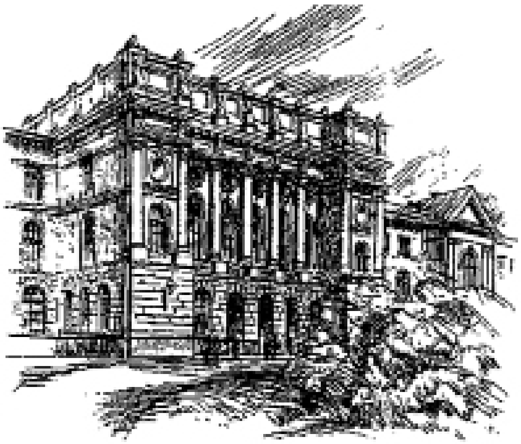
THE MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION



Computing, Telecommunications and Control

Vol. 13, no. 1
2020

Peter the Great St. Petersburg
Polytechnic University
2020

COMPUTING, TELECOMMUNICATIONS AND CONTROL

EDITORIAL COUNCIL

EDITORIAL BOARD

The journal is included in the List of Leading PeerReviewed Scientific Journals and other editions to publish major findings of PhD theses for the research degrees of Doctor of Sciences and Candidate of Sciences.

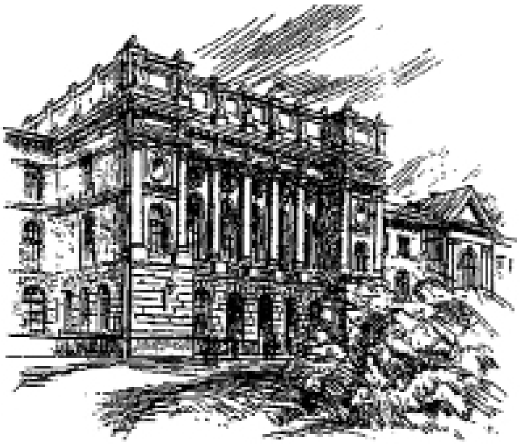
The journal is indexed by Ulrich's Periodicals Directory, Google Scholar, EBSCO, ProQuest, Index Copernicus, VINITI RAS Abstract Journal (Referativnyi Zhurnal), VINITI RAS Scientific and Technical Literature Collection, Russian Science Citation Index (RSCI) database © Scientific Electronic Library and Math-Net.ru databases.

The journal is registered with the Federal Service for Supervision in the Sphere of Telecom, Information Technologies and Mass Communications (ROSKOMNADZOR). Certificate ЭЛ No. ФС77-77378 issued 25.12.2019.

The views of the authors can contradict the views of the Editorial Board.

The address: 195251 Polytekhnicheskaya Str. 29,
St. Petersburg, Russia.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ



Информатика, телекоммуникации и управление

Том 13, № 1
2020

Санкт-Петербургский политехнический
университет Петра Великого
2020

ИНФОРМАТИКА, ТЕЛЕКОММУНИКАЦИИ И УПРАВЛЕНИЕ

РЕДАКЦИОННЫЙ СОВЕТ ЖУРНАЛА

РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА

Журнал с 2002 года входит в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

Сведения о публикациях представлены в Реферативном журнале ВИНИТИ РАН, в международной справочной системе «Ulrich`s Periodical Directory», в базах данных Российский индекс научного цитирования (РИНЦ), Google Scholar, EBSCO, Math-Net.Ru, ProQuest, Index Copernicus

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ЭЛ № ФС77-77378 от 25.12.2019.

При перепечатке материалов ссылка на журнал обязательна.

Точка зрения редакции может не совпадать с мнением авторов статей.

Адрес редакции: Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29.

Тел. редакции (812) 552-62-16.

Contents

Telecommunication Systems and Computer Networks

Software of Computer, Telecommunications and Control Systems

Hardware of Computer, Telecommunications and Control Systems

Circuits and Systems for Receiving, Transmitting, and Signal Processing

System Analysis and Control

Information and Telecommunication Technology in Education

Содержание

Телекоммуникационные системы и компьютерные сети

Программное обеспечение вычислительных, телекоммуникационных и управляющих систем

Аппаратное обеспечение вычислительных, телекоммуникационных и управляющих систем

Устройства и системы передачи, приёма и обработки сигналов

Системный анализ и управление

Инфокоммуникационные технологии в образовании

Dear Readers and Colleagues,

The Editorial Board has informed you previously about the changing of the Journal format and its publication totally in English from the year 2020. It is necessary to note that we have got a such experience (see, for example, June 2018, March 2018), but this is the first issue which has been prepared in English on the regular basis.

We would like to thank all authors, reviewers, and editorial board members who took part in the issue preparation. Thus, we hope that our Journal is on the right way to be included to the Scopus database.

Editor in Chief,

Prof. Dr. Alexander Korotkov

DOI: 10.18721/JCSTCS.13101
УДК 004.75

AN APPROACH FOR AUTOMATED DEPLOYMENT OF CLOUD APPLICATIONS IN THE EDGE-TO-CLOUD COMPUTING CONTINUUM SATISFYING HIGH QUALITY OF SERVICE REQUIREMENTS

P. Kochovski, P.D. Drobintsev

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

Modern component-based software engineering environments allow deployment of cloud applications on various computing infrastructures, such as Edge-to-Cloud infrastructures. The heterogeneous nature of such computing resources results in variable Quality of Service (QoS). Therefore, the deployment decision can seriously affect the application's overall performance. This study presents an approach for automated deployment of cloud applications in the Edge-to-Cloud computing continuum that considers non-functional requirements (NFRs). In addition, the authors explore multiple methods for selection of optimal cloud infrastructure, such as IaaS. The paper presents an experimental evaluation performed using a cloud application for storing data under different workloads. For the purposes of the experimental evaluation, a Kubernetes cluster composed of 44 computing nodes was used. The cluster nodes were geographically distributed computing infrastructures hosted by several service providers. The proposed approach allows a reliable selection of infrastructures, which satisfy high QoS requirements for cloud applications, from heterogeneous Edge-to-Cloud computing environments.

Keywords: cloud computing, cloud application deployment, Quality of Service, Infrastructure as a Service, Edge-to-Cloud.

Citation: Kochovski P., Drobintsev P.D. An approach for automated deployment of cloud applications in the Edge-to-Cloud computing continuum satisfying high Quality of Service requirements. Computing, Telecommunications and Control, 2020, Vol. 13, No. 1, Pp. 8-18. DOI: 10.18721/JCSTCS.13101

ПОДХОД К АВТОМАТИЗИРОВАННОМУ РАЗВЕРТЫВАНИЮ ОБЛАЧНЫХ ПРИЛОЖЕНИЙ В ВЫЧИСЛИТЕЛЬНОМ КОНТИНУУМЕ EDGE-TO-CLOUD, УДОВЛЕТВОРЯЮЩИХ ВЫСОКИМ ТРЕБОВАНИЯМ К КАЧЕСТВУ ОБСЛУЖИВАНИЯ

П. Кочовски, П.Д. Дробинцев

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Современные среды разработки программного обеспечения на основе компонентно-ориентированного программирования позволяют беспрепятственно развертывать облачные приложения в различных вычислительных инфраструктурах, таких как Edge-to-Cloud. Неоднородная природа таких вычислительных ресурсов приводит к непостоянному качеству обслуживания (QoS). Поэтому решение о развертывании приложения может серьезно повлиять на его общую производительность. В статье рассмотрен подход к автоматизированному развертыванию облачных приложений в вычислительном континууме Edge-to-Cloud, учитывающий нефункциональные требования (NFR). Исследованы способы выбора оптимальной услуги с точки зрения ожидаемого качества обслуживания. Экспериментальная оценка проведена с помощью облачного приложения для хранения данных в трех случаях с разной нагрузкой. Проведены эксперименты на кластере Kubernetes, состоящем из 44 вычислительных узлов (облачных инфраструктур). Узлы кластера были географически распределены в нескольких местах и размещались несколькими поставщиками услуг. Подход позволит надежно выбирать инфраструктуры из гетерогенных Edge-to-Cloud сред, удовлетворяющих требованиям к качеству обслуживания облачных приложений.

Ключевые слова: облачные вычисления, развертывание облачных приложений, качество обслуживания, инфраструктура как услуга, Edge-to-Cloud.

Ссылка при цитировании: Кочовски П., Дробинцев П.Д. Подход к автоматизированному развертыванию облачных приложений в вычислительном континууме Edge-to-Cloud, удовлетворяющих высоким требованиям к качеству обслуживания // Информатика, телекоммуникации и управление. 2020. Т. 13. № 1. С. 8-18. DOI: 10.18721/JCSTCS.13101

Introduction

Intensive development of the Internet-of-Things (IoT), has led towards the development of smart applications in different domains (e.g. smart cities, smart environments, industry 4.0). In order to assure high Quality of Service (QoS), a plethora of non-functional requirements (NFRs), such as computing and network performance must be addressed throughout application's life-cycle.

Novel software development environments support software development based on component-based development (CBD). In other words, they allow to compose software from existing microservices, discovery and selection of computing resources, deployment and resource orchestration in heterogeneous computing environments. The deployment process of microservices in such heterogeneous environments, covering the complete computing Edge-Fog-Cloud continuum, is a difficult problem. At this stage the microservice needs to be placed on optimal or near to optimal infrastructure from a large number of options, whilst considering multiple quality constraints.

The goal of this study is to describe an approach for resource balancing that increases application's performance by ranking and automatically deploying applications on optimal Edge-to-Cloud computing infrastructure.

Related work

The selection of optimal computing infrastructure (i.e. Infrastructure as a Service – IaaS) has been a point of interest in various studies related to load balancing [1–4], resource management and allocation [5–8], resource provisioning [9–11] or service placement and management systems [12, 13]. Since in such cases is necessary to consider large number of NFRs, the reviewed studies recommend implementing various multi-criteria approaches for different placement scenarios in Edge-to-Cloud computing environments.

The authors [14–16] describe the implementation of a multi-criteria decision-making method called the Analytic Hierarchy Process (AHP) for ranking various computing infrastructures. In order to perform the ranking, the AHP executes pairwise comparison of infrastructures instead of considering the software engineer's QoS requirements.

Zheng et al. [17] proposed a framework that ranks the infrastructures according to QoS requirements. In order to perform the ranking, the framework implements two forecasting algorithms, which calculate the ranking results based on the software engineer's QoS requirements. However, the framework's results are only based on network-level measurements data from prior usage experience.

Another commonly used method that is used to compute optimal cloud infrastructure is the Pareto method, which is used to find the optimal set of solutions by performing a trade-off between the conflicting objectives. Guerrero et al. [18] described an approach for resource allocation that is based on the Pareto optimization and implements Non-dominated sorting genetic algorithm (NSGA-II). In addition, another study [19] also utilizes the Pareto optimization for a trade-off of non-functional requirements at the earliest stages of the software development process and thus place the software on an optimal cloud infrastructure. However, the reviewed Pareto-based solutions suggest their results are based on no more than three criteria. Although some studies proposed more than three criteria, they all combined them into two or three main criteria upon which a trade-off was performed and a decision was derived. Using Pareto optimization for more than three criteria is also computationally expensive [20]. Moreover, it is impossible to visualize the Pareto curve on one figure for more than three criteria.

In comparison to the studies considered above, where cloud computing is considered as deterministic, multiple studies consider cloud computing as stochastic [21, 22]. In particular, they investigate the infrastructure's dependability on uncertainty. Moreover, numerous studies in various domains utilize Markov decision-process (MDP) to make decisions in random cases where unforeseen situations may arise. In this context, MDP is also suitable for applications in the field of cloud computing due to its stochastic nature. Yang et al. [23] present an MDP-based method to select a deployment infrastructure that provides optimal performance for applications. Su et al. [24] also proposed an MDP-based planning mechanism that maintains a compromise between the three attributes (accuracy, data usage, and computational cost) by implementing an iterative approach for decision making. In addition, the studies of Tsoumakos et al. [27] and Naskos et al. [28] applied MDP to the problem of horizontal scaling of virtual machines. However, their computational complexity hinders the integration of such methods in the main software practices, thus this challenge has not been addressed in existing studies. MDP also allows to formally verify the correctness of the deployment decision placement. Llerena et al. [25] developed a methodology for analyzing the influence of probability perturbations by checking the reachability properties of MDP models with applications for cloud computing.

Nevertheless, as far as we know, the use of MDP to ensure high quality of service when deploying a software component in the context of containers within the Edge-to-Cloud computing environments has not yet been considered.

Approach for automated deployment of microservices in Edge-to-Cloud computing environments

The foundation of this work is the hypothesis of implementing MDP as an effective decision-making mechanism for deploying microservices on an optimal cloud infrastructure by taking into account specific quality requirements, relevant infrastructure and network measurements.

Fig. 1 illustrates the process of ranking all available computing infrastructures and automated deployment of microservices by considering NFRs and their utilization context. The process is composed of five consecutive steps that are described as follows.

At the first step, the software engineer composes an application from containerized microservices that need to be deployed. The engineer selects the important NFRs, such as: location, cost, network performance, infrastructure performance and etc. In addition, at this step the engineer also defines threshold values for the chosen NFRs. However, this choice may vary significantly between different types of microservices. In other words, the engineer determines which requirements must be continuously met at run time (i.e. hard constraints), and which requirements are desirable but not mandatory (i.e. soft constraints). Once hard and soft constraints are defined, they are used in the two final stages of the automated decision-making process. Hard constraints are used as input parameters for the equivalence classification (second step), while soft constraints are used at the stage of generation and verification of the probabilistic model (third step).

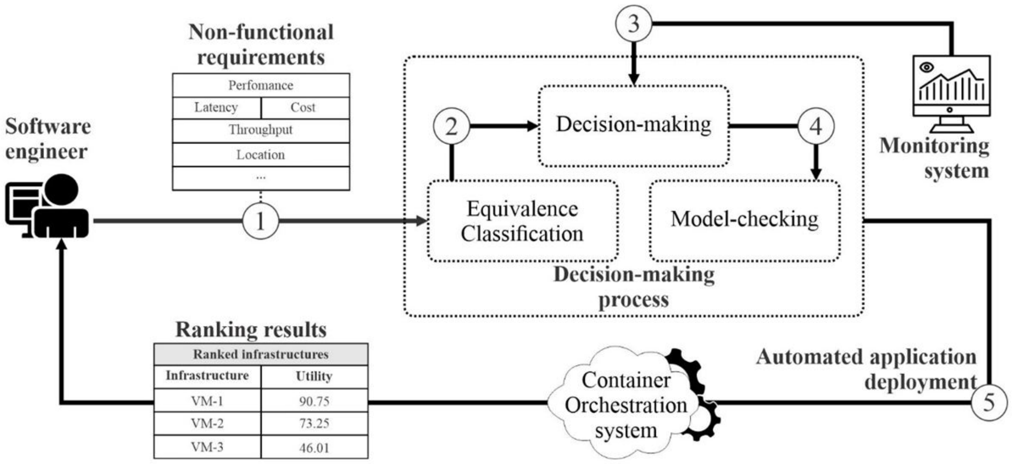


Fig. 1. Approach for automated deployment of cloud applications in the Edge-to-Cloud computing continuum

The second step is responsible for reducing the computational complexity of the method by reducing the number of computations for the decision-making process. As a result, the decision-making process considers only infrastructures that meet the hard constraints. At the beginning of this step, an automaton from all available deployment infrastructures is built. Each state of the automaton represents a deployment infrastructure. Once the automaton is built, an automated process, which classifies infrastructures into classes, is initiated. For instance, an equivalence class can be composed of all available deployment infrastructures that contain at least 8 CPU, or that are located in the territory of Russia.

The goal of the third step is to build a probabilistic model. In order to build the probabilistic model, multiple QoS metrics that represent the past and present performance of the infrastructures are used. These metrics are collected and stored in databases using a multi-tier monitoring system. However, this step only utilizes NFRs that are defined as soft constraints. At the end, the decision-making mechanism calculates the rank scores and sort all of the deployment infrastructures, which were included in this step.

The forth step verifies the results that are obtained from the previous. Using formal criteria and model-checking method, this step verifies the number of NFRs that are satisfied in the equivalence class. The estimated verification value is the output of the probabilistic model and represents a formal guarantee for achieving high QoS.

At the fifth step, the top-ranking infrastructure is automatically selected, where the microservice is going to be deployed using an orchestration tool (e.g. Kubernetes). This step is carried out under the assumption that the formal guarantee obtained for the top-ranking infrastructure, that is, the one with the highest score, is acceptable to the software developer.

Implementation

Applications in smart environments constantly generate and utilize large amounts of different formats and sizes of unstructured data. As a result, traditional cloud computing infrastructures cannot achieve the desired QoS. However, implementing a datacentric architecture, which offers moving the computing in close proximity to data sources, can be a solution to this problem.

Fig. 2 depicts a multi-tier architecture that was developed throughout the period of this research. It complies with the interoperability standards set by organizations such as: Cloud Native Computing Foundation (CNCF), Edge Computing Consortium Europe (ECCE), and OpenFog Consortium. The proposed design is composed of three tiers: Graphical User Interface, decision-making tier and computing tier with available Edge-to-Cloud infrastructures for the deployment of containerized microservices.

The Graphical User Interface (GUI) is an entry point for the software engineer, which is used to compose an application from containerized microservices, manage QoS requirements and input parameters for the deployment process, and initiate the deployment process. The GUI is implemented using EmberJS framework, which offers several views (e.g. component creation view and application composition view).

The decision-making tier is responsible for estimating an optimal deployment infrastructure based on MDP. In addition, this module also verifies the deployment decision and analyzes possible scenarios of the redistribution of microservices from one infrastructure to another at some point in time in the future. This tier also incorporates a container-orchestration system for automated application deployment, which initiates the deployment process after MDP estimates and verifies optimal deployment infrastructure.

The Edge-to-Cloud computing tier is composed of IoT devices, monitoring components and infrastructures that are used for deployment of containerized microservices and data. The infrastructures in this tier are used to store and process data in the computational continuum. Depending on the purpose and requirements of the deployed application, the proposed architecture allows to deploy application's containers in close proximity to data resources (i.e. Edge), the Fog or the Cloud infrastructures.

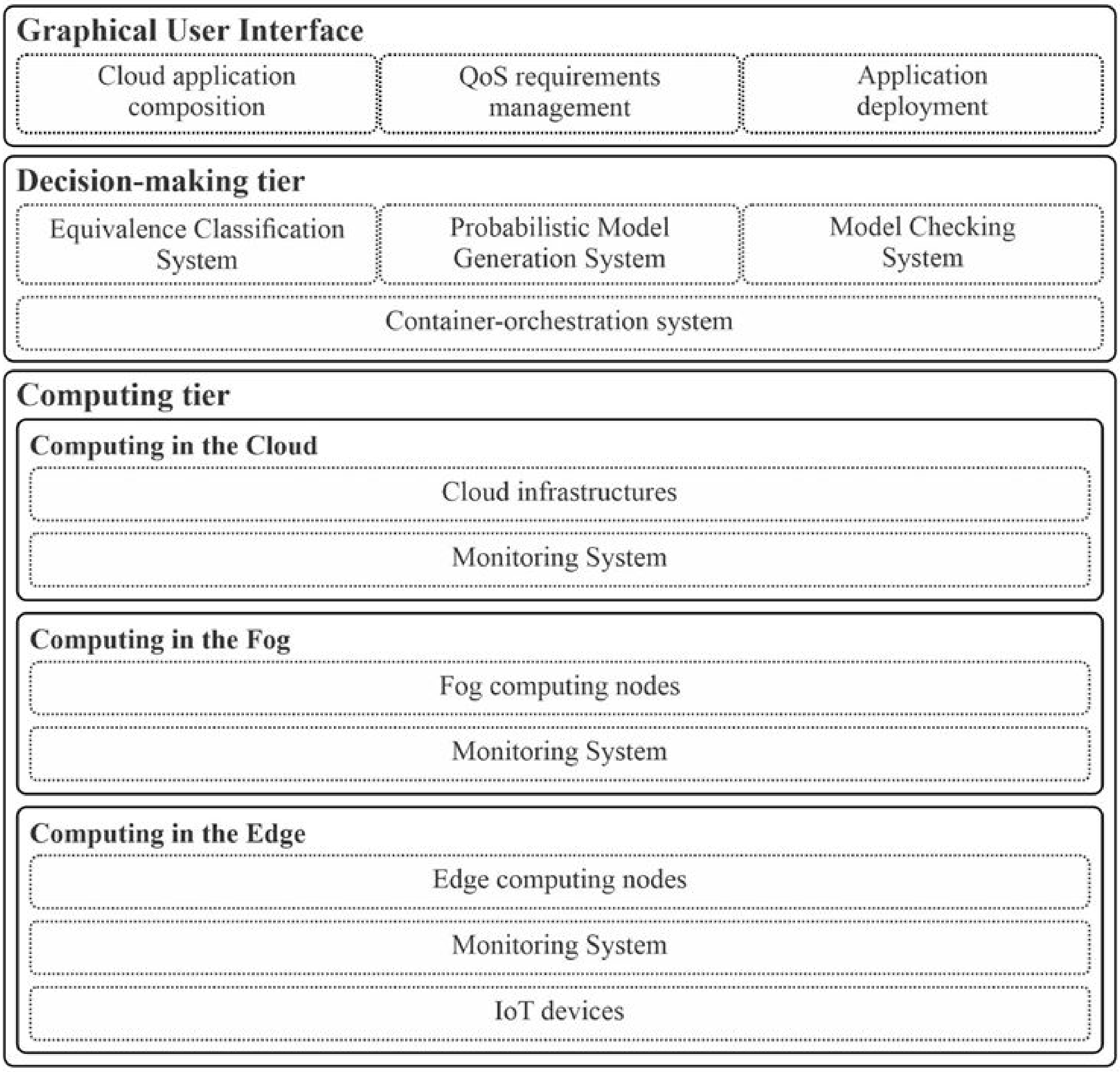


Fig. 2. High level system architecture

Deployment decision-making mechanism

The deployment decision-making mechanism is composed of two systems: monitoring system and MDP-based system for decision-making and verification.

The MDP-based system for decision-making and verification is composed of three subsystems: Equivalence Classification Subsystem (ECS), Probabilistic Model Generation Subsystem (PMGS) and Model Checking Subsystem (MCS). ECS initiates the decision-making process. First it retrieves all available infrastructures and generates an initial model, which is used as an input parameter for equivalence classification. Then, ECS discovers the infrastructures that satisfy the hard constraints and assigns them to the equivalence class. The composed equivalence class as an output from ECS is forwarded to PMGS. PMGS performs decision-making based on MDP method. First, PMGS generates a finite probabilistic model, where each state in the model is a different member of the equivalence class. Then, PMGS calculates the transition probability values, state rewards and state utility. Both subsystems, ECS and PMGS are developed using Java-based development technologies, such as: Java Jersey for RESTful web services and Apache Maven for software management.

MCS performs model-checking over the model and the output from PMGS based on probabilistic computation tree logic (PCTL). In other words, MCS checks the extent to which the selected optimal infrastructure will satisfy application's NFRs regarding the specific restrictions that were set by the software engineer. Therefore, MCS assures the engineer that the mechanism provides the optimal infrastructure for the application. This subsystem integrates the PRISM model-checker into the mechanism, which is used to analyze probabilistic models. To execute the model-checking, PRISM imports models from configuration scripts.

When the deployment infrastructure is selected and verified, the deployment decision-making mechanism generates a YAML script with deployment instructions for the container-orchestration system. The instructions described in the YAML script provide information on the deployment infrastructure, applications for deployment, backup and replication policies.

The proper work of the MDP-based system for decision-making and verification strongly depends on the monitoring system. The monitoring system is a set of monitoring components, such as: monitoring probes, monitoring agents, monitoring server, databases and knowledge bases. It plays an important role in the proposed mechanism, because it is used to collect input data for the decision-making process (e.g. throughput, latency, CPU and memory utilization) and ensures that any application satisfies the QoS requirements at runtime. Usually, the monitoring system begins to work once an infrastructure becomes available to the system. The monitoring system was implemented by using Jcatascopia, NetData and Prometheus monitoring systems.

Monitoring Agents are lightweight components that control the collection of metrics from virtual machine and container instances. Monitoring Probes are metric collectors managed by the Monitoring Agents. They are designed to collect low-level and high-level metrics. Monitoring Probes send metrics to the appropriate Monitoring Agent either periodically or when a specific event occurs. The Monitoring Server is used to collect the metrics from the Monitoring Agent and forward them to a database. The Monitoring Server must be installed on a host that meets the database hardware requirements. In other words, the host must provide enough memory, processor and disk resources.

To store the metrics from the monitoring system, it is necessary to implement a time series database (TSDB). For the purposes of this study, we integrated Apache Cassandra, which is an open source TSDB.

A Knowledge Base (KB) is used to collect complex information, which is required by ECS and PMGS as input parameters. The KB that was implemented was Apache Jena Fuseki, which collects information about the selected infrastructure, such as: location of the infrastructure, information about the type of application, assessment of the quality of experience and information about the deployment, in the form of RDF semantic triples. Utilizing such a KB allows to perform analysis of long-term trends or conduct a variety of strategic analyzes, such as, utilization trends.

Experimental evaluation

The approach described in this study was experimentally evaluated with a typical cloud scenario for uploading and storing files in the Cloud. The file storage application is designed to be deployed on an infrastructure, used and terminated each time the user needs to use it. The File Storage application is a Java servlet web application that processes requests to upload files to a server. The approach in this work allows one to choose the thresholds of different NFRs for each upload operation, since a container instance can be initiated in a different cloud infrastructure for each file upload operation.

For this experimental evaluation, the application was implemented as a Docker container, which is an advanced technology for application virtualization. Furthermore, the following NFRs were used: infrastructure location (Europe), latency (less than 100 ms), throughput (more than 4 Gbit/s), packet loss (less than 2 %) and quality of experience (more than 4). In this evaluation, infrastructure location was used as a hard constraint, whilst other attributes were used as soft constraints. Also, it was assumed that there are several deployment scenarios. After the initial deployment of the application, the software engineer had two different workload requirements, such as 1000 (deployment 2), 1500 (deployment 3) requests every five seconds. The experimental workload was created using the *httperf* tool. The software engineer was able to deploy the application in one of 42 available Fog-Cloud infrastructures or in one of 2 Edge infrastructures. Edge infrastructures were hosted near to the data sources (i.e. the application user), and Fog-Cloud infrastructures were hosted on the Google Cloud Platform, Amazon AWS EC2 and ARNES in 6 different locations: Ljubljana, Frankfurt, London, Tokyo, Sydney and Oregon. The experimental evaluation results are enlisted in Table.

Experimental evaluation deployment results

Infrastructure	Deployment 1		Deployment 2		Deployment 3	
	Rank	Utility	Rank	Utility	Rank	Utility
RaspberryPi 3	11	0.0	11	0.0	12	0.74010
RaspberryPi 4	5	1.02093	4	1.02093	4	1.02093
arnes	12	0.0	12	0.0	11	0.80221
g1-small	10	0.93593	8	0.93593	7	0.93593
n1-standard-1	7	1.01909	6	1.01909	6	1.01909
n1-standard-2	**1**	**1.12443**	10	0.86495	3	1.02220
n1-standard-4	6	1.01965	5	1.01965	5	1.01965
n1-standard-8	4	1.03024	3	1.03023	2	1.03024
a1.medium	8	1.01688	9	0.93204	8	0.93204
a1.large	3	1.11038	2	1.11038	**1**	**1.11038**
a1.xlarge	2	1.11571	**1**	**1.11571**	10	0.85815
a1.2xlarge	9	1.01473	7	1.01473	9	0.85853

According to the results of the experimental evaluation, the infrastructure n1-standard-2 offered the highest utility value for deploying the application. However, after an additional workload was applied to this infrastructure (1000 requests every five seconds), several quality thresholds were violated, so the application was transferred to the a1.xlarge infrastructure. Following the same steps, with a workload of 1,500 requests, the application was again redeployed to another infrastructure (i.e. a1.large), which guaranteed high QoS.

Conclusion

The goal of this work was to design a QoS-aware approach that guarantees high QoS for cloud applications in highly dynamic and heterogeneous Edge-to-Cloud environments. QoS are relevant requirements that software engineers at the deployment stage of cloud applications have to comply with. This study offers an approach that can be used for this purpose and which can be integrated into software development tools.

The results of the experimental evaluation elaborate that the proposed approach is universal enough for the automated deployment of applications with different QoS requirements in various infrastructures. This means that the proposed approach is not limited to a specific set of NFRs or types of applications.

The approach can be further expanded by improving deployment algorithms to implement multi-tier application deployment operations across multiple infrastructures, where each application tier is deployed in a different Edge-to-Cloud infrastructure.

REFERENCES

1. **Hu J., Gu J., Sun G., Zhao T.** A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. *Proceedings of the 3rd International Symposium on Parallel Architectures, Algorithms and Programmin*g, 2010, Pp. 89–96.

2. **Li L.E., Woo T.** Dynamic load balancing and scaling of allocated cloud resources in an enterprise network, *US Patent App. 12/571,271*, Mart 31, 2011.

3. **Randles M., Lamb D., Taleb-Bendiab A.** A comparative study into distributed load balancing algorithms for cloud computing. *Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, 2010, Pp. 551–556.

4. **Chaczko Z., Mahadevan V., Aslanzadeh S., Mcdermid C.** Availability and load balancing in cloud computing. *Proceedings of the International Conference on Computer and Software Modeling.* Singapur, 2011, Vol. 14.

5. **Manvi S., Shyam G.K**. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications*, 2014, Vol. 41, Pp. 424–440.

6. **Jennings B., Stadler R.** Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, 2015, Vol. 23, No. 3, Pp. 567–619.

7. **Luong N.C., Wang P., Niyato D., Wen Y., Han Z.** Resource management in cloud networking using economic analysis and pricing models: A survey. *IEEE Communications Surveys & Tutorials*, 2017, Vol. 19, No. 2, Pp. 954–1001.

8. **Jain N., Menache I**. Resource management for cloud computing platforms, *US Patent 9,450,838*, Sept. 20, 2016.

9. **Singh S., Chana I.** Q-aware: Quality of service based cloud resource provisioning. *Computers & Electrical Engineering*, 2015, Vol. 47, Pp. 138–160.

10. **Chaisiri S., Lee B.S., Niyato D.** Optimization of resource provisioning cost in cloud computing. *IEEE Transactions on Services Computing*, 2012, Vol. 5, No. 2. Pp. 164–177.

11. **Zhang L., Li Z., Wu C.** Dynamic resource provisioning in cloud computing: A randomized auction approach. *Proceedings of the IEEE Conference on Computer Communications*, 2014, Pp. 433–441.

12. **Paščinski U., Trnkoczy J., Stankovski V., Cigale M., Gec S.** QoS-aware orchestration of network intensive software utilities within software defined data centres. *Journal of Grid Computing*, 2018, Vol. 16, No. 1, Pp. 85–112.

13. **Mijumbi R., Serrat J., Gorricho J.L., Latre S., Charalambides M., Lopez D.** Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*, 2016, Vol. 54, No. 1, Pp. 98–105.

14. **Karim R., Ding C., Miri A.** An end-to-end QoS mapping approach for cloud service selection. *Proceedings of the IEEE 9th World Congress on Services*, 2013, Pp. 341–348.

15. **Garg S.K., Versteeg S., Buyya R.** A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 2013, Vol. 29, No. 4, Pp. 1012–1023.

16. **Goncalves Junior R., Rolim T., Sampaio A., Mendonca N.C.** A multi-criteria approach for assessing cloud deployment options based on non-functional requirements. *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015, Pp. 1383–1389.

17. **Zheng Z., Wu X., Zhang Y., Lyu M.R., Wang J.** QoS ranking prediction for cloud services. *IEEE Transactions on Parallel and Distributed Systems*, 2012, Vol. 24, No. 6, Pp. 1213–1222.

18. **Guerrero C., Lera I., Juiz C.** Genetic algorithm for multi-objective optimization of container allocation in cloud architecture. *Journal of Grid Computing*, 2018, Vol. 16, No. 1, Pp. 113–135.

19. **Štefanič P., Kimovski D., Siciu G., Stankovski V.** Non-functional requirements optimisation for multi-tier cloud applications: An early warning system case study. *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, 2017, Pp. 1–8.

20. **Guo X., Wang Y., Wang X.** Using objective clustering for solving many-objective optimization problems. *Mathematical Problems in Engineering*, 2013, Vol. 2013.

21. **Trenz M., Huntgeburth J., Veit D.** The role of uncertainty in cloud computing continuance: Antecedents, mitigators, and consequences. *ECIS*, 2013, P. 147.

22. **Tchernykh A., Schwiegelsohn U., Alexandrov V., Talbi E.G.** Towards understanding uncertainty in cloud computing resource provisioning. *Procedia Computer Science*, 2015, Vol. 51, Pp. 1772–1781.

23. **Yang J., Lin W., Dou W.** An adaptive service selection method for cross-cloud service composition. *Concurrency and Computation: Practice and Experience*, 2013, Vol. 25, No. 18, Pp. 2435–2454.

24. **Su G., Chen T., Feng Y., Rosenblum D., Thiagarajan P.** An iterative decision-making scheme for Markov decision processes and its application to self-adaptive systems. *Proceedings of the International Conference on Fundamental Approaches to Software Engineering*, 2016, Pp. 269–286.

25. **Llerena Y.R.S., Su G., Rosenblum D.S.** Probabilistic model checking of perturbed mdps with applications to cloud computing. *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 2017, Pp. 454–464.

26. **Mardani A., Jusoh A., Nor K., Khalifah Z., Zakwan N., Valipour A.** Multiple criteria decision-making techniques and their applications - A review of the literature from 2000 to 2014. *Economic Research*, 2015, Vol. 28, No. 1, Pp. 516–571.

27. **Tsoumakos D., Konstantinou I., Boumpouka C., Sioutas S., Koziris N.** Automated, elastic resource provisioning for nosql clusters using tiramola. *Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 2013, Pp. 34–41.

28. **Naskos A., Stachtiari E., Gounaris A., Katsaros P., Tsoumakos D., Konstantinou I., Sioutas S.** Dependable horizontal scaling based on probabilistic model checking. *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2015, Pp. 31–40.

Received 29.01.2020.

СПИСОК ЛИТЕРАТУРЫ

1. **Hu J., Gu J., Sun G., Zhao T.** A scheduling strategy on load balancing of virtual machine resources in cloud computing environment // Proc. of the 3rd Internat. Symp. on Parallel Architectures, Algorithms and Programming. 2010. Pp. 89–96.

2. **Li L.E., Woo T.** Dynamic load balancing and scaling of allocated cloud resources in an enterprise network // US Patent App. 12/571,271, Mart 31, 2011.

3. **Randles M., Lamb D., Taleb-Bendiab A.** A comparative study into distributed load balancing algorithms for cloud computing // Proc. of the IEEE 24th Internat. Conf. on Advanced Information Networking and Applications Workshops. 2010. Pp. 551–556.

4. **Chaczko Z., Mahadevan V., Aslanzadeh S., Mcdermid C.** Availability and load balancing in cloud computing // Proc. of the Internat. Conf. on Computer and Software Modeling. Singapur, 2011. Vol. 14.

5. **Manvi S., Shyam G.K.** Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey // J. of Network and Computer Applications. 2014. Vol. 41. Pp. 424–440.

6. **Jennings B., Stadler R.** Resource management in clouds: Survey and research challenges // J. of Network and Systems Management. 2015. Vol. 23. No. 3. Pp. 567–619.

7. **Luong N.C., Wang P., Niyato D., Wen Y., Han Z.** Resource management in cloud networking using economic analysis and pricing models: A survey // IEEE Communications Surveys & Tutorials. 2017. Vol. 19. No. 2. Pp. 954–1001.

8. **Jain N., Menache I.** Resource management for cloud computing platforms // US Patent 9,450,838. Sept. 20, 2016.

9. **Singh S., Chana I.** Q-aware: Quality of service based cloud resource provisioning // Computers & Electrical Engineering. 2015. Vol. 47. Pp. 138–160.

10. **Chaisiri S., Lee B.S., Niyato D.** Optimization of resource provisioning cost in cloud computing // IEEE Transactions on Services Computing. 2012. Vol. 5. No. 2. Pp. 164–177.

11. **Zhang L., Li Z., Wu C.** Dynamic resource provisioning in cloud computing: A randomized auction approach // Proc. of the IEEE Conf. on Computer Communications. 2014. Pp. 433–441.

12. **Paščinski U., Trnkoczy J., Stankovski V., Cigale M., Gec S.** QoS-aware orchestration of network intensive software utilities within software defined data centres // J. of Grid Computing. 2018. Vol. 16. No. 1. Pp. 85–112.

13. **Mijumbi R., Serrat J., Gorricho J.L., Latre S., Charalambides M., Lopez D.** Management and orchestration challenges in network functions virtualization // IEEE Communications Magazine. 2016. Vol. 54. No. 1. Pp. 98–105.

14. **Karim R., Ding C., Miri A.** An end-to-end QoS mapping approach for cloud service selection // Proc. of the IEEE 9th World Congress on Services. 2013. Pp. 341–348.

15. **Garg S.K., Versteeg S., Buyya R.** A framework for ranking of cloud computing services // Future Generation Computer Systems. 2013. Vol. 29. No. 4. Pp. 1012–1023.

16. **Goncalves Junior R., Rolim T., Sampaio A., Mendonca N.C.** A multi-criteria approach for assessing cloud deployment options based on non-functional requirements // Proc. of the 30th Annual ACM Symp. on Applied Computing. 2015. Pp. 1383–1389.

17. **Zheng Z., Wu X., Zhang Y., Lyu M.R., Wang J.** QoS ranking prediction for cloud services // IEEE transactions on parallel and distributed systems. 2012. Vol. 24. No. 6. Pp. 1213–1222.

18. **Guerrero C., Lera I., Juiz C.** Genetic algorithm for multi-objective optimization of container allocation in cloud architecture // J. of Grid Computing. 2018. Vol. 16. No. 1. Pp. 113–135.

19. **Štefanič P., Kimovski D., Siciu G., Stankovski V.** Non-functional requirements optimisation for multi-tier cloud applications: An early warning system case study // 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation. 2017. Pp. 1–8.

20. **Guo X., Wang Y., Wang X.** Using objective clustering for solving many-objective optimization problems // Mathematical Problems in Engineering. 2013. Vol. 2013.

21. **Trenz M., Huntgeburth J., Veit D.** The role of uncertainty in cloud computing continuance: Antecedents, mitigators, and consequences // ECIS. 2013. P. 147.

22. **Tchernykh A., Schwiegelsohn U., Alexandrov V., Talbi E.G.** Towards understanding uncertainty in cloud computing resource provisioning // Procedia Computer Science. 2015. Vol. 51. Pp. 1772–1781.

23. **Yang J., Lin W., Dou W.** An adaptive service selection method for cross-cloud service composition // Concurrency and Computation: Practice and Experience. 2013. Vol. 25. No. 18. Pp. 2435–2454.

24. **Su G., Chen T., Feng Y., Rosenblum D., Thiagarajan P.** An iterative decision-making scheme for Markov decision processes and its application to self-adaptive systems // Proc. of the Internat. Conf. on Fundamental Approaches to Software Engineering. 2016. Pp. 269–286.

25. **Llerena Y.R.S., Su G., Rosenblum D.S.** Probabilistic model checking of perturbed mdps with applications to cloud computing // Proc. of the 2017 11th Joint Meeting on Foundations of Software Engineering. 2017. Pp. 454–464.

26. **Mardani A., Jusoh A., Nor K., Khalifah Z., Zakwan N., Valipour A.** Multiple criteria decision-making techniques and their applications - A review of the literature from 2000 to 2014 // Economic Research. 2015. Vol. 28. No. 1. Pp. 516–571.

27. **Tsoumakos D., Konstantinou I., Boumpouka C., Sioutas S., Koziris N.** Automated, elastic resource provisioning for nosql clusters using tiramola // Proc. of the 13th IEEE/ACM Internat. Symp. on Cluster, Cloud, and Grid Computing. 2013. Pp. 34–41.

28. **Naskos A., Stachtiari E., Gounaris A., Katsaros P., Tsoumakos D., Konstantinou I., Sioutas S.** Dependable horizontal scaling based on probabilistic model checking // Proc. of the 15th IEEE/ACM Internat. Symp. on Cluster, Cloud and Grid Computing. 2015. Pp. 31–40.

Статья поступила в редакцию 29.01.2020.

THE AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

KOCHOVSKI Petar
КОЧОВСКИ Петар
E-mail: petako_bt@hotmail.com

DROBINTSEV Pavel D.
ДРОБИНЦЕВ Павел Дмитриевич
E-mail: drob@ics2.ecd.spbstu.ru

DOI: 10.18721/JCSTCS.13102
УДК 004.896

APPLICATION OF THE COMPUTER VISION SYSTEM FOR CONTROLLING A MOBILE ROBOT IN A DYNAMIC ENVIRONMENT

F. Daeef

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

The article proposes a new approach to the use of computer vision when controlling a robot in a dynamic environment. While moving along an unchanged path to the target point, a robot can encounter any new object (static or moving). We describe a visual analysis to determine the detection distance of moving objects to prevent collisions with the robot in a timely manner. An obstacle detection algorithm in the robot zone was developed based on data from an RGB-D video camera using computer vision methods. Based on ROS in a Gazebo virtual environment with a Turtlebot robot kit and an open source library (opencv), we adopted software implementation of the developed approaches which confirmed their applicability to the detection of objects in the mobile robot environment.

Keywords: mobile robots, dynamic environment, navigation, vision system, object detection.

Citation: Daeef F. Application of the computer vision system for controlling a mobile robot in a dynamic environment. Computing, Telecommunications and Control, 2020, Vol. 13, No. 1, Pp. 19-30. DOI: 10.18721/JCSTCS.13102

ПРИМЕНЕНИЕ СИСТЕМЫ ТЕХНИЧЕСКОГО ЗРЕНИЯ ПРИ УПРАВЛЕНИИ МОБИЛЬНЫМ РОБОТОМ В ДИНАМИЧЕСКОЙ СРЕДЕ

Ф. Даееф

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Предложен новый подход к применению системы технического зрения при управлении мобильным роботом в динамической среде. Во время движения по неизменной траектории к целевой точке на пути робота может появиться любой новый объект (статический или подвижный). В статье описан визуальный анализ для определения расстояния, на котором должно происходить обнаружение объектов, для своевременного предотвращения столкновения с роботом. Алгоритм обнаружения препятствий в зоне робота разработан на основе визуальных данных от RGB-D видеокамеры с помощью методов компьютерного зрения. На основе фреймворка ROS в виртуальной среде Gazebo, а также при помощи комплекта Turtlebot и библиотеки с открытым исходным кодом (opencv) написана программная реализация разработанных подходов, подтвердившая их применимость к обнаружению объектов в среде мобильного робота.

Ключевые слова: мобильный робот, динамическая среда, навигация, система технического зрения, обнаружение объектов.

Ссылка при цитировании: Даееф Ф. Применение системы технического зрения при управлении мобильным роботом в динамической среде // Информатика, телекоммуникации и управление. 2020. Т. 13. №1. С. 19-30. DOI: 10.18721/JCSTCS.13102

Introduction

Moving object detection in video streams is an interesting problem with a very wide range of vision-based application fields, such as action recognition [1], traffic control [2], industrial control [3], identification of human behavior [4] and intelligent video surveillance [5]. The general idea of detecting moving objects is to represent a set of related image pixels in a video sequence having a coherent motion in time (time aspect) and semantic similarity in the image space (spatial aspect) [6].

In this article we try to solve this problem for the purpose of robot navigation. Global path planning algorithm usually uses priori information to build a complete model of a structured environment, and then tries to find the best possible solution. But information is scarce in unknown or unstructured environments, so users need to combine the route planning method with local or reactive navigation using built-in sensors to locally observe small fragments of the environment at any time. A problem of detecting moving objects and responding to obstacles arises in this scenario. The most common approaches are: firstly, a proximity sensor belt (ultrasonic, infrared, ...) mounted on the vehicle, allowing for discrete scanning of the space around the robot; secondly, a rotating laser beam, often associated with a viewing system, which leads to a continuous assessment of the free area around the vehicle. The issue of accounting for moving obstacles in the management of a mobile robot still requires further research. We have a look at different ways to solve the problems presented in a number of studies. In the papers [7, 8] where laser scanners are used to detect obstacles the approaches described have limitations on the type of object's movement and the environment in which the robot can move; laser scanners are also rather expensive. In other studies, robot navigation is described using an on-board camera and video data. The purpose of detecting a moving object is to take a video sequence from a fixed / moving camera and output a binary mask representing moving objects for each frame of the sequence. However, this is not an easy task due to many problems and difficulties that arise when using a camera to capture a video sequence of moving objects. We categorize the existing methods to solve the problem in Table 1.

Table 1

Moving object detection from moving camera

Category	Positive points	Negative points
Background modeling [9-11]	• Moderately complex • Good for real time applications • Providing good object's silhouette	• Not good for free camera motion • Accuracy is highly dependent on background model
Trajectory classification [12, 13]	• Providing good object's trajectory over time • Moderately complex	• Very sensible to noise • Does not provide information on object's silhouette • Accuracy is highly dependent on motion tracker mode
Object tracking [14-16]	• Good performance with all camera motion • Moderately complex	• Does not provide information on object's silhouette • Needs initial good selection of the object
Low rank and sparse representation [17-19]	• Highly accurate • Providing good object's silhouette	• Requires a collection of frames • Not suitable for real-time applications • Highly complex

In this work, we try to solve these problems for navigation of a mobile robot in a dynamic environment on an unchanged path using the on-board vision system (an RGB-D Camera).

Problem statement

Let's have a transport robot on a flat underlying surface. There are static obstacles (walls, columns, tables, chairs, etc.), as well as moving obstacles (people, other robots), which are physical bodies that conform to the laws of dynamics. Let us also have a planned path for the robot, which cannot be changed during movement, in order to transfer material to the target point in the coordinates of the room. Such tasks arise during automation of transport operations in warehouses in which there are mobile objects –robots, people, etc. As a result, the system must position the robot at the target point, moving along the planned route and avoiding moving obstacles using high-speed adaptation method. It uses the computer vision system for this purpose while minimizing the movement time. Such transport robots, as a rule, can move at the maximum speed V_{rmax} = 1÷1.5 m/s and acceleration A_{rmax} = 0.1÷0.7 m/s^2. In our task we consider that the maximum speed and maximum acceleration of the robot V_{rmax} = 1.5 m/s, A_{rmax} = 0.3 m/s^2.

During the task of transportation, the robot should not deviate from the route, and has to avoid collisions with objects (Fig. 1), which can appear on the way. For this, we need to detect all objects (moving and static) on the way by using a vision system. In our task, we have no condition on the movement direction of objects, so objects can move in any direction. At the same time, they are solids with an unchanged shape, the maximum value of speed is V_{omax} = 1.5 m/s.

First of all, we have to calculate the distance for detecting the object, on which we have to slow down, because this is one of the most important factors for evaluating our work.

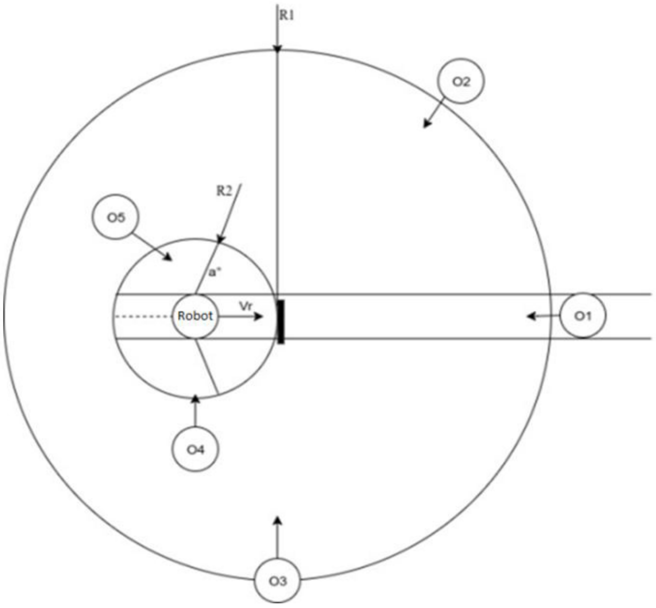


Fig. 1. Example of an environment in which a mobile robot moves, containing moving obstacle objects O1–O5, here R1 = S_n (the distance at which a moving obstacle is to be detected), R2 = S_r (distance to the robot full braking)

Suppose we have robot acceleration A_r, speed V_r, then the braking distance S_r is determined by the formula:

$$V_r = \sqrt{2*(-A_r)*S_r} \tag{1}$$

$$S_r = \frac{V_r^2}{2*(-A_r)}. \tag{2}$$

In our study the maximum velocity of the robot is V_{rmax} = 1.5 m/s, the maximum acceleration is $A_r = \pm$ 0.3 m/s^2, by the formula we get S_r = 3.75 m. In this paper, we consider a dynamic environment with moving objects. So, in our case, the formula for determining the distance S_n, for detecting obstacles moving at a constant speed, should be:

$$S_n = S_r + V_o * T_t \, ,$$

where V_o is the maximum speed of the moving object; T_t – time to stop, which can be calculated:

$$T_t = \frac{V_r}{-A_r} \, .$$

At the maximum acceleration and speed of the robot, the time to stop T_t = 1,5 / 0,3 = 5 s. Thus, the detection of objects should be at a distance not less than S_n = 3,75 + 1,5˙5 = 11,25 m, in case the object moves towards the robot.

Methodology development

Suppose we have a point with coordinates [X_w, Y_w, Z_w] in the world coordinate system. We want to translate these coordinates into a robot camera coordinate system [X_c ,Y_c , Z_c] and into image coordinate system [X_{img} ,Y_{img}]. To do this we have to build matrix P, to convert the world 3D coordinates to 2D image coordinates. Since we are using a digital camera, we will use a "pinhole" – a camera with a small hole instead of a lens or with a lens that simulates this effect. So we can describe the projection matrix in this way:

$$P = K * [\, R|t \,] \, , \tag{3}$$

where K is the internal matrix of transformation of three-dimensional coordinates of the camera into two-dimensional coordinates of the image (Fig. 2); [$R|t$] – the extrinsic matrix transformation of the camera which describes the camera in the world coordinate system, and the direction of its view.

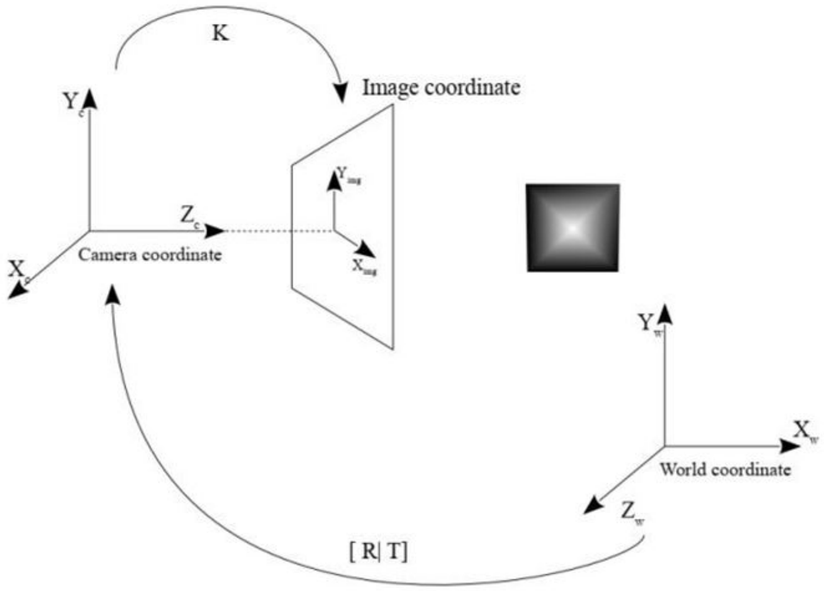


Fig. 2. Conversion from world coordinates to image coordinates

$$\begin{bmatrix} X_{img} \\ Y_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{F}{Z_c} & 0 & {}^{w}\!/_{2} \\ 0 & \frac{F}{Z_c} & {}^{h}\!/_{2} \\ 1 \; 0 & & 0 \end{bmatrix} * \begin{bmatrix} X_c \\ Y_c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{F}{Z_c}(X_c) + {}^{w}\!/_{2} \\ \frac{F}{Z_c}(Y_c) + {}^{h}\!/_{2} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{F}{X_w + Z^`}\left( Z_w + X^` \right) + {}^{w}\!/_{2} \\ \frac{F}{X_w + Z^`}\left( Y_w + Y^` \right) + {}^{h}\!/_{2} \\ 1 \end{bmatrix}. \quad (4)$$

Thus, we have a projection from the world coordinate system to the camera coordinate system; therefore, we can accept that $Z_c = X_w + Z`$ as the sum $X_w$ , $Z`$ also for X_c , Y_c.

Since we assume we know the movement characteristics of the robot (position, speed and acceleration), so we will use the camera coordinate system [X_c , Y_c , Z_c].

Analysis of the object movement in the camera coordinates

We accept that objects move on a flat surface; therefore, we consider the movement of the objects in the XY plane in the robot coordinate system. We will try to describe how the real move of the objects in the robot coordinates system reflected on the image coordinates.

Depth axis movement (object speed on Z axis). To find how the movements of objects along the Z axis (depth axis) can be reflected in the image from the robot camera, we consider the relationship between the change in length of the line segment from object and the change in depth of the object. Let's consider two points on the image with coordinates (X_{1img} , Y_{1img}) (X_{2img} , Y_{2img}), the distance between these two points L_t :

$$\left(L_t\right)^2 = \left(X_{1timg} - X_{2timg}\right)^2 + \left(Y_{1timg} - Y_{2timg}\right)^2.$$

Using equation (4) we can rewrite as:

$$\left(L_t\right)^2 = \left(\frac{F * X_{1tc}}{Z_{1tc}} - \frac{F * X_{2tc}}{Z_{2tc}}\right)^2 + \left(\frac{F * Y_{1tc}}{Z_{1tc}} - \frac{F * Y_{2tc}}{Z_{2tc}}\right)^2.$$

Suppose that we observe two points belonging to one object, which does not change its shape or rotate around any of its axes, in other words the object is either static or moving in a straight line, so that $Z_{1c} = Z_{2c}$ in the camera coordinate system. Then we can rewrite:

$$\left(L_t\right)^2 = \left(\frac{F * \left(X_{1tc} - X_{2tc}\right)}{Z_{tc}}\right)^2 + \left(\frac{F * \left(Y_{1tc} - Y_{2tc}\right)}{Z_{tc}}\right)^2 = \frac{\left(F * Lx\right)^2 + \left(F * Ly\right)^2}{{Z_{tc}}^2}. \quad (5)$$

Now, let's look at the change of the length of this line segment between two frames L_t , L_{t+1}

$$\frac{\left(L_t\right)^2}{\left(L_{t+1}\right)^2} = \frac{\dfrac{\left(F * Lx\right)^2 + \left(F * Ly\right)^2}{{Z_{tc}}^2}}{\dfrac{\left(F * Lx\right)^2 + \left(F * Ly\right)^2}{{Z_{t+c}}^2}} = \frac{{Z_{t+c}}^2}{{Z_{tc}}^2}. \quad (6)$$

From this equation (6) we notice that $(Z)^` = \frac{-(L)^`}{L^2}$ which means that change in the length of a line segment from one object is inversely related to the change of the depth of the object. We can calculate a change of object coordinate in Z axis (depth information), but it is relative to depth value. Thus, we require information about depth (approximate) to calculate the change. These changes of depth information represent the speed of the object along Z axis. For this purpose, we use pixel depth data from the matrix of a digital RGB-D camera.

Vertical axis movement (object speed on X axis). To demonstrate the way object movement on the vertical axis is displayed on the image in the robot camera, consider one point with image coordinates X_{img}, Y_{img}.

Based on equation (4), we can write X_{img} , X_{img} as follows:

$$X_{img} = \frac{F}{Z_c}(X_c) + {}^{w}\!/_{2} \qquad => \qquad \frac{F}{Z_c}X_c = X_{img} - {}^{w}\!/_{2}.$$

Consider the change in its coordinates between frames:

$$\frac{\frac{F}{Z_{ct}}X_{ct}}{\frac{F}{Z_{ct+1}}X_{ct+1}} = \frac{X_{imgt} - {}^{w}\!/_{2}}{X_{imgt+1} - {}^{w}\!/_{2}},$$

$$\frac{X_{ct}}{X_{ct+1}} = \frac{X_{imgt} - {}^{w}\!/_{2}}{X_{imgt+1} - {}^{w}\!/_{2}} * \frac{Z_{ct}}{Z_{ct+1}}. \qquad (7)$$

From this equation we can conclude that there are two factors affecting the movement of points in the image on axis X_{img} . The first factor is the movement of the point itself in space along *X* axis. The second one is its movement along the depth axis in such a way that both approximation and removal of the image of the controlled point eliminates the effect on the image of its movement along the depth axis.

The proposed approach

The general concept of our proposed method (detection of moving objects based on featured fragments) is shown in Fig. 3. As seen, the first step is the initialization of the featured fragments using the first set of frames. Next, we go in a cycle:

- Find comparisons of the featured fragments on the next frame.
- Then calculate the change in the position and length of each fragment between two frames.
- Update the fragment data.

After every round we send the fragment data to the robot control unit, where changes are analyzed and decisions are made for control of the robot speed. Color and depth frame sequences from an RGB-D camera comprise the inputs of the system. The goal is to provide information about the presence of moving or non-moving objects to the robot controller in order to decide (increase, decrease or not change) the speed of the robot.

Initialize featured fragments. In our task we propose that we don't have to find the whole moving object in the frame but the most interesting part of the object only for our task. We suppose this part is the edges that separate the body of the object from the background or other objects. That is a very widely studied problem called edge detection. To solve this kind of problem we use the Canny edge detector[1] developed in 1986 by John F. Canny. It uses a multi-stage algorithm to detect a wide range of edges in images. As a result of the Canny algorithm, we get a black and white picture where white pixels form the borders or edges that separate the objects. For better precision we have to filter the results of edge detection.

[1] https://en.wikipedia.org/wiki/Canny_edge_detector

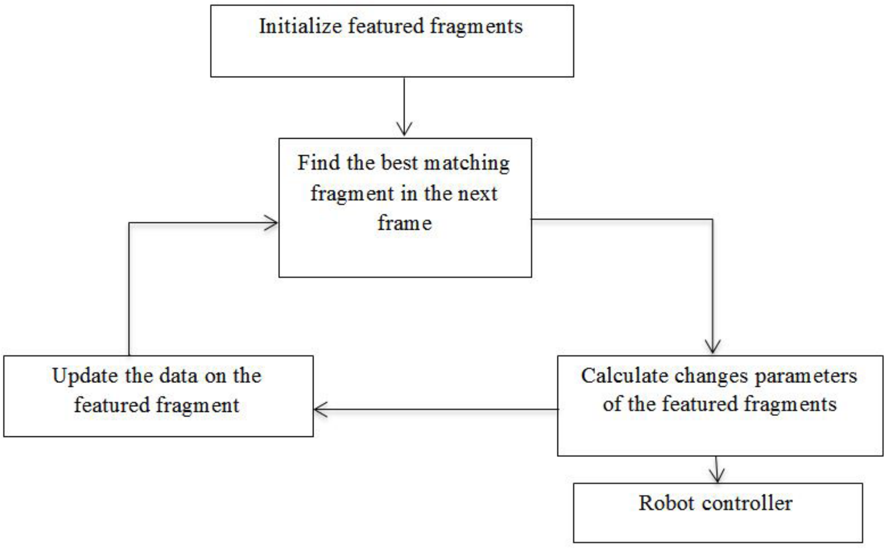

Initialize featured fragments
Find the best matching fragment in the next frame
Update the data on the featured fragment
Calculate changes parameters of the featured fragments
Robot controller

Fig. 3. Scheme of the proposed approach

Filtering featured fragments. The task of filtering is to remove false and unused points and fragments from image results after the edge detection phase. In order to apply such filtering, we need contours which represent curves connecting all the continuous points (along the boundary of an object) of the same color or intensity. Filtering is carried out on two bases:

- The length of the fragment itself (using fragments with length starting from 32 pixels).
- The depth of the point (up to 10 m).

Find the best matching fragment in the next frame. For the next step we propose to convert these fragments of the images into a set of segment lines and represent their by the two points ends segment. We use two methods for this step:

- The Hough transform to find imperfect instances of objects within a certain class of shapes (segment line) by a voting procedure.
- We use two ends of the contours that we found earlier as the two ends of a line segment.

As a result, we get a list of pairs of points we track between frames using an optical flow algorithm, based on the Lucas–Kanade method[1].

Calculating changes parameters of the featured fragments between two frames (determining the speed of objects)

We represent each pair of points (single segment line) as a separated object and try calculating the change in the length and position of this object (segment line) between two frames to find the speeds on vertical axis V_{ox} and depth V_{oz}.

1. Speed V_{oz}

We use an RGB-D camera that gives us depth data. For our task it is very important to measure depth error and how it increases with depth value. Khoshelham [2] studied a model for measuring depth error in the Kinect model camera (RGB-D) and concluded that the uncertainty of depth measurement is proportional to the square of the depth value $\sigma_z = \frac{1}{f}\sigma_d d^2$, where d is depth σ_d standard deviation, f is the focal length. For Intel® RealSense™ Depth Camera D435i errors increase quadratically from a few millimeters at a distance of 0.5 m to ~5 cm at the maximum distance of 10 m.

Therefore, we cannot get depth changes between frames from the depth data of the camera, but we can use the depth data from equation (6) to find the speed of an object:

[1] https://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method

$$\frac{(L_t)^2}{(L_{t+1})^2} = \frac{{Z_{t+c}}^2}{{Z_{tc}}^2}, \tag{8}$$

where L_t is the length of a segment line at frame t and Z_{tc} is the depth of the object from the robot at the frame t.

2. Speed V_{ox}

We can calculate the projection of the velocity of object V_{ox} relative to the coordinate system of the mobile robot by the formula:

$$\Delta s_{yr} = \frac{X_{ct}}{X_{ct+1}} = \frac{X_{imgt} - {}^{w}\!/_{2}}{X_{imgt+1} - {}^{w}\!/_{2}} * \frac{Z_{ct}}{Z_{ct+1}}$$

$$V_{ox} = \Delta s_{yr} / t\,. \tag{9}$$

Testing the developed approaches

To test our method, we performed a precision-recall curve. Using virtual implementation in an environment (ROS), we examined two metrics (precision-recall) for detecting moving objects at the pixel level and at the object level. The results are shown in Fig. 4.

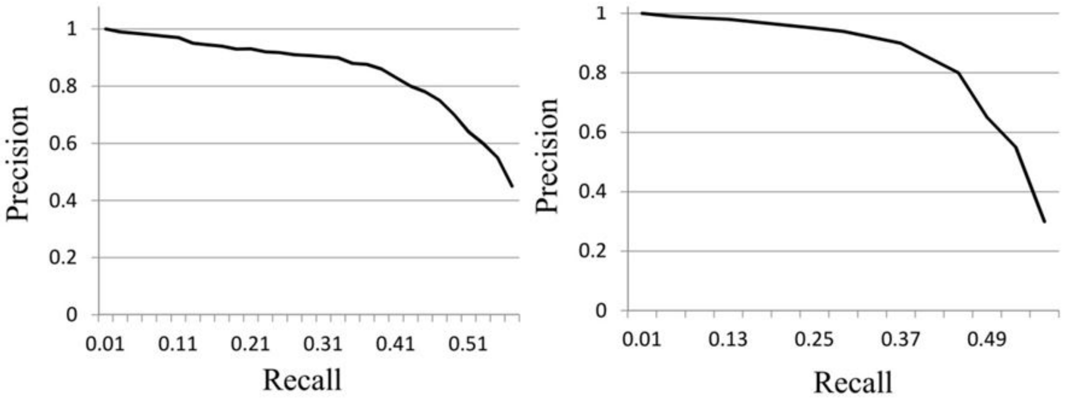


Fig. 4. Quantitative analysis of our approach
Precision-recall curve at the pixel level (left) and at the object level (right)

We also compared our results to a study conducted with the use of lidar [20]. Table 2 shows our result do not deviate much from lidar results, although we had only an RGB-D camera instead of some very expensive pieces of equipment.

Table 2

Comparison of the proposed method to lidar detection

	#Grd TruthObj	#Correct Obj	#Wrong Obj	Precision	Object Rcl	Sec./frame
Lidar [20]	52	48	0	1	0.9231	0.26
Proposed method	50	40	2	0.96	0.8	0.3

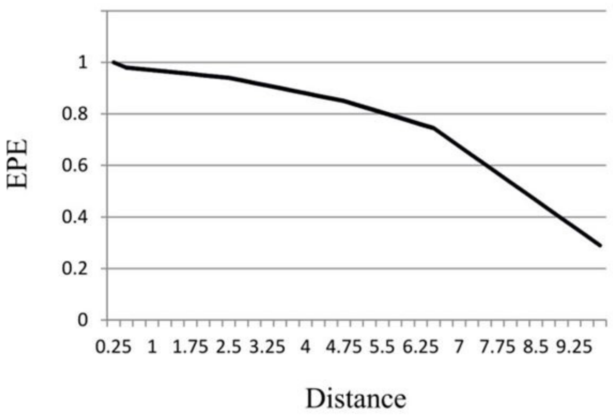


Fig. 5. Endpoint error (EPE) and the distance of the object from the robot

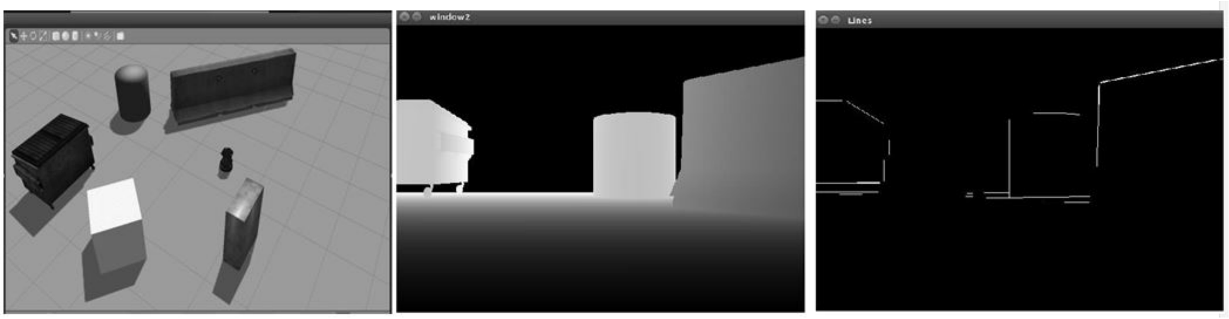


Fig. 6. Example of application based on ROS in Gazebo virtual environment with a Turtlebot robot kit. There are a general view of the scene (left), 3D image (center), and the result of the proposed method (right)

To analyze the tracking method, we consider a metric (EPE) that determines the error of the end-to-end point and how the distance of the object from the robot affects it. The end-point error is calculated by comparing the calculated optical flow vector V_{est} with the true optical flow vector V_{gt} for each point. We calculated this metric for each point in the list that we received in the previous step, in a static scene, because we know how they should move in the image (surface truth). Endpoint error is defined as the Euclidean distance between the two points: $A = | V_{est} - V_{gt} |$.

Using a virtual implementation, we were able to calculate the optical flow error EPE as a function of the distance of the object from the robot. We examined the values of the optical flow of static objects, the coordinates of which we know in the absolute system. We can calculate their coordinates on the images in each frame, then we calculate the error of the optical flow. Note in Fig. 5 that the closer is the object to the robot, the greater the error.

Conclusions

A detailed analysis of the environment of the mobile robot allowed us to solve the problem of calculating the braking distance and the distance of detection of obstacles to prevent collisions with them. The developed software implementation of the proposed approaches uses ROS (Robot Operating System) middleware (Fig. 6), which provides developers with libraries and tools for creating robotic applications. Our study conducted on a mobile robot with a caterpillar chassis confirms the applicability of our approach to analysis of a dynamic environment in real conditions when navigating transport robots in warehouse and workshop premises.

REFERENCES

1. **Wu S., Oreifej O., Shah M.** Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories. *IEEE International Conference on Computer Vision*, Nov. 2011, Pp. 1419–1426.

2. **Cucchiara R., Grana C., Piccardi M., Prati A.** Statistic and knowledge-based moving object detection in traffic scenes. *IEEE Intell. Transp. Syst.*, 2000, Pp. 27–32.

3. **Malamas E.N., Petrakis E.G., Zervakis M., Petit L., Legat J.D.** A survey on industrial vision systems, applications and tools. *Image Vis. Comput.*, 2003, Vol. 21 (2), Pp. 171–188.

4. **Hu W., Tan T., Wang L., Maybank S.** A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Syst Man Cybern.*, 2004, Vol. 34 (3), Pp. 334–352.

5. **Kim J.S., Yeom D.H., Joo Y.H.** Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems. *IEEE Trans. Consum. Electron.*, 2011, Vol. 57 (3), Pp. 1165–1170.

6. **Yilmaz A., Javed O., Shah M.** Object tracking: A survey. *ACM Comput. Surv.*, 2006, Vol. 38 (4), P. 13.

7. **Khelloufi A., Achour N., Passama R., Che-rubini A.** Tentacle-based moving obstacle avoidance for omnidirectional robots with visibility constraints. *30th IEEE/RSJ Internat. Conf. on Intelligent Robots and Systems.*, 2017.

8. **Gerasimov V.N.** The motion control system of the mobile robot in environment with dynamic obstacles. *St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunication and Control Systems*, 2013, Vol. 5 (181), Pp. 100–108. (rus)

9. **Minaeian S., Liu J., Son Y.J.** Effective and efficient detection of moving targets from a UAV's camera. *IEEE Trans. Intell. Transp. Syst.*, 2018, Vol. 19 (2), Pp. 497–506.

10. **Wu Y., He X., Nguyen T.Q.** Moving object detection with a freely moving camera via background motion subtraction. *IEEE Trans. Circuits Syst. Video Technol.*, 2017, Vol. 27 (2), Pp. 236–248.

11. **Gong L., Yu M., Gordon T.** Online codebook modelling based background subtraction with a moving camera. *3rd International Conference on Frontiers of Signal Processing*, ICFSP, 2017, Pp. 136–140.

12. **Singh S., Arora C., Jawahar C.V.** Trajectory aligned features for first person action recognition. *Pattern Recognit.*, 2017, No. 62, Pp. 45–55.

13. **Yin X., Wang B., Li W., Liu Y., Zhang M.** Background subtraction for moving camera based on trajectory-controlled segmentation and label inference, KSIITrans. *Internet Inf. Syst.*, 2015, Vol. 9 (10), Pp. 4092–4107.

14. **Chen J., Sheng H., Zhang Y., Xiong Z.** Enhancing detection model for multiple hypothesis tracking. *Conference on Computer Vision and Pattern Recognition Workshops*, 2017, Pp. 2143–2152.

15. **Bagherzadeh M.A., Yazdi M.** Regularized least-square object tracking based on ℓ2,1 minimization. *3rd RSI International Conference on Robotics and Mechatronics*, ICROM, Oct. 2015, Pp. 535–539.

16. **Bouwmans T., Silva C., Marghes C., Zitouni M., Bhaskar H., Frelicot C.** On the role and the importance of features for background modeling and foreground detection. *Comput. Sci. Rev.*, 2018, No. 28, Pp. 26–91.

17. **Chau G., Rodriguez O.** Panning and jitter invariant incremental principal component pursuit for video background modelling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, Pp. 1844–1852.

18. **Chen C., Li S., Qin H., Hao A.** Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis. *Pattern Recognit.*, 2016, No. 52, Pp. 410–432.

19. **Thomaz L., Jardim E., da Silva A., da Silva E., Netto S., Krim H.** Anomaly detection in moving-camera video sequences using principal subspace analysis. *IEEE Trans. Circuits Syst. I. Regul. Pap.*, 2018, Vol. 65 (3), Pp. 1003–1015.

20. **Ma Y., Anderson J., Crouch S., Shan J.** Moving object detection and tracking with doppler LiDAR. *Remote Sens*, 2019, No. 11, P. 1154.

Received 27.01.2020.

СПИСОК ЛИТЕРАТУРЫ

1. **Wu S., Oreifej O., Shah M.** Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories // IEEE Internat. Conf. on Computer Vision. Nov. 2011. Pp. 1419–1426.

2. **Cucchiara R., Grana C., Piccardi M., Prati A.** Statistic and knowledge-based moving object detection in traffic scenes // IEEE Intell. Transp. Syst. 2000. Pp. 27–32.

3. **Malamas E.N., Petrakis E.G., Zervakis M., Petit L., Legat J.D.** A survey on industrial vision systems, applications and tools // Image Vis. Comput. 2003. Vol. 21 (2). Pp. 171–188.

4. **Hu W., Tan T., Wang L., Maybank S.** A survey on visual surveillance of object motion and behaviors // IEEE Trans. Syst Man Cybern. 2004. Vol. 34 (3). Pp. 334–352.

5. **Kim J.S., Yeom D.H., Joo Y.H.** Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems // IEEE Trans. Consum. Electron. 2011. Vol. 57 (3). Pp. 1165–1170.

6. **Yilmaz A., Javed O., Shah M.** Object tracking: A survey // ACM Comput. Surv. 2006. Vol. 38 (4). P. 13.

7. **Khelloufi A., Achour N., Passama R., Che-rubini A.** Tentacle-based moving obstacle avoidance for omnidirectional robots with visibility constraints // 30th IEEE/RSJ Internat. Conf. on Intelligent Robots and Systems. 2017.

8. **Герасимов В.Н.** Система управления движением мобильного робота в среде с динамическими препятствиями // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2013. № 5 (181). С. 100–108.

9. **Minaeian S., Liu J., Son Y.J.** Effective and efficient detection of moving targets from a UAV's camera // IEEE Trans. Intell. Transp. Syst. 2018. Vol. 19 (2). Pp. 497–506.

10. **Wu Y., He X., Nguyen T.Q.** Moving object detection with a freely moving camera via background motion subtraction // IEEE Trans. Circuits Syst. Video Technol. 2017. Vol. 27 (2). Pp. 236–248.

11. **Gong L., Yu M., Gordon T.** Online codebook modelling based background subtraction with a moving camera // 3rd Internat. Conf. on Frontiers of Signal Processing. 2017. Pp. 136–140.

12. **Singh S., Arora C., Jawahar C.V.** Trajectory aligned features for first person action recognition // Pattern Recognit. 2017. No. 62. Pp. 45–55.

13. **Yin X., Wang B., Li W., Liu Y., Zhang M.** Background subtraction for moving camera based on trajectory-controlled segmentation and label inference, KSIITrans // Internet Inf. Syst. 2015. Vol. 9 (10). Pp. 4092–4107.

14. **Chen J., Sheng H., Zhang Y., Xiong Z.** Enhancing detection model for multiple hypothesis tracking // Conf. on Computer Vision and Pattern Recognition Workshops. 2017. Pp. 2143–2152.

15. **Bagherzadeh M.A., Yazdi M.** Regularized least-square object tracking based on ℓ2,1 minimization // 3rd RSI Internat. Conf. on Robotics and Mechatronics, ICROM. Oct. 2015. Pp. 535–539.

16. **Bouwmans T., Silva C., Marghes C., Zitouni M., Bhaskar H., Frelicot C.** On the role and the importance of features for background modeling and foreground detection // Comput. Sci. Rev. 2018. No. 28. Pp. 26–91.

17. **Chau G., Rodriguez O.** Panning and jitter invariant incremental principal component pursuit for video background modelling // Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. Pp. 1844–1852.

18. **Chen C., Li S., Qin H., Hao A.** Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis // Pattern Recognit. 2016. No. 52. Pp. 410–432.

19. **Thomaz L., Jardim E., da Silva A., da Silva E., Netto S., Krim H.** Anomaly detection in moving-camera video sequences using principal subspace analysis // IEEE Trans. Circuits Syst. I. Regul. Pap. 2018. Vol. 65 (3). Pp. 1003–1015.

20. **Ma Y., Anderson J., Crouch S., Shan J.** Moving object detection and tracking with doppler LiDAR // Remote Sens. 2019. No. 11. P. 1154.

Статья поступила в редакцию 27.01.2020.

THE AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

DAEEF Feras
ДАЕЕФ Ферас
E-mail: ferasit87@gmail.com

DOI: 10.18721/JCSTCS.13103
УДК 004

EFFICIENCY ANALYSIS OF HIGH-LEVEL SYNTHESIS TOOLS FOR HARDWARE IMPLEMENTATION OF SORTING ALGORITHMS

A.P. Antonov, D.S. Besedin, A.S. Filippov

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

The article is devoted to the research of efficiency of Xilinx's high-level synthesis tools, the Vivado HLS package version 2019.2, for synthesis of a hardware implementation of sorting algorithms. The relevance of creating hardware implementation of sorting algorithms is determined by modern approaches to building high-performance heterogeneous computing systems and modern criteria for the efficiency of such systems – the ratio of performance to power consumption and the ratio of real performance to peak performance. The authors carried out a comparative analysis of the implementation of the selected sorting algorithms on a universal processor and on the basis of the VLSI Xilinx submarine research. The article discusses approaches to optimize the description of algorithms and control the Vivado HLS package to achieve optimal performance of the resulting hardware solutions. The article shows that the main performance gain is provided by parallelizing of the source arrays processing, which is achieved both by the settings of the design tool, the Vivado HLS package, the selected description style, as well as the features of the sorting algorithm selected for hardware implementation.

Keywords: hardware acceleration, sorting algorithms, high-level synthesis, reconfigurable hardware accelerator, FPGA.

Citation: Antonov A.P., Besedin D.S., Filippov A.S. Efficiency analysis of high-level synthesis tools for hardware implementation of sorting algorithms. Computing, Telecommunications and Control, 2020, Vol. 13, No. 1, Pp. 31-41. DOI: 10.18721/JCSTCS.13103

АНАЛИЗ ЭФФЕКТИВНОСТИ СРЕДСТВ ВЫСОКОУРОВНЕВОГО СИНТЕЗА ДЛЯ АППАРАТНОЙ РЕАЛИЗАЦИИ АЛГОРИТМОВ СОРТИРОВКИ

А.П. Антонов, Д.С. Беседин, А.С. Филиппов

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Статья посвящена исследованию эффективности средств высокоуровневого синтеза компании Xilinx, пакета Vivado HLS версии 2019.2, для создания аппаратной реализации алгоритмов сортировки. Актуальность создания аппаратной реализации алгоритмов сортировки определяется современными подходами к построению высокопроизводительных гетерогенных вычислительных систем и современными критериями эффективности таких систем: отношению производительности к энергопотреблению и отношению реальной производительности к пиковой производительности. Проведен сравнительный анализ реализации выбранных алгоритмов сортировки на универсальном процессоре и на базе СБИС ПЛ компании Xilinx. Рассмотрены подходы и способы оптимизации описания алгоритмов и управления пакетом Vivado HLS для достижения оптимальных показателей эффективности полученных аппаратных решений. Показано, что основной выигрыш в производительности дает возможность частичного распараллеливания процесса обработки исходных массивов, что достигается как настройками средства проектирования – пакета Vivado HLS, выбранным стилем описания, так и особенностями алгоритма сортировки, выбранного для аппаратной реализации.

Ключевые слова: аппаратное ускорение, алгоритмы сортировки, высокоуровневый синтез, реконфигурируемый аппаратный вычислитель, СБИС программируемой логики.

Ссылка при цитировании: Антонов А.П., Беседин Д.С., Филиппов А.С. Анализ эффективности средств высокоуровневого синтеза для аппаратной реализации алгоритмов сортировки // Информатика, телекоммуникации и управление. 2020. Т. 13. № 1. С. 31-41. DOI: 10.18721/JCSTCS.13103

Introduction

A modern trend in the development of computing systems is the creation of heterogeneous distributed hardware reconfigurable systems that provide a solution to the problem of hardware adaptation and reconfiguration for the algorithm to solve the main problem [1]. Such approach to the construction of computing systems allows us to create temporarily highly specialized hardware devices using the computing resources available as part of the system, in accordance with the logic of the problem to be solved. It provides a more efficient solution of computationally complex algorithms than universal processors and devices with SIMD architecture [2]. The most important modern performance criteria for high-performance computing systems are energy efficiency, i.e. performance-to-power ratio, and computational efficiency – the ratio of real performance to peak performance [3, 4].

The traditional procedure for developing specialized hardware devices based on the use of hardware description languages, for example, VHDL, Verilog HDL, System Verilog, is laborious and requires significant time both at the stage of device development and at the stage of debugging [5].

A modern approach for developing specialized hardware devices is to use the capabilities of high-level synthesis tools that are provided by leading VLSI manufacturers of programmable logic, such as Xilinx [6] and Intel PSG [7], and companies engaged in the development of electronic device development tools, for example, Mentor Graphics [8].

High-level synthesis tools allow both to synthesize hardware solutions to problems described in high-level programming languages, such as C or C ++, and to verify the correct operation of the algorithm and the synthesized device using a single test described in C or C ++.

The use of high-level synthesis tools to create computing systems is a new approach and there are currently no reliable data on its effectiveness in the implementation of many data processing algorithms with high computational complexity and significant memory requirements.

In order to analyze the efficiency of using high-level synthesis tools, it is necessary to carry out a comparative analysis of the implementation of the same algorithm described in C or C ++ based on a universal processor and its hardware implementation obtained as a result of high-level synthesis. A comparative analysis can be performed according to such an efficiency criterion as performance, or runtime, when solving a task of a given dimension, since the architecture of the universal processor and reconfigurable hardware are different.

Object to research

Sorting algorithms were chosen as the object of this research paper due to their widespread use for solving problems associated with processing large data, their computational complexity, memory requirements and the relevance of the task of accelerating their execution for many applications related to database processing.

A simplified classification of sorting algorithms is shown in Fig. 1. Among the variety of sorting algorithms [9], several typical algorithms were selected for this research: comb sorting, gnome sorting and merge sorting.

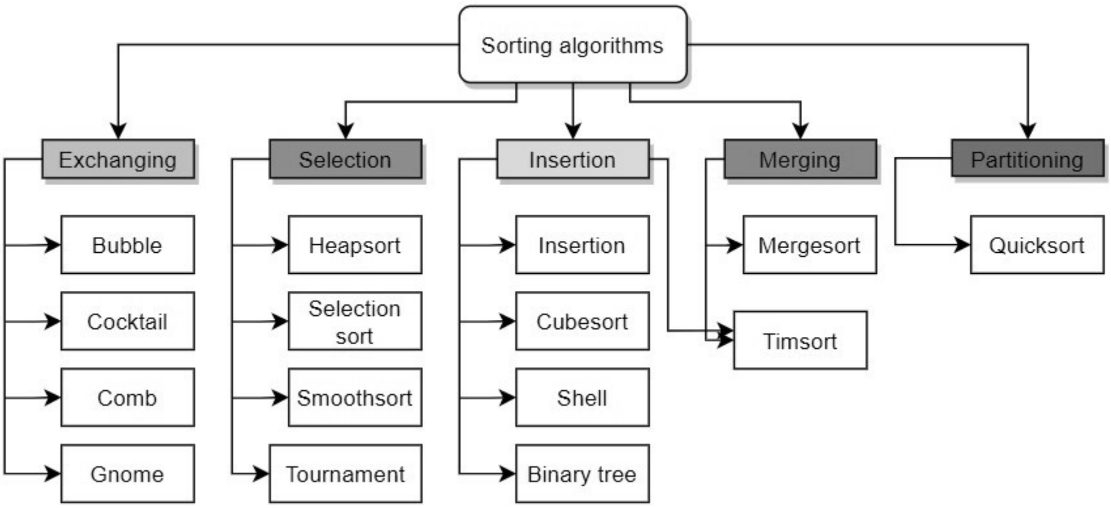


Fig. 1. A simplified classification of sorting algorithms

These algorithms are typical representatives of the classes shown in Fig. 1, and that is the reason to choose them for the research. Moreover, for choosing objects of research such as sorting algorithms, we need to allow for the significant limitation of modern high-level synthesis tools due to the impossibility of implementing recursive algorithms.

Comb sorting and gnome sorting belong to the class of exchange sorting algorithms, they are simple to implement, are considered the slowest when implemented on universal processors, have high O (n^2) computational complexity and do not require additional memory costs, like all representatives of this class (O(1)) [10, 11].

Merge sorting is an algorithm with sorting principle significantly different from exchange sorting algorithms, but it is also simple to implement. This sorting algorithm is faster than exchange algorithms when running on universal processors, since it has less computational complexity O (n log n), but significantly higher memory costs O(n) [12].

Method and research methodology

The research method is simulation of solving sorting problems on computational structures with different architectures and conducting a comparative analysis according to the selected criteria.

As a criterion for a comparative analysis, we selected performance parameters related to each other, that is, the number of operations performed in a given unit of time, and speed, which is the time spent on the task. The selection of these criteria for comparative analysis is justified by the fact that the goal of creating hardware solutions is to increase speed and productivity in solving computationally complex problems and, as a result, increase the computational efficiency of the entire high-performance system.

Here is a list of selected hardware and software tools used in this research, simulation and comparative analysis:

- For the software implementation on a universal processor:
  - IDE – JetBrains CLion;
  - Hardware part – PC based on Intel Core i7-4710HQ 2.50 GHz, with 12 GB RAM, type DDR3.
- For the hardware implementation based on FPGA:
  - IDE – Vivado HLS (High level synthesis) [6];
  - Hardware part – FPGA family Virtex UltraScale by Xilinx [13]: XCVU 125-flvc2104-3-e.

The JetBrains development environment CLion is a cross-platform C and C ++ development environment developed by JetBrains. It allows you to easily compile and run any programs using popular compilers (GCC, Clang, MinGW, Cygwin) and pre-installed libraries. It means the ability to work with the same source code of the program with the addition of standard C libraries operators for calculating the expended time.

For this purpose, the description of the sorting algorithms in the C language was made using the functions of the library time.h, which allows us to estimate the time interval between two control points during program execution, which, thus, when simulating a solution to a problem on a universal processor, provides data on speed and performance for comparative analysis.

The Vivado HLS (High level synthesis) development environment synthesizes the description of the device operation algorithm presented in C or C ++ into a hardware implementation; evaluates the performance and speed of a synthesized device; displays the expected hardware "cost" for its implementation on the basis of the selected element base – the selected FPGA part. This development environment allows optimization of the created device during synthesis, setting up its implementation to use various resources available in the target FPGA part; pipelining and parallelizing hardware implementation according to user-defined criteria.

To assess the performance and speed of a synthesized device, Vivado HLS offers the calculation of the minimum possible period of the clock frequency synchronizing the operation signal of the device, and an estimate of the number of periods of the clock frequency for the complete execution of the algorithm, in other words, the number of clock cycles through which the input of the device that implements the synthesized algorithm can be fed new data. It is possible to calculate the time of one sorting, by multiplying the estimate of the period of the clock frequency by the number of required clock cycles based on these data.

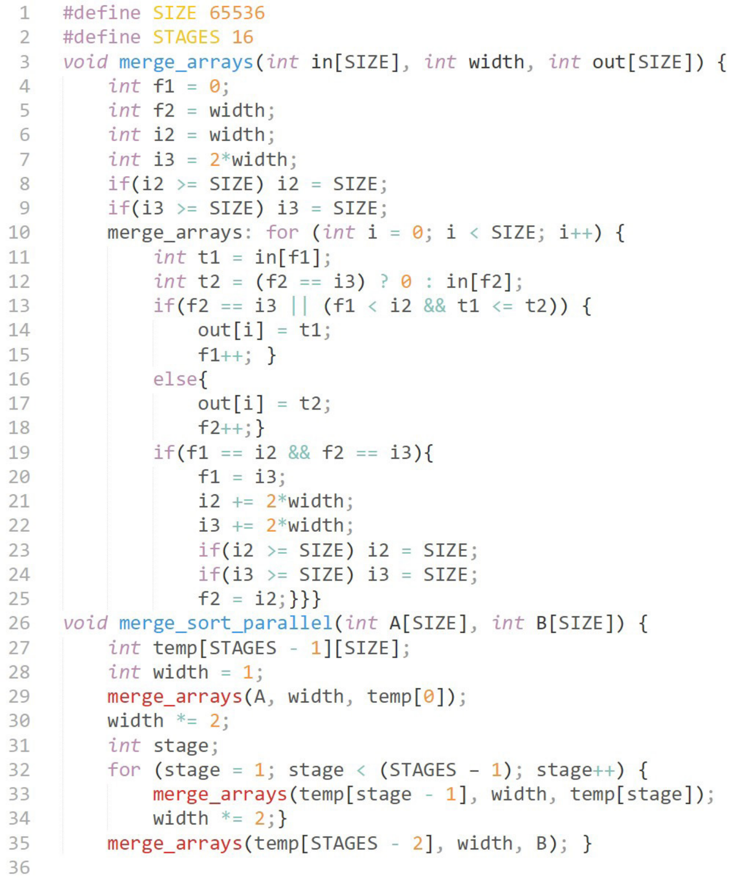
The research methodology includes the following steps:

- The creation of a text code description of an algorithm suitable for both a software implementation based on a universal processor and for the synthesis of a reconfigurable hardware solution. In the created description, the means of controlling the runtime on the basis of a universal processor are used. The description should allow to process arrays of input data of arbitrary size.
- The creation of a test text code description that will be used to verify both the correct operation of the initial description of the algorithm in the C language and the model of the synthesized hardware solution. In the created test description, it is necessary to launch the function of the tested algorithm several times, since this allows you to simulate a continuous data stream characteristic of a hardware implementation. The description should allow you to create arrays of source data of arbitrary size. The source arrays must be initialized random, with a uniform distribution, integers.
- Simulation based on a universal processor:
  - Test of the initial description of the algorithm based on a universal processor for a given set of array sizes;
  - Software implementation of an algorithm based on a universal processor for a given set of input array sizes. Obtaining a set of characteristics for the execution time of the algorithm.
- Simulation and optimization of hardware implementation of the algorithm:
  - Test of the initial description of the algorithm in the framework of a high-level synthesis system on a given set of array sizes;
  - Iteratively conducting synthesis-optimization stages for a given set of array sizes and a selected set of control directives for a high-level synthesis system. The goal is to achieve maximum performance for each set of array sizes if there is a limit – the logical capacity selected of the selected FPGA part;
  - Hardware and software testing, based on a common test for hardware and software implementations, of each optimal hardware implementation of the algorithm for each set of array sizes.
- Comparative analysis of software and hardware implementations of the same algorithm.

Conducting research

The following sets of array sizes were selected for the research: 128; 1024; 16384; 32768; 65536; 131072. All numbers were of type Integer (signed integer 32 bits).

The description in C language of the merge sorting algorithm used for simulation, both for a software implementation based on a universal processor and for synthesizing a hardware implementation of an algorithm, is shown in Fig. 2. In this code, for clarity, the directives for optimizing hardware implementation and design are omitted, providing an assessment of productivity and performance in software implementation.



```c
#define SIZE 65536
#define STAGES 16
void merge_arrays(int in[SIZE], int width, int out[SIZE]) {
    int f1 = 0;
    int f2 = width;
    int i2 = width;
    int i3 = 2*width;
    if(i2 >= SIZE) i2 = SIZE;
    if(i3 >= SIZE) i3 = SIZE;
    merge_arrays: for (int i = 0; i < SIZE; i++) {
        int t1 = in[f1];
        int t2 = (f2 == i3) ? 0 : in[f2];
        if(f2 == i3 || (f1 < i2 && t1 <= t2)) {
            out[i] = t1;
            f1++; }
        else{
            out[i] = t2;
            f2++;}
        if(f1 == i2 && f2 == i3){
            f1 = i3;
            i2 += 2*width;
            i3 += 2*width;
            if(i2 >= SIZE) i2 = SIZE;
            if(i3 >= SIZE) i3 = SIZE;
            f2 = i2;}}}
void merge_sort_parallel(int A[SIZE], int B[SIZE]) {
    int temp[STAGES - 1][SIZE];
    int width = 1;
    merge_arrays(A, width, temp[0]);
    width *= 2;
    int stage;
    for (stage = 1; stage < (STAGES - 1); stage++) {
        merge_arrays(temp[stage - 1], width, temp[stage]);
        width *= 2;}
    merge_arrays(temp[STAGES - 2], width, B); }
```

Fig. 2. Description of merge sort algorithm in C language

During the iterative stages of synthesis optimization, for each given set of array sizes, the following sets of control directives of the Vivado HLS high-level synthesis system were selected:

- Directives for choosing an interface architecture for implementing reading raw data and writing sorted data:
  - This allows the synthesizer with certain interface architectures to automatically use BRAM blocks for intermediate storage of an array of numbers;
  - This allows you to speed up the steps of reading the source data and writing sorted values in some cases, depending on the features of the algorithm.
- Pipeline directives for both internal and external loops in the description of algorithms:
  - Pipelining allows parallelization of both reading the source data, performing individual steps of data processing, and recording sorted data in certain cases, depending on the features of the algorithm.
- Dataflow directives for pipelining at the level of data flows, that is, in relation to the considered implementations of the algorithm, at the level of data processing between cycles:

- Pipelining at the data flow level allows the device to compose an output array during the sorting procedure in some cases, depending on the features of the algorithm. That can make the device more adaptive to the features of the input data, for example, for the case if the array is sorted before the algorithm passes completely.

Research results

As was pointed above, a set of sorting algorithms were considered for hardware implementations synthesized by HLS tool. There are several limitations imposed by the HLS tool on the description of the investigated algorithms. These are:

- Programming language must be C or C ++.
- It must be a non-recursive description.
- Dynamic memory allocation should not be used.

As a result of the research, it was found that for the sorting algorithms of comb and gnome sorting, the optimal hardware implementations, devices obtained as a result of synthesis, have a similar architecture, the features of which are:

- Using dual port RAM memory for input array.
- Using the DataFlow directive, which provides for "forwarding" of the processed data between the internal cycles of the algorithm with the implementation of ping-pong mode.

For the merge sorting algorithm, the hardware solution obtained as a result of synthesis and optimization, a device that implements the specified algorithm, has the following architectural features:

- Dual port RAM memory is used to store the sorted array.
- Implement pipelining of the merge cycle of two arrays.
- During the optimization of this device, the following steps were applied:
  - Arrays for storing intermediate data are divided into separate memory blocks, which allows simultaneous merging of different arrays;
  - Merge loop in the main sorting function are unrolled for parallel implementation of all loop iterations. In addition to the previous paragraph, this allows you to merge different parts of the entire array of numbers at the same time, that is, to maximize parallelizing of the sorting process.

Table 1 shows the estimates of hardware "cost" for the implementation of optimal hardware devices (optimality criteria were considered above) for the indicated sorting algorithms: Comb, Gnome, Merge.

Table 1

Hardware "cost" of synthesized devices

Array size (number of samples)	Algorithm					
	Comb		Gnome		Merge	
	LCELL, num.	BRAM, num.	LCELL, num.	BRAM, num.	LCELL, num.	BRAM, num.
128	634	0	340	0	6260	12
1024	688	0	346	0	9032	18
16384	741	0	354	0	12812	277
32768	798	0	356	0	13877	812
65536	813	0	351	0	14626	1740
131072	849	0	353	0	15591	3712

Where:

- LCELL – logical blocks (cells) of FPGA part, that contains look-up tables (LUT) used to implement logical functions, and synchronous triggers (FF) used to store data.
- BRAM – built-in memory blocks that are used to store intermediate data when implementing the sorting algorithm. These embedded memory blocks are taken into account while estimating hardware "cost" of the algorithm. The external memory, which is necessary for storing the source and sorted arrays is not taken into account because this hardware "cost" is a constant for all sorting algorithms.

For clarity, the data shown in Table 1 about the logic blocks used to implement each of the algorithms (LCELL) are summarized in one graph, shown in Fig. 3.

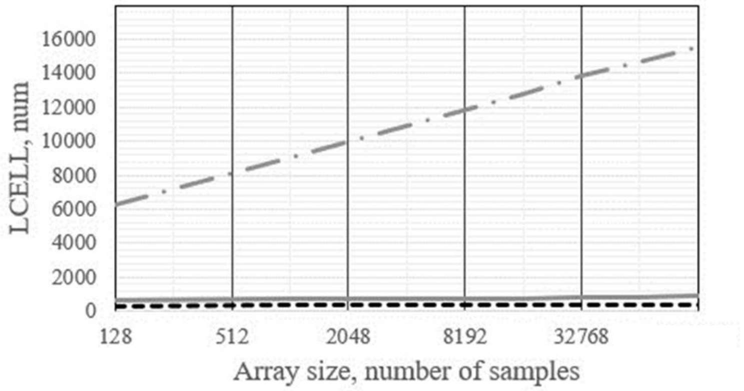


Fig. 3. Hardware "cost" of LCELL synthesized devices
(—) – Comb; (---) – Gnome; (— ·) – Merge

An analysis of the graphs in Fig. 3 shows that the hardware "cost" for implementing sorting algorithms with comb and gnome sorting are significantly lower than the hardware "cost" for implementing the merge sorting algorithm. In this case, there is a directly proportional relationship between the size of the sorted array and the number of logical cells used to implement the merge sort algorithm.

Table 2 shows the performance estimates for the optimal implementation of all synthesized devices obtained in the framework of the Vivado HLS development environment.

Table 2

Performance assessment of sorting devices

Array size (number of samples)	Algorithm					
	Comb		Gnome		Merge	
	Period, ns	Latency, num. of clocks	Period, ns	Latency, num. of clocks	Period, ns	Latency, num. of clocks
128	4.066	35219	6.229	32770	5.176	916
1024	4.066	2141219	6.229	2097154	5.176	10269
16384	4.066	537575459	6.229	536870914	5.176	229417
32768	4.066	2148892707	6.229	2147483650	5.176	491804
65536	4.066	8592752675	6.229	4294967300	5.176	1048623
131072	4.066	34371665987	6.229	8589934600	5.176	2228274

Where:

- Latency is the number of clock cycles of the synchronization signal required to obtain a ready, sorted, array at the device output.
- Period is the minimum possible period of the synchronization signal.

Therefore, you can calculate the time to complete the array sorting, and, therefore, you can determine the performance of the synthesized hardware implementation of the sorting algorithm by multiplying the period and the number of delay ticks.

Table 3 shows the estimates obtained in the framework of the study on the execution time of the selected sorting algorithms on a universal processor and on the basis of synthesized hardware computers.

Table 3

Estimation of sorting time

Array size (number of samples)	Algorithm					
	Comb		Gnome		Merge	
	CPU, s	FPGA, s	CPU, s	FPGA, s	CPU, s	FPGA, s
128	0.000021	0.000157	0.000046	0.0002041	0.00002	0.0000047
1024	0.00102	0.009651	0.00211	0.01306	0.00016	0.0000531
16384	0.3204	2.3895	0.514394	3.3442	0.00252	0.001187
32768	1.3156	9.5581	2.06334	13.3767	0.00496125	0.002544
65536	5.4017	38.2324	8.2765	53.5067	0.0099225	0.005427
131072	22.18	152.782	33.1937	214.0268	0.019845	0.01085

For clarity and simplification of the analysis, the data given in Table 3 are summarized in the graphs presented in Fig. 4 and Fig. 5. All graphs in the figures are presented in a logarithmic scale: base 10 for the ordinate axis, base 2 for the abscissa axis.

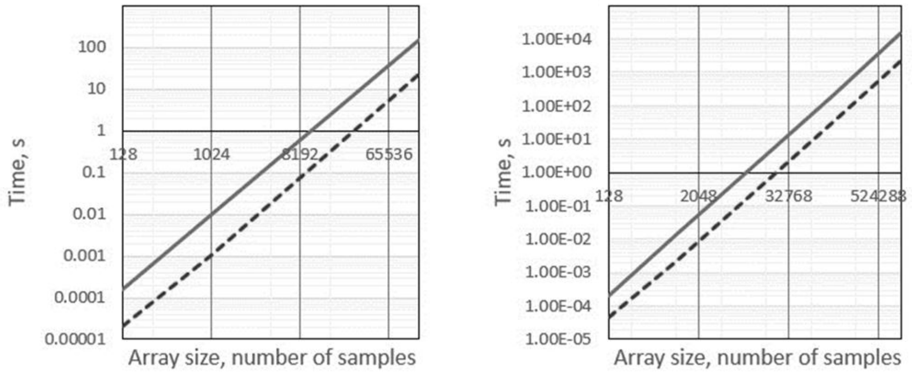


Fig. 4. Dependence of the sorting time with a comb (left) and a gnome (right) on the size of the array (---) – CPU; (—) – FPGA

Conclusions

Analysis of the research results allowed us to draw the conclusions below.

The hardware implementation of the algorithm does not always provide greater performance compared to the execution of the algorithm on a universal processor. So, the graphs in Fig. 4 show that the execution time of sorting algorithms with a comb and gnome on a universal processor is an order of magnitude shorter than the time achievable with the hardware implementation of these algorithms. These results can be explained:

- These algorithms involve sequential operations that are difficult to parallelize using high-level synthesis tools.
- The clock speed of the universal processor is about an order of magnitude higher than the clock frequency for hardware implementation: the processor used for the study has a clock frequency of 2.5 GHz, and the synthesized device, as it is not difficult to calculate from the data given in Table 2, is about 250 MHz. Therefore, in the sequential implementation of the algorithm, the universal processor demonstrates a tenfold gain in the execution time of the algorithm.

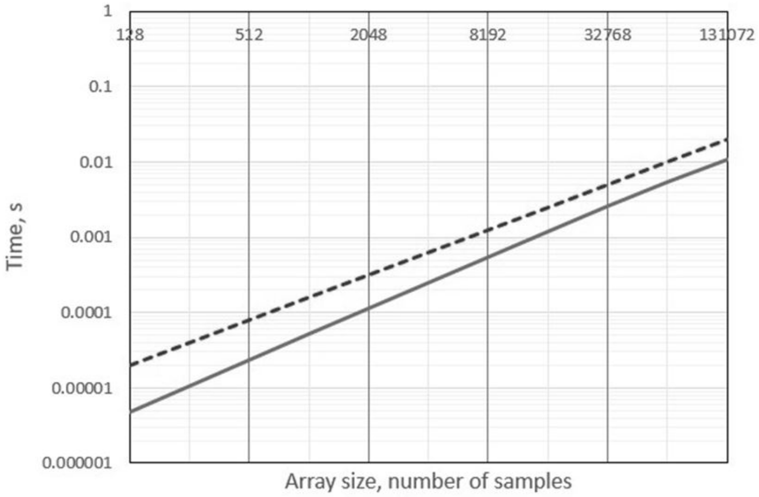


Fig. 5. Dependence of merge sorting time on array size
(---) – CPU; (—) – FPGA

The merge sorting algorithm allows you to parallelize its execution, so the synthesized hardware calculator allows you to simultaneously perform several sorting stages, stream parallelization, and several operations of one sorting stage, pipelining, which provides it with an advantage in speed compared to running the algorithm on a universal processor, see Table 3 and Fig. 5. Moreover, as can be seen from Table 1 and Fig. 3, while the number of samples of the processed arrays are increasing, the hardware expenses are growing as well. The limitation for parallelization is the number of logic elements in modern FPGA circuits.

The further direction of the research is related to the expansion of the number of sorting algorithms covered and the search for such algorithms that will maximize the capabilities of modern FPGAs and high-level synthesis tools, providing a significant performance increase in solving sorting problems in comparison with the best, with the fastest, implementations of sorting algorithms on universal processors.

REFERENCES

1. **Antonov A.P., Zaborovskiy V.S., Kalyayev I.A.** Architecture of reconfigurable heterogeneous distributed supercomputer system for solving problems of intelligent data processing in the era of digital transformation of the economy. *Voprosy Kiberbezopasnosti*, 2019, Vol. 33, No. 5, Pp. 2–11. (rus). DOI:10.21681/2311-3456-2019-5-02-11

2. **Antonov A.P., Zaborovskiy V.S., Kiselev I.O.** The reconfigurable computational modules in network-centric supercomputer systems. *Sistemy vysokoy dostupnosti*, 2018, Vol. 14, No. 3, Pp. 57–62. (rus). DOI:10.18127/j20729472-201803-09

3. **Mantovani F., Calore E.** Performance and power analysis of HPC workloads on heterogeneous multi-node clusters. *Low Power Electron*, 2018, Vol. 2, No. 8, Pp. 1–14. DOI:10.3390/ jlpea8020013

4. **Usman A., Fathy A., Aiiad A., Abdullah A.** Performance and power efficient massive parallel computational model for HPC heterogeneous exascale systems. *IEEE Access*, 2018, No. 6, Pp. 23095–23107. DOI:10.1109/ACCESS.2018.2823299

5. **Kobayashi R., Oobata Y., Fujita N., Yamaguchi Y., Boku T.** OpenCL-ready high speed FPGA network for reconfigurable high performance computing. *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, 2018, Pp. 192–201. DOI:10.1145/3149457.3149479

6. Sreda Vivado HLS. Available: *https://www.xilinx.com/video/hardware/vivado-hls-tool-overview.html* (Accessed: 30.01.2020).

7. Intel HLS compiler. Available: *https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/hls-compiler.html?wapkw=HLS* (Accessed: 30.01.2020).

8. Catapult HLS. Available: *https://www.mentor.com/hls-lp/catapult-high-level-synthesis/* (Accessed: 30.01.2020).

9. Algoritmy sortirovki. Available: *https://en.wikipedia.org/wiki/Sorting_algorithm* (Accessed: 30.01.2020).

10. Sorting Algorithm – Comb Sort. Available: *https://www.ideserve.co.in/learn/comb-sort* (Accessed: 31.01.2020).

11. Gnomya sortirovka. Available: *https://ru.wikipedia.org/wiki/Gnomya_sortirovka* (Accessed: 30.01.2020). (rus)

12. Sortirovka sliyaniyem. Available: *https://ru.wikipedia.org/wiki/Sortirovka_sliyaniyem* (Accessed: 06.02.2020). (rus)

13. UltraScale and UltraScale + FPGA. Available: *https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale.html#productTable* (Accessed: 31.01.2020).

Received 16.01.2020.

СПИСОК ЛИТЕРАТУРЫ

1. **Антонов А.П., Заборовский В.С., Каляев И.А.** Архитектура реконфигурируемой гетерогенной распределенной суперкомпьютерной системы для решения задач интеллектуальной обработки данных в эпоху цифровой трансформации экономики // Вопросы кибербезопасности. 2019. Т. 33. № 5. С. 2–11. DOI:10.21681/2311-3456-2019-5-02-11

2. **Антонов А.П., Заборовский В.С., Киселев И.О.** Специализированные реконфигурируемые вычислители в сетецентрических суперкомпьютерных системах // Системы высокой доступности. 2018. Т. 14. № 3. С. 57–62. DOI:10.18127/j20729472-201803-09

3. **Mantovani F., Calore E.** Performance and power analysis of HPC workloads on heterogeneous multi-node clusters // Low Power Electron. 2018. Vol. 2. No. 8. Pp. 1–14. DOI:10.3390/ jlpea8020013

4. **Usman A., Fathy A., Aiiad A., Abdullah A.** Performance and power efficient massive parallel computational model for HPC heterogeneous exascale systems // IEEE Access. 2018. No. 6. Pp. 23095–23107. DOI:10.1109/ACCESS.2018.2823299

5. **Kobayashi R., Oobata Y., Fujita N., Yamaguchi Y., Boku T.** OpenCL-ready high speed FPGA network for reconfigurable high performance computing // Proc. of the Internat. Conf. on High Performance Computing in Asia-Pacific Region. 2018. Pp. 192–201. DOI:10.1145/3149457.3149479

6. Среда Vivado HLS // URL: https://www.xilinx.com/video/hardware/vivado-hls-tool-overview.html (Дата обращения: 30.01.2020).

7. Intel HLS compiler // URL: https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/hls-compiler.html?wapkw=HLS (Дата обращения: 30.01.2020).

8. Catapult HLS // URL: https://www.mentor.com/hls-lp/catapult-high-level-synthesis/ (Дата обращения: 30.01.2020).

9. Алгоритмы сортировки // URL: https://en.wikipedia.org/wiki/Sorting_algorithm (Дата обращения: 30.01.2020).

10. Sorting Algorithm – Comb Sort // URL: https://www.ideserve.co.in/learn/comb-sort (Дата обращения: 31.01.2020).

11. Гномья сортировка // URL: https://ru.wikipedia.org/wiki/Гномья_сортировка (Дата обращения: 30.01.2020).

12. Сортировка слиянием // URL: https://ru.wikipedia.org/wiki/Сортировка_слиянием (Дата обращения: 06.02.2020).

13. UltraScale and UltraScale+ FPGA // URL: https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale.html#productTable (Дата обращения: 31.01.2020).

Статья поступила в редакцию 16.01.2020.

THE AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Antonov Alexander P.
Антонов Александр Петрович
E-mail: antonov@eda-lab.ftk.spbstu.ru

Besedin Denis S.
Беседин Денис Сергеевич
E-mail: 1310nero@mail.ru

Filippov Alexey S.
Филиппов Алексей Семенович
E-mail: alexey.s.filippov@gmail.com

DOI: 10.18721/JCSTCS.13104
УДК 621.39

TRANSMISSION EFFICIENCY OF MULTI-FREQUENCY SIGNALS IN MBC USING AMPLITUDE LIMITATION ON THE TRANSMITTING MODULE

D.C. Nguyen, S.V. Zavyalov, S.V. Volvenko

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

The ability to receive electromagnetic waves reflected from meteor traces is the basis for the construction of meteor burst communication (MBC) systems. MBC has a low data rate (about a few dozen kilobits per second). Therefore, from the point of view of increasing the information transfer rate, we can use multi-frequency signals, for example, OFDM signals. These signals occupy a smaller frequency band, but have a high peak-to-average power ratio. In this article, we used simulation modeling to transmit information in an MBC system using multi-frequency signals under the condition of amplitude limitation on the transmitting module.

Keywords: meteor burst communications, peak-to-average power ratio, amplitude limitation, MBC, OFDM, PAPR.

Citation: Nguyen D.C., Zavyalov S.V., Volvenko S.V. Transmission efficiency of multi-frequency signals in MBC using amplitude limitation on the transmitting module. Computing, Telecommunications and Control, 2020, Vol. 13, No. 1, Pp. 42-52. DOI: 10.18721/JCSTCS.13104

ЭФФЕКТИВНОСТЬ ПЕРЕДАЧИ МНОГОЧАСТОТНЫХ СИГНАЛОВ В МЕТЕОРНОМ КАНАЛЕ ПРИ УСЛОВИИ АМПЛИТУДНОГО ОГРАНИЧЕНИЯ НА ПЕРЕДАЮЩЕМ МОДУЛЕ

Д.К. Нгуен, С.В. Завьялов, С.В. Волвенко

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Возможность приема отраженных от метеорных следов электромагнитных колебаний лежит в основе построения систем метеорной связи. Метеорная радиосвязь имеет низкую скорость передачи данных (единицы-десятки бит в секунду). С точки зрения повышения скорости передачи информации мы можем использовать многочастотные сигналы, например, сигналы OFDM. Эти сигналы занимают меньшую полосу частот, но обладают высоким значением пик-фактора. Выполнено имитационное моделирование передачи информации в метеорном канале при использовании многочастотных сигналов при условии амплитудного ограничения на передающем модуле.

Ключевые слова: метеорная радиосвязь, пик-фактор, амплитудное ограничение, MBC, OFDM, PAPR.

Ссылка при цитировании: Нгуен Д.К., Завьялов С.В., Волвенко С.В. Эффективность передачи многочастотных сигналов в метеорном канале при условии амплитудного ограничения на передающем модуле // Информатика, телекоммуникации и управление. 2020. Т. 13. № 1. С. 42-52. DOI: 10.18721/JCSTCS.13104

Introduction

When the Earth moves in the orbit, cosmic particles known as meteoroids daily enter the Earth's atmosphere. The meteoroid intrusion into the Earth's atmosphere occurs at a speed ranging from 11 to 73 km/s. It is accompanied by heating and complex destruction processes: melting, spraying, crushing and evaporation. Evaporated meteor molecules collide with air molecules and atoms and decompose into atoms, causing their excitation and ionization. Meteor atoms are mainly ionized, since their ionization potential is lower than the ionization potential of gas atoms. The size of the ionized trails formed along the trajectory of the meteoroid depends on the mass and speed of the meteoroid [1].

The length of the trail can reach tens of kilometers. The radius of the trail at its formation is about a meter and increases over time due to diffusion. Ionized trail of meteors is able to reflect radio waves, and therefore they can be observed in the radio spectrum. The ability to receive electromagnetic waves reflected from meteor trails is the basis for constructing meteor burst communication (MBC) systems (Fig. 1).

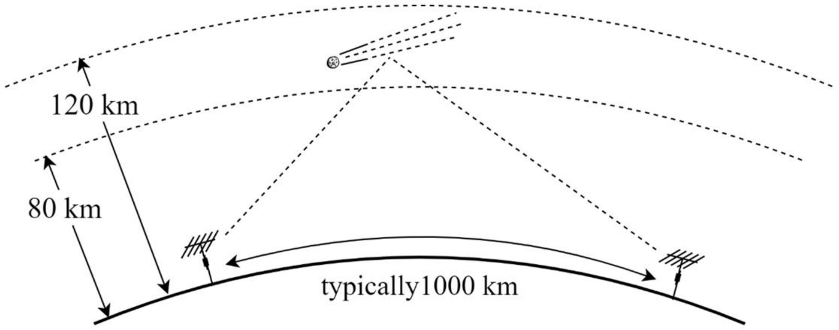


Fig. 1. Illustration of the principle of MBC [2]

As a rule, MBC uses a frequency ranging from 40 MHz to 100 MHz [3, 4]. The distance, at which communication can be established, is mainly determined by the height, at which the meteor trail is created, and the radius of curvature of the Earth's surface. The maximum communication distance between radio stations can be about 2000 km [1–3].

When transmitting messages over long distances, MBC systems provide a simpler solution to the problems of organizing communication than, for example, short-wave communication systems. So, in the short-wave communication systems, to ensure high quality of message reception, various independent carrier oscillation frequencies are used during daytime and nighttime operation, which is caused by a change in the height of the ionospheric layer during the day. Due to interference at the main frequencies, the introduction of reserve frequencies is required in these systems. Serious problems in short-wave systems arise due to the presence of selective fading, limitation on the maximum applicable carrier frequency, the choice of the optimal transmission frequency, and exposure to aurora in the northern regions. The indicated drawbacks are deprived of MBC systems, in which messages are transmitted on one fixed carrier frequency, and the range may vary from 300 to 2000 km. The transmission channel allows one frequency to automatically temporarily separate the work of hundreds of peripheral stations, which eliminates the need for transitions to other frequencies.

It should be noted that MBC has its drawbacks. The most obvious disadvantage of using MBC is a very low data rate (units to tens of bits per second). MBC is also limited by distance (no more than 2000 km). The third disadvantage of MBC is the impossibility of organizing voice communication and video communication due to the longer waiting time for the formation of a suitable meteor trail.

Usually, a single-frequency signal is used in an MBC system [5–9]. Frequency modulation signals are often used, since the peak-to-average power ratio (PAPR) of the emitted oscillations in this case will be minimal, which allows to maximize the efficiency of power amplifiers on the transmitting module. From the point of view of increasing the information transfer rate, we can use multi-frequency signals, for example, orthogonal frequency division multiplexing (OFDM) signals [10]. OFDM is a digital modulation scheme that uses a set of orthogonal frequency subcarriers. OFDM signals occupy a smaller frequency bandwidth, but have a high PAPR value [11, 12]. PAPR is the ratio of the maximum power P_{max} of the signal $s(t)$ to the average power P_{avg}:

$$\mathrm{PAPR} = \frac{P_{max}}{P_{avg}}. \tag{1}$$

Reducing the PAPR value to increase the efficiency of the amplifier stages of the transmitting devices is possible by various methods of limiting the emitted oscillations, for example, by using the amplitude limit (i.e., the generated signal first passes through the amplitude limiter and only then through the power amplifier). An additional positive aspect of the amplitude limitation is an increase in the average signal power, which should have a positive effect on the transmission range of information at a fixed error probability or on a decrease in the probability of error at a fixed transmission distance. However, the introduction of a PAPR limitation leads to additional signal distortion. Signals located at adjacent subcarrier frequencies begin to influence each other, which ultimately leads to an increase in the probability of errors in reception.

Thus, the aim of this work is to assess the feasibility and effectiveness of the use of multi-frequency signals in the meteor burst communications under the condition of amplitude limitation of signal on the transmitting module.

Description of multi-frequency signals

We consider OFDM signals as a general form of multi-frequency signals. The OFDM signal is a multi-frequency signal with N subcarriers and complex modulation symbols C_k on each subcarrier without a cyclic prefix. OFDM symbols with a duration T_s have the following form [13]:

$$s(t) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} C_k e^{j2\pi k\Delta f t}, t \in [0; T_s]. \tag{2}$$

In formula (2), the frequency spacing between adjacent subcarriers is $\Delta f = 1 / T_s$. In this article, we use the methods of generating and receiving OFDM signals presented in Fig. 2 [1].

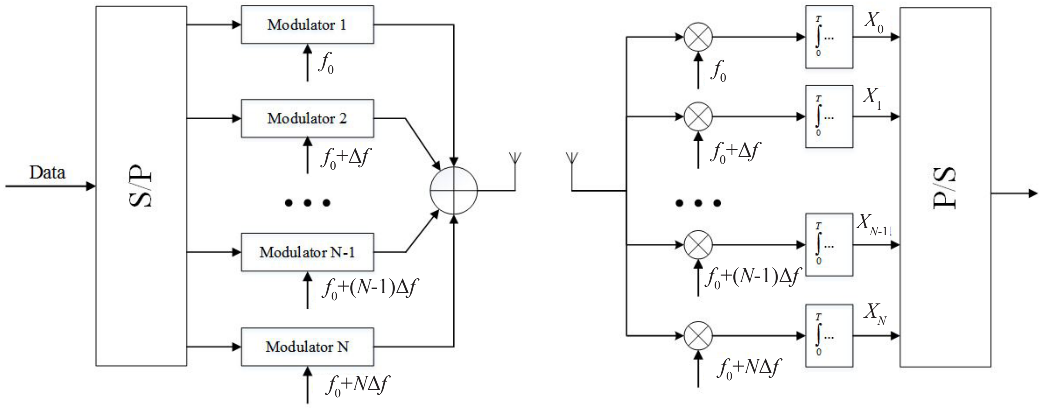


Fig. 2. Scheme of formation and reception of OFDM signals

Characteristics of meteor burst communications

To describe the processes occurring in the meteor channel, it is required to use the distribution of the duration, intensity of occurrence and power of meteor tracks. The intensity of the appearance of traces depends on the time of day and year. The average value of the duration of the meteor trail T_c and the interval between two consecutive traces ΔT_c are determined by the formulas:

$$T_c = \frac{\sum trails \quad duration}{number \quad of \quad trails}; \qquad \Delta T_c = \frac{\sum duration \quad between \quad two \quad trails}{number \quad of \quad trails}.$$

Depending on the concentration of electrons, meteor trails are divided into two main types: underdense and overdense. For underdense trail, the linear electron concentration n on the trace axis is less than $2.4 \cdot 10^{14}$ electrons per meter, and the incident radio wave is reflected in view of the effect of the secondary electron reemission [2]. The amplitude-time characteristic (ATC) of the radio reflection from an underdense trail after a rapid increase in amplitude has an exponential decline due to the recombination of free electrons and ions of the trail. For overdense trail ($n > 2.4 \cdot 10^{14}$ electrons per meter), the incident wave is reflected, as if from a metal cylinder of some critical radius [15]. The ATC of radio reflection from a classic overdense trail is as follows: after a phase of rapid growth, the reflection amplitude then slowly increases and, having reached its maximum, slowly drops to zero.

In this paper, we consider a model of a meteor channel based on an underdense trail, because this is the most frequently observed MBC mechanism [6, 16, 17]. In this case, the energy per bit to noise power spectral density ratio E_b/N_0 in the transmission channel will change in the same way as the ATC reflection, i.e. after a rapid increase in amplitude has an exponential decline.

According to the recommendation ITU-R F.1113[1], for a power level of a transmitting module of 200 W, when modeling, we select:

- mean duration of a meteor trail: $T_c = 330$ ms;
- mean duration between two meteor trails: $\Delta T_c = 800$ ms.

An example of the dependence of E_b/N_0 on time t at the input of the receiving module using the above parameters is shown in Fig. 3.

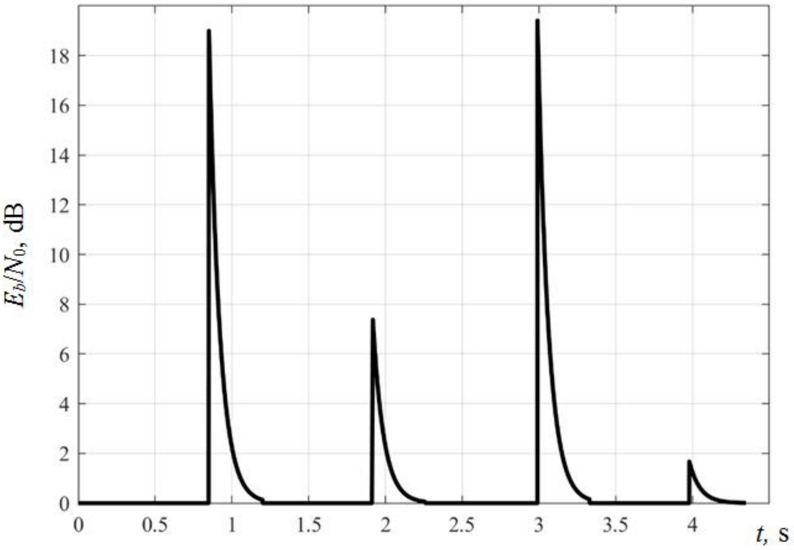


Fig. 3. An example of the dependence of E_b/N_0 on time for MBC

[1] Recommendation ITU-R F.1113. Radio systems employing meteor-burst propagation, 1994.

Simulation model

A simulation model for analyzing the effectiveness of the proposed signals under the condition of amplitude limitation on the transmitting module is presented in Fig. 4 and implemented in the Matlab system.

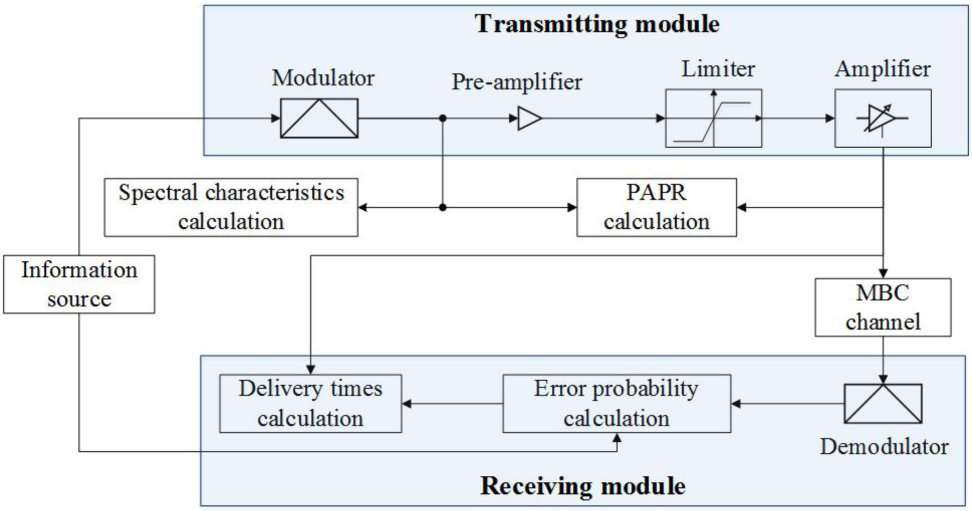


Fig. 4. Structural scheme of the simulation model

It includes an information source, a transmitting module block, spectral characteristics and PAPR calculation blocks, an MBC channel simulation block, a demodulator, error probability and delivery time calculation blocks.

In the information source a pseudo-random sequence of zeros and units of a given volume is formed depending on the number of subcarriers and equiprobable symbols used.

The transmitting module includes a modulator, pre-amplifier, limiter and power amplifier. The pseudo-random sequence received from the information source is fed to the input of the modulator. We used modulation scheme QPSK. The number of frequency subcarriers is 8. The generated symbols from the output of the modulator are fed to the input of the preamplifier to increase the average radiation power. Then, the received signal is sent to the limiter to limit the amplitude of the signal to the maximum amplitude of the output signal of the modulator. This means that the average signal power increases, while the peak signal power does not change, which leads to a decrease in PAPR.

The procedure for restricting PAPR of symbols is shown in Fig. 5. As long as level A, which determines the PAPR reduction value, does not exceed the A_{lim} value, the output value of U_{out} linearly depends on the input U_{in}. Starting from value A_{lim}, U_{out} stops changing.

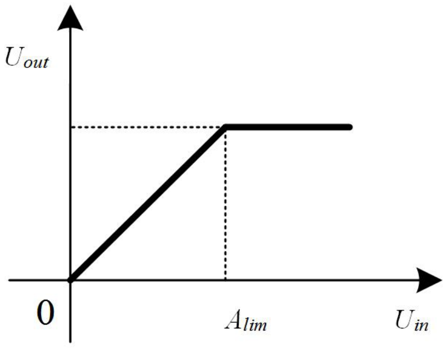


Fig. 5. Characteristic of the limiter

The amplified signal is transmitted to the blocks for calculating the spectral characteristics and PAPR values. The ratio of the PAPR of the received signal at the output of the modulator (PAPR_{orig}) to the PAPR of the signal received at the output of the power amplifier (PAPR_{red}) is defined as the value of PAPR reduction (PR):

$$\mathrm{PR} = \mathrm{PAPR}_{orig}\,(\mathrm{dB}) - \mathrm{PAPR}_{red}\ (\mathrm{dB}).$$

The signals received after passing through the MBC channel will be processed in the demodulator. Then they will be transferred to the following blocks to calculate the values of the probability of error and the delivery time.

In modeling, a broadcast protocol is used, which is usual for the meteor burst communication [18]. This approach provides the ability to transmit information without establishing a communication session between stations. Note that in this option there is no way to control the delivery of information. With the comparative inefficiency of this protocol, it is used for broadcast transmission of small amounts of information. The advantage of the broadcast protocol is that the receiving station does not waste energy, working only on reception, in contrast to the half-duplex protocol [18]. When using a broadcast protocol, the desired value is the delivery time T_d. We assume that the delivery time T_d is the time from the start of emission of a physical layer packet to the reception of this packet without errors. After receiving the packet, the received delivery time is recorded without errors and the simulation is restarted. Further, the obtained values are averaged. In the modeling process, at least 1000 iterations were used per one fixed set of modeling parameters. In an MBC system with transmission speed of 100 kbps, the FM signal used for transmission occupies a frequency band of 100 kHz. The choice of parameters of the OFDM signals is determined in such a way that the occupied frequency band without amplitude limitation does not exceed the 100 kHz band. The number of frequency subcarriers of the OFDM signal is selected equal to $N = 8$. The final simulation parameters:

- Duration of one OFDM symbol: $T_{OFDM\,symbol} = 10^{-4}$ s = 0.1 ms.
- Number of symbols in one packet: $N_{OFDM\,symbol} = 10^2$, $2 \cdot 10^2$ or $3 \cdot 10^2$ symbols.
- Duration of the package $T = 0.01$, 0.02 or 0.03 s (depend on the value $N_{OFDM\,symbol}$).
- Modulation scheme: QPSK.
- Sampling frequency $F_s = 10$ MHz.

The assumption of simulation is the absence of clock and symbol synchronization, that is, at the reception the moments of the beginning of OFDM symbols and the phase are always known. Also, no checksum was used to verify the integrity of the packet. Under these conditions, we are able to obtain a “lower estimate”, that is, minimum estimates of the delivery time.

Results

In the beginning, we consider the case without amplitude limitation (PR = 0 dB). In this case, the signal has a great PAPR value (more than 11 dB). The transmission of such signals will be inefficient in terms of the use of power amplifiers due to their low energy conversion efficiency. As you can see, it is required to reduce the PAPR.

Fig. 6 and 7 show examples of normalized instantaneous power $p(t)/\max(p(t)$ and energy spectrum of signals $|S(f)|^2/|S(0)|^2$ for various values of the PR value. Note that the energy spectrum of a random sequence of OFDM signals with a rectangular envelope has a low decay rate of out-of-band emissions (OOBE). With a stricter restriction on PAPR, significant additional distortions begin to appear due to an increase in the mutual influence of signals from neighboring frequency subcarriers due to an increase in OOBE of OFDM signals. At the level of energy spectrum up to 10 dB, we received a bandwidth of 100 kHz.

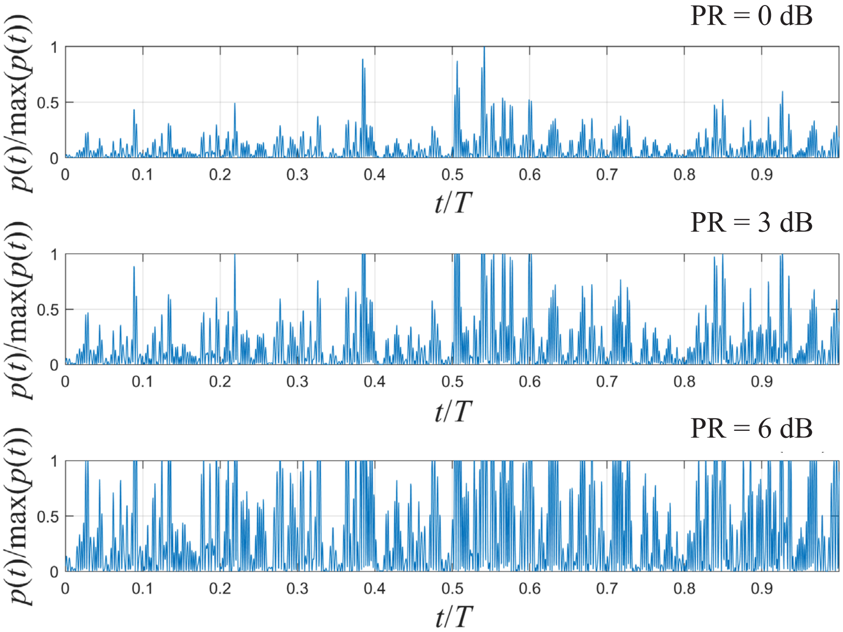


Fig. 6. Examples of normalized instantaneous signal power at different values of PR

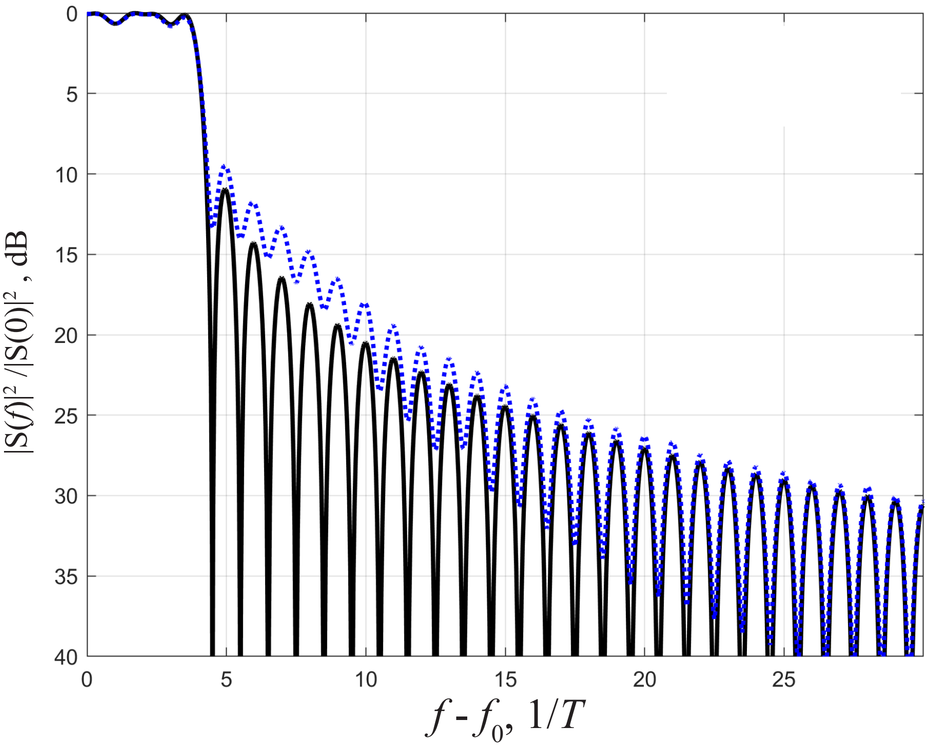


Fig. 7. Energy spectrum of signals at different values of PR
(—) – PR = 0 dB; (···) – PR = 6 dB

Fig. 8 *a* shows the dependence of the delivery time T_d on the level of the PR values for various values of the packet duration *T*. From the analysis of these graphs, we can conclude that with a decrease in PAPR by 5 dB, the delivery time is almost the same for OFDM signals. The resulting peak factor value is reduced to 5 dB, which significantly increases the overall efficiency of amplifiers and transmitting modules.

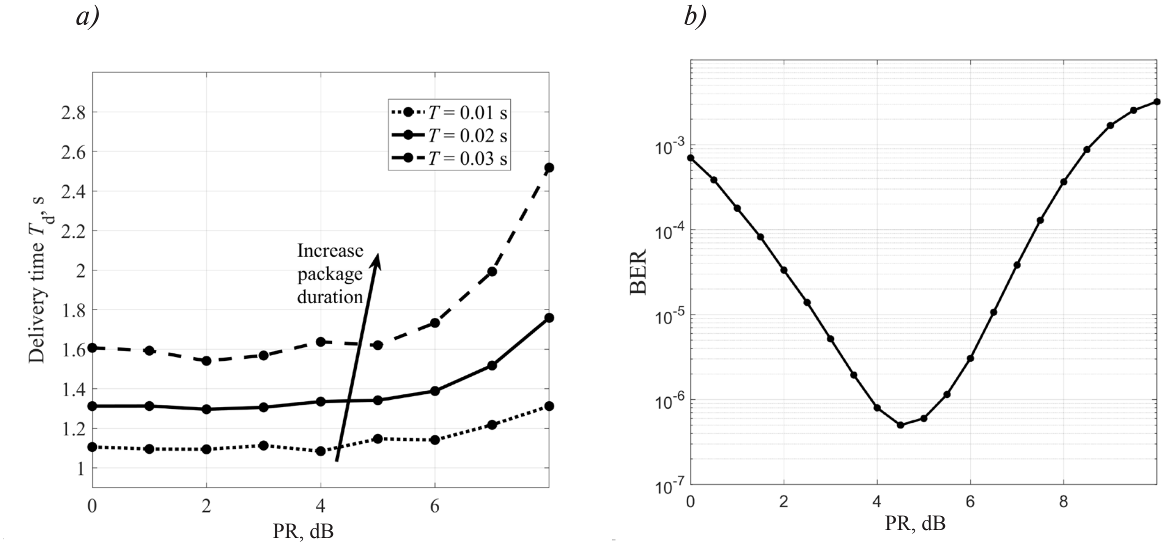


Fig. 8. Dependence of delivery time on the level of PR for various values of the duration of the packet *T*

In addition, the results also allow us to choose the appropriate values of the duration of the packet *T*, which helps to increase the speed of information transmission. For example, when increasing the duration of the packet *T* by 3 times, the delivery time increases from 1.1 s to 1.6 s, that is, it increases only 1.5 times. At the same time, the amount of transmitted information increases by 2 times. In [19], we have already found that, from the point of view of minimizing the probability of errors, the effective PR value for OFDM signals is 4.5–5 dB. This result is shown in Fig. 8 *b*. In the considered scheme of the transmitting module there is both a limiter and an amplifier. When limiting the value of the PAPR, the average power of the emitted oscillations increases, which reduces the error probability value for fixed signal-to-noise ratio values. However, with a further increase in the level of limitation, the signals from neighboring subcarrier frequencies start mutually affecting each other, which ultimately leads to an increase in the probability of error. These effects are also manifested in an increase in the message delivery time in the MBC channel (Fig. 8 *a*) when trying to limit the PAPR by more than 5 dB.

Combining the results of these researches, we can conclude that the combined use of amplifiers and limiters on the transmitting module can reduce the PAPR of the emitted oscillations. For the considered parameters of multi-frequency signals, the optimal PAPR reduction values to ensure a given delivery time in the MBC channel are about 5 dB.

Conclusion

In the process of simulation, the authors obtained estimates of the lower confine of the delivery time for the MBC channel in the case of the use of multi-frequency signals. We show it is possible to reduce the PAPR of the emitted oscillations by 5-6 dB for OFDM signals with 8 subcarrier frequencies. Also, with an increase in the packet size (the number of symbols transmitted in the OFDM packet) by 3 times, the delivery time increases only by 1.5 times. As a result, the total amount of information transmitted will increase by 2 times.

Further research will be associated with the addition of clock and symbol synchronization mechanisms, checksums, etc. to get closer to real conditions. In addition, we intend to test options for using smoothed optimal envelopes of multi-frequency signals.

REFERENCES

1. **Sugar G.R.** Radio propagation by reflection from meteor trails. *Proceedings of the IEEE*, 1964, Vol. 52, No. 2, Pp. 116–136.

2. **Glover I.A.** Meteor burst communications. I. Meteor burst propagation. *Electronics & Communication Engineering Journal*, 1991, Vol. 3, No. 4, Pp. 185–192.

3. **Weitzen J.A., Ralston W.T.** Meteor scatter – an overview. *IEEE Transactions on Antennas and Propagation*, 1988, Vol. 36, No. 12, Pp. 1813–1819.

4. **Yavuz D.** Meteor burst communications. *IEEE Communications Magazine*, 1990, Vol. 28, No. 9, Pp. 40–48.

5. **Mukumoto K., Fukuda A.** Development of software modem for the MBC experiment in Antarctica. *Proceedings of the 5th Annual International Conference on Global Research and Education*, The Alexandru Ioan Cuza University in Iasi, Romania, Sept. 2006.

6. **Forsyth P.A., Vogan E.L., Hansen D.R., Hines C.O.** The principles of JANET - a meteor-burst communication system. *Proceedings of the IRE*, 1957, Vol. 45, No. 12, Pp. 1642–1657.

7. **Bartholome P., Vogt I.** COMET – a new meteor-burst system incorporating ARQ and diversity reception. *IEEE Transactions on Communication Technology*, 1968, Vol. 16, No. 2, Pp. 268–278.

8. **Schaefer G.L.** SNOTEL: The world's first and largest data collection system using meteor burst technology. *Proc. of the Symposium on the Hydrological Basis for Water Resources Management*, Beijing, 1990, Pp. 229–238.

9. **Weitzen J., Larsen J., Mawrey R.** Design of a meteor scatter communication-network for vehicle tracking. *Vehicular Technology Society 42nd VTS Conference – Frontiers of Technology: From Pioneers to the 21st Century*, 1992, Vol. 1, 2, Pp. 75–78.

10. **Makarov S.B., Rashich A.V.** Formirovanie i priem spektral'no-jeffektivnyh signalov s OFDM [The generation and reception of spectrally effective signals with OFDM]. *St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control System*s, 2011, Vol. 138, No. 6-2, Pp. 19–26. (rus)

11. **Makarov S.B., Rashich A.V.** Snizheniye pik-faktora signalov s ortogonal'nym chastotnym uplotneniyem [Reducing the peak factor of signals with orthogonal frequency multiplexing]. *State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*, 2008, No. 2(55), Pp. 79–84. (rus)

12. **Antonov E.O., Rashich A.V., Fadeev D.K., Tan N.** Reduced complexity tone reservation peak-to-average power ratio reduction algorithm for SEFDM signals. *Proceedings of the 39th International Conference on Telecommunications and Signal Processing*, 2016, Pp. 445–448.

13. **Rashich A., Kislitsyn A., Fadeev D., Nguyen N.T.** FFT-based trellis receiver for SEFDM signals. *Proceedings of 2016 IEEE Global Communications Conference*, 2017, Pp. 1–6.

14. **He Q., Schmeink A.** Comparison and evaluation between FBMC and OFDM systems. *Proceedings of the 19th International ITG Workshop on Smart Antennas*, 2015.

15. **Eshleman V.R.** Theory of radio reflections from electron-ion clouds. *Transactions of the IRE Professional Group on Antennas and Propagation*, 1955, Vol. 3, No. 1, Pp. 32–39.

16. **Weitzen J.A., Bourque S., Horton M., Bench P.M., Baily A.D., Ostergaard J.C.** Distributions of underdense meteor trail amplitudes and application to meteor scatter communication-system design. *Proceedings of the 9th Annual International Phoenix Conference on Computers and Communications*, 1990, Pp. 237–240.

17. **Weitzen J.A., Hibshoosh E., Schilling D.L.** Some observations on the distributions of amplitude and duration of underdense meteor trails and its application to design of meteor scatter protocols. *Military Communications Conference 1988. Conference record. 21st Century Military Communications – What's Possible?* IEEE, 1988, Vol. 2, Pp. 571–576.

18. **Miller S.L., Milstein L.B.** A comparison of protocols for a meteor-burst channel based on a time-varying channel model. *IEEE Transactions on Communications*, 1989, Vol. 37, No. 1, Pp. 18–30.

19. **Nguyen D.C., Zavjalov S.V., Ovsyannikova A.S.** The effectiveness of application of multi-frequency signals under conditions of amplitude limitation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, Pp. 681–687.

Received 07.01.2020.

СПИСОК ЛИТЕРАТУРЫ

1. **Sugar G.R.** Radio propagation by reflection from meteor trails // Proc. of the IEEE. 1964. Vol. 52. No. 2. Pp. 116–136.

2. **Glover I.A.** Meteor burst communications. I. Meteor burst propagation // Electronics & Communication Engineering J. 1991. Vol. 3. No. 4. Pp. 185–192.

3. **Weitzen J.A., Ralston W.T.** Meteor scatter – an overview // IEEE Transactions on Antennas and Propagation. 1988. Vol. 36. No. 12. Pp. 1813–1819.

4. **Yavuz D.** Meteor burst communications // IEEE Communications Magazine. 1990. Vol. 28. No. 9. Pp. 40–48.14

5. **Mukumoto K., Fukuda A.** Development of software modem for the MBC experiment in Antarctica // Proc. of the 5th Annual Internat. Conf. on Global Research and Education. The Alexandru Ioan Cuza University in Iasi, Romania, Sept. 2006.

6. **Forsyth P.A., Vogan E.L., Hansen D.R., Hines C.O.** The principles of JANET - a meteor-burst communication system // Proc. of the IRE. 1957. Vol. 45. No. 12. Pp. 1642–1657.

7. **Bartholome P., Vogt I.** COMET – a new meteor-burst system incorporating ARQ and diversity reception // IEEE Transactions on Communication Technology. 1968. Vol. 16. No. 2. Pp. 268–278.

8. **Schaefer G.L.** SNOTEL: The world's first and largest data collection system using meteor burst technology // Proc. of the Symp. on the Hydrological Basis for Water Resources Management. Beijing, 1990. Pp. 229–238.

9. **Weitzen J., Larsen J., Mawrey R.** Design of a meteor scatter communication-network for vehicle tracking // Vehicular Technology Society 42nd VTS Conf. – Frontiers of Technology: From Pioneers to the 21st Century. 1992. Vol. 1, 2. Pp. 75–78.

10. **Макаров С.Б., Рашич А.В.** Формирование и прием спектрально-эффективных сигналов с OFDM // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2011. № 6-2 (138). С. 19–26.

11. **Макаров С.Б., Рашич А.В.** Снижение пик-фактора сигналов с ортогональным частотным уплотнением // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2008. № 2 (55). С. 79–84.

12. **Antonov E.O., Rashich A.V., Fadeev D.K., Tan N.** Reduced complexity tone reservation peak-to-average power ratio reduction algorithm for SEFDM signals // Proc. of the 39th Internat. Conf. on Telecommunications and Signal Processing. 2016. Pp. 445–448.

13. **Rashich A., Kislitsyn A., Fadeev D., Nguyen N.T.** FFT-based trellis receiver for SEFDM signals // Proc. of the 2016 IEEE Global Communications Conf. 2017. Pp. 1–6.

14. **He Q., Schmeink A.** Comparison and evaluation between FBMC and OFDM systems // Proc. of the 19th Internat. ITG Workshop on Smart Antennas. 2015.

15. **Eshleman V.R.** Theory of radio reflections from electron-ion clouds // Transactions of the IRE Professional Group on Antennas and Propagation. 1955. Vol. 3. No. 1. Pp. 32–39.

16. **Weitzen J.A., Bourque S., Horton M., Bench P.M., Baily A.D., Ostergaard J.C.** Distributions of underdense meteor trail amplitudes and application to meteor scatter communication-system design // Proc. of the 9th Annual Internat. Phoenix Conf. on Computers and Communications. 1990. Pp. 237–240.

17. **Weitzen J.A., Hibshoosh E., Schilling D.L.** Some observations on the distributions of amplitude and duration of underdense meteor trails and its application to design of meteor scatter protocols //

Military Communications Conf. 1988. Conference record. 21st Century Military Communications - What's Possible? IEEE, 1988. Vol. 2. Pp. 571–576.

18. **Miller S.L., Milstein L.B.** A comparison of protocols for a meteor-burst channel based on a time-varying channel model // IEEE Transactions on Communications. 1989. Vol. 37. No. 1. Pp. 18–30.

19. **Nguyen D.C., Zavjalov S.V., Ovsyannikova A.S.** The effectiveness of application of multi-frequency signals under conditions of amplitude limitation // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2019. Pp. 681–687.

Статья поступила в редакцию 07.01.2020.

THE AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

NGUYEN Dac Cu
НГУЕН Дак Кы
E-mail: daccu91.spb@gmail.com

ZAVYALOV Sergey V.
ЗАВЬЯЛОВ Сергей Викторович
E-mail: zavyalov_sv@spbstu.ru

VOLVENKO Sergey V.
ВОЛВЕНКО Сергей Валентинович
E-mail: volk@cee.spbstu.ru

DOI: 10.18721/JCSTCS.13105
УДК 004.896

NEURAL NETWORK COMPENSATION OF DYNAMIC ERRORS IN A POSITION CONTROL SYSTEM OF A ROBOT MANIPULATOR

E.N. Rostova[1], *N.V. Rostov*[2], *Z. Yan*[2]

[1]St. Petersburg Institute for Informatics and Automation of RAS,
St. Petersburg, Russian Federation;
[2]Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

This paper considers a position control system of a 3-link robot manipulator. The authors reviewed publications on nonlinear compensation of dynamic errors with the use of neural networks in robot manipulator control systems. The paper presents mathematical description of the control system with the compensation of nonlinear dynamics of the robot mechanism. We carried out training of multivariable neural network compensators of dynamic errors occurring because of the influence of inertia, Coriolis and gravity load torques. We developed computer models of the control system with different types of neural network compensators which are included in feedforward and feedback of the system and carried out a computer simulation of control systems with prototype and different kinds of neural network compensators. We also conducted a comparative analysis of dynamic errors in the system with different combinations of neural network compensators and gave recommendations on program realization of neural network compensators for real robot manipulator position control systems.

Keywords: robot manipulator, position control system, neural networks, nonlinear multivariable compensators, simulation, dynamic errors.

Citation: Rostova E.N., Rostov N.V., Yan Z. Neural network compensation of dynamic errors in a position control system of a robot manipulator. Computing, Telecommunications and Control, 2020, Vol. 13, No. 1, Pp. 53-64. DOI: 10.18721/JCSTCS.13105

НЕЙРОСЕТЕВАЯ КОМПЕНСАЦИЯ ДИНАМИЧЕСКИХ ОШИБОК В ПОЗИЦИОННОЙ СИСТЕМЕ УПРАВЛЕНИЯ МАНИПУЛЯЦИОННЫМ РОБОТОМ

Е.Н. Ростова[1], *Н.В. Ростов*[2], *Ч. Янь*[2]

[1]Санкт-Петербургский институт информатики и автоматизации РАН,
Санкт-Петербург, Российская Федерация;
[2]Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Рассмотрена система позиционного управления трёхзвенным манипуляционным роботом. Проведен обзор публикаций по вопросам нелинейной компенсации динамических ошибок в системах программного управления манипуляционными роботами с использованием нейронных сетей. Представлено математическое описание системы управления с нелинейной компенсацией динамики исполнительного механизма робота. Проведено обучение многомерных нелинейных нейросетевых компенсаторов динамических ошибок, обусловленных действием инерционных, кориолисовых и гравитационных нагрузочных моментов в приводах робота. Разработаны компьютерные модели системы управления с различными вариантами многомерных нейросетевых компенсаторов, включаемые в разомкнутый канал управления и замкнутый контур системы управления. Проведено компьютерное моделирование систем с прототипными и нейросетевыми компенсаторами рассматриваемых типов. Сделан сравнительный анализ динамических ошибок в системах управления с различными комбинациями нейросетевых компенсаторов. Даны рекомендации по программной реализации нейросетевых компенсаторов для реальных позиционных систем управления манипуляционными роботами.

Ключевые слова: робот-манипулятор, позиционная система управления, нейронные сети, нелинейные многомерные компенсаторы, моделирование, динамические ошибки.

Ссылка при цитировании: Ростова Е.Н., Ростов Н.В., Янь Ч. Нейросетевая компенсация динамических ошибок в позиционной системе управления манипуляционным роботом // Информатика, телекоммуникации и управление. 2020. Т. 13. № 1. С. 53-64. DOI: 10.18721/JCSTCS.13105

Introduction

These days the new problems of robot manipulator (RM) motion control are given significant consideration. One of such problems is increasing the precision characteristics of robots executing various technological operations. When the motion of robot links along program trajectories is performed, substantial dynamic errors occur mainly because of considerable nonlinearity of RM dynamics and because of robot links' interaction [1–4].

The design of RM motion control systems (CS) requires a very precise description of system dynamics and knowing technical parameters of a RM. Existing control methods, including the control based on the calculation of torques in link joints and other control methods based on the direct solving of the inverse dynamics problem, require the use of precise RM dynamics models [5–7]. However, mathematical models of real control systems contain some parameter uncertainties. Therefore, RM control requires taking into account not only the links' interaction but also the influence of various load torques.

Lately, algorithms based on fuzzy logic and neural networks have been implemented for the purpose of increasing robot precision characteristics [8–10]. In particular, neural network calculators can be used in robot drives as nonlinear quasi time-optimal regulators [11]. Neural network interpolators of robot link trajectories can be implemented instead of traditional spline interpolators in computer numerical control (CNC) systems [12, 13]. Also, neural network can be used for solving inverse kinematics of a robot [14, 15].

In this paper, we explore robot manipulator motion control systems with neural network (NN) compensators of dynamic errors occurring when robot drives execute program trajectories of the robot links. These errors are caused by torque loads resulting from nonlinear robot dynamics. Similar investigations were carried out in [16, 17].

The purpose of this work is the choice of structures and the training of multivariable nonlinear NN compensators of dynamic errors occurring on program gripper trajectories of a 3-link robot that operates in an angular coordinate system.

The main problems we approach in this paper are listed below:

1) mathematical modeling of control systems with prototype compensators of dynamic errors;
2) training of NN compensators which occur on given program trajectories of RM links;
3) computer simulation of control systems with prototype and NN compensators;
4) evaluation and comparative analysis of dynamic errors in the systems under consideration.

Mathematical models of prototype nonlinear compensators

Multivariable compensators of dynamic errors are described by expressions corresponding to the nonlinear dynamic model of a robot, which is presented in the form of Lagrange equations [1–4]:

$$A(q)\ddot{q}+B(q,\dot{q})\dot{q}+C(q)=Q_d-Q_L, \tag{1}$$

where (q , $\dot{q}$, $\ddot{q}$) – N×1 vectors of position, speed and acceleration coordinates of the robot links; N – the number of robot links.

The main loads of the robot drives are the elements of the vectors in the left part of the equations (1), where: $Q_{iner} = A(q)\ddot{q}$ – N×1 vector of inertia torques caused by accelerated links' motion; $A(q)$ – N×N matrix of the robot's kinetic energy; $Q_{cor} = B(q,\dot{q})\dot{q}$ – N×1 vector of Coriolis and centrifugal torques; $B(q,\dot{q})$ – N×N matrix; $Q_{grav} = C(q)$ – N×1 vector of gravity torques and other potential forces.

In the right part of equations (1) Q_d – N×1 vector of torques generated by the robot drives; Q_L – N×1 vector of additional loads occurring in the drives because of frictions in the joints and the influence of external forces on the robot gripper:

$$Q_L = J^T(q)F_L, \tag{2}$$

where $J^T(q)$ – N×N matrix transposed to the robot's Jacobi matrix; $F_L = (F_x, F_y, F_z, M_x, M_y, M_z)^T$ – 6×1 vector of coordinates of an external force and an external torque that influence the gripper.

In motion control systems, trajectories of robot links are calculated by solving the inverse kinematics problem in the base points of a robot gripper trajectory. After that, the link trajectories are interpolated using spline or other polynomials. Thus, the torques which the drives must overcome can be calculated with the program values of position, speed and acceleration vectors of the links (q_p , $\dot{q}_p$, $\ddot{q}_p$):

$$Q_{iner}=A(q_p)\ddot{q}_p,\quad Q_{cor}=B(q_p,\dot{q}_p)\dot{q}_p,\quad Q_{grav}=C(q_p), \tag{3}$$

$$Q_{ff}=Q_{iner}+Q_{cor}+Q_{grav}. \tag{4}$$

Expressions (3) and (4) can be used directly for compensation of dynamic errors in CS with torque drives.

Fig. 1 shows the functional diagram of a system with compensators included in a feedforward (FF) circuit of the system, where PCU is a program control unit.

In case there are additional torques of loads Q_L, there might be significant errors in a system with FF compensation. These errors can be reduced by adding a linear compensator into the position PID regulator:

$$U_{pid}=(K_p e+K_i\int edt-K_d\dot{q})+K_{com}\dot{q}_p, \tag{5}$$

where $K_{com} = diag\{K_{(com,i)}\}$ – the diagonal matrix of linear compensator coefficients.

Nonlinear compensators can be also included into a closed loop of a system with position and speed feedback (FB), in a combination with a PID regulator or a more complex nonlinear regulator. In this case the FB compensator uses signals of real links' positions and speeds that are measured by sensors:

$$Q_{iner}=A(q)U_{pid},\quad Q_{cor}=B(q,\dot{q})\dot{q},\quad Q_{grav}=C(q), \tag{6}$$

$$Q_{fb}=Q_{iner}+Q_{cor}+Q_{grav}, \tag{7}$$

where the output vector of the PID regulator U_{pid} is interpreted as a vector of program accelerations.

A functional diagram of a system with a nonlinear FB compensator is shown in Fig. 2.

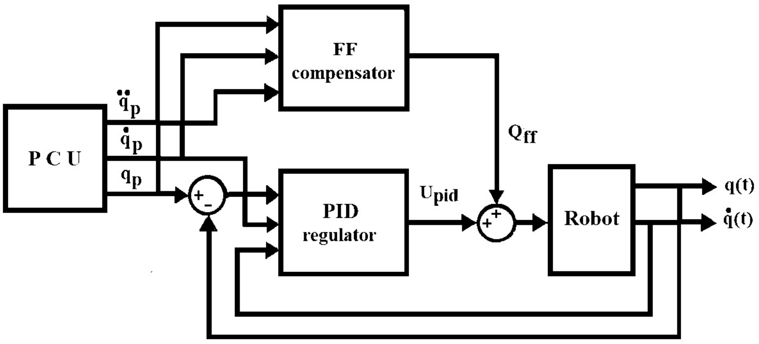


Fig. 1. A control system with feedforward compensation

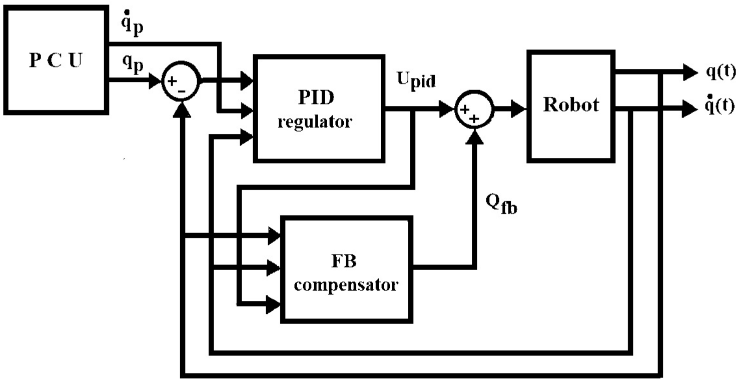


Fig. 2. A control system with feedback compensation

Multivariable compensators (6) and (7) perform the functional linearization of the robot's nonlinear dynamics described by equations (1), and thus, help make the robot drives work more accurately.

Simulation of a system with prototype nonlinear compensators

Certain Simulink models were built for the investigation of control systems with prototype FF and FB compensators. Parameters of a 3-link RM model corresponded to the parameters of the first three links of the 6-link PUMA-560 robot. Frictions in the robot joints were not taken into consideration [18–20].

A helical (spiral-shaped) trajectory was chosen as a standard program trajectory of a robot gripper. Fig. 3 illustrates the animation of the robot gripper movement along a helical trajectory made with functions from Robotics Toolbox 10.3.1 [20].

The dynamic errors on the gripper trajectory were calculated using the expression below:

$$E = \sqrt{(X_p - X_r)^2 + (Y_p - Y_r)^2 + (Z_p - Z_r)^2}, \tag{8}$$

where $(X_p, Y_p, Z_p)^T$ – the program coordinates of gripper positions; $(X_r, Y_r, Z_r)^T$ – the real coordinates of gripper positions.

Fig. 4 shows dynamic errors obtained in the process of simulation for the two types of systems: 1 – for the system with a PID regulator only, without a linear compensator (dash line); 2 – for the system with a PID regulator and an additional linear FF compensator (solid line).

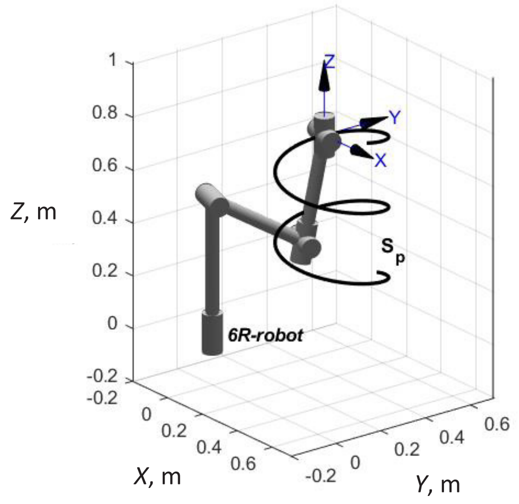


Fig. 3. The helical program trajectory of the robot gripper

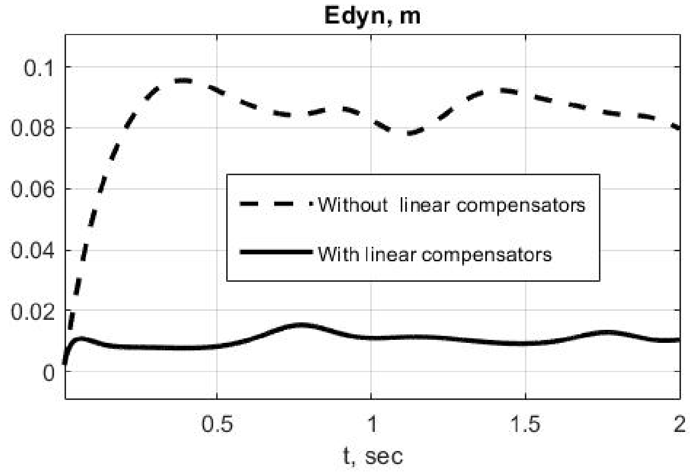


Fig. 4. Errors in the systems without nonlinear compensators

Fig. 5 shows dynamic errors in the systems with a linear compensator and additional nonlinear FF compensators of different types:

1 – FF compensation of gravity torques;

2 – FF compensation of Coriolis and centrifugal torques;

3 – FF compensation of inertia torques;

4 – full FF compensation of load torques.

Fig. 6 shows dynamic errors in the systems with a linear compensator and different combinations of nonlinear FF compensators:

1 – FF compensation of gravity torques only;

2 – FF compensation of gravity and Coriolis torques;

3 – FF compensation of gravity, Coriolis and inertia torques.

From what we can see in Fig. 5 and 6, the minimal dynamic errors are achieved with the use of a linear compensator together with all types of nonlinear FF compensators.

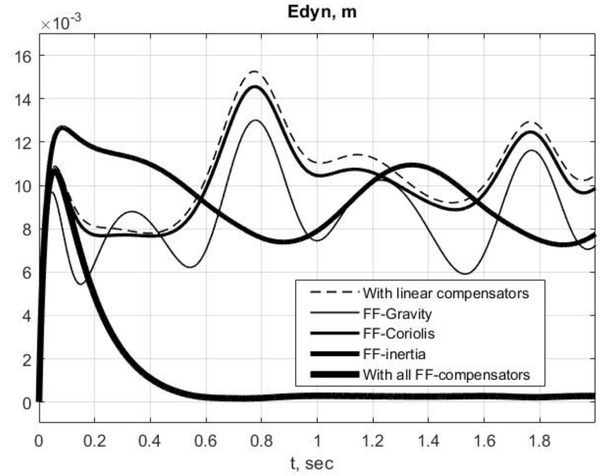


Fig. 5. Errors in the systems with nonlinear FF compensators

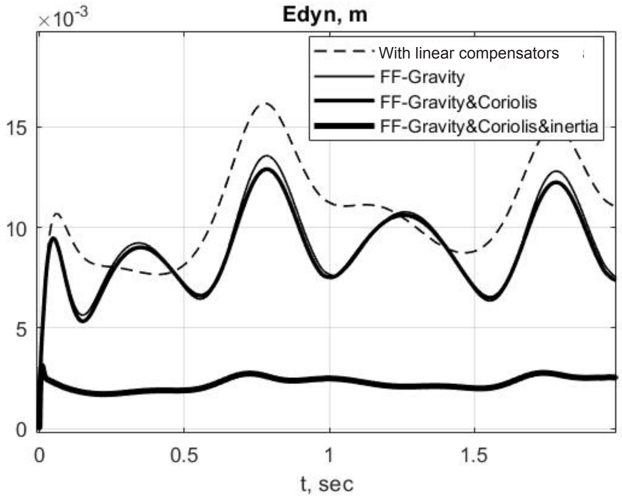


Fig. 6. Errors in the systems with a combination of FF compensators

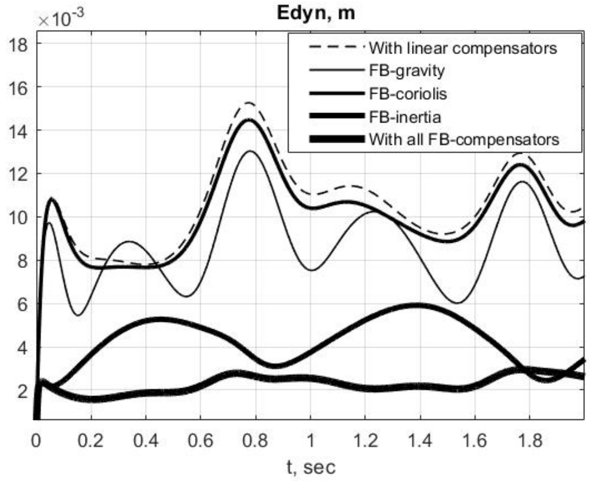


Fig. 7. Errors in the systems with nonlinear FB compensators

Fig. 7 illustrates dynamic errors in the systems with a linear compensator and additional nonlinear FB compensators of different types:

1 – FB compensation of gravity torques;

2 – FB compensation of Coriolis and centrifugal torques;

3 – FB compensation of inertia torques;

4 – full FB compensation of load torques.

Training of nonlinear neural network compensators

Neural network compensators can be realized with the use of different types of neural networks [21–26]. In this work, models of prototype compensators were used for the training of NN compensators. Program links' coordinates $(q_p, \dot{q}_p, \ddot{q}_p)$ on a given helical gripper trajectory were chosen as input data for the training of NN compensators. Calculated load torques (Q_{iner}, Q_{cor}, Q_{grav}) were used as output data. Radial basis neural networks were chosen as NN compensators. The training was performed using functions from Neural Network Toolbox in MATLAB.

Simulation of a system with nonlinear neural network compensators

Simulink models shown in Fig. 8, 9 and 10 were developed for the analysis of processes in systems with neural network FF and FB compensators.

Dynamic errors obtained during the simulation of the system shown in Fig. 8 are close to the errors shown in Fig. 5.

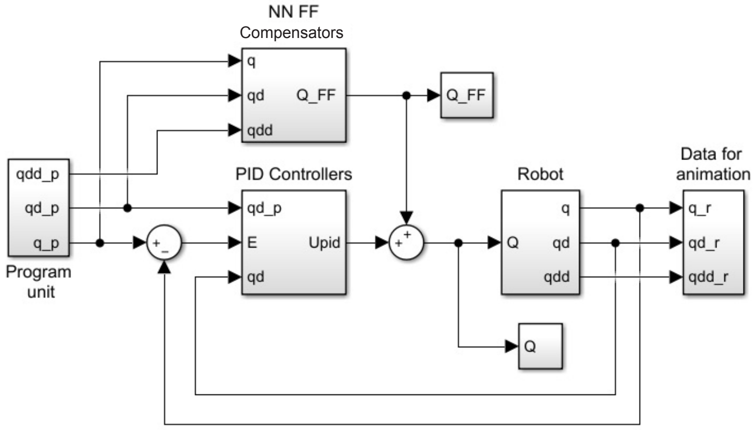


Fig. 8. The model of a control system with neural network FF compensators

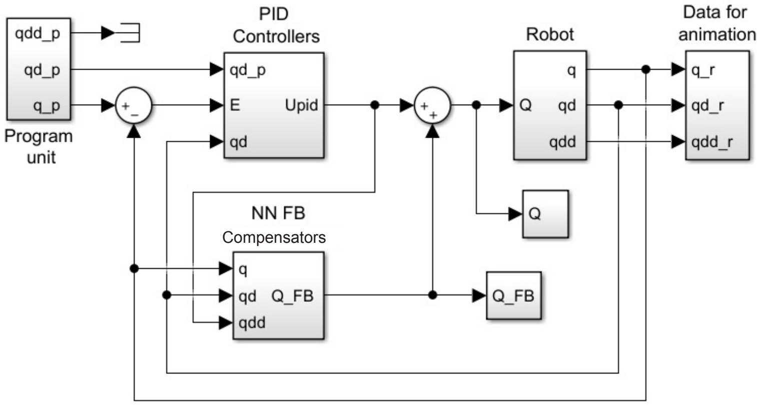


Fig. 9. The model of a control system with neural network FB compensators

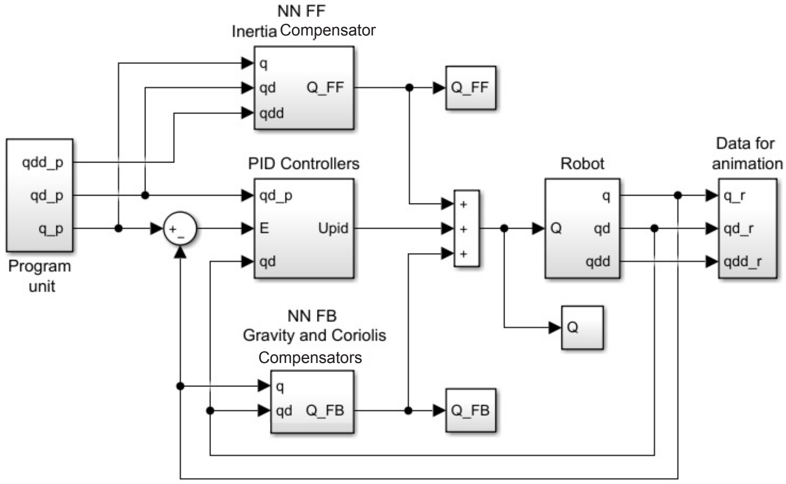


Fig. 10. The model of a control system with two neural network FB compensators and one FF inertia compensator

Dynamic errors obtained during the simulation of the system shown in Fig. 9 are different from the errors shown in Fig. 7.

The model in Fig. 10 includes two neural network FB compensators (of gravity and Coriolis torques) and one neural network FF inertia torques compensator.

Fig. 11 shows dynamic errors in three systems with a linear compensator and different combinations of neural network FF compensators:

1 – full neural network FF compensation;

2 – neural network FB compensation of gravity and Coriolis torques;

3 – neural network FF compensation of inertia torques and FB compensation of gravity and Coriolis torques.

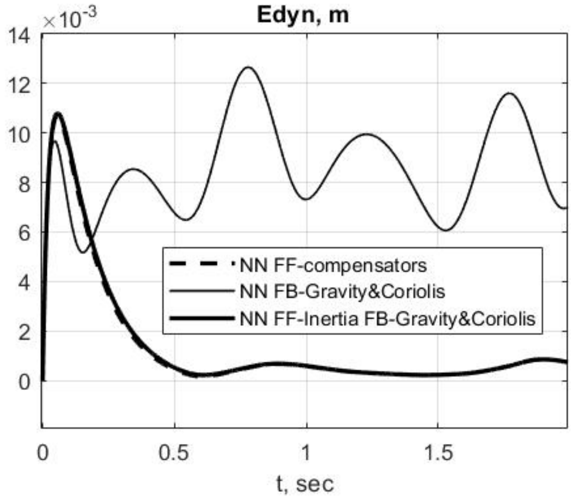


Fig.11. Errors in the systems with neural network FF and FB compensators

The evaluation of errors in Fig. 11 leads to the conclusion that the minimal dynamic errors are achieved in the system with two neural network FB compensators of gravity and Coriolis torques and one additional neural network FF compensator of inertia torques, just as in the system with full neural network FF compensation.

Conclusion

The dynamical precision of traditional CNC systems is achieved with combined control of the robot drives that includes regulation with feedbacks and one-variable linear compensation of errors caused by derivatives of drives' inputs. But in case of robot manipulators decreasing dynamical errors requires nonlinear multivariable compensation of the influences of inertia, Coriolis, centrifugal and gravity torques in the robot link drives when moving along complex program trajectories.

In this work, the authors explore robot manipulator control systems when the robot operates in determined conditions. In this case construction parameters and the robot's dynamic model are known. This allows to calculate torques of load that occur on given trajectories of the robot gripper beforehand in order to compensate for the influence of torques during the operation.

NN compensators of dynamic errors are an alternative to prototype nonlinear compensators, which have specific structures and parameters described with complex nonlinear equations. Training of NN compensators should be performed in the autonomous (offline) mode based on the results of computer simulation of prototype compensators for certain program trajectories of the robot gripper. The minimal time of training is achieved when using NN compensators with RBF neurons.

NN compensators are effective for increasing the robot's precision in modern CNC systems. Also, the functional linearization of complex and nonlinear multivariable robot dynamics that they give ensures more stable work of robot link drives. In systems without nonlinear compensation, the robustness of high-precision drives can be obtained only with the use of adaptive regulators, which are rather complicated to implement.

The necessary condition for NN compensators use is that the computer model used for their training corresponds to the real robot dynamics. In this case compensators trained for certain program trajectories of the robot links will not cause instability of closed-loop drive systems. In situations with some parameter uncertainty of real robot dynamics, NN compensation will be incomplete, and the drives robustness can be improved with adaptive regulators. However, the problems of synthesis of such kind of regulators require special consideration.

The estimates of dynamic errors on the gripper helical trajectory obtained by computer simulation show that errors in systems with multivariable RBF NN compensators and standard PID position regulators are much smaller compared to systems with linear compensators. The results of simulation show that the highest precision is achieved in the control system with two NN FB compensators and one NN FF inertia compensator (Fig. 10). Such structure can be recommended for implementation in real systems.

For NN approximation of different torques of load with necessary precision, the number of RBF neurons and training time depend on the type of trajectories, operation time and singularity points on the continuous-path gripper trajectory. Program realization of nonlinear NN compensators requires high-speed microprocessors, such as digital signal processors (DSP). However, the problems of implementation of NN compensators require more detailed consideration that is beyond the scope of this paper.

REFERENCES

1. **Zenkevich S.L., Yuschenko A.S.** *The basis of robot manipulator control*. Moscow: MGTU named by N.E. Bauman Publ., 2004. (rus)

2. **Jurevich E.I.** *The basis of robotics*. St. Petersburg: BHV–Petersburg Publ., 2018. (rus)

3. **Ignatova E.I., Lopota A.V., Rostov N.V.** *Robot motion control systems: computer-aided design*. St. Petersburg: Polytechnic University Publ., 2014. 301 p. (rus)

4. **Lewis F.L., Abdallah C.T., Dawson D.M.** *Control of robot manipulators*. New York: Macmillan, 1993.

5. **Santibañez V., Camarillo K., Moreno-Valenzuela J., Campa R.** A practical PID regulator with bounded torques for robot manipulator. *Int. J. Control Autom. Syst.*, 2010, Vol. 8, No. 3, Pp. 544–555.

6. **Lee G., Wand Cheng F.T.** Robust control of manipulators using the computed torque plus H∞ compensation method. *IEEE Proc. Control Theory Appl.*, 1996, Vol. 143, No. 1, Pp. 64–72.

7. **Islam S., Liu X.P.** Robust sliding mode control for robot manipulators. *Proceedings of IEEE Trans Ind Electron.*, 2011, Pp. 2444–2453.

8. **Zabihifar S., Markazi A., Yuschenko A.** Control of a two link manipulator with the use of fuzzy logic sliding mode control. *Vestnik MGTU named by N.E.Bauman*, 2015, No. 6 (105), Pp. 30–45. (rus). DOI: 10.18698/0236-3933-2015-6-30-45

9. **Lewis F.L., Jaganathan S., Yesildirek A.** *Neural network control of robot manipulators and nonlinear systems*. London: Taylor & Francis, 1999.

10. **Kim Y.H., Lewis F.L.** Neural network output feedback control of robot manipulators. *Proceedings of IEEE Trans Robot Autom.*, 1999, Pp. 301–309.

11. **Ishmuratov V.N., Rostov N.V.** Computer training of neural network quasi time-optimal digital regulators. *Proceedings of International Scientific and Practical Conference*, 2007, Pp. 134–136.

12. **Terpukhov S.Yu., Rostov N.V.** Neural network interpolation of program trajectories of links of a robot manipulator. *Proceedings of International Scientific and Practical Conference*, 2010, Pp. 70–72.

13. **Yan Z., Rostov N.V.** Error analysis of neural network interpolators of program trajectories of links of a robot manipulator. *Proceedings of ComCon-2018*, 2018, Pp. 114–119.

14. **Bhattacharjee T., Bhattacharjee A.** A study of neural network based inverse kinematics solution for a planar three joint robot with obstacle avoidance. *Assam University Journal of Science & Technology: Physical Sciences and Technology*, 2010, Vol. 5, No. 2, Pp. 1–7.

15. **Chiddarwar S.S., Babu N.R.** Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach. *Engineering Applications of Artificial Intelligence*, 2010, Vol. 23, Pp. 1083–1092.

16. **Duc M.N, Trong T.N.** Neural network structures for identification of nonlinear dynamic robotic manipulator. *Proceedings of IEEE International Conference on Mechatronics and Automation*, 2014, Pp. 1575–1580.

17. **Singh H.P., Sukavanam N., Panwar V.** Neural network based compensator for robustness to the robot manipulators with uncertainties. *International Conference on Mech. Electr. Technol. Proc.*, 2010, Pp. 444–448.

18. **Dixon W.E., Moses D., Walker L.D., Dawson D.M.** A Simulink-based robotic toolkit for simulation and control of the PUMA 560 robot manipulator. *Intelligent Robots and Systems. Proceedings of IEEE/RSJ International Conference*, 2001, Vol. A, Pp. 2202–2207.

19. **Corke P.I.** *Robotics, vision and control. fundamental algorithms in MATLAB*. Berlin: Springer-Verlag Berlin Heidelberg, 2017. DOI: 10.1007/978-3-319-54413-7

20. **Corke P.I.** Robotics toolbox for MATLAB. Release 10. 2017. Available: *http://www.petercorke.com/robot*

21. **Seshagiri S., Khalil H.K.** Output feedback control of nonlinear systems using RBF neural networks. *Proceedings of IEEE Trans Neural Networks*, 2000, Pp. 69–79.

22. **Tai N.T., Ahn K.K.** A RBF neural network sliding mode controller for SMA actuator. *Proceedings of International J. Control Autom. Syst.*, 2010, Pp. 1296–1305.

23. **Kumar N., Panwar V., Borm J.H., Chai J.** Enhancing precision performance of trajectory tracking controller for robot manipulators using RBFNN and adaptive bound. *Appl. Math. Comput.*, 2014, Vol. 231, Pp. 320–328. DOI: 10.1016/j.amc.2013.12.082

24. **Yan Z., Rostova E.N., Rostov N.V.** Neural network compensation of dynamic errors in a robot manipulator programmed control system. *Lecture Notes in Networks and Systems. Cyber-Physical Systems and Control*, 2020, Vol 95, Pp. 554–563. DOI: 10.1007/978-3-030-34983-7_543

25. **Farzam T., Nafise F.R.** Robust control of a 3-DOF parallel cable robot using an adaptive neuro-fuzzy inference system. *Artificial Intelligence and Robotics (IRANOPEN)*, 2017, Pp. 97–101. DOI: 10.1109/RIOS.2017.7956450

26. **Zabihifar S., Yuschenko A.** Hybrid force/position control of a collaborative parallel robot using adaptive neural network. *Lecture Notes in Computer Science. Interactive Collaborative Robotics*, 2018, Vol. 11097, Pp. 280–290. DOI: 10.1007/978-3-319-99582-3_29

Received 10.01.2020.

СПИСОК ЛИТЕРАТУРЫ

1. **Зенкевич С.Л., Ющенко А.С.** Основы управления манипуляционными роботами. М.: Изд-во МГТУ им. Н.Э. Баумана, 2004.

2. **Юревич Е.И.** Основы робототехники. СПб.: BHV–Петербург, 2018.

3. **Игнатова Е.И., Лопота А.В., Ростов Н.В.** Системы управления движением роботов. Компьютерное проектирование. СПб.: Изд-во Политехн. ун-та, 2014. 301 с.

4. **Lewis F.L., Abdallah C.T., Dawson D.M.** Control of robot manipulators. New York: Macmillan, 1993.

5. **Santibañez V., Camarillo K., Moreno-Valenzuela J., Campa R.** A practical PID regulator with bounded torques for robot manipulator // Int. J. Control Autom. Syst. 2010. Vol. 8. No. 3. Pp. 544–555.

6. **Lee G., Wand Cheng F.T.** Robust control of manipulators using the computed torque plus H∞ compensation method // IEEE Proc. Control Theory Appl. 1996. Vol. 143. No. 1. Pp. 64–72.

7. **Islam S., Liu X.P.** Robust sliding mode control for robot manipulators // Proc. of IEEE Trans Ind Electron. 2011. Pp. 2444–2453.

8. **Забихифар С., Маркази А., Ющенко А.** Управление двухзвенным манипулятором с использованием нечеткого управления скользящего типа // Вестник МГТУ им. Н.Э. Баумана. 2015. № 6 (105). С. 30–45. DOI: 10.18698/0236-3933-2015-6-30-45.

9. **Lewis F.L., Jaganathan S., Yesildirek A.** Neural network control of robot manipulators and nonlinear systems. London: Taylor & Francis, 1999.

10. **Kim Y.H., Lewis F.L.** Neural network output feedback control of robot manipulators // Proc. of IEEE Trans Robot Autom. 1999. Pp. 301–309.

11. **Ishmuratov V.N., Rostov N.V.** Computer training of neural network quasi time-optimal digital regulators // Proc. of Internat. Scientific and Practical Conf. 2007. Pp. 134–136.

12. **Terpukhov S.Yu., Rostov N.V.** Neural network interpolation of program trajectories of links of a robot manipulator // Proc. of Internat. Scientific and Practical Conf. 2010. Pp. 70–72.

13. **Yan Z., Rostov N.V.** Error analysis of neural network interpolators of program trajectories of links of a robot manipulator // Proc. of ComCon-2018. 2018. Pp. 114–119.

14. **Bhattacharjee T., Bhattacharjee A.** A study of neural network based inverse kinematics solution for a planar three joint robot with obstacle avoidance // Assam University J. of Science & Technology: Physical Sciences and Technology. 2010. Vol. 5. No. 2. Pp. 1–7.

15. **Chiddarwar S.S., Babu N.R.** Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach // Engineering Applications of Artificial Intelligence. 2010. Vol. 23. Pp. 1083–1092.

16. **Duc M.N, Trong T.N.** Neural network structures for identification of nonlinear dynamic robotic manipulator // Proceedings of IEEE Internat. Conf. on Mechatronics and Automation. 2014. Pp. 1575–1580.

17. **Singh H.P., Sukavanam N., Panwar V.** Neural network based compensator for robustness to the robot manipulators with uncertainties // Internat. Conf. on Mech. Electr. Technol. Proc. 2010. Pp. 444–448.

18. **Dixon W.E., Moses D., Walker L.D., Dawson D.M.** A Simulink-based robotic toolkit for simulation and control of the PUMA 560 robot manipulator // Intelligent Robots and Systems. Proc. of IEEE/RSJ Internat. Conf. 2001. Vol. A. Pp. 2202–2207.

19. **Corke P.I.** Robotics, vision and control. fundamental algorithms in MATLAB. Berlin: Springer-Verlag Berlin Heidelberg, 2017. DOI: 10.1007/978-3-319-54413-7

20. **Corke P.I.** Robotics toolbox for MATLAB. Release 10. 2017 // URL: http://www.petercorke.com/robot

21. **Seshagiri S., Khalil H.K.** Output feedback control of nonlinear systems using RBF neural networks // Proc. of IEEE Trans Neural Networks. 2000. Pp. 69–79.

22. **Tai N.T., Ahn K.K.** A RBF neural network sliding mode controller for SMA actuator // Proc. of Internat. J. Control Autom. Syst. 2010. Pp. 1296–1305.

23. **Kumar N., Panwar V., Borm J.H., Chai J.** Enhancing precision performance of trajectory tracking controller for robot manipulators using RBFNN and adaptive bound // Appl. Math. Comput. 2014. Vol. 231. Pp. 320–328. DOI: 10.1016/j.amc.2013.12.082

24. **Yan Z., Rostova E.N., Rostov N.V.** Neural network compensation of dynamic errors in a robot manipulator programmed control system // Lecture Notes in Networks and Systems. Cyber-Physical Systems and Control. 2020. Vol. 95. Pp. 554–563. DOI: 10.1007/978-3-030-34983-7_543

25. **Farzam T., Nafise F.R.** Robust control of a 3-DOF parallel cable robot using an adaptive neuro-fuzzy inference system // Artificial Intelligence and Robotics (IRANOPEN). 2017. Pp. 97–101. DOI:10.1109/RIOS.2017.7956450

26. **Zabihifar S., Yuschenko A.** Hybrid force/position control of a collaborative parallel robot using adaptive neural network // Lecture Notes in Computer Science. Interactive Collaborative Robotics. 2018. Vol. 11097. Pp. 280–290. DOI: 10.1007/978-3-319-99582-3_29

Статья поступила в редакцию 10.01.2020.

THE AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Rostova Ekaterina N.
Ростова Екатерина Николаевна
E-mail: rostovae@mail.ru

Rostov Nikolay V.
Ростов Николай Васильевич
E-mail: rostovnv@mail.ru

Yan Zhengjie
Янь Чжэнцзе
E-mail: yanzhengjie1019@gmail.com

DOI: 10.18721/JCSTCS.13106
УДК 004.85, 004.62, 378.147

THE ASSESSMENT OF THE RESULTS OF A MASSIVE OPEN ONLINE COURSE USING DATA MINING METHODS

S.A. Nesterov, E.M. Smolina

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

The paper presents the results of a grade reports analysis for five sessions of a massive open online course "Data Management" at openedu.ru. For our research, we used clustering and classification in the R programming environment. Clustering showed the presence of four groups of course participants with nearly similar course results. These clusters were similar for all five sessions of the course we analyzed. We also showed it is possible to predict whether a participant completes the course or drops out, based on the test results during the first half of the course. The course lecturers can use the results to plan measures for keeping the students in the course. Also, such a type of analysis helps to understand the reasons why the students drop out of the course. The lecturers can take them into account to modify the course structure and learning content. This new knowledge about the course participants can be used during the next course sessions. We expect that for other courses with a similar structure, the clustering results will be also similar. The approach to predict whether a student drops out or completes the course used in the paper is applicable for other courses as well.

Keywords: MOOC, learning management systems, Data Mining, clustering, classification.

Citation: Nesterov S.A., Smolina E.M. The assessment of the results of a massive open online course using Data Mining methods. Computing, Telecommunications and Control, 2020, Vol. 13, No. 1, Pp. 65-78. DOI: 10.18721/JCSTCS.13106

ОЦЕНКА РЕЗУЛЬТАТОВ ПРОВЕДЕНИЯ МАССОВОГО ОТКРЫТОГО ОНЛАЙН КУРСА С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

С.А. Нестеров, Е.М. Смолина

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Представлены результаты исследования отчетов об оценках пяти сессий дистанционного массового онлайн курса «Управление данными» на портале Открытого образования openedu.ru. В ходе исследования решались задачи кластеризации и классификации. Исследование проводилось с использованием языка программирования R. Кластеризация показала наличие четырех групп слушателей курса, сходных по результатам прохождения курса. Характеристики этих групп близки для всех рассмотренных сессий курса. Показано, что на основании результатов прохождения тестов в первой половине курса можно с высокой точностью предсказать, бросит ли слушатель изучение курса или будет учиться до его окончания. Полученные результаты можно использовать при планировании мероприятий с целью удержания слушателей на курсе. Подобный анализ помогает понять причины, по которым студенты бросают изучение курса, и учесть это при корректировке его структуры.

Ключевые слова: МООК, системы дистанционного обучения, интеллектуальный анализ данных, кластеризация, классификация.

Ссылка при цитировании: Нестеров С.А., Смолина Е.М. Оценка результатов проведения массового открытого онлайн курса с использованием методов интеллектуального анализа данных // Информатика, телекоммуникации и управление. 2020. Т. 13. № 1. С. 65-78. DOI: 10.18721/JCSTCS.13106

Introduction

As e-learning is currently developing at an accelerating pace, massive open online courses (MOOCs) providing simultaneous education to thousands of students are becoming more and more popular. At the same time, this type of e-learning has a common drawback of only a small percent of students completing the courses [1, 2].

Information systems of distance online education accumulate large amounts of data on course participants and the results of their studies. We can analyze these data and give recommendations to increase the quality of the courses. For example, some e-learning platforms such as Moodle provide inbuilt tools for statistics analyses of completed tests [3–5]. In other instances, the data in the form of students' reports or event log files get downloaded from the education monitoring system and analyzed by additional external means, including those using Data Mining algorithms.

Nowadays, an emerging direction of data analysis called Educational Data Mining is developing new methods of data analysis in education [6–8]. This type of analysis helps to identify characteristic groups of students [8, 9], predict students' course results [10–12], as well whether or not the participants will finish the course [11, 13], determine the most difficult tasks [4] and general behavioral patterns the participants display [1, 14].

This paper analyses grade reports of the participants of "Data management" course on an open education platform openedu.ru [15]. We engaged in clustering (to identify characteristic groups of course participants) and binary classification (to predict whether or not the participants will finish the course) tasks.

The purpose of this research is to enhance MOOC efficiency using Data Mining results accessible for lecturers from standard reports. The research tasks include:

- obtaining new knowledge on the course participants based on the data found in the course reports;
- analyzing possibilities of applying the obtained results to the future sessions of the course, as well as other courses at openedu.ru and similar platforms.

Preliminary data analysis

As we mentioned before, the research presents an analysis of grade reports of the participants of "Data management" MOOC at openedu.ru. The course takes up one semester and starts twice a year, in fall and spring. We analyzed five sessions of the course: the fall of 2016, the spring and fall of 2017 and 2018.

The course lasts for 16 weeks, each week presenting a new topic to study. The course content includes video lectures, lecture notes, workshops, weekly tests (Homework in reports). The students have a Midterm Exam after the 8th week and a Final Exam after the 16th week. The final course grade consists of the homework results (an average for all weeks), the midterm and final exams combined. During the first session of the course these grades are summed up with weighting factors: 0.3 for the homework results, 0.35 for the midterm exam and 0.35 for the final exam. The subsequent sessions were subject to some changes: the final exam required only participant's identity authentication and its contribution to the final grade increased significantly. The new weighting factors now were 0.2, 0.2 and 0.6 respectively.

We chose R programming language for analysis since it presents a variety of tools for statistical data processing, visualization and machine learning [16–18]. We downloaded delimited text files of reports from openedu.ru and imported them into R, where they were displayed as data frames. The absent grade data were replaced by zero. Thus, we assumed the difference between the case a student failed to complete the task and a case a student failed the test scoring zero was insignificant. Table 1 presents a fragment of a grade report after the above-mentioned replacement.

Table 1

Fragment of the report under consideration

id	Grade	Homework 1	...	Homework 16	Midterm Exam	Final Exam
217782	0.0	0.0	...	0.0	0.0	0.0
181077	0.05	0.8	...	0.0	0.0	0.0
180553	0.94	1.0	...	1.0	0.933	0.9

For each session of the course, the authors calculated a percentage of the participants enrolling in the course, but failing to complete any tasks. We used the final course grade for the purpose (defined as Grade at openedu.ru). If a participant's Grade amounts to zero, the participant never commenced performing the tasks. It is noteworthy that the Grade is presented with values between 0 and 1 rounded to two decimal places in the report, so a roundoff error may occur: the students completing only a small part of a task of only one week may fall into the group of the students who never commenced to perform the tasks. Table 2 shows the results.

Table 2 demonstrates that the largest number of participants enrolled in the first session of the course. This was probably due to the interest in the new course at the website. In the fall of 2018 the enrollment deadlines were prolonged significantly which also had a positive impact on the number of the applicants. For five sessions of the course, only 31, 32, 23, 19 and 23% of the participants (respectively) commenced to perform the tasks.

Table 2

Number of students failing to perform any task

The course starting period	Enrolled in the course	Commenced performing the tasks	Failed to perform any task, %
class 2016	2547	798	69
class 2017 (spring)	1572	499	68
class 2017 (fall)	1823	427	77
class 2018 (spring)	1504	279	81
class 2018 (fall)	2346	529	77

A bar chart in Fig. 1 demonstrates a number of students of the first course session who commenced performing the tasks. The x-axis shows the number of the task, while the y-axis presents the number of students. 798 students performed the tasks of the first week, then a steep drop occurs with only 435 proceeding to perform the second week tasks. In the course of the subsequent weeks, the number of active participants continues to fall gradually, however we observe a slight increase in numbers on the midterm exam week.

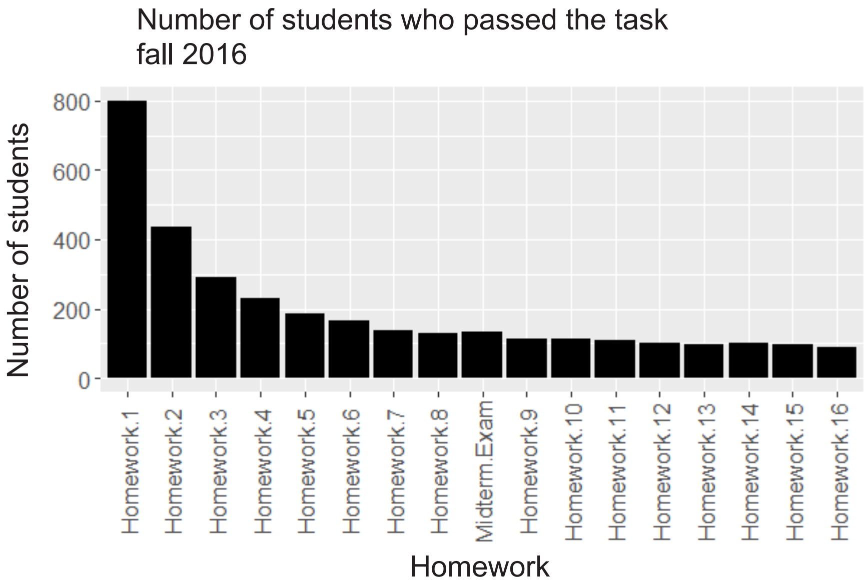


Fig. 1. Number of students completing the tasks (fall 2016)

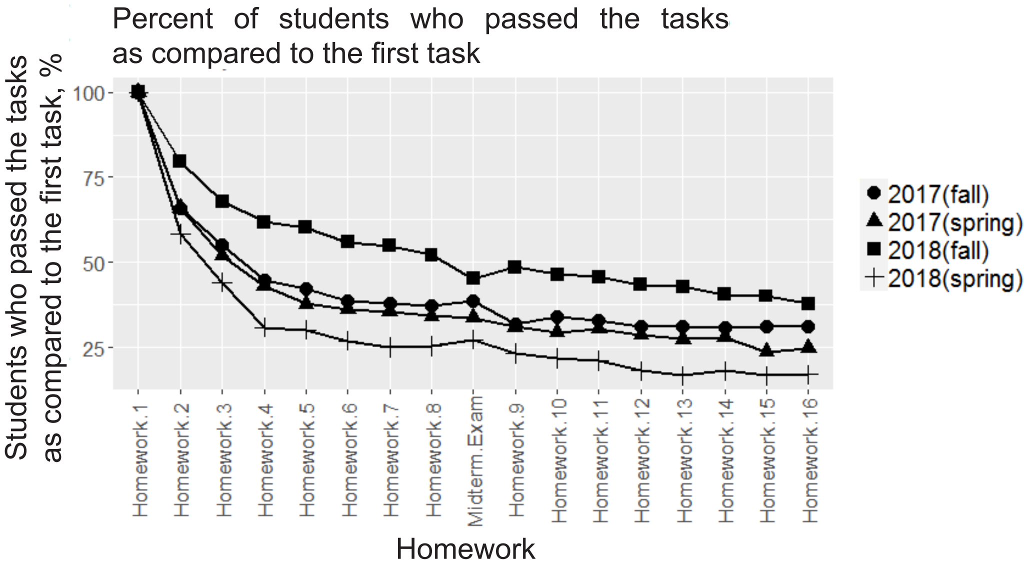


Fig. 2. Percentage of the students performing the task

The spring start of 2017 had 499 participants commencing to perform the first task. Similarly to the first start, the second week exhibited a dramatic fall down to 329 participants. The following weeks saw a smooth drop in the number of participants. The subsequent sessions showed a similar dependency. We should note that the midterm exam marks the moment after which the number of active participants remains virtually the same.

Fig. 2 presents graphs for all five course sessions under consideration with the number of students taking tests weekly displayed as percentage of the students performing the first week tasks. Figure shows that all the course sessions exhibit a sharp fall in the number of active participants after the first week. This can be due to a number of reasons, for example:

- the content turned out to be uninspiring or too difficult;
- the participants realized the course required a lot of time;
- the participants had no intention to study and wanted to "have a look" only;
- the course is too extensive and long.

The next task of this research was to find groups of participants similar in their level of activity in the course.

Clustering

Based on the reports of the progress in the course we can divide the participants into groups according to their results. This is a clustering task we can describe in the following manner. Let *I* be a multitude of the course participants:

$$I = \{ i_1 , i_2 , \ldots, i_n \},$$

where each of the participants possesses a set of attributes:

$$i_j = \{ x_1 , x_2 , \ldots, x_m \},$$

x_k is an independent variable which can assume values from a certain multitude (usually numerical values).

We need to form a multitude of clusters

$$C = \{ c_1 , c_2 , \ldots, c_g \},$$

where each cluster includes similar objects from *I* multitude of participants under consideration:

$$c_h = \{ i_j , i_p \mid i_j , i_p \in I , d (i_j , i_p) < \sigma \}.$$

Here $d(i_j , i_p)$ is a measure of closeness between objects (distance), σ is a boundary value of distance to include objects in one cluster [19].

The task was to divide the multitude of the course participants into groups with similar attributes (clusters) and compare the clusters obtained for different course sessions.

We based the choice of clusters on the research of the dependency of the change of the total mean squared deviation (squared distance between each element and the cluster center) on the number of clusters [16, 20]. This approach uses a number of clusters corresponding to the elbow of the curve (the so-called "elbow method"). According to this criterion each course session had a value of 4 clusters. We used the *k*-means clustering algorithm. Our clustering algorithm did not take the results of the participants who failed to commence performing any task at all into account.

To define each cluster, the authors constructed graphs describing average cluster grades for the weekly tests and exams. Fig. 3 demonstrates clustering for the course session starting in the fall of 2018. The graphs for the other course sessions are visually very similar [9].

Thus, in the course of the study we defined four major groups of active course participants present in every session under consideration:

1. students with a stable performance (Fig. 3, cluster 3);
2. students with high performance in the first half of the course, low performance in the second half who still completed the course (cluster 4);
3. students who attended the first two weeks with occasional attendance in the following weeks (cluster 2);
4. students with high performance in the first weeks and low further *attendance* who dropped out of the course after the midterm (cluster 1).

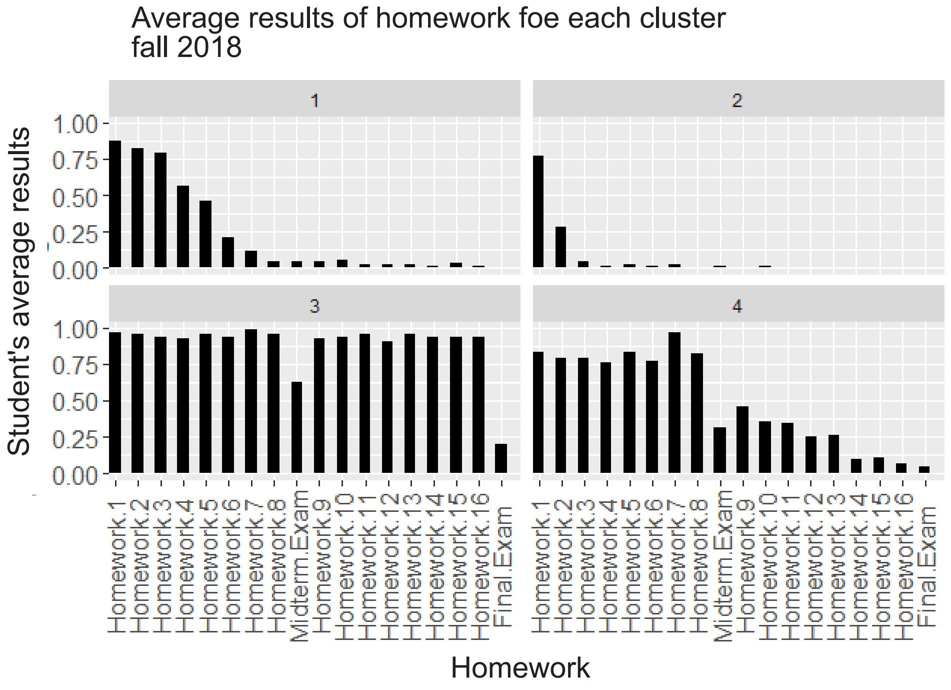


Fig. 3. Average cluster grades for the course session of the fall of 2018

Classification

The most interesting and essential analysis task is obtaining a predicative model which uses the results of the previous courses to benefit the upcoming sessions. In this research, we used the results of students participating in 4 sessions of the course (2016, the spring and fall of 2017, spring of 2018) as a training sample, while applying the classification model obtained to the fifth course session (the fall of 2018) to predict the completion rate for the course. The purpose of this prediction was to determine whether a participant drops out or continues to study in the course until the end. Thus, we reduced the task to a binary classification.

We could not use the final exam results as the target attribute, because the exam with remote identity authentication and proctoring requires payment at openedu.ru, so only a small percentage of participants engages in it. Therefore, to form the target attribute (hereinafter referred to as *targetAttr*) we used grade values for the homework of the last (15th and 16th) weeks. If a student has a grade exceeding zero for a test of the 15th or 16th week, *targetAttr* acquires a value of 1 (completed the course), otherwise 0 (dropped out). Then we had to determine the weeks most suitable for predicting. For a pooled sample of the participants of the first four course sessions, we calculated a number of *n* week participants as well as their percentage in relation to the number of students who managed to complete the course (Table 3).

Table 3 shows results up to the midterm. As we can see, the majority of the participants who reached the midterm continued their studies. Thus, we decided it makes sense to predict “whether a participant finishes the course or drops out” before the midterm. Then we trained the classification models using the data of the first 4–7 and 8 weeks of the course.

Using an R tool, *sample()*, we randomly formed a training dataset (with 75 % of the initial data) and a test dataset (including the rest 25 %). It is important to note, that the training dataset included approximately 20 % of the *targetAttr* values equalled 1, while the rest 80 % equalled 0, thus making the dataset unbalanced. Table 4 displays the results of solving the classification task using three methods: *k*-nearest neighbors, Naive Bayes and decision trees. We used R packages class, naivebayes and rpart, respectively.

Table 3

Number of *n* week participants

Homework	Number of *n* week participants	Percentage of the students who completed the course, %
1	1869	22
2	1199	35
3	901	46
4	719	58
5	638	66
6	585	72
7	544	77
8	528	80
Midterm Exam	543	77
Completed the course	423	

Table 4

Classification results (unbalanced sample)

Algorithm	Week	Characteristics			
		accuracy	*precision*	*recall*	*f*1
k-nearest neighbors	4	0.79	0.75	0.73	0.74
	5	0.8	0.73	0.77	0.75
	6	0.81	0.75	0.77	0.76
	7	0.81	0.75	0.83	0.79
	8	0.82	0.76	0.81	0.79
Naive Bayes	4	0.79	0.68	0.9	0.78
	5	0.82	0.71	0.94	0.81
	6	0.82	0.71	0.94	0.81
	7	0.84	0.73	0.95	0.82
	8	0.85	0.75	0.95	0.84
Decision trees	4	0.79	0.74	0.73	0.73
	5	0.79	0.7	0.82	0.75
	6	0.83	0.78	0.79	0.78
	7	0.81	0.74	0.86	0.8
	8	0.84	0.77	0.87	0.82

The quality characteristics of the classification model presented in Table 4 are defined in the following way [20]. Let's assume that after the training dataset the binary classifier showed:

- TP – a number of true positive predictions;
- TN – a number of true negative predictions;
- FP – a number of false positive predictions;
- FN – a number of false negative predictions.

Then we can calculate the characteristics as:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}};$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}};$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}};$$

$$f1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

To improve the classification results we balanced the training dataset using random undersampling [16]. The obtained sample had 40 % of the *targetAttr* values equalling 1. Table 5 contains the classification results.

Table 5

Classification results (balanced sample)

Algorithm	Week	Characteristics			
		accuracy	*precision*	*recall*	*f*1
k-nearest neighbors	4	0.79	0.67	0.9	0.77
	5	0.77	0.66	0.88	0.75
	6	0.81	0.7	0.89	0.78
	7	0.82	0.7	0.93	0.8
	8	0.86	0.76	0.94	0.84
Naive Bayes	4	0.78	0.67	0.9	0.77
	5	0.81	0.7	0.94	0.8
	6	0.82	0.71	0.94	0.81
	7	0.84	0.73	0.96	0.83
	8	0.86	0.75	0.96	0.84
Decision trees	4	0.78	0.66	0.89	0.76
	5	0.79	0.68	0.88	0.76
	6	0.81	0.7	0.91	0.79
	7	0.83	0.71	0.95	0.82
	8	0.85	0.75	0.93	0.83

Tables 4 and 5 show that the balanced sample does not significantly improve the results. Moreover, *precision* characteristic demonstrates a rise of false responses, meaning the share of true positive objects is reduced. Thus, we cannot deem the balanced training feasible for the current dataset.

Combining the classification and clustering results poses a more interesting task. Fig. 3 shows results of dividing the participants into characteristic groups (clusters) for the fall of 2018. The groups displayed in the Figure and named in the list below it include the following numbers of students: 201 in group 1; 84 in group 2; 98 in group 3; 146 in group 4.

For the prediction purposes we chose models based on the prediction results for 8 weeks of learning as the values of the generalized metric $f1$ (see Table 4) are the highest specifically by the midterm. Table 6 shows confusion matrices for each group.

Table 6

Confusion matrices for each group

Method	Group															
	1				2				3				4			
k-nearest neighbors		fact				fact				fact				fact		
	pred		0	1	pred		0	1	pred		0	1	pred		0	1
		0	0	32		0	29	7		0	98	0		0	141	2
		1	11	158		1	40	8		1	0	0		1	2	1
Naive Bayes		fact				fact				fact				fact		
	pred		0	1	pred		0	1	pred		0	1	pred		0	1
		0	0	1		0	14	7		0	98	0		0	143	3
		1	11	189		1	55	8		1	0	0		1	0	0
Decision trees		fact				fact				fact				fact		
	pred		0	1	pred		0	1	pred		0	1	pred		0	1
		0	0	16		0	29	9		0	98	0		0	139	2
		1	11	174		1	40	6		1	0	0		1	4	1

The classifier demonstrates acceptable precision for the first group. That is probably connected with the fact the group includes the students with the highest grades on a permanent basis: 190 out of 201 students completed the course. Interestingly enough, responses of each algorithm to the same dataset were different. The k-nearest neighbors determined a smaller share of positive objects than it should have. The Naive Bayes classifier, on the contrary, defined more values as positive falsely. The decision trees algorithm demonstrated the best prediction results for the first group: almost all the responses were correct.

We can see, that any classifier formed using any of the chosen algorithms is underperforming in terms of predictions for the second group showing a high rate of false positive results. 84 students of the group displayed high academic performance in the first half of the session, but then it dropped: only 15 participants completed the course. At the same time, the models predict that more than a half of the students would complete it. By the end of the 8th week most of the students had sufficiently high grades. We can attribute the classification errors to this reason.

The results for the third group turned out to be the most accurate. This is probably due to the fact the cluster includes the students who dropped out exclusively.

While predicting the reply to the question of the fourth group students dropping out the k-nearest neighbors algorithm committed no errors. The Naive Bayes algorithm ignored the values equaling 1, and the decision trees predicted more positive results than there were in reality. This cluster included 146 students, 3 of them completed the course.

Analysis of the results

While analyzing the obtained results it is important to note that the grade reports are the main source of information on students' progress for the course lecturer at openedu.ru. The platform currently fails to provide any inbuilt analytical means for lecturers. For this reason, analyses of grade reports similar to the one described in this article may be vital for any course at the website, and possibly for other courses using the same Open edX platform. This feature distinguishes this system from LMS Moodle wide-spread in universities, as the latter one offers e-learning analysis tools [3–5].

We analyze the obtained results starting with the participants clustering. Firstly, solving this problem gave us a better understanding of the peculiarities of the course participants' behavior. Dividing the participants into 4 described groups was not obvious, this result can be useful for the subsequent course sessions. Secondly, the repetitive nature of the clustering results shows that small changes to the course procedures (similar to the prolongation of the enrollment deadlines in the fall of 2018 we previously described) had little impact. This may testify that all the sessions of the course are equal. If we choose to analyze the data of other courses, the number and characteristics of the clusters may be different, but the most important factor is the repetitive nature for different sessions, provided the course content was subject to no major changes.

In other cases, we might face a reverse problem: if the course content is significantly revised, we need to understand whether the participants exhibit different behavior. We can expect the number and characteristics of clusters into which the participants fall to change as well.

Let us compare the clusters obtained with the results of other researchers. There is a following classification of typical groups of MOOC participants [14, 21]:

- "Ghosts" – participants who enrolled in a course, but never accessed any course content, i.e. never actually participated in the course.
- "Observers" – participants who enrolled in a course, accessed course content (video, lecture notes), but ignore any tests or tasks.
- "Non-completers" – participants who use MOOC content as auxiliary in their studies or work. They have no intention to complete the course, so the majority of such students drops out.
- "Passive participants" – these participants access course content, watch the video lectures, take tests, communicate with other students and lecturers online, but ignore difficult tests or bigger projects.
- "Active Participants" – participants with a high motivation level working on any type of course content, participating in projects, actively communicating with other students and lecturers.

To identify these groups of participants, the analysis requires not only the grades, but also the information on their access to course content. The lecturers of openedu.ru have no direct access to this information. Nevertheless, we can assume that the first two groups consist of 70–80 % of the participants from Table 2 who enrolled in the course, but never performed any of the tasks. Previously, we described two clusters of the participants who were active in the first weeks, but dropped out at different stages of the course, thus corresponding to the "Non-completers" group. We can assume that the cluster of the participants who had stable performance throughout the course corresponds to the "Active Participant" group. The "Passive participants" apparently consists of the remaining cluster.

As for the prediction on the results of the participant's studies, in most cases the researchers use a different approach and initial data. For example, the predication mechanism uses information on the results of this particular student in other courses [10]. Or along with the grades the researchers engaged additional information, such as household income, the participant's sex, etc. [12], which is usually impossible to access for MOOCs. The papers also present statistics for the participants accessing the content (links referrals, time and mode of watching video, video paused, etc.) [13].

However, we should note that the approach to predicting whether a participant finishes the course or drops out based only on grades for the accomplished tasks presented in this paper showed that it can still be of interest. Lecturers can carry out this kind of analysis in a timely manner. At the same time, it is only applicable to the courses with strict deadlines for test paper admittance during the course. If the deadlines are absent or the main deadline is the date of the course ending, this approach is irrelevant.

Conclusion

This paper analyses grade reports of the participants of "Data management" course at an open education website openedu.ru. Grade reports are the kind of data lecturers use when running courses at the website. This MOOC system does not offer any tools to analyze students' progress in the courses yet.

As a result of clustering we identified 4 characteristic groups of participants. Moreover, the same clustering pattern persisted in all 5 sessions of the course analyzed.

We also demonstrated the grade reports obtained in the first half of the course are sufficient enough to predict whether students drop out or complete the course with high accuracy. To improve the prediction accuracy, we attempted to train the models using balanced samples. However, this approach did not result in any significant improvement of classification accuracy.

The classifiers show different accuracy for various groups of students. This allows to assume that using different algorithms for various groups of participants can benefit the prediction accuracy.

The lecturers can use the results to keep the participants in the course. For example, certain students may require some measures taken beforehand to provide incentives for them to remain in the course and complete the studies. These measures can include new content offers or task notifications sent to them via e-mail.

This kind of analysis can also help to indentify the reasons the students drop out of the course which can be taken into account to correct its structure. For instance, the lecturer can change difficult tasks and recommend additional content to certain groups of students. This can increase the number of participants who complete the course.

Thus, as a result of the research we obtained new knowledge on the course participants useful for the lecturers. They can take it into account while planning next MOOC sessions. The lecturers of openedu.ru and similar systems can apply this approach to conduct the analysis of grade reports for their courses. We can assume that, provided the courses have structures close to the one described in this paper, the results of identifying characteristic groups of participants may be similar. We can also expect the approach to predict whether a student drops out or completes the course used in the paper to be applicable for other courses as well.

REFERENCES

1. **Gelman B., Revelle M., Domeniconi C., Johri A., Veeramachaneni K.** Acting the same differently: A cross-course comparison of user behavior in MOOCs. *Proceedings of the 9th International Conference on Educational Data Mining*, EDM 2016, Pp. 376–381. Available: http://www.educationaldatamining.org/EDM2016/proceedings/paper_136.pdf (Accessed: 01.11.2019).

2. **Nesterov S.A., Smolina E.M.** Analysis of the results of distance learning in the format of the massive open online course. *Proceedings of XXII International Conference on Systems Analysis in Engineering and Control*, St. Petersburg: Politeh-Press, 2018, Vol. 2, Pp. 379–383. (rus)

3. **Protasova I.V., Tolstobrov A.P., Korzhik I.A.** Metodika analiza i povysheniya kachestva testov v sisteme elektronnogo obucheniya MOODLE [Method of test quality analysis and improvement in MOODLE e-learning system]. *Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Sistemnyy analiz i informatsionnyye tekhnologii*, 2014, No. 3, Pp. 61–72. (rus)

4. **Tolstobrov A.P., Protasova I.V., Korzhik I.A.** Sistema analiza statistiki testirovaniya kak sredstvo samootsenki prepodavatelem elektronnogo obrazovatelnogo resursa [The system of analysis of testing statistics as a means of self-assessment by a teacher of an electronic educational resource]. *Sovremennyye informatsionnyye tekhnologii i IT-obrazovaniye* [*Modern Information Technologies and IT-Education*], 2013, No. 9, Pp. 133–141. (rus)

5. **Nesterov S.A.** Analiz statistiki vypolneniya testovykh zadaniy v srede distantsionnogo obucheniya MOODLE [Analysis of quiz statistics in LMS MOODLE]. *Modern Information Technologies and IT-Education*, 2016, Vol. 12, No. 4, Pp. 62–67. (rus)

6. **Romero C., Ventura S.** Educational Data Mining: A review of the state of the art. *Journal IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews Archive*, 2010, Vol. 40, Issue 6, Pp. 601–618.

7. **Belonozhko P.P., Karpenko A.P., Khramov D.A.** Analiz obrazovatelnykh dannykh: napravleniya i perspektivy primeneniya [Analysis of educational data: directions and prospects of application]. *Journal Naukovedenie*, 2017, Vol. 9, No. 4. Available: http://naukovedenie.ru/PDF/15TVN417.pdf (Accessed: 01.11.2019). (rus)

8. **Algarni A.** Data Mining in education. *International Journal of Advanced Computer Science and Applications*, 2016. Vol. 7(6), Pp. 456–461. DOI: 10.14569/IJACSA.2016.070659

9. **Nesterov S.A., Smolina E.M.** Methods of Data Mining in analysis of the results of distance learning. *Proceedings of XXIII International Conference on Systems Analysis in Engineering and Control*, June 10–11, 2019. St. Petersburg: Politeh-Press, 2019, Vol. 3, Pp. 407–412. (rus)

10. **Sweeney M., Lester J., Rangwala H., Johri A.** Next-term student performance prediction: A recommender systems approach. *JEDM*, 2016, Vol. 8, Issue 1, Pp. 22–51.

11. **Villanueva A., Moreno L.G., Salinas M.J.** Data Mining techniques applied in educational environments: Literature review. *Digital Education Review*, 2018, No. 33, Pp. 235–266.

12. **Salal Y.K., Abdullaev S.M.** Using of Data Mining techniques to predict of student's performance in Industrial Institute of Al-Diwaniyah, Iraq. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control & Radioelectronics*, 2019, No. 19, Pp. 121–130. DOI: 10.14529/ctcr190111

13. **Yang D., Kraut R., Rose C.** Exploring the effect of student confusion in massive open online courses. *JEDM*, 2016, Vol. 8, Issue 1, Pp. 52–83.

14. **Tabaa Y., Medouri A.** LASyM: A learning analytics system for MOOCs. *International Journal of Advanced Computer Science and Applications*, 2013, Vol. 4, Issue 5, Pp. 113–119. DOI: 10.14569/IJACSA.2013.040516

15. **Andreyeva N.V., Nesterov S.A.** Data management: online course. Available: *https://openedu.ru/course/spbstu/DATAM/* (Accessed: 01.11.2019). (rus)

16. **Bruce A., Bruce P.** *Practical statistics for Data Scientists*. O'Reilly Media, 2017.

17. **Lantz B.** *Machine learning with R*. Packt Publishing, 2015.

18. **Wickham H., Grolemund G.** *R for Data Science: Import, tidy, transform, visualize and modeling data*. St. Petersburg: Alfa-kniga Publ., 2018. 592 p. (rus)

19. **Barsegyan A.A., Kupriyanov M.S., Stepanenko V.V., Kholod I.I.** *Metody i modeli analiza dannykh: OLAP i Data Mining* [*Methods and models of data analysis: OLAP and Data Mining*]. St. Petersburg: BHV–Petersburg Publ., 2004. 336 p. (rus)

20. **Gras J.** *Data Science. Nauka o dannykh s nulya* [*Data Science from Scratch*]. St. Petersburg: BHV–Petersburg Publ, 2017. 336 p. (rus)

21. **Hill P.** Emerging student patterns in MOOCs: A (Revised) graphical view. Available: *https://eliterate.us/emerging-student-patterns-in-moocs-a-revised-graphical-view/* (Accessed: 01.11.2019).

Received 10.11.2019.

СПИСОК ЛИТЕРАТУРЫ

1. **Gelman B., Revelle M., Domeniconi C., Johri A., Veeramachaneni K.** Acting the same differently: A cross-course comparison of user behavior in MOOCs // Proc. of the 9th Internat. Conf. on Educational Data Mining. 2016. Pp. 376–381 // URL: http://www.educationaldatamining.org/EDM2016/proceedings/paper_136.pdf (Дата обращения: 01.11.2019).

2. **Нестеров С.А., Смолина Е.М.** Анализ результатов дистанционного обучения в формате массового открытого онлайн-курса // Системный анализ в проектировании и управлении: сб. науч. тр. XXII Междунар. науч.-практ. конф. СПб.: Изд-во Политехн. ун-та, 2018. Ч. 2. С. 379–383.

3. **Протасова И.В., Толстобров А.П., Коржик И.А.** Методика анализа и повышения качества тестов в системе электронного обучения MOODLE // Вестник Воронежского государственного университета. Сер.: Системный анализ и информационные технологии. 2014. № 3. С. 61–72.

4. **Толстобров А.П., Протасова И.В., Коржик И.А.** Система анализа статистики тестирования как средство самооценки преподавателем электронного образовательного ресурса // Современные информационные технологии и ИТ-образование. 2013. № 9. С. 133–141.

5. **Нестеров С.А.** Анализ статистики выполнения тестовых заданий в среде дистанционного обучения MOODLE // Современные информационные технологии и ИТ-образование. 2016. Т. 12. № 4. С. 62–67.

6. **Romero C., Ventura S.** Educational Data Mining: A review of the state of the art // J. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews Archive. 2010. Vol. 40. Issue 6. Pp. 601–618.

7. **Белоножко П.П., Карпенко А.П., Храмов Д.А.** Анализ образовательных данных: направления и перспективы применения // Науковедение. 2017. Т. 9. № 4 // URL: http://naukovedenie.ru/PDF/15TVN417.pdf (Дата обращения: 01.11.2019).

8. **Algarni A.** Data Mining in education // Internat. J. of Advanced Computer Science and Applications. 2016. Vol. 7(6). Pp. 456–461. DOI: 10.14569/IJACSA.2016.070659

9. **Нестеров С.А., Смолина Е.М.** Методы интеллектуального анализа данных в задачах оценки результатов дистанционного обучения // Системный анализ в проектировании и управлении: сб. науч. тр. XXIII Междунар. науч.-практ. конф. СПб.: Политех-Пресс, 2019. Ч. 3. С. 407–412.

10. **Sweeney M., Lester J., Rangwala H., Johri A.** Next-term student performance prediction: A recommender systems approach // JEDM. 2016. Vol. 8. Issue 1. Pp. 22–51.

11. **Villanueva A., Moreno L.G., Salinas M.J.** Data Mining techniques applied in educational environments: Literature review // Digital Education Review. 2018. No. 33. Pp. 235–266.

12. **Salal Y.K., Abdullaev S.M.** Using of Data Mining techniques to predict of student's performance in Industrial Institute of Al-Diwaniyah, Iraq // Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control & Radioelectronics. 2019. No. 19. Pp. 121–130. DOI: 10.14529/ctcr190111

13. **Yang D., Kraut R., Rose C.** Exploring the effect of student confusion in massive open online courses // JEDM. 2016. Vol. 8. Issue 1. Pp. 52–83.

14. **Tabaa Y., Medouri A.** LASyM: A learning analytics system for MOOCs // Internat. J. of Advanced Computer Science and Applications. 2013. Vol. 4. Issue 5. Pp. 113–119. DOI: 10.14569/IJACSA.2013.040516

15. **Андреева Н.В., Нестеров С.А.** Управление данными: онлайн-курс // URL: https://openedu.ru/course/spbstu/DATAM/ (Дата обращения: 01.11.2019).

16. **Bruce A., Bruce P.** Practical statistics for Data Scientists. O'Reilly Media, 2017.

17. **Lantz B.** Machine learning with R. Packt Publishing, 2015.

18. **Уикем Х., Гроулмунд Г.** Язык R в задачах науки о данных: импорт, подготовка, обработка, визуализация и моделирование данных. Пер. с англ. СПб.: Изд-во «Альфа-книга», 2018. 592 с.

19. **Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И.** Методы и модели анализа данных: OLAP и Data Mining. СПб.: БХВ-Петербург, 2004. 336 с.

20. **Грас Дж.** Data Science. Наука о данных с нуля. СПб.: БХВ-Петербург, 2017. 336 с.

21. **Hill P.** Emerging student patterns in MOOCs: A (Revised) graphical view // URL: https://eliterate.us/emerging-student-patterns-in-moocs-a-revised-graphical-view/ (Дата обращения: 01.11.2019).

Статья поступила в редакцию 10.11.2019.

THE AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Nesterov Sergei A.
Нестеров Сергей Александрович
E-mail: nesterov@saiu.ftk.spbstu.ru

Smolina Elena M.
Смолина Елена Михайловна
E-mail: smolensk9595@mail.ru

ИНФОРМАТИКА, ТЕЛЕКОММУНИКАЦИИ И УПРАВЛЕНИЕ

COMPUTING, TELECOMMUNICATIONS AND CONTROL

Том 13, № 1, 2020

Учредитель — Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный политехнический университет»

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ЭЛ № ФС77-77378 от 25.12.2019 г.

Редакция журнала

д-р техн. наук, профессор *А.С. Коротков* — главный редактор
Е.А. Калинина — литературный редактор, корректор
Г.А. Пышкина — ответственный секретарь, выпускающий редактор

Телефон редакции (812)552-62-16

E-mail: infocom@spbstu.ru

Компьютерная верстка *А.А. Кононова*

Перевод на английский язык *Д.Ю. Алексеева*

Лицензия ЛР № 020593 от 07.08.97

Дата выхода 27.03.2020. Формат 60×84 1/8

Санкт-Петербургский политехнический университет Петра Великого
Адрес университета и редакции: 195251, Санкт-Петербург, ул. Политехническая, д. 29.