduction**

Currently, active research and exploration of space is underway, including through the International Space Station (ISS). The work of cosmonauts at the station involves performing tasks not only inside the modules, but also on their surface (extravehicular activity), which requires spacewalk [1, 2]. In this case, cosmonaut attaches himself to spacecraft by means of two safety tethers cling to the station's handrails. Some jobs involve a long walk from the exit of space module to destination point. In the course of moving along the handrails, cosmonaut must fix and unfix carabines of two tethers multiple times [3]. Therefore there is a risk of unsuccessful carbine fixation or tether malfunction, in which cosmonaut is detached from the station and cannot return. To solve this problem, it is important to use cosmonaut rescue system (for example, Simplified Aid For EVA Rescue or abbreviated SAFER [4]), which represents a space jetpack with thrusters, put on spacesuit. Manual or automatic control of the jetpack is performed by special control panel. An important and actual task is to train cosmonauts in terrestrial conditions in the skills of using rescue equipment.

In recent years, an actively developing area is creation of training systems using virtual reality technologies. These systems replace real objects with their virtual prototypes, and training is performed by immersing the operator in virtual environment. An example is simulators [5, 6] designed to train cosmonauts to control spacecrafts and perform tasks during extravehicular activity. It is important that using VR technologies can improve the quality of visual perception for those tasks and operations that cannot be realized in terrestrial condi-

tions. Paper [7] presents solutions for training cosmonauts in spacewalk, that include several tracking systems that allow determining movements of human body's various parts (arms, legs, head). Authors of paper [8] developed a simulator for control of manned spacecraft by joysticks as part of the planned mission of the Moon exploration. Also hardware solutions [9] created in the NASA laboratory and based on cosmonaut training by means of VR technologies are widely used.

Several papers are devoted to the task of cosmonaut rescue using jetpack [10, 11, 12]. In paper [10], MATLAB package with control implementation in Simulink environment is used for simulation of space jetpack. This approach is intended to demonstrate cosmonaut motion, but not suitable for training. Alternative solution consists in using external control devices, as was shown in [11] and [12] by the example of creating a test bench for space jetpack control with joystick or gamepad. In publications considered, simulation of cosmonaut motion was carried out without VR technologies.

This paper presents methods and approaches for training cosmonauts to control jetpack, which are based on using VR technologies and developed virtual environment system. The scientific novelty is that a control of jetpack model is executed by interaction of real cosmonaut with elements of virtual three-dimensional control panel. To that end, a virtual model of cosmonaut with a jetpack was created, in which movement of his hands copies movement of the operator's real hands. With this approach, control of virtual jetpack, and hence motion of cosmonaut model, is carried out by interaction of real person's hands with elements of jetpack's three-dimensional virtual control panel through virtual hands. Proposed solution includes implementing dynamics and control for virtual model of cosmonaut, his hands, as well as operation of control panel and thrusters of space jetpack. Approbation of the results obtained in this paper was carried out in our virtual environment system VirSim [13] developed at the SRISA RAS and demonstrated their effectiveness by the example of solving the problem of cosmonaut rescue.

## 2. Virtual environment system

Virtual environment system (VES) that we created for solving the task of cosmonaut training to use rescue equipment in the form of jetpack includes hardware block, software complex and digital visual models (DVM). Its structure is illustrated in Figure 1.

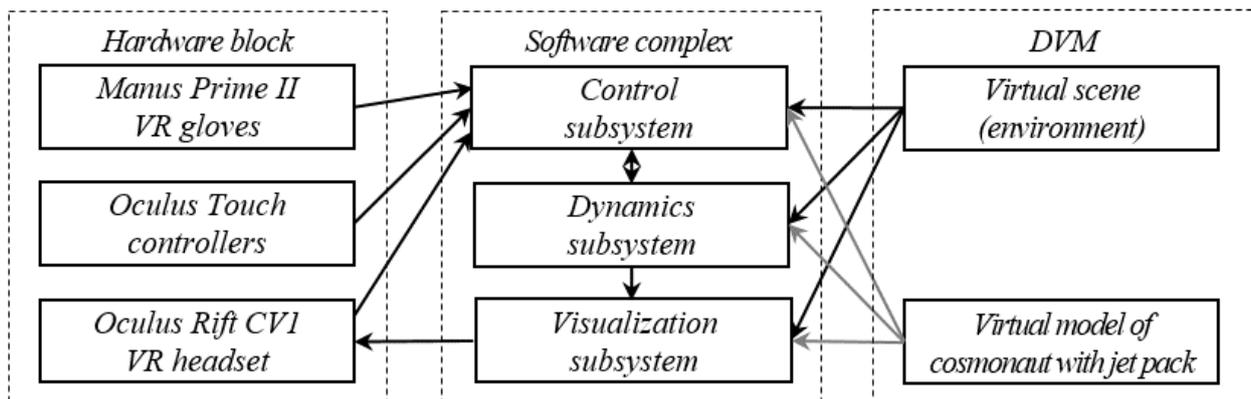


Fig. 1. VES structure for cosmonaut rescue training.

Hardware block, which consists of Manus Prime II VR gloves, Oculus Touch controllers and Oculus Rift CV1 VR headset, has two functions. The first is tracking of the operator's head and hands. Despite good quality of finger tracking, Manus gloves don't support determination of hand position in space but only their orientation. In this regard, we propose to implement the missing functionality using Oculus Touch controllers. Original and quite ergonomic placement of the Touch on hand with Manus glove is shown in Figure 2. The second

function of hardware block is to display the synthesized stereopair to the operator's eyes. The Rift VR headset provides this capability.



Fig. 2. Device configuration of Oculus Touch and Manus Prime II on operator's hand.

Software complex is own original product and doesn't use third-party solutions. It includes three subsystems, the tasks of which are implementation of virtual object control, simulation of their dynamics and virtual environment visualization. Subsystems run as separate processes controlled by a common shell. The shell also provides for the transfer of most of the data between the subsystems using special protocols. Control subsystem receives at its input information from external devices included in the hardware block. In addition, it receives data from state sensors of virtual control elements (buttons, toggle switches, joysticks, etc.) which are parts of DVM. Based on obtained information, computing of functional schemes [14] for virtual environment's dynamic object control, and synthesis of control signals are performed. Functional schemes are stored together with DVM and loaded to control subsystem during VES initialization. Generated signals are transmitted to dynamics subsystem which computes new positions and orientations of virtual environment objects, and also performs collision detection and response between them. After that, it is necessary to synthesize stereo image of virtual space and transfer it to hardware block for display. This functionality is implemented in visualization subsystem. Stereopair rendering with simulation of realistic object lighting and shadowing in virtual environment is performed on multi-core graphics processor (GPU) in real time, i.e. with frame rate of at least 25 times per second. The result is stereopair that is transmitted into the Rift VR headset.

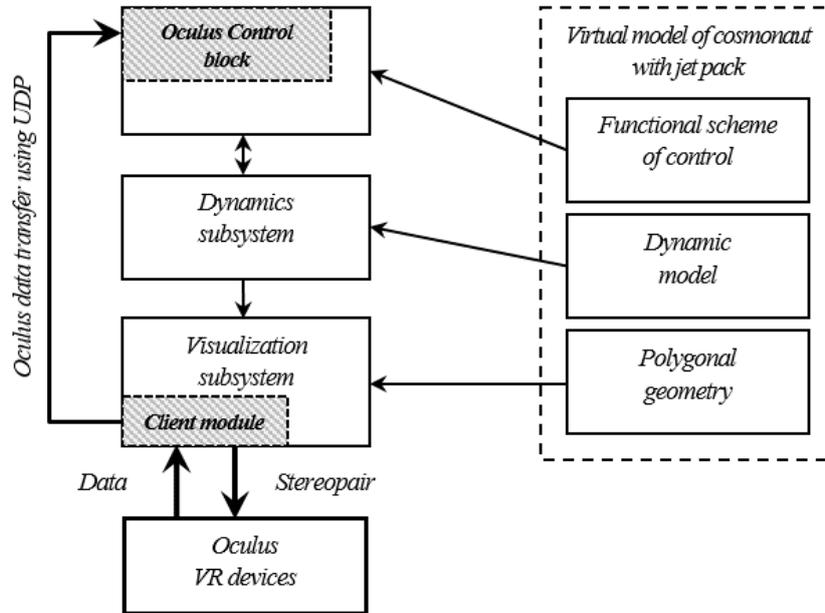


Fig. 3. Interaction of Oculus devices with subsystems of software complex.

One of the challenges encountered when implementing interconnection between hardware block and software complex is that the initialization of channel for receiving/transmitting information with Oculus devices is possible only within one process. So there's no way to simultaneously access the devices both in control subsystem (for reading tracking data from the Touch and the Rift) and visualization subsystem (for sending stereopair to the Rift). To solve the problem, we propose an approach that implements direct information link between Oculus VR equipment and visualization subsystem, as well as transit channel for data required by control subsystem from Oculus devices through visualization system to functional scheme of cosmonaut model control via UDP network protocol (fig. 3). To do this, a client module is implemented in visualization subsystem. It interacts with VR devices via Oculus SDK and sends network packets to control subsystem. And furthermore, special Oculus Control block is implemented in the functional scheme editor's library, which is a part of scheme loaded to control subsystem, and provides server function for receiving data from the client module.

The DVM created by us includes highly detailed three-dimensional virtual scene of cosmonaut environment (models of the ISS and the Earth), which has about million polygons, and virtual model of cosmonaut with a jetpack. Control scheme of cosmonaut model motion (head rotations, translations and rotations of hands, bending of fingers) and operation of rescue equipment's thrusters was also developed. Signals from hardware block, as well as toggle switches and joystick located on virtual control panel of a jetpack are inputs of the scheme. To receive signals from Oculus devices, our Oculus control block considered above is added to this scheme.

### 3. Mathematical model and control of virtual hand

The interaction of trained cosmonaut with virtual environment objects consists in copying the movements of his hands with their virtual prototypes. Based on VR technologies, this is done through hand tracking with VR gloves and controllers. To implement this approach in developed VES, virtual models of cosmonaut's hands were created, design of which was carried out in the computer simulation system 3ds Max. Figure 4 shows a model of right hand, which consists of a set of bones and geometry. Changing bones position and orientation leads

to compressions and stretches of the fabric upper layers. This behavior is specified using a special *skin* modifier by applying the coefficients of bones influence on vertices of model geometry.

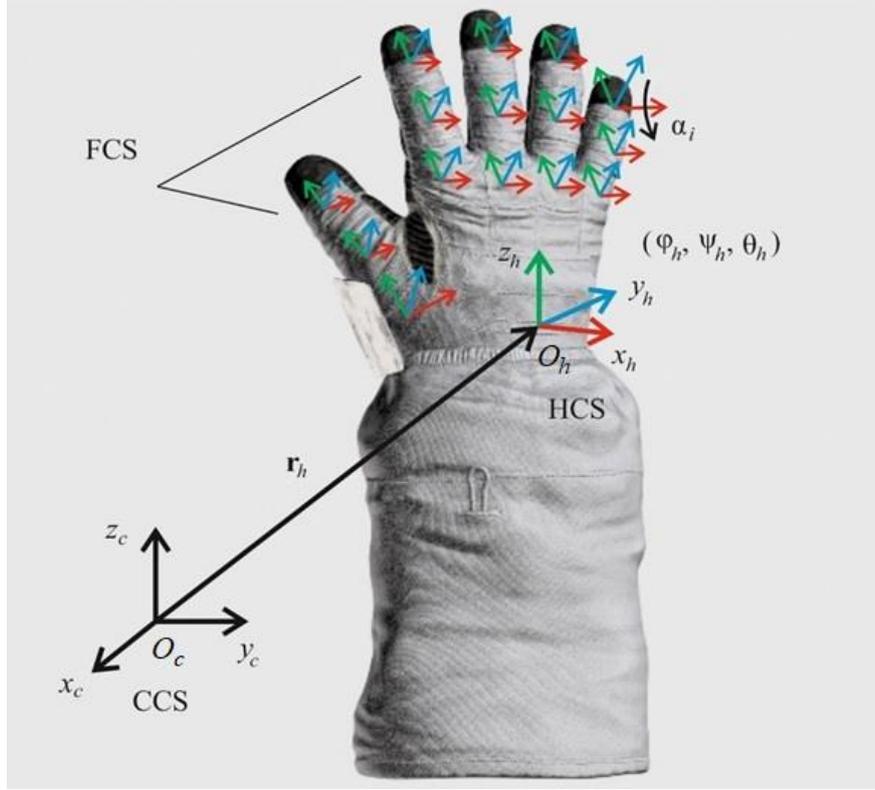


Fig. 4. Virtual model of cosmonaut's hand.

Hand motion simulation is implemented in developed VES with using its dynamic model. In this model, the hand is represented as an articulated body system. To describe the motion of virtual hand, consider the CCS (*the Cosmonaut Coordinate System*) of cosmonaut model's torso, the HCS (*the Hand coordinate system*) of hand itself, and the FCS (*the Finger coordinate systems*) of its fingers. Hand position is defined relative to the torso with radius-vector  $\mathbf{r}_h = \mathbf{O}_c \mathbf{O}_h$ , and attitude – with rotations first around the  $z_h$ -axis on the angle  $\theta_h$ , then around the  $x_h$ -axis on the angle  $\varphi_h$  and finally around the  $y_h$ -axis on the angle  $\psi_h$ . In proposed model, the motion of fingers is described by the bend angles  $\alpha_i$ ,  $i = \overline{1, K}$ , where  $K$  is the number of hand finger links. Combine vector  $\mathbf{r}_h$  and angles  $\varphi_h$ ,  $\psi_h$ ,  $\theta_h$ ,  $\alpha_i$ , we get that hand motion is described by generalized coordinates  $\mathbf{q} = (q_1, \dots, q_N)^T$ ,  $N = K + 6$ .

The task of virtual hand simulation is to ensure its motion, including in the presence of collisions with virtual environment objects. In doing so, it is required to carry out simulation of the contact, impact and friction between bodies, which are formulated in the form of constraints imposed on the coordinates and velocities of bodies. Then task described above is to find  $\mathbf{q}$  for every time instant  $t$  under such constraints.

The articulated body dynamics with generalized coordinates is described using differential equations [15]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{Q}_e + \mathbf{J}(\mathbf{q})^T \boldsymbol{\lambda}, \quad (1)$$

where  $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{N \times N}$  is the mass matrix,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{N \times N}$  is the Coriolis matrix,  $\mathbf{Q}_e \in \mathbb{R}^N$  is the vector of generalized forces (external torques that applied at hand's joints),

$\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{M \times N}$  is the constraint Jacobian matrix,  $\boldsymbol{\lambda} \in \mathbb{R}^M$  is the vector of Lagrange multipliers,  $M$  is the number of constraints.

Differential equations (1) describe the dynamics of virtual hand with constraints, where  $\ddot{\mathbf{q}}$  and  $\boldsymbol{\lambda}$  are unknown variables. To solve this task, an approach is proposed that is based on the articulated-body method [16] and its extended version [17] for constraint enforcing. The forward dynamics problem for solving equations (1) in the articulated-body method is reduced to the following form

$$\ddot{q}_i = f_i(\mathbf{q}, \dot{\mathbf{q}}, \mathcal{Q}_{e,i}, \boldsymbol{\lambda}), \quad i = \overline{1, N}, \quad (2)$$

where functions  $f_i$  are computed recursively.

Use the semi-implicit Euler scheme for numerical integration of equations (2). According to this scheme, first, the generalized velocities with  $\boldsymbol{\lambda} = \mathbf{0}$  for the time instant  $t + \Delta t$  are computed:

$$\dot{q}_i'(t + \Delta t) = \dot{q}_i(t) + \Delta t f_i(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathcal{Q}_{e,i}, \mathbf{0}). \quad (3)$$

The search for a vector  $\boldsymbol{\lambda}$  is formulated as a Mixed Linear Complementarity Problem (MLCP) [18]. This task is for a given matrix  $\mathbf{A} \in \mathbb{R}^{M \times M}$  and vector  $\mathbf{b} \in \mathbb{R}^M$  to find  $\boldsymbol{\lambda} \in \mathbb{R}^M$  and  $\mathbf{w} \in \mathbb{R}^M$  such that

$$\mathbf{w} = \mathbf{A}\boldsymbol{\lambda} + \mathbf{b}, \quad \boldsymbol{\lambda}^{\min} \leq \boldsymbol{\lambda} \leq \boldsymbol{\lambda}^{\max}, \quad (4)$$

where one of three conditions is satisfied:

$$\lambda_i = \lambda_i^{\min}, \quad w_i \geq 0;$$

$$\lambda_i = \lambda_i^{\max}, \quad w_i \leq 0;$$

$$\lambda_i^{\min} < \lambda_i < \lambda_i^{\max}, \quad w_i = 0.$$

In task (4), vector elements  $b_i$ ,  $i = \overline{1, M}$  are computed by differentiating the constraints on body velocities, and matrix elements  $A_{ij}$ ,  $i, j = \overline{1, M}$  are determined on the basis of extended articulated body method [17, 19]. Vector elements  $\lambda_i^{\min}$  and  $\lambda_i^{\max}$  are set depending on the type of constraint (for example,  $\lambda_i^{\min} = 0$  and  $\lambda_i^{\max} = \infty$  for contact constraint). In this work, to solve task (4), the Projected Gauss-Seidel method is used [20]. The idea is that  $\lambda_i$  are computed iteratively by solving equations  $w_i = 0$ . If  $\lambda_i$  does not satisfy (4), then projection is performed on the feasible set. After computing  $\boldsymbol{\lambda}$ , the generalized velocities are corrected as follows

$$\dot{q}_i(t + \Delta t) = \dot{q}_i'(t + \Delta t) + \Delta t f_i(\mathbf{q}(t), \dot{\mathbf{q}}'(t + \Delta t), \mathbf{0}, \boldsymbol{\lambda}). \quad (5)$$

Finally, at the last stage of the semi-implicit Euler scheme, new generalized coordinates for the time instant  $t + \Delta t$  are computed:

$$q_i(t + \Delta t) = q_i(t) + \Delta t \dot{q}_i(t + \Delta t). \quad (6)$$

Virtual hand control is realized in copying mode by means of Manus VR glove and Oculus Touch controller. With this approach, the readings of devices set the vector  $\mathbf{q}_d$  of desired generalized coordinates. Proposed solution is that the torques  $\mathcal{Q}_{e,i}$  are formed in the following form

$$\mathcal{Q}_{e,i} = -k_{p,i}(q_i - q_{d,i}) - k_{d,i}(\dot{q}_i - \dot{q}_{d,i}), \quad (7)$$

where  $k_{p,i} > 0$  and  $k_{d,i} > 0$  are given coefficients.

Thus, the task of virtual hand motion simulation is implemented by applying toques according to Eq. (7) and integrating equations (2) with the semi-implicit Euler scheme using expressions (3), (5) and (6), where the vector  $\boldsymbol{\lambda}$  is computed by solving (4).

#### 4. Mathematical model for motion of cosmonaut with jetpack

The motion of cosmonaut in Earth's orbit is described relative to the ISS. To do this, consider the world coordinate system  $Oxyz$  (WCS) and the local coordinate system (LCS), which is rigidly attached to the ISS and cosmonaut, respectively (fig. 5). Cosmonaut position is defined with the radius-vector  $\mathbf{r} = \mathbf{OO}_c = (x, y, z)^T$ , and the attitude – with transformation matrix  $\mathbf{R}$  from the LCS of cosmonaut to the WCS.

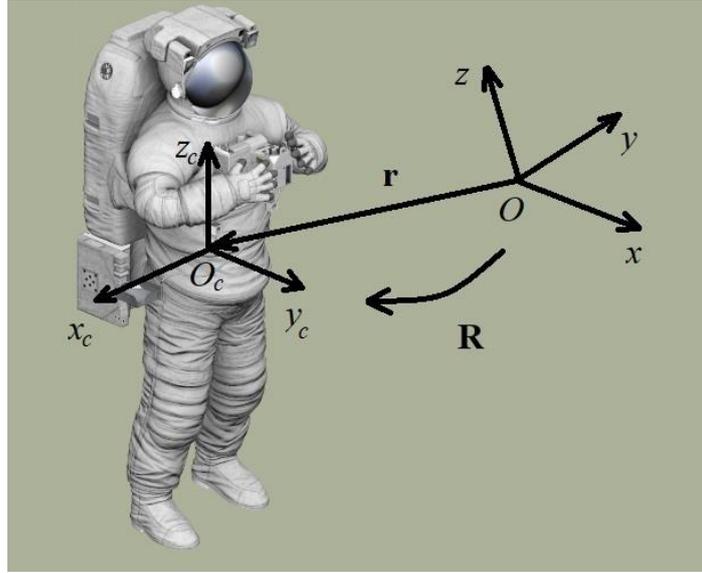


Fig. 5. Virtual model of cosmonaut with jetpack.

The translational (linear) motion of cosmonaut is provided under the action of thrusters, the direction of which coincides with axes of the LCS. Since there are four thrusters with the equal thrust  $f$  on each side of the jetpack, total thrust force from all the thrusters forms a vector  $\mathbf{f}_t = (F_x, F_y, F_z)^T$  relative to the LCS, where  $F_x, F_y, F_z \in \{-F, 0, F\}$ ,  $F = 4f$ .

In the absence of gravity and without taking into account the ISS influence on cosmonaut motion, the dynamics of its linear motion is described by differential equations in the form of Newton's second law:

$$m\dot{\mathbf{v}} = \mathbf{R} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}, \quad (8)$$

where  $m$  is the total mass of cosmonaut, spacesuit and jetpack,  $\mathbf{v} = \dot{\mathbf{r}} = (v_x, v_y, v_z)^T$  is the linear velocity of cosmonaut in the WCS.

The rotational motion of cosmonaut is provided under the action of torques created by jetpack's thrusters. It is assumed that the thrusters are located symmetrically relative to cosmonaut's center of mass. Therefore, we obtain the total moment  $\boldsymbol{\tau}_t = (\tau_x, \tau_y, \tau_z)^T$  relative

to the LCS, where  $\tau_x \in \{-T_x, 0, T_x\}$ ,  $\tau_y \in \{-T_y, 0, T_y\}$ ,  $\tau_z \in \{-T_z, 0, T_z\}$ ,  $T_x$ ,  $T_y$  and  $T_z$  are the torques of jetpack's thrusters.

The dynamics for rotational motion of cosmonaut is described by the Euler differential equations [21]:

$$\begin{aligned} I_x \dot{\omega}_x - (I_y - I_z) \omega_y \omega_z &= \tau_x; \\ I_y \dot{\omega}_y - (I_z - I_x) \omega_x \omega_z &= \tau_y; \\ I_z \dot{\omega}_z - (I_x - I_y) \omega_x \omega_y &= \tau_z, \end{aligned} \quad (9)$$

where  $I_x$ ,  $I_y$  and  $I_z$  are the principal moments of inertia,  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$  is the angular velocity of cosmonaut in its LCS.

In that way, the mathematical model for motion of cosmonaut with jetpack is described by the equations (8) and (9), in which  $F_x$ ,  $F_y$ ,  $F_z$ ,  $\tau_x$ ,  $\tau_y$  and  $\tau_z$  are the control variables.

## 5. Motion control of cosmonaut model

In this paper, an approach is proposed in which automatic and manual motion control of cosmonaut model is implemented using virtual three-dimensional control panel, the prototype of which is the hand controller module of the SAFER rescue system [4]. Figure 6 shows created model of this control panel, which contains three toggle switches, one button and four-axis joystick with a button. Two-position toggle switch labeled "PWR" turns control panel on and off, while two-position toggle switch labeled "MODE" is used to select mode for rotational and translational motion of cosmonaut model ("ROT" and "TRAN"). The joystick has four degree of freedom: displacement along the X-axis and rotations around the Z, Y and X axes (in fig. 6, rotation around the X-axis is denoted as Rot). The button on the joystick of control panel is used to turn on the mode of automatic attitude hold, in which angular velocity of cosmonaut is damped. Created control panel also includes the "RTRN" button to activate the mode of cosmonaut automatic return to predetermined position.

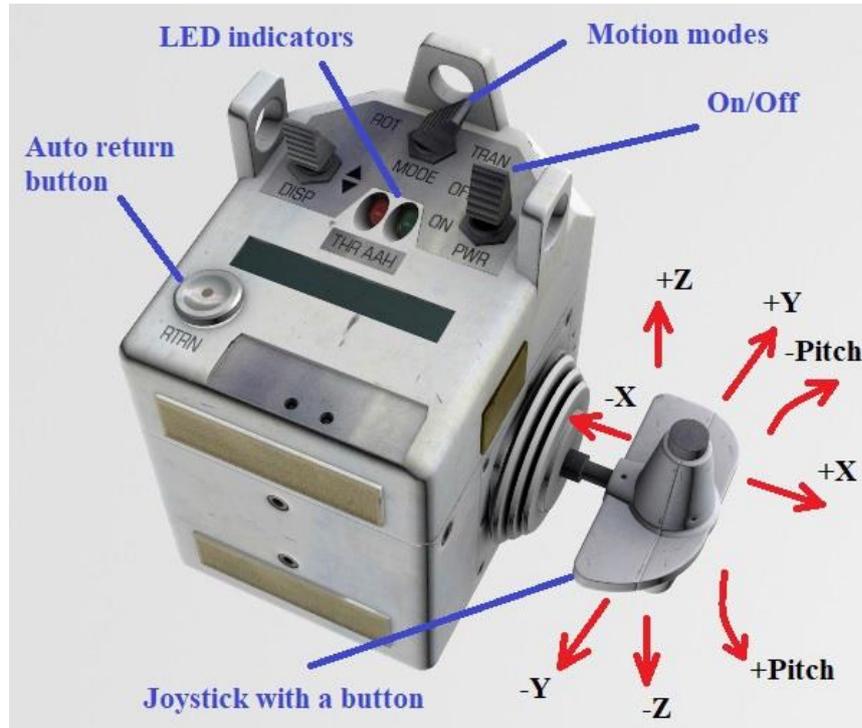


Fig. 6. Virtual model of space jet pack's control panel.

Table 1. Control parameters of space jetpack in manual mode.

MODE = «ROT»		MODE = «TRAN»	
X-	$\tau_Z = T_Z$	X-	$F_X = -F$
X+	$\tau_Z = -T_Z$	X+	$F_X = F$
X <sub>o</sub>	$\tau_Z = 0$	X <sub>o</sub>	$F_X = 0$
Z-	$\tau_Y = T_Y$	Z-	$F_Z = -F$
Z+	$\tau_Y = -T_Y$	Z+	$F_Z = F$
Z <sub>o</sub>	$\tau_Y = 0$	Z <sub>o</sub>	$F_Z = 0$

Depending on motion mode, changing the state of joystick to manual control sets the commands to turn on and off the thrusters. The joystick offset along the  $Y$ -axis and its rotation around the  $X$ -axis are available in both modes. When the joystick is shifted along the  $Y$ -axis, thrust  $F_Y = F \operatorname{sgn}(Y)$  is generated, and when the joystick is turned, torque  $\tau_X = T_X \operatorname{sgn}(\operatorname{Rot})$  is generated. The control law of jetpack for changing joystick positions in other directions is given in Table 1, where signs “+” and “-” indicate positive and negative directions, respectively, and “o” is neutral position.

The attitude stabilization of cosmonaut model (damping of the angular velocity) is provided in automatic mode by pressing the joystick’s button. Let us use the nonlinear control

theory to solve this task. Consider the positive definite Lyapunov function  $V(\boldsymbol{\omega}) = \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\omega}$ .

From the theory of Lyapunov functions [22] it follows that for the occurrence of sliding mode around the surface  $\boldsymbol{\omega} = \mathbf{0}$ , it is necessary and sufficient that the inequality

$$\dot{V} = \dot{\boldsymbol{\omega}}^T \boldsymbol{\omega} < 0 \quad (10)$$

is satisfied.

Substituting equations (9) into (10) yields

$$\dot{V} = \omega_X I_X^{-1} (\tau_X + (I_Y - I_Z) \omega_Y \omega_Z) + \omega_Y I_Y^{-1} (\tau_Y + (I_Z - I_X) \omega_X \omega_Z) + \omega_Z I_Z^{-1} (\tau_Z + (I_X - I_Y) \omega_X \omega_Y)$$

Let's form the relay control in the following form

$$\tau_X = -T_X \operatorname{sgn}(\omega_X), \quad \tau_Y = -T_Y \operatorname{sgn}(\omega_Y), \quad \tau_Z = -T_Z \operatorname{sgn}(\omega_Z). \quad (11)$$

Then inequality (10) will be satisfied under the conditions

$$T_X > |(I_Y - I_Z) \omega_Y \omega_Z|, \quad T_Y > |(I_Z - I_X) \omega_X \omega_Z|, \quad T_Z > |(I_X - I_Y) \omega_X \omega_Y|. \quad (12)$$

To reduce the number of switches, it is proposed to use a dead zone with a given threshold  $\varepsilon > 0$ . Then the control law for attitude stabilization around the  $x_c$ -axis of the LCS is

$$\tau_X = \begin{cases} -T_X, & \omega_X > \varepsilon; \\ 0, & |\omega_X| \leq \varepsilon; \\ T_X, & \omega_X < -\varepsilon. \end{cases} \quad (13)$$

Expressions for the torques  $\tau_Y$  and  $\tau_Z$  of attitude stabilization around  $y_c$  and  $z_c$  axes of the LCS are formed in a similar way.

In turn, proposed solution for automatic return of cosmonaut model in target position  $\mathbf{r}_d = (x_d, y_d, z_d)^T$  is to ensure the motion velocity towards the target:

$$\mathbf{v}_d = \begin{cases} v_r \frac{\mathbf{r}_d - \mathbf{r}}{\|\mathbf{r}_d - \mathbf{r}\|}, & \|\mathbf{r}_d - \mathbf{r}\| > l; \\ \mathbf{0}, & \|\mathbf{r}_d - \mathbf{r}\| \leq l, \end{cases}$$

where  $v_r$  is constant velocity for cosmonaut model,  $l$  is distance.

Use the theory of Lyapunov function and equations (8). Then the control law for the motion of cosmonaut model along the  $x_c$ -axis of its LCS is

$$F_x = \begin{cases} -F, & e_{v,x} > \chi; \\ 0, & |e_{v,x}| \leq \chi; \\ F, & e_{v,x} < -\chi, \end{cases} \quad (14)$$

where  $\mathbf{e}_v = \mathbf{R}^{-1}(\mathbf{v}_d - \mathbf{v})$  is the residual vector in the LCS,  $\chi > 0$  is given tolerance.

Similar relations are valid for  $F_y$  and  $F_z$ .

## 6. Results

Methods and approaches proposed in this paper were implemented in our VES VirSim prototype to train cosmonaut for rescue skills. Hardware platform of this system includes the following components: high-performance computing unit based on Intel Core i7-8700K CPU, NVidia RTX 2080 graphics card, sensors, Oculus Rift CV1 headset, Oculus Touch controllers, Manus Prime II gloves, and display. Software complex were implemented with using object-oriented language C++, the OpenGL 3D graphics library, shader language GLSL 4.3 and the CUDA parallel computing architecture.

Figure 7 illustrates a structure of functional scheme created for control of cosmonaut model. In this scheme, hardware connection is implemented in the form of blocks interacting with hand and head tracking devices. At the output of Manus Prime II block and ‘‘Touch’’ output of Oculus Control block (see section 2), target coordinates  $\mathbf{q}_d^l$  and  $\mathbf{q}_d^r$  are formed for virtual models of left and right hands. Obtained coordinates together with virtual sensor readings participate in torque computation according to Eq. (7). This torques are transmitted to performing devices (controlled joints). In turn, head rotation angles at ‘‘Rift’’ output of Oculus Control block are used to compute voltages  $\mathbf{U}$ , which are transmitted to virtual camera's electric motors. Computation of  $\mathbf{U}$  is performed by means of PD controllers, taking into account virtual camera's orientation sensor readings. When virtual hands interact with virtual control panel's elements, vector  $\mathbf{s}$  of control commands is formed, which sets control of jet-pack in manual or automatic mode. These commands are then used to compute the thrust vector  $\mathbf{f}_t$  and torque  $\boldsymbol{\tau}_t$ , according to which corresponding thrusters of space jetpack are switched on and off. In the automatic mode, control algorithm is synthesized by formulas of the form (13) and (14) based on sensor readings.

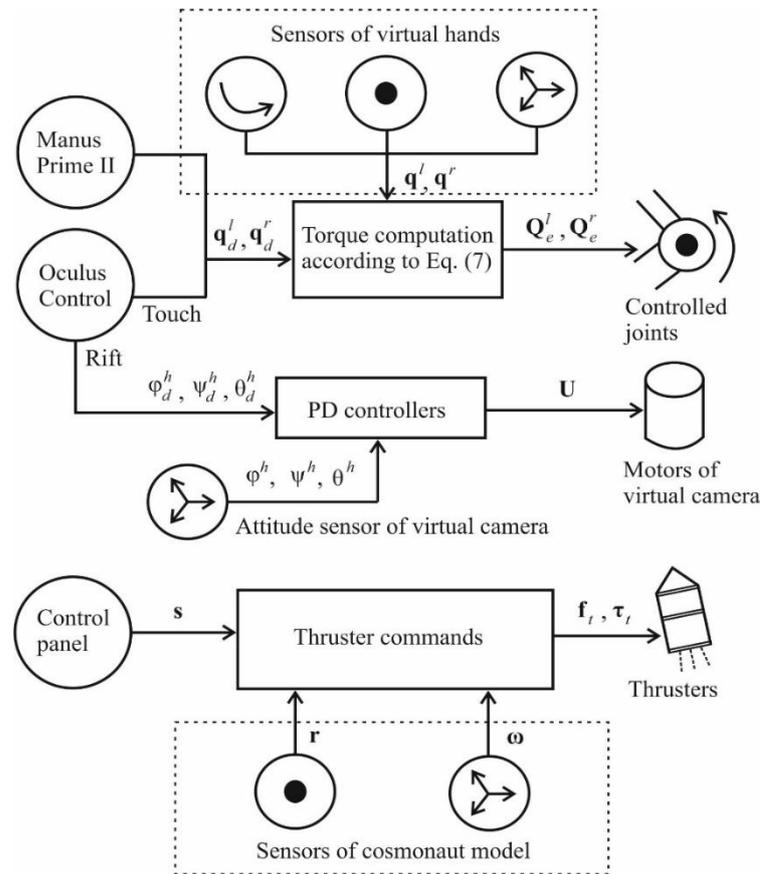


Fig. 7. Control scheme structure.

Approbation of proposed methods and approaches was performed on the example of emergency simulation when cosmonaut has separated from the ISS surface and has non-zero speeds. Training is implemented through immersing the operator in virtual environment (see fig. 8, 9) by means of VR headset and gloves. Using his hands, movements of which are copied by the virtual ones, the operator interacts with virtual panel elements and thus controls cosmonaut model motion. Approbation results showed the adequacy and effectiveness of solutions proposed in this paper for training cosmonaut skills of space jetpack control, which in the future can contribute to their rescue when separating from the ISS.



Fig. 8. Operator training with use of VR technologies.

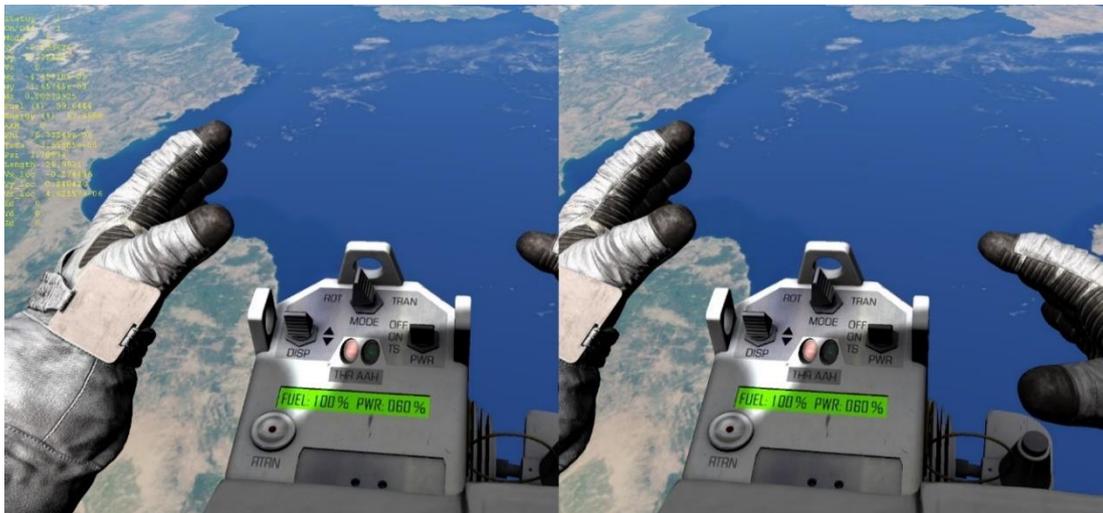


Fig. 9. Stereopair observed by operator.

## 7. Conclusions

In the paper we propose solutions for simulation of cosmonaut rescue process, based on modern technologies of virtual reality. The use of a stereo headset provides a high level of immersion of the operator in virtual environment and thereby improves the quality of training his skills to control a space jetpack. Results obtained in this paper can be used for cosmonaut training with practicing actions in emergencies and various scenarios.

## Acknowledgements

The publication is made within the state task of Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences” on “Carrying out basic scientific researches (47 GP)” on topic No. FNEF-2021-0012 “Virtual environment systems: technologies, methods and algorithms of mathematical modeling and visualization. 0580-2021-0012” (Reg. No. 121031300061-2).

## References

1. Fullerton R.K. EVA Tools and Equipment Reference Book // NASA Johnson Space Center, JSC-20466 Rev. B, Nov. 20, 1993.
2. Ryzhkov E. Hronika poleta jekipazha MKS [Chronicle of the ISS crew flight] // Russkij kosmos, No. 7, 2017, pp. 8-13 [in Russian].
3. Cygankov O.S. Pjatidesjatiletie vnekorabel'noj dejatel'nosti [Fifty years of extravehicular activity] // Kosmicheskaja tehnika i tehnologii, Vol. 1, No. 8, 2015, pp. 3-16 [in Russian].
4. Kelly J.C., Kemp K. Formal methods, specification and verification guidebook for software and computer systems, volume II: A practitioner’s companion, planning and technology insertion // Technical Report NASA-GB-001-97, NASA, Washington, DC 20546, May 1997.
5. Stone R., Panfilov P., Shukshunov V. Evolution of aerospace simulation: From immersive virtual reality to serious games // In Recent Advances in Space Technologies (RAST), 2011 5th International Conference on, June 2011, pp. 655–662.
6. Cater J.P., Huffman S.D. Use of the remote access virtual environment network (RAVEN) for coordinated IVA-EVA astronaut training and evaluation // Presence: Teleoperators & Virtual Environments, Vol. 4, No. 2, 1995, pp. 103-109.
7. Liu Yuqing and others. VR simulation system for EVA astronaut training // Proceedings of AIAA Space 2010 Conference & Exposition, Anaheim California, 2010.
8. Bruguera M.B., Ilk V., Ruber S., Ewald R. Use of virtual reality for astronaut training in future space missions – spacecraft piloting for the Lunar Orbital Platform – Gateway (LOP-G) // 70th International Astronautics Congress, Washington D.C., 2019.

9. Garcia A. D., Schlueter J., Paddock E. Training astronauts using hardware-in-the-loop simulations and virtual reality // AIAA SciTech Forum, Orlando, FL, 2020.
10. Wen J., Zhang J., Gao L., Li X. Modeling and simulation of Simplified Aid for EVA Rescue using virtual prototype technology // Open Automation and Control Systems Journal, No. 6, 2014, pp. 1532-1540.
11. Handley P.M., Robinson S.K., Duda K.R., Prasov Z., York S.P., West J.J. Real-time performance metrics for SAFER self-rescue // In 45<sup>th</sup> International Conference on Environmental Systems, 12-16 July 2015, Bellevue, Washington, pp. 1-14.
12. Rize J.P., Hoffman J., Carpenter M.D., Cohanin B. Real-time virtual reality environment for MAJIC attitude control system development and integration // In Proceedings of the 2014 IEEE Aerospace Conference, pp. 1-11.
13. Mihaylyuk M.V., Maltcev A.V., Timokhin P.Ju., Strashnov E.V., Krjuchkov B.I., Usov V.M. Sistema virtual'nogo okruzheniya VirSim dlja imitacionno-trenazhernyh kompleksov podgotovki kosmonavtov [The VirSim virtual environment system for the simulation complexes of cosmonaut training] // Pilotiruemye polety v kosmos, Vol. 4, No. 37, 2020, pp. 72-95 [in Russian].
14. Mikhaylyuk M.V., Torgashev M.A. Vizualnyi redaktor i modul rascheta funktsionalnykh skhem dlja imitatsionno-trenazhernykh kompleksov [The visual editor and calculation module of block diagrams for simulation and training complexes] // Programmnye produkty i sistemy, No. 4, 2014, pp. 10-15 [in Russian].
15. Shabana A.A. Computational dynamics, 3rd edition, John Wiley & Sons Ltd, 2010.
16. Featherstone R. Rigid body dynamics algorithms, New York: Springer-Verlag, 2008.
17. Mikhaylyuk M.V., Strashnov E.V., Timokhin P.Yu. Algorithms of multibody dynamics simulation using articulated-body method // Mathematica Montisnigri, No. 39, 2017, pp. 133-145.
18. Garstenauer H. A unified framework for rigid body dynamics. Master's thesis, Johannes Kepler Universitat, Linz, 2006.
19. Kokkevis E. Practical physics for articulated characters // Proceedings of Game Developers Conference, 2004.
20. Stepien J. Physics-based animation of articulated rigid body systems for virtual environments. Gliwice, 2013.
21. Landau L.D., Lifshitz E.M. Mechanics, 2<sup>nd</sup> edition. Course of theoretical physics. Vol. 1, Pergamon Press, 1969.
22. Shtessel Y., Edwards C., Fridman L., Levant A. Sliding Mode Control and Observation. Birkhauser, New York: Springer, 2014, 356 p.