

# Processing Principles of Ionosphere Passive Monitoring Data

Dmitry M. Markov  
North-Caucasus Federal University, Russia  
dmitri13.1991@gmail.com

Alexandr F. Chipiga  
North-Caucasus Federal University, Russia  
chipiga.alexander@gmail.com

## Abstract

Modern technologies allow to carry out passive monitoring of the ionosphere with the possibility to track changes in ionospheric parameters with a high sampling rate and influence of these parameters on quality of navigation and spatial positioning. With the growth of the sampling frequency the amount of data being processed also increases. The article deals with the principles that have been selected to handle the large amounts of passive monitoring ionospheric data and software solution, which was developed on the basis of these principles.

## 1 Introduction

The Science of Big Data is a young field of knowledge, which is developing dynamically. It covers the processing of any data: word processing, processing of financial statistics or results of long-term researches. This is so large amount of data, which cannot be processed with traditional methods within a reasonable time.

Data can come from various sources and in different formats, also the data could be received from sources at different time intervals. Development of processing methodology of such data, and ensuring of maximum performance is quite a challenge because in the course of the design of such systems is necessary to solve a number of problems which will be determined by technical requirements of the product.

Hardware and software complex of real-time monitoring of ionospheric parameters was developed during the study of ionospheric parameters, which are described in the articles [PEP06, F.13b, F.13a, AFPI15, FV14, F.14, FMV15, M.16b, M.16a, F.15, PFV14, MFV16]. Separate software was developed in order to investigate the possibility of applying machine learning methods. The main purpose of the software is post-processing of collected ionospheric parameters data.

In this paper we introduce the new software programs which are used in studying of ionosphere in North-Caucasus Federal University. The purpose of the article is the systematization and analysis of developed software solutions, principles and algorithms that are used in real-time processing and post-processing of the large amount of the ionospheric parameters measurements.

## 2 Shortcomings of Existing Software Solutions

Working with the receiver of satellite navigation signals implies continuous processing of the incoming data stream. Some research and experience of stream processing were described in the articles [ScZ05, CcC<sup>+</sup>02,

---

*Copyright © 2017 by the paper's authors. Copying permitted for private and academic purposes.*

In: S. Hölldobler, A. Malikov, C. Wernhard (eds.): *YSIP2 – Proceedings of the Second Young Scientist's International Workshop on Trends in Information Processing, Dombai, Russian Federation, May 16–20, 2017*, published at <http://ceur-ws.org>.

CDTW00]. These solutions [ScZ05, CcC+02, CDTW00] provide to developers a complete tools with their advantages and disadvantages. The main advantage of these solutions is their flexibility for different tasks of streaming data processing, but the universality in this case is also a disadvantage because the use of universal application programming interface (API) imposes additional costs that would affect to the speed of data processing. A key factor for the construction of high-precision navigation systems and spatial positioning is the high speed of production of the processed data. Therefore, the use of ready-made solutions for the processing of data is poorly applicable to research problems of ionospheric parameters, which are described in the articles [PEP06, F.13b, F.13a, AFPI15, FV14, F.14, FMV15, M.16b, M.16a, F.15, PFV14].

Most part of modern systems for processing streaming data are commercial and open source projects are not developing. For our study of the ionosphere problems there is no need for a specialized API, which would allow to execute queries with parameters on the server-side. All identified disadvantages lead to the conclusion that in this case independent development of software solutions for the implementation of the research objectives requires.

### 3 Hardware and Software Environment

Hardware and software complex consists of a receiver of satellite navigation signals Novatel GPStation-6, notebook, server of real-time data processing and client software (Figure 1). Data server in real-time is controlled by GNU/Linux with a specially designed software. The notebook is used for connection of the receiver and the server. Using a laptop is a necessity because the receiver is not able to connect directly to the network, as well as the receiver and the server are physically separated. The client software is an application which receives data via TCP-connections and processes the stream of the calculated values. In order to monitor the real-time state of the ionosphere a special application that is able to cache data observations for the last 30 minutes was developed.

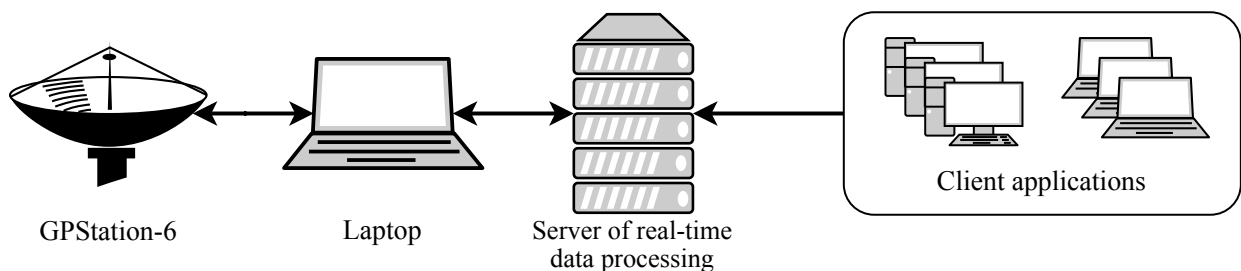


Figure 1: The scheme of hardware and software complex for monitoring of the ionospheric parameters in real-time

Everyday receiver Novatel GPStation-6 generates  $\approx 7$ GB of data in a binary format. Daily at 00:00 UTC the server of real-time data processing creates a new file of daily measurements, after that the old files are archived and transferred to the post-processing server. General scheme of the organization of the two systems is shown in Figure 2.

The developed software that is installed on both servers for processing measurement data of Novatel GPStation-6, uses a common source code for the same calculations.

The main purpose of post-processing data server is a training of machine learning models. Using of machine learning algorithms is designed for identifying new dependencies or improve existing mathematical apparatus, which has been accumulated over the years of research of the ionosphere [AP06, VVS13, PEP06, F.13b, F.13a, AFPI15, FV14, F.14, PFV14]. The main purpose of creation of hardware and software is studying of the ionosphere small-scale irregularities impact on the signal between the satellite and the receiver, which are described in detail in previous articles [IMV15, MI15, MIS15, MF15, M.15, FMV15] so the main mathematical tools are taken from the materials of previous studies [AP06, VVS13, PEP06, F.13b, F.13a, AFPI15, FV14, F.14, PFV14].

### 4 Server of Real-Time Data Processing

The operating system (OS) on the server has been selected from GNU/Linux family, because it provides high stability of its work. This family of operating systems allows to develop applications which used all available computing resources with maximum efficiency [Gre13]. For OS GNU/Linux family there is a full stack of open source license tools for development of any kind software.

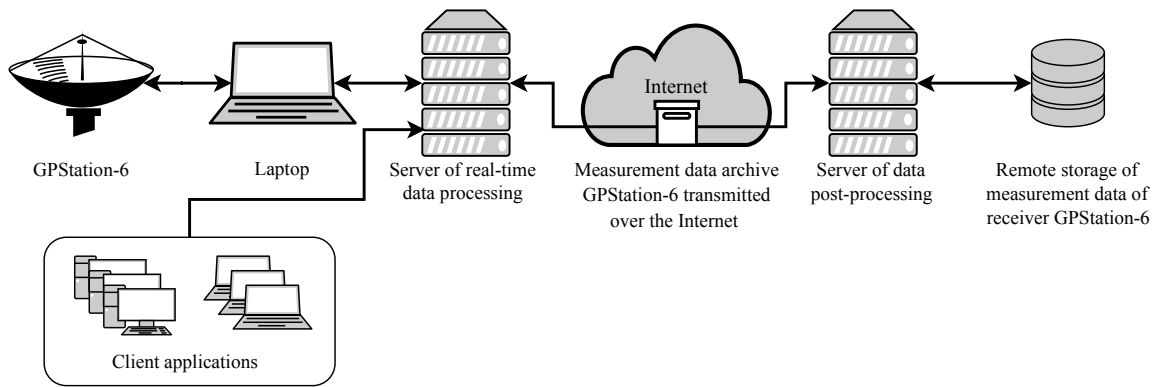


Figure 2: General scheme of the organization of the two systems for the study of ionospheric parameters

The main component of the software part of the hardware-software complex is designed software that is installed on the data server. The server software is a classical application server that accepts all incoming data flows from customers, provides processing and sends the result to clients. In common case the client applications can act as monitors for the current state of the ionosphere. Within the complex a client application for monitoring the state of the ionosphere was developed because all basic calculations are provided on the server side. Despite the fact that the application server implements most of the necessary algorithms for data processing and computing finite values the client application can perform their own calculations. The client applications can duplicate calculations using the part of values obtained from server. Also they can perform any other calculations with obtained values with help of new methods. In this case, the data server acts as a source of reference values.

When designing an application server first concept assumed that all values will be recorded in the database. Database management system (DBMS) will produce a calculation of all values. The first practical results have shown that the idea failed because the allocated computing resources were insufficient for such tasks. As a result, an application server was developed, which performs the calculations only in real-time and stores original data stream for post-processing in the file.

This file contains measurements of the data flow not more than from one day and which begins not earlier than 00:00 UTC for the selected day. If during the day there was a break connection between the laptop and the server, the new file will be created. Thus, each file comprises a continuous stream of data. Once the data file in the previous period is complete, the file is archived in order to save disk space. In practice the standard GNU Zip algorithm compresses the file by 25% on average. By using the algorithm LZMA2 compression reaches 50%, but it requires one core of CPU, 1GB of RAM and a considerable amount of time. When unpacking LZMA2 it also requires one CPU core. Server of post-processing data occupied all processor cores for data processing. As a result, for file compression algorithm GNU Zip has been selected, so size of one archive file for a full day is  $\approx 5$ GB.

Unlike existing solutions for the processing of streaming data, which are described in articles [ScZ05, CcC<sup>+</sup>02, CDTW00] the developed application server does not provide any kind of API for sending query to the server. The server sends all values to the client applications which perform necessary data rework.

As a programming language for developing applications server programming language C/C++ was selected, because it allows to maintain a balance between the written efficiency code and speed of development. An external application, which is written in Python 3, provides formation of plots.

## 5 Real-time Data Processing Principles of Application Server

Receiver Novatel GPStation-6 has the ability to measure a sufficiently large number of values [nov] that can be used in various scenarios of studies or geolocation. However, these opportunities also have a negative effect. The main problem with processing data of Novatel GPStation-6 is the heterogeneity of the measurement formats. Some results are conveniently processed data slice for all satellites at a predetermined sampling frequency and other data slice are separated by individual logs which grouped by type of satellite system. Some measurements are grouped slice of results for period of time with the delta of measured values.

This representation of the data automatically leads to the fact that calculations of data is always delayed

because it needs to expect all results in various formats, then synchronize data and only after that perform the necessary calculations. Table 1 shows an example of calculation algorithm input data. The data that will be skipped and not used for calculations, called inconsistent, as this data is not complete. The data that will be used for calculations, called coherent because each tuple contains all the necessary values for a single algorithm.

Table 1: An example of coherent and inconsistent data

	Time	Elevation angle	Pseudorange L1, m	Pseudorange L2, m
Inconsistent data	10:00:00.040		20152153,623	
	10:00:00.060		20152154,123	
	10:00:00.080		20152154,623	20152156,306
	10:00:00.100		20152155,123	20152157,200
	10:00:00.120		20152155,623	20152158,094
	10:00:00.140	20,001		20152158,988
Coherent data	10:00:00.160	20,011	20152156,623	20152159,882
	10:00:00.180	20,012	20152157,123	20152160,776
	10:00:00.200	20,013	20152157,623	20152161,670
	10:00:00.220	20,014	20152158,123	20152162,564
	10:00:00.240	20,015	20152158,623	20152163,458
	10:00:00.260	20,016	20152159,123	20152164,352

On the basis of the existing features of Novatel GPStation-6 receiver data formats we offer the following guidelines for data processing:

1. Each type of calculation must not depend on other calculations, therefore, every time when there is need to perform the calculations the copy of all data should be created. In a multi-threaded processing it allows to avoid the need of multiple data lock and free the memory immediately after executing of calculation which used this data.
2. Any type of calculation must be a class that contains all required buffers for storing time series of measurements. This requirement is specified by using the pattern "expert information" [GHJ<sup>+</sup>94].
3. Some values cannot be measured by the receiver with a high sampling rate (the desired measurement sampling rate is 50Hz), so that data is automatically duplicated to the desired sampling rate. One of these values is the elevation angle of the satellite. The receiver can perform the measurement values required for the calculation of the satellite elevation angle with a frequency of 1Hz, but for the convenience of further calculations, this value is duplicated 50 times. It allows to reduce delaying in the calculations.

Using the principles of data processing, based on the well-known design patterns software [Fow02, GHJ<sup>+</sup>94] the procedure of data was developed (Figure 3). Multi-threaded processing begins at the moment of the creation of independent tuples that will be send to threads, which will perform the calculations. Each thread uses a class that implements one kind of calculation, by means of what the principle of sole responsibility is achieving.

Special container was designed for multi-threaded processing of the incoming data stream, which allows simultaneously record data and perform calculations using the data in the container. The container is implemented as a C++ template. Figure 4 shows the activity diagram that describes principle of the container functioning, which is used simultaneously by two threads.

Figure 4 shows that the central method of the container is the full locked movement from the writing buffer to the end of the reading buffer which will be used for blocking reading. Thus, the container allows to push new values without locking of reading buffer and reading of data don't block the pushing of new values. After reading the data can be safety removed from the container.

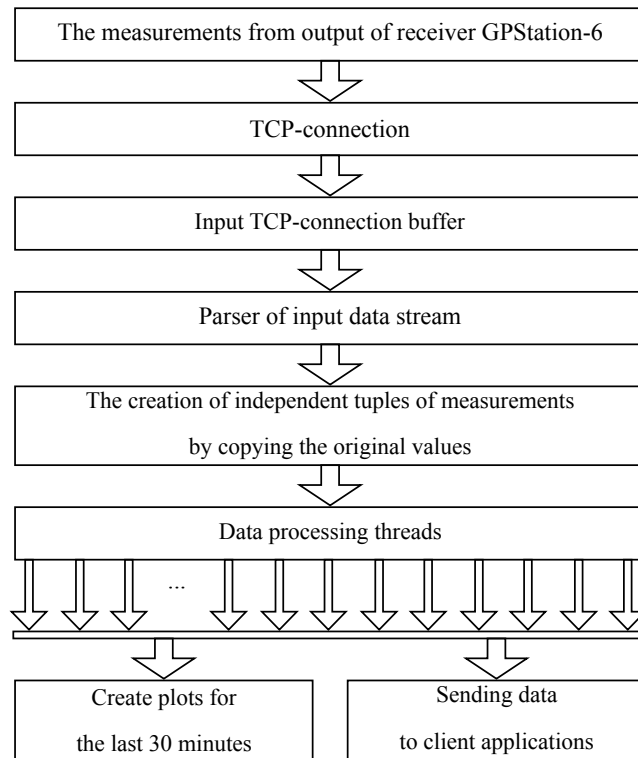


Figure 3: The processing procedure of measurements flow in real-time

## 6 Server of Post-Processing Data

For post-processing special software has been developed which performs multi-threaded processing of collected data. The speed of single file processing is the critical parameter, so the application uses aggressive optimization of code and large amounts of RAM. The basic principles of data processing have remained unchanged, but in comparison with processing in real-time there are additional mechanisms which are necessary for post-processing and require separate consideration.

For post-processing there are two variants of archived data, which divided into separate files:

1. sequential unpacking and file processing;
2. merging data in continuous stream of compressed data, which will be uncompressed and redirected to standard input (stdin) of post-processing server software.

The first option is easier in terms of debugging, but it requires additional disk space and processing time of each file is extended by the time required for the decompression of the file. The first option makes it easier to keep track of the specific file processing.

The second option is more abstract for post-processing software, as all data will look like a continuous data stream without file separation. Merging of compressed files as uncompressed data stream is possible with the help of standard tools `cat` and `tar` of GNU/Linux. Redirecting of standard error (`stderr`) from the `tar` program to the file allows to monitor the processed files.

Based on the advantages of each method of archived data processing, at the stage of development and debugging of software the first option was implemented. Then, the second option was implemented to perform post-processing tasks in the real server. Figure 5 shows a flow measurement processing procedure, which is implemented in the post-processing application.

Application has been optimized with the help of compiler for Intel Haswell processor architecture in the real server for maximum performance. Compiler optimization gave a further boost in performance. The developed software used special large memory pages (Hugepages) in size of 2MB instead of the usual 4KB, which allows to organize the use of memory buffers effectively and reduce the load in general, because the data of one thread is located within two or three pages of memory [hug].

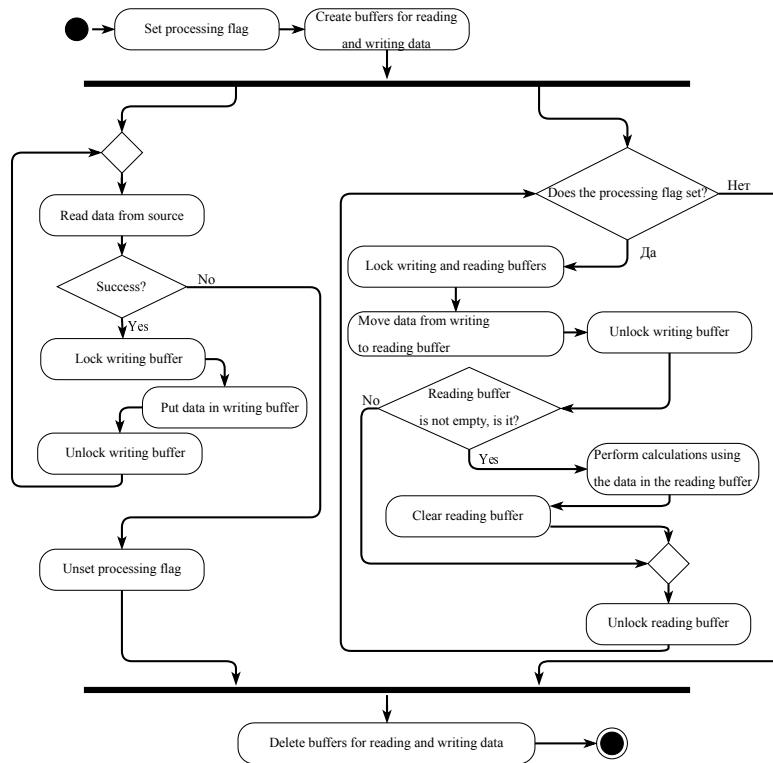


Figure 4: Activity diagram of the container that shared by two threads

Table 2 shows the approximate processing elapsed time of collected ionospheric parameters measurements. For post-processing server the following configuration of virtual private server is used:

- 16GB RAM DDR4;
- Intel Xeon Processor E5-2650 v3 (Haswell) (only 5 CPU cores is available);
- HDD 500GB;
- OS Ubuntu Server 16.04.

Table 2: The results of post-processing application test

The number of processed daily files	The elapsed time for processing
1	≈ 20 minutes
3	≈ 60 minutes
72	≈ 1440 minutes (twenty-four hours)
365	≈ 7300 minutes (5 twenty-four hours)

## 7 Summary

In this article we show that the real-time processing and post-processing of data from the receiver Novatel GPStation-6 is not a trivial task. We described successful implementation of hardware and software complex, which is used for real-time processing of time-series measurements. Post-processing complex has been developed, which allows to identify new mathematical relationships between the ionosphere parameters and their influence

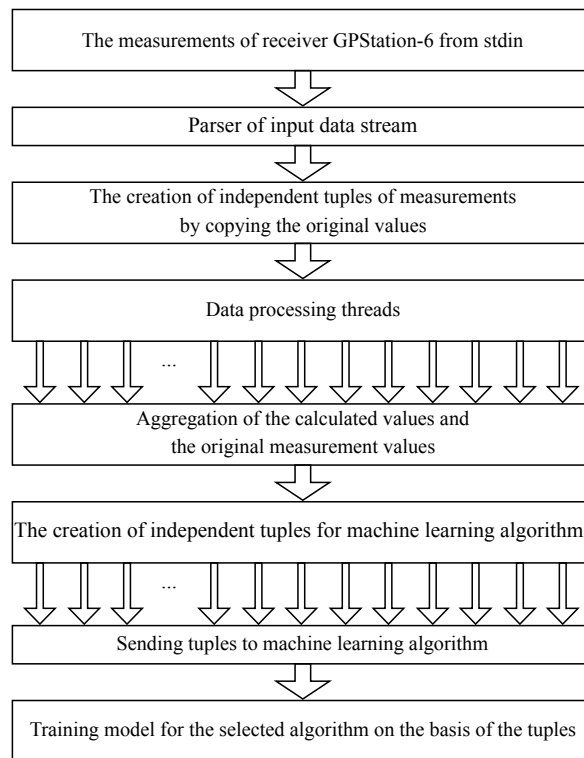


Figure 5: The procedure of post-processing measurements

on the parameters of satellite navigation and communications by using machine learning techniques. Using of post-processing complex allows to obtain an improved method for forecasting of ionospheric parameters. The results are used as input for the construction of high-precision navigation systems and spatial positioning systems.

During the development of software for processing data from the receiver Novatel GPStation-6 we found that existing solutions are not applicable for our scenarios. First of all, the modern solutions for processing of time-series are commercial. Then we found that open source solutions for time-series processing are not developed for some years. Than practice of DBMS using shows that small computing resources are insufficient for processing of time-series with high sampling rate. As expected development of special software for our tasks allows to use all available computing resources with maximum efficiency, but it takes some time.

We found that maximum performance of Novatel GPStation-6 data processing is possible with aggressive using of RAM and optimization of code. Using of hugepages allows to reduce the size of system memory pages table [hug] and improve the performance of data processing. All of used optimization techniques are available only in UNIX-like family, so the using of other operating systems are not possible.

## References

- [AFPI15] Shevchenko V. A., Chipiga A. F., Pashintsev V. P., and Toporkov K. I. Prediction of noise immunity of satellite communication on the results of monitoring ionospheric scintillation index. *Information and Communication technologies*, 13(4):365–375, 2015. (In Russian).
- [AP06] Afraimovich E. A. and Perevalova N. P. *GPS-monitoring the Earth's upper atmosphere*. RAS SB ISTP, 2006. (In Russian).
- [CcC+02] Don Carney, Uğur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Greg Seidman, Michael Stonebraker, and Stan Zdonik Nesime Tatbul. Monitoring streams – a new class of data management applications. In *Proceedings of the 28 th VLDB Conference*. Hong Kong, China, 2002.
- [CDTW00] Jianjun Chen, David J. DeWitt, Feng Tian, and Yuan Wang. NiagaraCq: A scalable continuous query system for internet databases. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 379–390. 2000.

- [con15] *I All-Russian Scientific and Technical Conference Fundamental and applied aspects of computer technology and information security*, Rostov-on-Don, 2015. SFU. (In Russian).
- [con16] *Student's science for the development of the Information Society: proceedings of V All-Russian Scientific and Technical Conference*, Stavropol, 2016. NCFU. (In Russian).
- [F.13a] Chipiga A. F. Analysis of possibilities for the practical implementation of the satellite communications system on a plot spacecraft - land when working at low frequencies. *The science. Innovation. Technologies*, (1):63–71, 2013. (In Russian).
- [F.13b] Chipiga A. F. The choice of parameters of technical means of satellite communication using low frequencies and dual receiving signals. *Bulletin of the North Caucasus Federal University*, (4):15–20, 2013. (In Russian).
- [F.14] Chipiga A. F. Analysis of the energy of low-frequency stealth satellite communication systems of signal detection. *Proceedings of SFU. Technical science.*, (2):209–217, 2014. (In Russian).
- [F.15] Chipiga A. F. Development of mathematical models of the transmission loss estimation due to absorption of waves in the ionosphere. In *Proceedings of the international scientific-technical conference it. Leonardo da Vinci*, page 2015. Verain "Wissenschaftliche Welt", Berlin, 2015.
- [FMV15] Chipiga A. F., Markov D. M., and Slyusarev G. V. Effect of packed formats protocols measurements on the speed of data processing in adaptive systems, satellite communications. *Fundamental researches*, 4(11), 2015. (In Russian).
- [Fow02] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002.
- [FV14] Chipiga A. F. and Slyusarev G. V. The noise value in pseudorange in the satellite radio navigation system under perturbations of the ionosphere. *Fundamental researches*, 2(12):263–268, 2014. (In Russian).
- [GHJ+94] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Grady Booch. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
- [Gre13] Brendan Gregg. *Systems performance: enterprise and the cloud*. Prentice Hall, 2013.
- [hug] *A brief summary of hugetlbpage support in the Linux kernel*.
- [IMV15] Toporkov K. I., Markov D. M., and Peskov M. V. Model of complex of software of monitoring ionosphere in real-time. In *I All-Russian Scientific and Technical Conference Fundamental and applied aspects of computer technology and information security* [con15], page 465. (In Russian).
- [M.15] Markov D. M. Selection and optimization of the structure of a database for storing and processing the measurement results of research equipment. In *Proceedings of the international scientific conference of students, graduate students and young scientists Youth and Science: Avenue Free*. SFU, Krasnoyarsk, 2015. (In Russian).
- [M.16a] Markov D. M. Methods of calculating the scintillation index with a high sampling rate for gpstation-6 receiver. In *Student's science for the development of the Information Society: proceedings of V All-Russian Scientific and Technical Conference* [con16], pages 365–368. (In Russian).
- [M.16b] Markov D. M. The technique of phase ambiguity resolution for post-processing tasks observing the ionosphere. In *Student's science for the development of the Information Society: proceedings of V All-Russian Scientific and Technical Conference* [con16], pages 369–372. (In Russian).
- [MF15] Markov D. M. and Chipiga A. F. Analysis of speed processing of compressed formats, protocols measurements of gpstation-6 receiver. In *Proceedings of the international scientific-practical conference Youth Forum: technical and mathematical sciences*. VSFU, Voronezh, 2015. (In Russian).
- [MFV16] Markov D. M., Chipiga A. F., and Stepanenko A. V. Determination of the exact coordinates of stationary gps/glonass receiver. *The science. Innovation. Technologies*, (1):47–62, 2016. (In Russian).



- [MI15] Markov D. M. and Toporkov K. I. A study known hardware and software systems for monitoring the state of the ionosphere. In *XXXIV All-Russian Scientific and Technical Conference Problems of efficiency and safety of complex technical and information systems*. Serpukhov, 2015. (In Russian).
- [MIS15] Markov D. M., Toporkov K. I., and Bondarenko O. S. Analysis of processing speed of data format of receiver gpstation-6. In *I All-Russian Scientific and Technical Conference Fundamental and applied aspects of computer technology and information security [con15]*, page 465. (In Russian).
- [nov] *Novatel OEM6® Family Firmware Reference Manual*.
- [PEP06] Pashintsev V. P., Solchatov M. E., and Gakhov R. P. *Influence of the ionosphere on the characteristics of space communication systems*. FIZMATLIT, Moscow, 2006. (In Russian).
- [PFV14] Pashintsev V. P., Chipiga A. F., and Galkin V. A. Lyakhov A. V. Determination of the intensity of ionospheric irregularities with the help of low-frequency satellite communication systems. In *Proceedings Sixth International Scientific and Technical Conference Info-communication technologies in science, business and education (Infocom-6)*, pages 204–210. NCFU, Stavropol, 2014. (In Russian).
- [ScZ05] Michael Stonebraker, Uğur Çetintemel, and Stan Zdonik. The 8 requirements of real-time stream processing. *ACM SIGMOD Record*, 34(4):42–47, 2005.
- [VVS13] Demyanov V. V., Yasukevich Y. V., and Dzin S. Control of the current propagation conditions of navigation satellites signals. *Solar-terrestrial physics*, (22):35–40, 2013. (In Russian).