https://doi.org/10.1051/e3sconf/202458304010

Development of automated system for receiving and analysing telemetry data from small spacecrafts

Roman Aleshko[1*], *Alexander Gordeychik*[1], *Vladimir* Berezovsky[1] and *Ksenia* Shoshina[1]

[1]Northern (Arctic) Federal University, Northern Dvina emb., 17, Arkhangelsk, 163000, Russia

Abstract. The system presented in the study is designed to increase the share of automation in the process of interaction with small spacecraft (SSC) at all stages of operation. Automatic operation of the system will reduce the cost of human resources, the number of errors in management, and increase the time of uninterrupted and continuous operation. The system independently calculates the time of all possible sessions of a group of small spacecrafts with a ground station; forms on its basis a schedule for the operation of a ground station, taking into account historical and future calculation sessions; implements control of the rotary antenna according to the compiled session schedule; decodes telemetry data, writes it to the database, carries out preprocessing and visualization. The system was developed taking into account possible scaling to several receiving stations.

1 Introduction

In recent decades, great interest in small spacecraft (SSC) has made it possible to look at the usual issues of Earth remote sensing (RS) in a new way [1, 2, 3, 4]. Low cost, speed of production, and improved technology expand the possibilities of using SSC. On their basis, groups consisting of dozens of spacecrafts are deployed. Due to the large number of devices, there is a need to develop new methods for controlling and monitoring the state of a group of SSC.

A typical cycle of work with a small spacecraft consists of three stages: loading a task onto the small spacecraft, executing the task using a payload, and transmitting data to Earth. The first and last stages are possible only at those moments when the satellite is located over the ground station for receiving and transmitting data and is in its radio visibility zone (RVZ). At the first stage, it is necessary to indicate the start and end time of the payload, and the time of transmission of the collected data to Earth.

Difficulties arise when operating a constellation of small spacecraft. Due to the specific nature of small spacecraft launches, several of them may be above the station at one time. In such situations, it is necessary to select the device with which the communication session will be carried out. For example, a constellation of 100 small spacecraft in polar orbit can

* Corresponding author: r.aleshko@gmail.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

generate up to 1000 possible sessions per day. Such volumes of work require an automated system for scheduling communication sessions.

With suboptimal planning, the resource of each spacecraft will be wasted, efficiency will decrease, and the financial costs of the company will increase. On the other hand, optimal scheduling will allow the maximum amount of data to be transmitted to fewer earth stations with greater efficiency [5]. An urgent task is to create a software system designed for automated planning of data transmission periods, taking into account the timing of spacecraft entry into the stations' visibility zones.

2 Methodology

The first and most important part of the system is the group of SSC with which the work is carried out. The target devices in this study are small spacecraft of the "cubesat" form factor of the Space-π project. The SSC data were chosen due to the openness of the project. Any interested user can receive and decrypt telemetry data.

Secondly, equipment for receiving signals from SSC plays an important role. For this purpose, a receiving and control station has been installed at the Center of Space Monitoring of the Arctic, Northern (Arctic) Federal University (CSMA NArFU). It contains a directional helical antenna with a rotating device 433-438 MHz.

LimeSDR is used as a directional antenna receiver. It is a high-performance software-defined transceiver designed for research, educational and amateur radio applications. LimeSDR has a USB port for communication with a computer and two SMA connectors for connecting to an antenna, which are necessary for receiving and transmitting signals, respectively. The workstation runs under the Ubuntu operating system. Using a distribution kit of the Linux family allows to flexibly configure different system parameters for efficient and convenient operation of the station.

Thirdly, special attention should be paid to accompanying software. When receiving data on the Ubuntu operating system, open-source software Gpredict is used to track the location of satellites, control the rotator and reception frequency, and GNU Radio is used to demodulate the signal [6]. The disadvantage of the system is that satellite selection must be done manually. The Windows operating system uses: SDR Sharp for signal reception, Orbitron for receiving frequency correction, Gpredict for PTZ control, Soundmodem for signal demodulation, other software for SSTV, digital image transmission, custom decoders.

Fourthly, the new part of the system being developed in the study is the automatic control system. It will independently calculate the time of communication sessions with the small spacecraft, create a schedule for the ground station based on them, and receive data.

From the point of view of software implementation, the main object with which interaction occurs is an instance of the Schedule program class. Through it, all other necessary classes are accessed and managed: DataProvider, Satellite, GroundStation, Task, Passes. The main task of the class is to fill auxiliary classes with data, calculate the intersection of satellites with the ground station, calculate the work schedule and compile it into a file for transmission and execution to the ground station.

Data Provider is a program class for working with all external files that the program needs: TLE.txt, config.env, Database.sqlite, ground_station.shp. Implemented a request to update a TLE file from the Internet. To work with a database, an auxiliary class Database has been created. It establishes a connection to the database, writes and reads tables to populate other classes with data.

Satellite is a program class for working with satellites. Inherited from the Orbital library, which implements functions for calculating the position of a satellite. Through this class it is possible to get all the information about the satellite: name, NORAD number, orbital parameters and others. When creating class objects, a reference to the object is placed in a special array, which allows to have access to all satellites in one place.

GroundStation is a program class for working with ground stations. It stores information about the location of the station, operating frequency, number in the database, mode of possible operations and schedule of the station.

Task is a program class for working with tasks for a communication session with a spacecraft. It stores information such as the task, its execution time and reset of the collected data.

Passes is a program class for working with sessions. Each instance of the class stores a reference to the Satellite, GroundStation and Task objects corresponding to a given session. The class object also contains information about the time, maximum satellite altitude, and session duration.

Figure 1 shows a block diagram of the system.

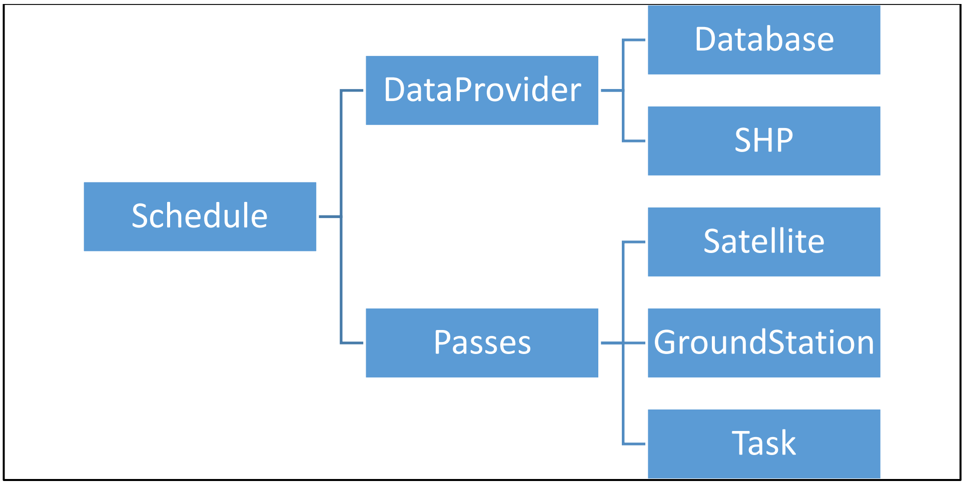

Fig. 1. System block diagram.

During system operation, the Schedule object accesses the DataProvider to load the schedule and satellites through the Database and the SHP to load information about ground stations. Sessions are saved to a database for later access.

Having the necessary tools for working with the grouping of small spacecraft and metrics for assessing the compiled schedule, it is possible to proceed to the development of an automated process for selecting sessions. Dividing the system into classes allows to flexibly customize the system

The TLE code is used to calculate the current and future positions of spacecraft in Earth's orbit. As an example, the ORBICRAFT-ZORKIY spacecraft will be used [7]. The zero line of the code contains the name of the spacecraft and its call sign. Next come two mandatory lines with information about the orbital parameters. Each of them contains 69 characters.

Mathematical models SGP4/SDP4/SDP8 are developed for use with the TLE code, and on its basis can calculate the positions of small spacecraft in orbit [8]. To calculate the position of satellites using the TLE code, the pyorbital library was used, in particular the Orbital module. The get_lonlatalt(utc_time) method was used to get the satellite position. It accepts a time parameter for which the satellite's position will be calculated.

The current position of the satellite will be displayed in the terminal: “(-22.3176522067, -12.14870354148, 530.371965356)”. The parameter has the format: longitude, latitude, altitude. Figure 2 shows the location of the satellite at the time of calculation.

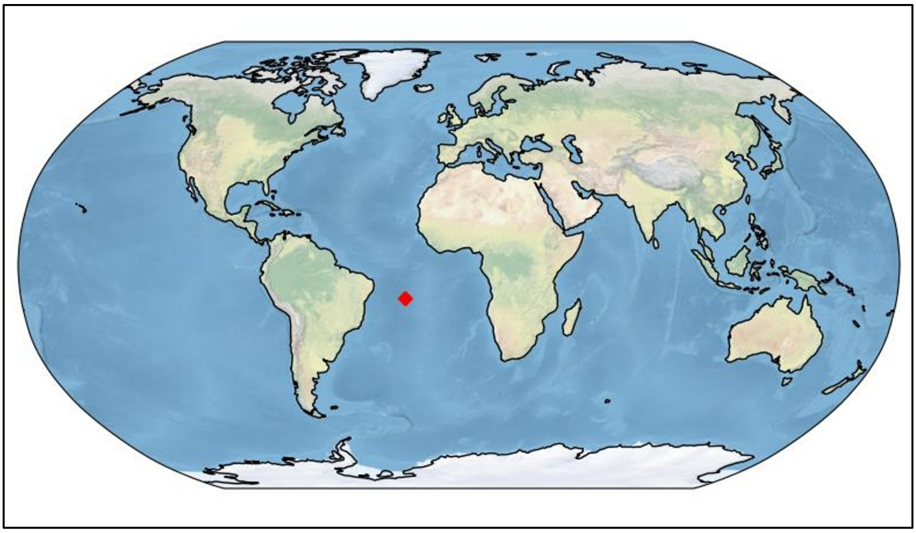

Fig. 2. Spacecraft location.

The pyorbital software library also allows to calculate the intersection of the satellite with the ground station. The get_next_pass() method is used for this. It takes the following parameters as input: time, interval for calculating intersections, station longitude, station latitude, station height, accuracy, minimum station height above the horizon. Figure 3 shows the trajectory of the satellite in the area of intersection with the ground station.

Fig. 3. Intersection of satellite path and ground station.

To calculate the intersections of all satellites with a ground station, the calculate_passes function is implemented. The current list of intersections is cleared, then intersections with the ground station are calculated for each satellite, and the data is recorded in the database. After the calculation is completed, the data is sent to the database. An example of data from the passes table is shown in Figure 4.

	NORAD	satellite	ground_station	time_start	time_end	time_max	max_el	id	used
1	47951	CUBESX-HSE	ARH	2024-04-12 20:40:00	2024-04-12 20:48:47	2024-04-12 20:44:24	14.2889055261363	0	False
2	57172	UMKA-1	ARH	2024-04-12 21:09:02	2024-04-12 21:18:16	2024-04-12 21:13:46	7.32983809129278	44	False
3	47951	CUBESX-HSE	ARH	2024-04-12 22:11:21	2024-04-12 22:21:20	2024-04-12 22:16:22	89.2258505497109	1	False
4	53376	UTMN	ARH	2024-04-12 22:13:33	2024-04-12 22:18:58	2024-04-12 22:16:16	2.56053179718175	22	False
5	53377	MIET-AIS	ARH	2024-04-12 22:28:21	2024-04-12 22:40:09	2024-04-12 22:34:16	10.6130451168509	33	False
6	57172	UMKA-1	ARH	2024-04-12 22:45:20	2024-04-12 22:57:56	2024-04-12 22:51:54	28.9772701944005	45	False
7	47951	CUBESX-HSE	ARH	2024-04-12 23:42:59	2024-04-12 23:51:56	2024-04-12 23:47:28	17.4477742082423	2	False
8	53376	UTMN	ARH	2024-04-12 23:46:11	2024-04-12 23:56:32	2024-04-12 23:51:22	19.2719194637103	23	False
9	53377	MIET-AIS	ARH	2024-04-13 00:08:29	2024-04-13 00:23:28	2024-04-13 00:16:00	36.1619105622895	34	False
10	57172	UMKA-1	ARH	2024-04-13 00:21:55	2024-04-13 00:35:06	2024-04-13 00:28:49	68.7700585118078	46	False

Fig. 4. Example of storing data on satellite flights over a receiving station.

To control the rotation of the antenna, the rotctld software library is used. It allows control various rotating devices via TCP sockets and has a simple command syntax. Commands are sent to the server on one line, and rotctld responds with one line to each request, returning the requested values or error messages. The use of TCP sockets allows multiple user programs to share a single rotator, which is especially useful for multi-user client systems.

An application in the form of a TCP client has been developed to control the signal receiving rotary station. The application connects to the server created by rotctld, sends commands and receives data.

The following user interface elements are used to interact with the antenna:

- two fields for entering azimuth and elevation values;
- button to send a command to the server;
- two indicators of the position of the rotary device.

GNU Radio is used to receive and decrypt data from the spacecraft. The interface of the configured workspace is shown in Figure 5.

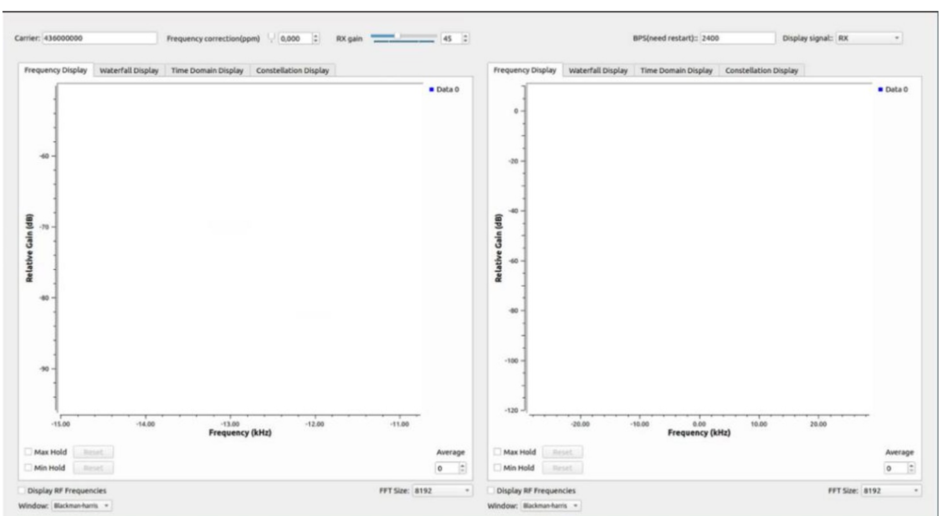

Fig. 5. Working area of the receiving module.

At the top of the screen there is a field for entering the operating frequency for receiving radio signals. The system automatically changes the frequency to match that used by each specific spacecraft. The default frequency is 436 MHz. To change it from third-party clients, a TCP socket is used.

3 Result

As a result of the research, a client-server ground station control system was developed. To work with the automatic scheduling system, a TCP server has been developed that is capable of sending json data packets. The server waits in a loop for incoming connections. When a client connects, a message is expected from it. If a command has been received, a function is called to obtain the schedule. The schedule data is packaged into a json package and sent to the client. The client part establishes a connection with the server and sends a message to receive the schedule. The loop reads the received data until it runs out. The resulting string is decoded into a dictionary format using the loads method of the json library.

The get_schedule() function retrieves from the database the generated schedule of sessions for which the "used" field is marked as "True". Adds the code used to calculate the satellite's motion to the TLE session data.

The client part of the system is designed to implement control of the ground receiving station according to a prepared schedule. The program image is shown in Figure 6.

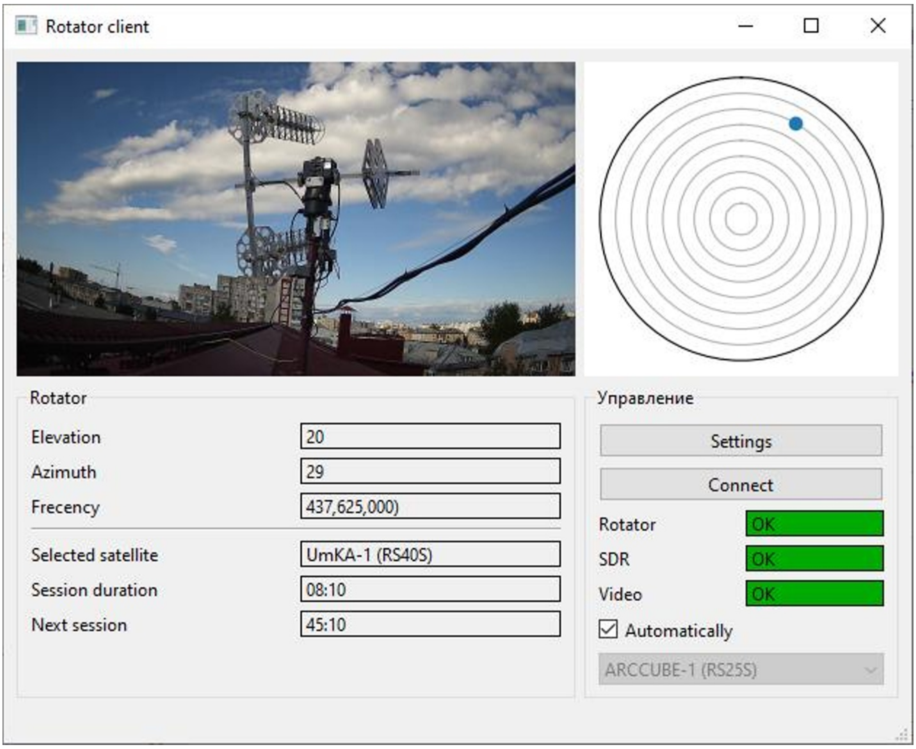

Fig. 6. Client part of the developed system.

Visually, the client is divided into four parts: a visual control window for the directional antenna, a polar graph of the position of the directional antenna and satellites, an information panel for the rotating device, and a control panel.

The control panel contains the following elements:

- "settings" button – to open the settings window;
- "connect" button – connection to the TCP control ports of the ground station and the video stream from the camera;
- information fields "rotator", "SDR" and "video" to display connection status;
- automatic operation system switch;

https://doi.org/10.1051/e3sconf/202458304010

- satellite selection window for manual mode.

The Rotator panel displays the main parameters of the status of the rotator and the scheduling system: the current position of the rotator in azimuth and elevation, the set reception frequency, the selected satellite for reception, duration and time until the next session. The visual control window is a display of streaming video images via the RTSP protocol. The polar plot is a graphical display of the position of the rotator and the trajectory of the satellite.

4 Discussion

To evaluate the operation of the system, metrics for an automatically generated schedule were introduced. The basic estimate is the arithmetic average of the number of successful sessions with SSC. However, this indicator does not reflect the distribution of received data among devices. The standard deviation will allow to evaluate the uniformity of the number of successful sessions from different SSC. The combination of these parameters already allows to compare schedules with each other. The efficiency calculation is determined by the formula:

$$\eta = \overline{x} - \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}},$$

where $\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ is the arithmetic mean of the number of successful sessions;
x_i– number of successful SSC sessions;
n– number of SSC in the constellation.

In addition, there may be other evaluation metrics:
– by time elapsed since the last communication session;
– according to the distribution of reception sessions during the day.

Let's compare two scheduling algorithms: the basic model of sequential session generation and the random scheduling system. The rules for drawing up the first schedule will be as follows:
– for each satellite in the sample, a schedule of intersections with the ground station is compiled;
– a satellite is selected, all its intersections with the ground station are scheduled;
– the next satellite is selected, if the crossing time is free, the crossing is recorded in the schedule;
– the previous point is repeated for all satellites.

A diagram will be presented to visualize the algorithm and check the distribution of stations. For each satellite, the percentage of effective operating time will be determined. The ratio of the scheduled transmission time to the maximum possible transmission time. Figure 7 shows such a diagram. The diagram shows that there are satellites that take up all the time without giving transmission time to others. The standard deviation of this case is 29.17, the arithmetic mean is 66.26. Using formula above, the value obtained was 37.09.

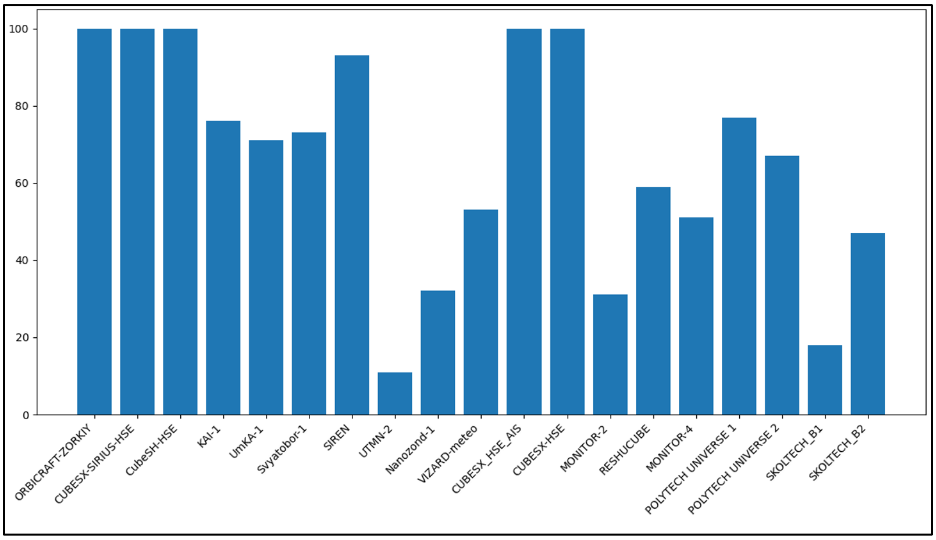

Fig. 7. Diagram of the basic model of reception time distribution

One way to improve performance is to fill the schedule with intersections in a random order. It is necessary to randomly select exactly the intersection of the satellite from the list of all intersections, and not the satellite and all its intersections. In this case, the time of reception of data by the ground station from satellites is equalized. Figure 8 shows the reception distribution plot for a random distribution of observations.

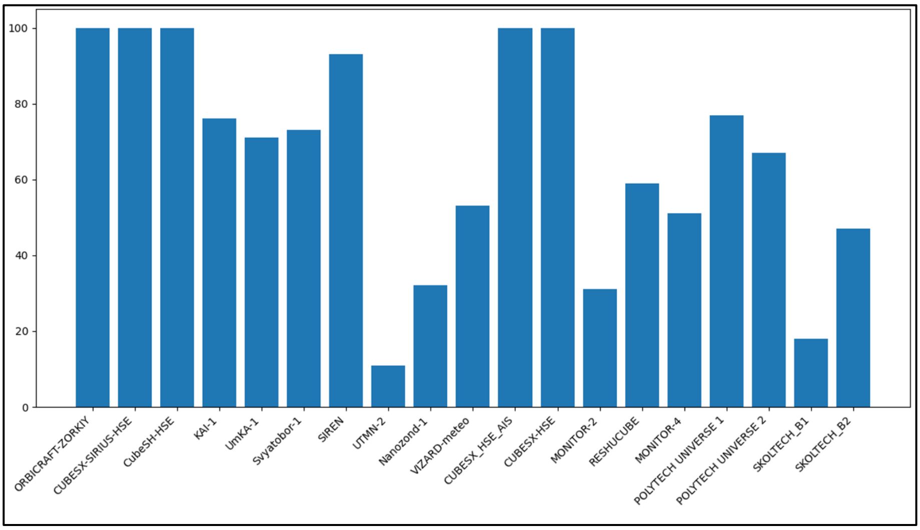

Fig. 8. Diagram of schedule distribution using randomness

In this case, the standard deviation is 24.52, the arithmetic mean is 67.15. Using formula above we get 42.63. Satellites with virtually no reception time received additional intervals for data transmission. This is accomplished by reducing the reception time of other satellites.

5 Conclusion

As a result of the research, a system was developed that is capable of independently generating and executing a schedule for the operation of a ground station for receiving data from small spacecraft. The paper presents examples of calculating satellite coordinates, time of intersection with a ground station, and direction to the satellite. Implemented control of the rotary device and reception frequency. The process of developing program modules and one of the possible schedule generation algorithms is shown. For the developed system, a server has been created that can send the generated schedule upon request. A client has been created to automatically control the operation of a ground station according to the received schedule. The work also shows ways to evaluate the effectiveness of the created system.

The system developed during the research is used by small spacecraft operators to interact with satellites orbiting the Earth.

6 Acknowledgment

The research was supported by the Russian Science Foundation and the Government of the Arkhangelsk Region, project No. 23-11-20026.

References

1. T. Villela, C.A. Costa, A.M. Brandão, F.T. Bueno, R. Leonardi, *Towards the thousandth CubeSat: A statistical overview*, International Journal of Aerospace Engineering, **2019(1)**, 5063145, (2019), doi: https://doi.org/10.1155/2019/5063145.
2. N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T.Y. Al-Naffouri, M.S. Alouini, *CubeSat communications: Recent advances and future challenges*, IEEE Communications Surveys & Tutorials, **22(3)**, pp. 1839-1862, (2020), doi: https://doi.org/10.1109/COMST.2020.2990499.
3. E. Peral, E. Im, L. Wye, S. Lee, S. Tanelli, Y. Rahmat-Samii, D. Hawkins, *Radar technologies for earth remote sensing from cubesat platforms*, Proceedings of the IEEE, **106(3)**, pp. 404-418, (2018), doi: https://doi.org/10.1109/JPROC.2018.2793179.
4. N. Ivliev, V. Evdokimova, V. Podlipnov, M. Petrov, S. Ganchevskaya, I. Tkachenko, V. Soifer, *First earth-imaging CubeSat with harmonic diffractive lens*, Remote Sensing, **14(9)**, 2230, (2022), doi: https://doi.org/10.3390/rs14092230.
5. D. Dumitriu, P.U. Lima, B. Udrea, *Optimal trajectory planning of formation flying spacecraft*, IFAC Proceedings Volumes, **38(1)**, pp. 313-318, (2005), doi: https://doi.org/10.3182/20050703-6-CZ-1902.02013.
6. A.R. Periyapatna, M.T. Murali, S.C. Basavaraju, S.M. Bagali, N.R. Goparaju, J. Manikandan, A. Purohith, *Automation of a Satellite Earth Station*, IEEE 8th International Conference for Convergence in Technology (I2CT), pp. 1-6, (2023), doi: https://doi.org/10.1109/I2CT57861.2023.10126424.
7. R.N. Zharkikh, A.V. Purikov, V.V. Kuznetsov, M.I. Shubin, A.M. Svinareva, L.L. Eletskaya, *Construction of wide-purpose space complexes based on the SXC6 platform*, AIP Conference Proceedings, **2318(1)**, (2021), doi: https://doi.org/10.1063/5.0038178.
8. S. Baghel, D. Mohanty, A. Tamrakar, T. Dewangan, *Accuracy Assessment of SGP4 and TLE Orbital Information*, Mathematical Statistician and Engineering Applications, **71(4)**, pp. 9675-9680, (2022), doi: https://doi.org/10.17762/msea.v71i4.1768.