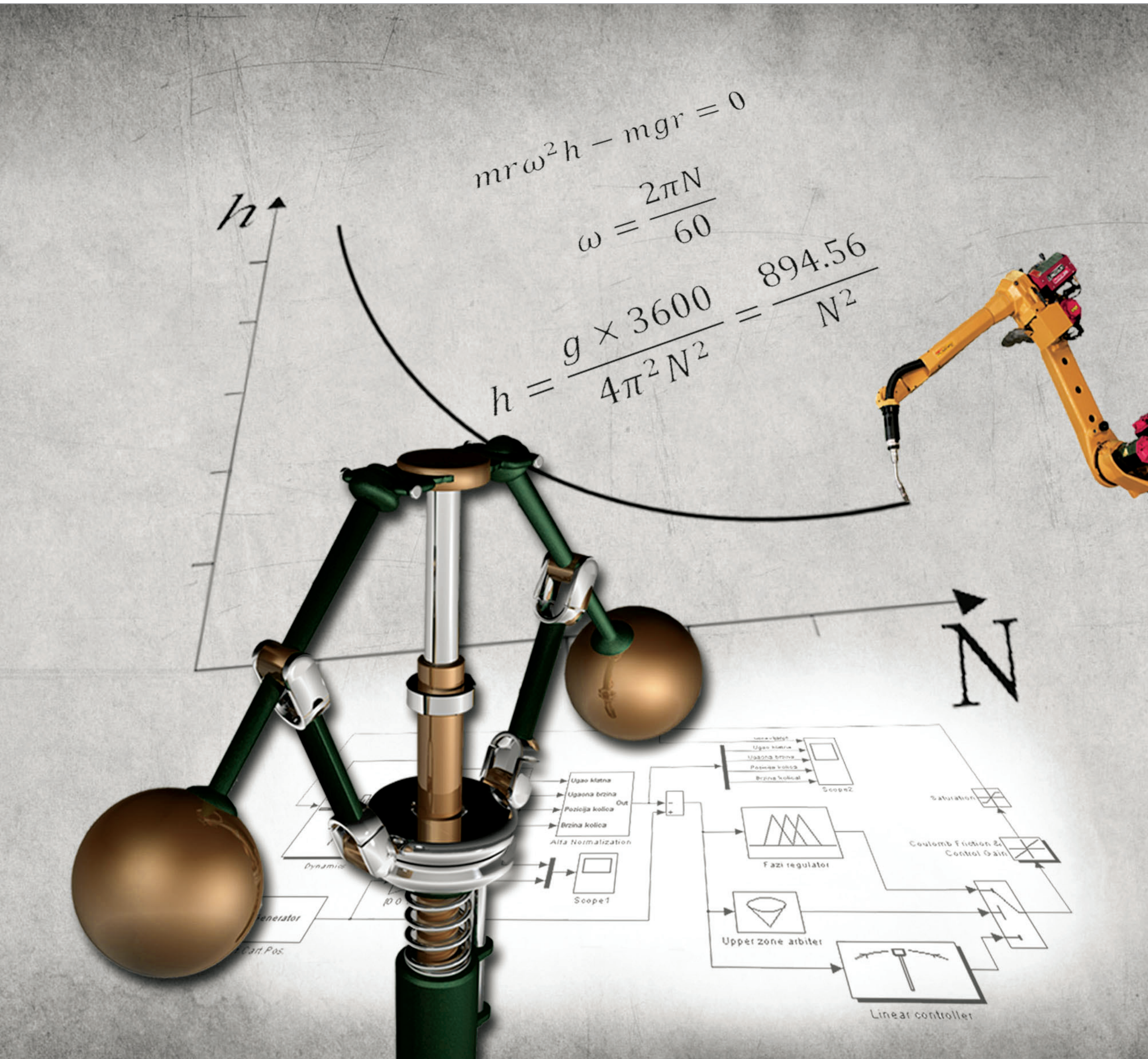
UNIVERSITY OF NIŠ

ISSN 1820 – 6417 (Print)
ISSN 1820 – 6425 (Online)
COBISS.SR-ID 158108940
UDC 62

FACTA UNIVERSITATIS

Series
AUTOMATIC CONTROL AND ROBOTICS
Vol. 21, Nº 3, 2022



Scientific Journal **FACTA UNIVERSITATIS**
UNIVERSITY OF NIŠ
Univerzitetski trg 2, 18000 Niš, Republic of Serbia
Phone: +381 18 257 095 Telefax: +381 18 257 950
e-mail: facta@ni.ac.rs http://casopisi.junis.ni.ac.rs/

Scientific Journal FACTA UNIVERSITATIS publishes original high scientific level works in the fields classified accordingly into the following periodical and independent series:

Architecture and Civil Engineering
Automatic Control and Robotics
Economics and Organization
Electronics and Energetics
Law and Politics
Linguistics and Literature
Mathematics and Informatics
Mechanical Engineering
Medicine and Biology
Philosophy, Sociology, Psychology and History
Physical Education and Sport
Physics, Chemistry and Technology
Teaching, Learning and Teacher Education
Visual Arts and Music
Working and Living Environmental Protection

SERIES: **AUTOMATIC CONTROL AND ROBOTICS**

Editor-in-Chief: **Marko Milojković**, e-mail: *fuacred@junis.ni.ac.rs*
University of Niš, Faculty of Electronic Engineering
Republic of Serbia, 18000 Niš, Aleksandra Medvedeva 14
Phone: +381 18 529 363, Fax: +381 18 588 399

Technical Assistance: **Staniša Lj. Perić**, University of Niš, Faculty of Electronic Engineering,
Department of Control Systems, Niš, Republic of Serbia, e-mail: *fuacrts@junis.ni.ac.rs*

Publication frequency – one volume, three issues per year.
Published by the University of Niš, Republic of Serbia

Financial support: Ministry of Education, Science and Technological Development of the Republic of Serbia
Printed by ATLANTIS DOO, Niš, Republic of Serbia
Circulation 60

Previous title: Scientific Journal FACTA UNIVERSITATIS, Series Mechanics, Automatic Control and Robotics. – Vol. 1, No 1 (1991) – Vol. 6, No 1(2007). – ISSN 0354 – 2009
Since 2007 devided in two Series:
Scientific Journal FACTA UNIVERSITATIS, Series Mechanics. – Vol. 7, No 1(2008) – . – ISSN
Scientific Journal FACTA UNIVERSITATIS, Series Automatic Control and Robotics. – Vol. 7, No 1(2008) – . – ISSN 1820 – 6417

ISSN 1820 – 6417 (Print)
ISSN 1820 – 6425 (Online)
COBISS.SR-ID 158108940
UDC 62

FACTA UNIVERSITATIS

SERIES **AUTOMATIC CONTROL AND ROBOTICS**

Vol. 21, N^o 3, 2022

UNIVERSITY OF NIŠ

INSTRUCTION FOR AUTHORS

As an author, you are kindly advised to use the paper template available for downloading on journal web site (section Download documents). This way you have nothing to change in terms of paper and text format. Simply applying the styles defined in the document will be sufficient.

Paper submitted for publication may be written in English, French or German (preferably in English) and submitted in the final camera-ready form. It is mandatory to submit your original work in Microsoft Word format (.doc not .docx) by using our online manuscript submission system. You have to make "New Submission" and upload your paper by using the online interface. All subsequent versions should be uploaded by using the same paper ID and your defined password. We are unable to process files sent by E-mail. We will do the final formatting and all necessary format conversions of your paper.

Articles are usually 10 to 25 type-written pages long. However, in special cases, shorter or longer articles may be accepted with appropriate reasoning. Author name, affiliation and complete address are to be placed underneath the title. Each paper should be preceded by a brief summary (50-150 words) in the same language. The text should be concise.

Letters, figures, and symbols should be clearly denoted so that no doubts about their meaning can arise. Symbols which may lead to confusion (e.g. letter I and figure 1, figure 0 and letter O) should be distinguished by marks which are explained in "Remarks for the typesetter".

Equations should be typewritten using MathType add-on (http://www.mathtype.com). For equations in your paper (Insert | Object | Create New | or MathType Equation), and, with the number, placed in parentheses at the right margin. Reference to equations should use the form "Eq. (2)" or simply (2). Each formula should occupy one line.

All figures should be numbered with consecutive Arabic numbers, have descriptive captions, and be mentioned in the text. Figures submitted must be of a standard high enough for direct reproduction. Line drawings should be prepared in electronic form. Figures should be planned in advance, so as to allow reduction to 12.75 cm in column width.

References should be quoted in the text by the corresponding number in square brackets and listed at the end of the manuscript in the order as they shown in the paper, in the same way as the following examples (for a journal, book, unpublished paper, proceeding, thesis, user manual, internet document):

[1] B. M. Danković, "A class of almost orthogonal filters," Journal of Circuits, Systems, and Computers, vol. 18, no. 5, pp. 923–931, 2009. [Online]. Available: http://dx.doi.org/10.1142/S0218126609005447
[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
[3] M. T. Milojković, D. S. Antić, S. S. Nikolić, Z. D. Jovanović, S. Lj. Perić, "On a new class of quasi-orthogonal filters," International Journal of Electronics, [Online]. Available: http://dx.doi.org/10.1080/00207217.2012.743087, to be published.
[4] M.-B. Radac, R.-A. Achimescu, R.-E. Precup, S. Preitl, C.-A. Dragos, A.-I. Stinean, "Design and experiments for model-free PI control of DC drives," in *Proceedings of 8th IEEE International Symposium on Applied Computational Intelligence and Informatics*, Timisoara, Romania, pp. 103–108, 2013. [Online]. Available: http://dx.doi.org/10.1225/sc.2013.018
[5] D. Mitić, Digital variable structure systems based on input-output model. PhD thesis, University of Niš, Faculty of Electronic Engineering, 2006.
[6] Inteco, "The laboratory anti-lock braking system controlled from PC," User's manual, 2008. [Online]. Available: www.inteco.com.pl
[7] MATLAB, The Language of Technical Computing, 2013. [Online]. Available: http://www.mathworks.com/products/matlab [Accessed on December 2013].

Electronic submission. Papers for consideration should be submitted to the Series Editor in electronic form via the Journal's home page: http://casopisi.junis.ni.ac.rs/index.php/FUAutContRob/index.

FACTA UNIVERSITATIS

Series

Automatic Control and Robotics

Vol. 21, Nº 3, 2022

Contents

FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 21, Nº 3, 2022, pp. 131 - 145
https://doi.org/10.22190/FUACR220623011J

Regular Paper

DEVELOPMENT OF GEO-DA BASED ANDROID APPLICATION FOR SPATIAL AND STATISTICAL ANALYSIS OF SERBIAN LONG-TERM PRECIPITATION DATA

UDC ((004.6:528.92) + (004.9:339.133))

Miljan Jeremić[1], Miljana Milić[2], Jelena Milojković[2], Milan Gocić[3]

[1]Knjaževac gymnasium, Knjaževac, Republic of Serbia
[2]University of Niš, Faculty of Electronic Engineering, Niš, Republic of Serbia
[3]University of Niš, Faculty of Civil Engineering, Niš, Republic of Serbia

Abstract. *Precipitation is one of the most important parameters to consider while analyzing risks from natural weather threats like floods and droughts. To prevent significant infrastructural and financial damage and loss, a good analysis of data and its visual presentation must be enabled. This also requires a reliable source of precipitation data for the region of interest. It is important to interpret and visualize the data obtained from meteorological stations to make an assessment, a forecast, or issue a warning. This paper describes the development of an Android application that enables visualization and analysis of average annual precipitation data for the period from 1946 to 2019, in the territory of the Republic of Serbia. Necessary precipitation data were acquired on the request from the official weather monitoring service for the Republic of Serbia (RHMSS).*

Key words: *Precipitation, Android application, Geo-Da environment, Visualization*

1. INTRODUCTION

The evolution of mankind in all areas of life produced some undesirable effects. They are the weather and climate changes. Climate change is characterized by an increase in temperatures, but also by changes in the precipitation pattern, its annual distribution as well as the intensity distribution, but also by the increased frequency of extreme weather events and periods with extreme climatic conditions [1]. Such changes strongly impact the environment, the economy, and people's health and safety [2], [3], [4], [5], [6], [7]. To assess the climate

Received June 23, 2022 / Revised October 11, 2022 / Accepted November 23, 2022
Corresponding author: Miljana Milić
University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia
E-mail: miljana.milic@elfak.ni.ac.rs

changes for some territories and make future climate projections, it is necessary to use available and reliable long-term climatic and meteorological data sets [8].

The climate change can only be detected as a long-term process [9], [10]. It requires the processing of large meteorological data sets that are acquired over a long period of time. These data sets are measured and assigned to a particular territory. Measured meteorological data consists of many different meteorological indices, which makes the amount of information quite large [11]. Meteorological data could be obtained manually or automatically from sensors at meteorological stations, satellite observations, radar data, and smart mobile devices [12]. In general, national meteorological centers or similar dedicated organizations are usually responsible for the systematic observation of weather conditions across a specific area [13].

To deal with a massive amount of information, for long-time periods, it is necessary to develop computing methods for easier, faster and more reliable processing and visualization. In the field of meteorological data, one of the most appropriate is the spatio-temporal analysis [14]. The spatio-temporal data analysis is a useful tool in the field of application development. It is a relatively new computational technique that enables the analysis of large spatio-temporal databases. Spatio-temporal patterns arise when data is collected over time and space and has at least one spatial and one temporal attribute. An event in a spatial data set describes a spatial and temporal phenomenon that exists at a given time and location [15]. It contains information about the value of the observed variable, and its time and coordinates of sampling. To assess such data structures a combination of time series and spatial-statistical analysis is required [16].

This paper describes a development of a specific Geo-Da based Android application that performs the spatio-temporal analysis of precipitation data for the territory of the Republic of Serbia for the 74 years long period. The application should identify statistical and geospatial relations over the acquired data and use the Geo-Da software tool to obtain certain meteorological and climatic conclusions, assumptions and forecasts that could be used for decision-making actions in the fields of science, economy, politics, agriculture, education etc.

2. Study Area

In the year 2014 the territory of the Republic of Serbia experienced one of the worst natural disasters i.e., an extreme flood [17]. The flood prevention system either failed to prevent damage or was never implemented. The riverbeds were rarely cleaned and maintained for decades, and the embankments were not renovated. The pumping stations were also neglected and their functionality was rarely checked. The authorities of the Republic of Serbia have conducted a post-disaster damage assessment after the floods. In general, the serious unpreparedness disturbed the lives of many people and did severe damage to urban systems, agriculture, and the local economy [18], [19]. These are the main reasons that confirm the necessity of developing, one easy-to-use, highly available and affordable, mobile phone-based application for following the precipitation trends for different areas in the Republic of Serbia.

The necessary data for development of such an application can be acquired from 28 national meteorological stations that are shown on the map in Figure 1 [20].

The meteorological data that are gathered for the analysis cover the area between 46°11′ N (Hajdukovo) to 41°51′ N (Dragaš, Kosovo) latitude and 23°01′ E (Senokos, Dimitrovgrad) to 18°51′ E (Bezdan, Vojvodina) longitude. This covers an area of approximately, 88,499km^2, and is bounded by a 2,114km borderline. Serbia's climate is a mix of continental in the north, with cold dry winters and warm, humid summers with well-distributed rainfall patterns, and the Mediterranean in the south, with hot, dry summers and autumns and average comparatively cool and rainy winters with substantial mountain snowfall [21]. Climate variances are caused by differences in height, closeness to the Adriatic and Aegean Seas, vast river basins, and exposure to the winds [22].

Northern Serbia has a continental climate, shaped by air masses from northern and western Europe. Strong Mediterranean influences can be noticed in mild winters and hot summers in Serbia's south and south-east [21]. The Dinaric Alps and other mountain ranges, on the other hand, help to cool a large portion of the warm air masses. Because of the mountains that encircle the plateau, the winters in Raška (region) are particularly harsh [23] [24]. For the period 1981 to 2017, the average annual air temperature for the area with a height of up to 300 m was 11.6°C. The average yearly temperature in areas between 300 and 500 meters in height is around 11.0°C, while over 1,000 m altitude is around 7.5°C [25]. According to the Republic Hydrometeorological Service of Serbia, the record of the coldest weather in Serbia measured the temperature of −39.5°C (January 26, 2006, Karajukića Bunari in Pešter), while the highest daily temperature was 45.2°C (July 24, 2007, Leskovac) [25].

Fig. 1 Map of national meteorological stations in the Republic of Serbia

Precipitation is one of the most essential climatic elements [21], [26]. Considering the atmospheric processes and the characteristics of the relief, the precipitation on the territory of Serbia is irregularly distributed in time and space. The typical annual amount of precipitation for the entire territory of Serbia is 896 mm. Annual average precipitation values increases with altitude. Dryer areas, with precipitation below 600 mm, are located in the north-east of the country, in the South Morava Valley and part of Kosovo [27]. The area of the Danube Region, the valley of the Great Morava and its continuation towards Vranje and Dimitrovgrad, has up to 650 mm of precipitation during the year [27]. Going east, in the area of the Homolje Mountains, the annual amount of precipitation reaches values up to 800 mm. Similar conclusions stand for the mountainous areas in south-eastern Serbia. The larger areas at the west and south-west are the rainiest areas of Serbia. According to the Pešterska plateau and Kopaonik, the values reach 1,000 mm per year, while some mountain peaks in south-western Serbia have even larger annual precipitation indices [27].

Figure 2 depicts the average annual precipitation in Serbia for the period 1946-2019. In the lowest parts of Serbia, annual precipitation values range from 540 to 820 mm. In locations over 1,000 m, the average precipitation is from 700 to 1,000 mm, and up to 1,200 mm on some mountain peaks in south-western areas.

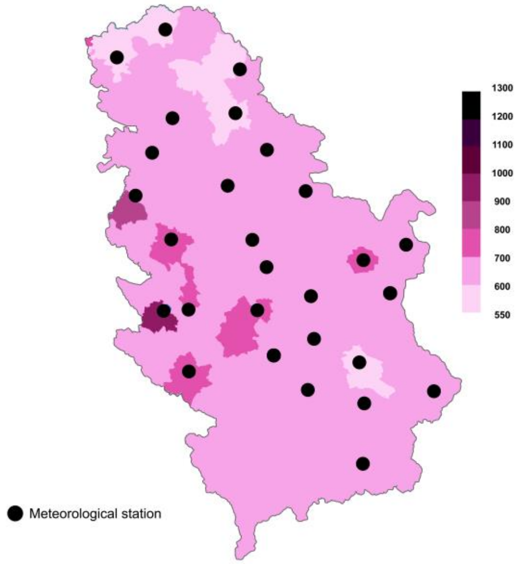


Fig. 2 Mean annual precipitation with a spatial distributionfor the period 1946-2019 in Serbia

Figure 3 provides the spatial distribution of the coefficient of annual precipitation variation for the precipitation data for the selected meteorological stations. Zlatibor and Loznica have a coefficient of variation of about 17%, while Kopaonik has the largest variation of over 32%. The coefficient of precipitation variance in Serbia is mostly less than 20%.

The majority of Serbia experiences the continental precipitation, with higher amounts of rainfall in the warmer months. The rainiest months are June and May. The month of June receives 12 to 13% of the total yearly rainfall. The months of February and October receive the least precipitation, accounting for 5 to 6% of the total yearly precipitation. The territory in southern Serbia has a Mediterranean precipitation regime, with a maximum in November, December, and January, and a minimum in August, due to relief, slopes of high mountain massifs, and the impact of the Mediterranean climate [28].

The snowfall and snow cover formation are characteristic for the period from November to March, and sometimes in April and October. For the mountain regions above 1,000 m these can appear during other months. The highest number of days with snow cover (30 to 40% of the total annual number of days with snow) is in January [29]

The extreme values of ever measured precipitation so far are:

- The driest year was 2000, when only 223.1 mm was measured in Kikinda.
- The rainiest year was 1937, when as much as 1,324.5 mm was measured in Loznica.
- The highest monthly amount of precipitation was registered in June 1954 in Sremska Mitrovica, 308.9 mm.
- The highest daily amount of precipitation was registered on October 10, 1955 in Negotin, 211.1 mm.

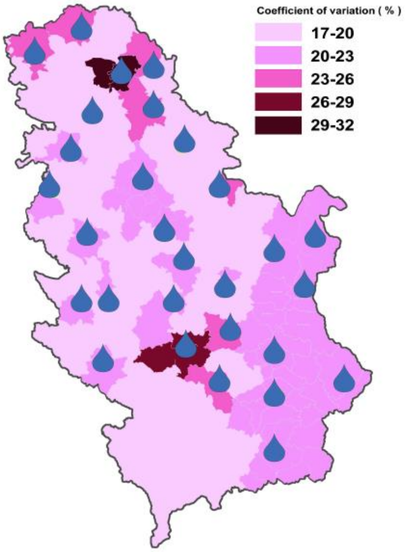


Fig. 3 Annual variations of precipitation data in Serbia

3. Development of Geo-Da Based Application using MIT App Inventor Environment

Figure 4 depicts the methodology for developing an Android application using the available data. It starts with the data gathering and finishes with an appropriate visualization of precipitation geo-data. In particular, the waterfall methodology was used in this study for the application development. The waterfall technique is a systematic and sequential paradigm for developing information systems [30]. The steps of the Waterfall approach are: 1 Definition and analysis of the system requirements, 2 Design of the software system, 3, Unit implementation and testing, 4 System integration, and 5 System operation and maintenance. This traditional project management flow, provides a project with a well-defined and simple structure. Waterfall is a linear and sequential method, that enables the team to divide the project into manageable, explainable phases with distinct outputs. Only once the previous phase has been finished and approved does the team move on to the following step.

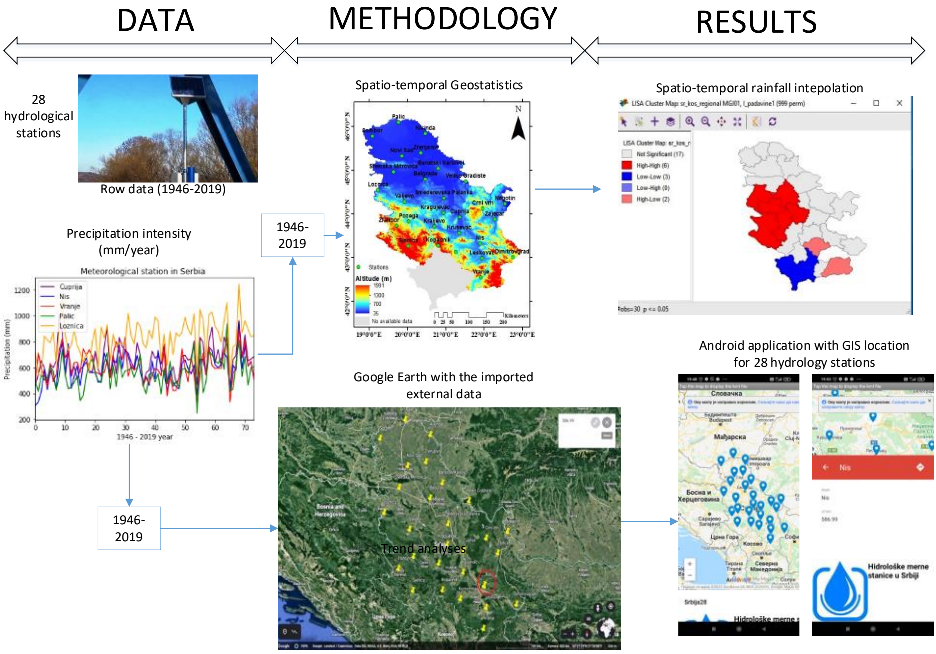


Fig. 4 Methodology for Android application development

The amount of raw meteorological data is quite large for a direct processing. This necessitates some feature extraction to reduce the amount of data to some meaningful and manageable size [31]. In particular, the necessary precipitation data was acquired on the request from the official weather monitoring service for the Republic of Serbia [Data source: Republic Hydrometeorological Service of Serbia; available online at www.hidmet.gov.rs, accessed on the 5th of December 2022]. Some parts of the data set are also available online at the web platform of this service [32]. Each data sample consists of information about the year, month, geographic location of the place, and the precipitation value for each month of the year. Using the analytical queries with the ORACLE database and its services, classification by months for 74 years was obtained over the data set. In this way the feature extraction process resulted in the parametric numerical and visual data systematization by the time period and location of the input signals. The extracted features are stored in the feature database and are used for further processing. Resulting data structures contain the average values of precipitation for all 28 locations that could later be linked to the KML database in order to enable their visualization. It should be emphasized that the main purpose of the application for precipitation visualization did not require any data adjustment. These data should be used and visualised in its “raw“ version, since the aim of the application is to visualize these values for the specified place and time and display it in the appropriate way.

Table 1 Geographic position of the meteorological measuring stations by their longitude and latitude

City	Longitude	Latitude
Kininda	20.47	45.85
Novi Sad	19.85	45.33
S. Mitrovica	19.63	44.97
Beograd	20.47	44.80
Loznica	19.23	44.55
Valjevo	19.92	44.28
Crni Vrh	21.97	44.13
Kragujevac	20.93	44.03
Zajecar	22.28	43.88
Kraljevo	20.70	43.72
Krusevac	21.35	43.57
Nis	21.90	43.33
Kopaonik	20.80	43.28
Vranje	21.92	42.55
Palic	19.77	46.10
Sombor	19.08	45.78
Zrenjanin	20.35	45.40
Dimitrovgrad	22.75	43.02
B. Karlovci	20.80	45.05
S. Palanka	20.95	44.37
V. Gradiste	21.52	44.75
Negotin	22.55	44.23
Zlatibor	19.72	43.73
Sjenica	20.02	43.27
Pozega	20.03	43.83
Kursumlija	21.27	43.13
Loznica	21.37	43.93
Valjevo	21.95	43.98

Two forms of acquired data are illustrated in Table 1, representing geographic positions of the meteorological measuring stations, and Table 2, representing the form of „raw" precipitation data that needs to be visualized [32]. Table 2 gives the exact values of measured precipitation values at 28 meteorological stations at the territory of the Republic of Serbia from April to September 2022.

The process and programs that enable described data synergy are shown in Figure 5.

Geo-Da software is a free, user-friendly platform, that enables different spatio-related data analysis. It is developed at the University of Illinois and Chicago, and was funded by federal agencies like the National Science Foundation (USA), and the National institute of Health (USA) [33], [34]. It is an open-source platform which can run under different operating systems. Its main areas of implementation are spatial data analysis, geovisualization, spatial modeling and autocorrelation.

A graphical, intuitively designed interface to methods of exploratory spatial data analysis is accomplished by incorporating spatial statistical tests into simple map representation, linking data views of spatial and non-spatial distributions, and enabling real-time exploration of spatial and statistical patterns. It can import, convert and export data in different formats, [35], [36]. Then, it can transfer variables and create new ones. The software allows one to get new insights from spatial data by interactively exploring the data.

Table 2 Measured precipitation values in mm from April to September 2022, from 28 meteorological stations in the Republic of Serbia

Measuring station/ Month 2022	April	May	Jun	July	August	September
Palic	33.5	35.3	68.6	14.2	73.5	126.3
Sombor	35.6	56.4	36.1	19.6	39.0	112.1
Novi Sad	54.5	17.9	43.6	13.8	103.9	159.0
Zrenjanin	45.1	25.0	19.9	1.5	87.4	132.9
Kikinda	37.7	53.2	61.5	10.0	80.7	60.2
B. Karlovac	68.1	18.1	17.4	24.3	78.4	157.5
Loznica	57.6	31.2	171.5	68.2	122.9	167.0
S. Mitrovica	65.5	11.8	42.8	12.4	129.0	134.4
Valjevo	42.7	32.8	94.2	24.6	42.6	64.2
Beograd	80.1	32.2	43.3	63.9	89.7	98.0
Kragujevac	35.6	77.6	103.3	66.3	66.2	56.9
S. Palanka	75.4	62.4	120.9	79.3	99.3	108.6
V. Gradiste	81.7	42.1	38.9	42.2	54.4	160.7
Crni Vrh	48.9	58.7	55.9	69.4	75.5	84.4
Negotin	70.0	33.7	47.2	44.1	61.5	40.2
Zlatibor	66.9	61.0	126.7	26.3	44.7	94.3
Sjenica	55.7	75.9	76.4	63.4	52.1	96.3
Pozega	38.2	45.9	92.9	8.5	30.6	59.8
Kraljevo	41.7	32.1	127.5	38.1	124.9	99.3
Kopaonik	105.4	78.6	133.0	106.6	118.3	156.5
Kursumlija	52.5	22.1	84.3	30.1	85.8	107.6
Krusevac	40.1	28.9	93.3	53.3	102.2	119.8
Cuprija	37.8	50.6	83.3	18.7	101.6	99.5
Nis	58.1	23.0	43.5	58.6	119.2	67.9
Leskovac	67.9	47.1	82.2	35.9	83.0	122.3
Zajecar	38.3	36.2	124.7	44.1	24.2	54.3
Dimitrovgrad	73.0	52.7	111.1	81.7	111.8	96.3
Vranje	55.0	35.6	81.5	43.5	38.9	100.1

One can detect patterns and correlations, find trends with different maps and statistical views, and detect outliers. At the end, after exploring the spatial data in multiple views, the software can produce global and local spatial autocorrelation statistics as well as spatial regression models.

MIT App Inventor is a suitable and worthy environment for creating Android applications [37], [38]. Its application is shown in Figure 6. This web-based environment is designed to operate with command blocks; similar to the Arduino [38], Mbot [40], [41] and Scratch [42], [43] environments where commands are dragged into the workspace and parameter values are then set in commands. The Java Script programming language is used in the background.

This data processing aims to visualize the average precipitation in millimeters for 74 years, for the selected hydrological measuring station with the developed application. Data is aggregated by year and month starting from 1946 until 2019 [20].

KML (Keyhole Markup Language) is a file format that displays geographic data in an Earth browser such as Google Earth. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard [44], [45].

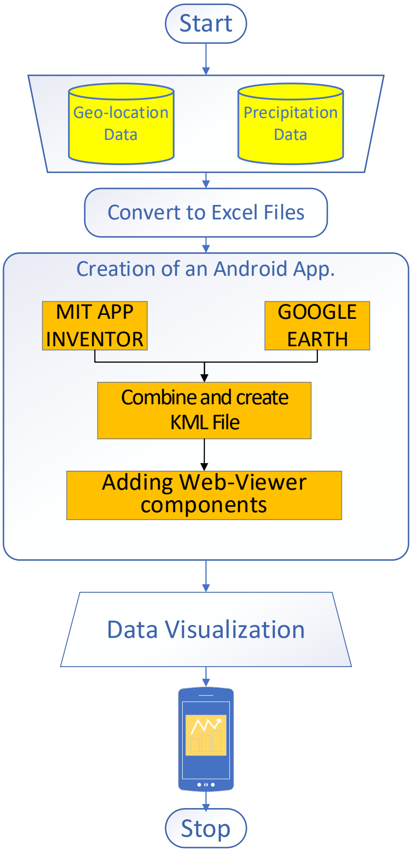


Fig. 5 Android application algorithm development

Using the Google Earth application at earth.google.com, we import a newly created KML file with geo-data on the locations of the hydrological measuring stations [46], [47], [48], [49]. Their average precipitations for 74 years are displayed on the map after importing the file (Figure 6). All stations are shown on the map of Serbia and are denoted with blue markings. By clicking on a particular hydrological measuring station, in the upper left corner, one can obtain the exact average amount of precipitation for 74 years. After all the imported files are located in the Google Earth web environment, one should now save the newly created project and download that link from the project for use in the MIT App Inventor Android environment.

The WebViewer component is used to display geospatial and numerical data from the Google Earth environment, which recognizes that a KML file with all the necessary data has been generated from the user. This component is certainly the most important in the created application because it regulates the selection and display of the measuring station, and the overall display of all measuring stations in the Republic of Serbia.

After selecting of the hydrological measuring station from the map, the application shows the name of the measuring station and the data related to the average amount of precipitation for the selected measuring station.

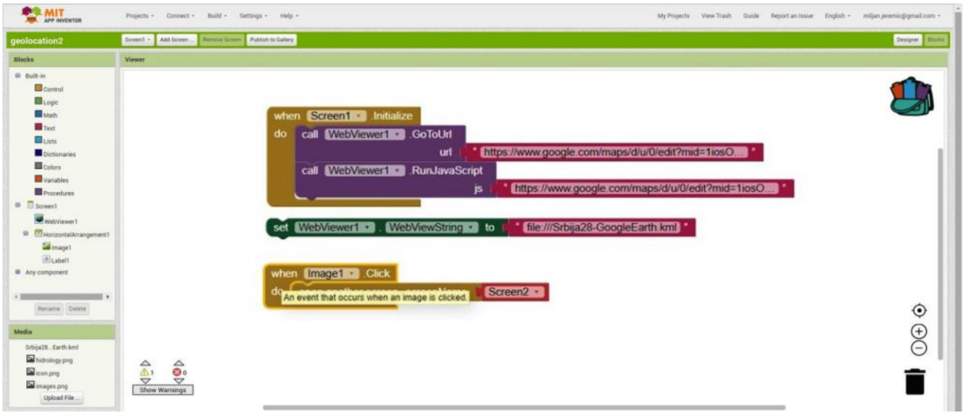
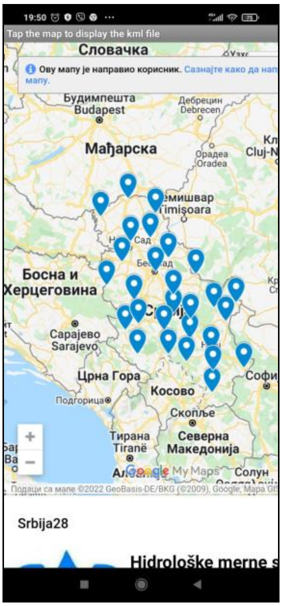


Fig. 6 MIT App Inventor

4. Visualization of Precipitation Data

Precipitation data can be loaded and viewed using the developed Android application, as illustrated in Figure 7. As mentioned before, there are 28 measuring stations displayed on the map of Serbia. The map for the selected station can provide information about a specific location, i.e., its yearly precipitation value.



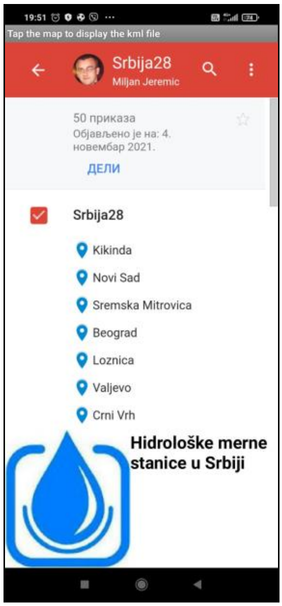


Fig. 7 Map of Serbian meteorological stations **Fig. 8** List of meteorological stations

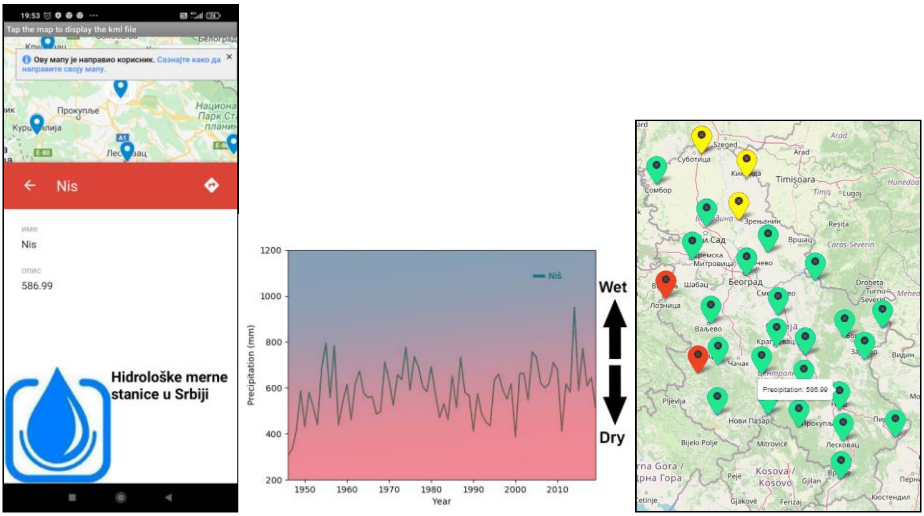
The annual amount of precipitation is depicted on the Serbian map in various colours, with yellow representing precipitation below 580 mm, green representing precipitation between 580 and 800 mm, and red representing precipitation exceeding 800 mm.

Figure 8 shows the list of the meteorological stations monitored by the application, as it appears in the Android application.

The time series of annual precipitation of the selected three meteorological stations in Serbia (Niš, Palić, and Loznica) are presented in Figures 9, 10 and 11, respectively. Each figure displays the selected meteorological location on the map of Serbia, with the corresponding average annual precipitation for the period 1946-2019 (Figures 9 a), 10 a) and 11 a)). For the given period, the average annual precipitation at selected locations ranged from 557 mm (Palić) to 830 mm (Loznica).

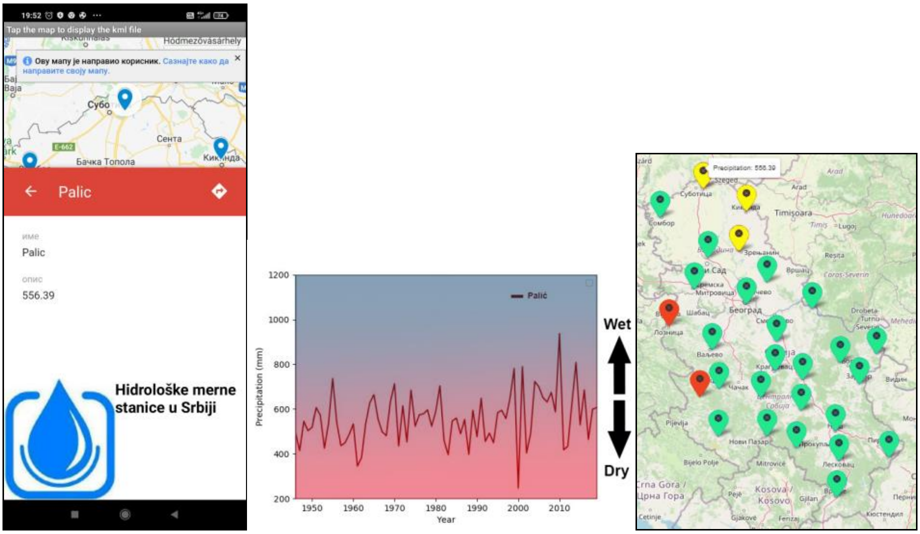
The time series of annual precipitation of the selected three meteorological stations in Serbia (Niš, Palić, and Loznica) are presented in Figures 9b), 10b) and 11b), respectively.

Precipitation data for cities of Niš, Palić, and Loznica on the map of the Serbian metheorological stations are shown in Figures 9c), 10c) and 11c), respectively.



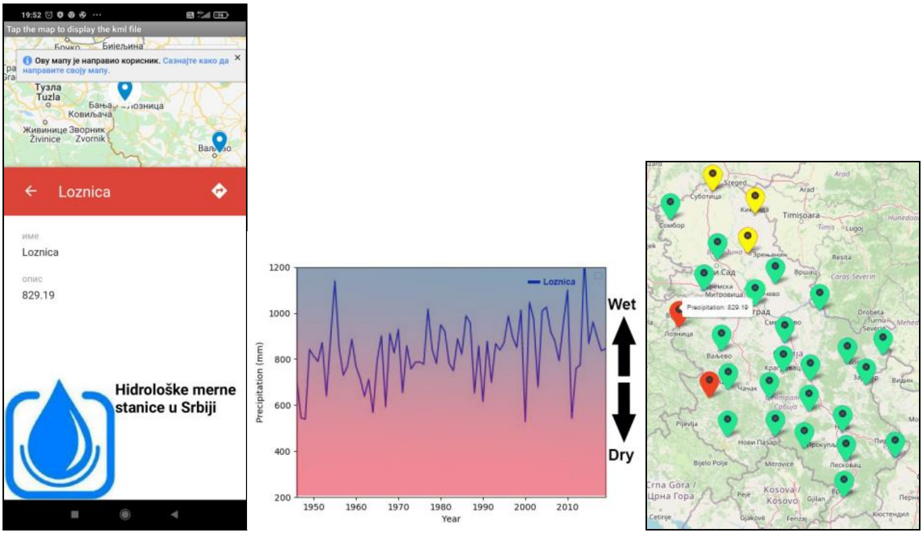
a) b) c)

Fig. 9 Precipitation data for the city of Niš obtained from the Android application: a) anual precipitation value for the selected city, b) time series of long-term annual precipitation for the selected city, and c) precipitation data for the city of Niš on the map of Serbian metheorological stations



a) b) c)

Fig. 10 Precipitation data for the city of Palić, obtained from the Android application: a) anual precipitation value for the selected city, b) time series of long-term annual precipitation for the selected city, and c) precipitation data for the city of Palić on the map of the Serbian metheorological stations



a) b) c)

Fig. 11 Precipitation data for the city of Loznica, obtained from the Android application: a) anual precipitation value for the selected city, b) time series of long-term annual precipitation for the selected city, and c) precipitation data for the city of Loznica on the map of the Serbian metheorological stations

5. Conclusion

The methodology and the Android Geo-Da based application for visualizing average annual precipitation data for 28 meteorological stations in Serbia from 1946 to 2019 is described in this paper. The application could be valuable in water resources as well as agricultural planning and management.

With the collection of new precipitation data, for the period after 2019, the application would have to be updated. Changes of the data sources should not influence the functionality of the application if their structure remains the same. On the other hand, the KML file would also have to be updated in order to keep the application functional.

The focus on further research in this field should be put on the development of advanced application options that enable monitoring of natural hazards such as drought and flood that also rely on precipitation data. Further, in our future research a newer version of the described application should be developed with the implementation of some mathematical and soft-computing approaches that predict extreme weather conditions, and enable early alarming and timely reacting.

Acknowledgement: *This work has been supported by The Ministry of Education, Science and Technological Development of the Republic of Serbia under the contract No. 451-03-68/2022-14/ 200102 of February, 04. 2022.*

References

[1] V. Đurđević, A. Vuković, and M. Vujadinović Mandić, “Climate changes observed in Serbia and future climate projections based on different scenarios of future emissions”, Report under the Project ”Second Biennial Update Report and Third National Communication under the UNFCCC for the Republic of Serbia”, United Nations Development Programme, ISBN-978-86-7728-301-8, 2018.
[2] N. S. Gruda, M. Bisbis, and J. Tanny, “Influence of climate change on protected cultivation: Impacts and sustainable adaptation strategies-A review”, Journal of Cleaner Production, vol. 225, pp. 481-495, 2019.
[3] J. Singh, R. P. Singh, and R. Khare, “Influence of climate change on Antarctic flora”, Polar Science 18, doi.org/10.1016/j.polar.2018.05.006, pp. 94-101, 2018.
[4] U. Husain, and S. Javed, “Impact of climate change on agriculture and Indian economy: a quantitative research perspective from 1980 to 2016”, Industrial Engineering & Management, vol. 8, no. 2, pp. 2-5, 2019.
[5] F. Darus, H. I. M. Zuki, and H. Yusoff, “The path to sustainability: Understanding organizations’ environmental initiatives and climate change in an emerging economy”, European Journal of Management and Business Economics, vol. 29, ISSN: 2444-8494, 2019.
[6] C. J. Sorensen et al., “Clinical implications of climate change on US emergency medicine: challenges and opportunities”, Annals of Emergency Medicine, vol. 76, no. 2, doi.org/10.1016/ j.annemergmed.2020.03.010, 168-178, 2020.
[7] P. L. Kinney, “Interactions of climate change, air pollution, and human health”, Current environmental health reports, vol. 5, no.1, pp. 179-186, 2018.
[8] M. A. Amiri, M. Gocić, “Innovative trend analysis of annual precipitation in Serbia during 1946–2019” Environmental Earth Sciences, vol. 80, no. 23, DOI:10.1007/s12665-021-10095-w, 2021.
[9] M. Collins et al., “Long-term climate change: projections, commitments and irreversibility”, In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, 2013.
[10] A. Moazami et al., “Impacts of future weather data typology on building energy performance–Investigating long-term patterns of climate change and extreme weather conditions”, Applied Energy vol. 238, DOI:10.1016/j.apenergy.2019.01.085 ,pp. 696-720, 2019.
[11] C. Donald Ahrens, Essentials of Meteorology, An invitation to the Atmosphere, ISBN: 9781285462363, Brooks/Cole Publishing Company, Stamford, USA, January 2014,

[12] M. V. Kiselev, V. Nadezdha, E. A. Dyukarev, and S. A. Kurakov, “Automatic meteorological measuring systems for microclimate monitoring”, IOP Conference Series Earth and Environmental Science, 190:012031, DOI:10.1088/1755-1315/190/1/01203, October 2018.
[13] “The global observing system for climate: implementation needs”, World Meteorological Organization, 2016., available online at https://library.wmo.int/doc_num.php?explnum_id=3417; accessed on the 7th of September 2022.
[14] J. Han, M. Kamber, and J. Pei, Data Mining Trends and Research Frontiers, Chapter in book: Data Mining, Morgan Kaufmann Elsevier, pp. 585-631, 2012, ISBN 9780123814791.
[15] “Population health methods spatiotemporal analysis”, available online at https://www.publichealth.columbia.edu/research/population-health-methods/spatiotemporal-analysis, accessed on the 9th of September 2022.
[16] R. S. Bivand, E. Pebesma, and V. Gómez-Rubio, Applied Spatial Data Analysis with R, Part of the book series USE R, vol. 10, Springer, ISBN: 978-1-4614-7617-7, 2013.
[17] “Floods in Serbia - European Union continues to provide support” (in Serbian) – available online https://europa.rs/poplave-u-srbiji-evropska-unija-nastavlja-da-pruza-podrsku/, accessed on the 5th of October 2022.
[18] “Serbia floods 2014” (in Serbian) – available online at https://www.ilo.org/wcmsp5/groups/public/---ed_emp/documents/publication/wcms_397685.pdf, accessed on the 5th of October 2022.
[19] F. Trgovčević, A. Stupar, M. Ivanović, and R. Susman, “Article Toward Flood Resilience in Serbia: The Challenges of an (Un)Sustainable Policy”, Sustainability, vol. 12, no. 17, 2020, doi:10.3390/su12177228.
[20] Republic Hydrometeorological Service of Serbia www.hidmet.gov.rs, accessed on the 5th of December 2022.
[21] D. Milosevic, and M. Grujic, Geography 8 (in Serbian), Vulkan znanje, Belgrade, ISBN: 978-86-10-03430-1, 2021.
[22] M. Lukić et al., “Assessment of Outdoor Thermal Comfort in Serbia’s Urban Environments during Different Seasons” *Atmosphere,* vol. 12, no. 8, DOI:10.3390/atmos12081084, 2021.
M.Thomas, “Serbia”, Encyclopedia Britannica, 13 Aug. 2022, available online at https://www.britannica.com/place/Serbia, accessed on the 5th October 2022.
[23] M. Radovanović, and V. Dučić, 2002, Variability of Climate in Serbia in the Second Half of the 20th century, EGS XXVII General Assembly, Nice, 21 - 26 April 2002, Biblcode 2002EGSGA.27.2283R
[24] Republic Hydrometeorological Service of Serbia, “Basic Climate Characteristics for the Territory of Serbia (standard normal period 1961-1990)” (in Serbian), available online at: https://www.hidmet.gov.rs/data/klimatologija_static/eng/Klima_Srbije.pdf , accessed on the 5th of October 2022.
[25] M. Airey, and M. Hulme, “Evaluating climate model simulations of precipitation: methods, problems and performance”, Progress in Physical Geography: Earth and Environment, vol. 19, no. 4, pp. 427–448. doi.org/10.1177/030913339501900401, 1995.
[26] M. Blagojević, “Soil weed seed banks in vineyards established by intensive and extensive cultivation methods”, Doctoral Dissertation, (in Serbian), National Repository of Dissertations in Serbia, 2018.
[27] M. Novaković-Vuković, and S. Eremija, “Florističke i edafske karakteristike šuma crnog i belog bora na serpentinitu i peridotitima u zapadnoj Srbiji”, Beograd 2020, ISBN 978-86-80439-41-9.
Republic Hydrometeorological Service of Serbia, “Precipitation regime in Serbia” (in Serbian), available online at: https://www.hidmet.gov.rs/data/klimatologija_static/latin/Padavinski_rezim_u_Srbiji.pdf, accessed on the 5th of October 2022.
[28] D. R. Lucitasari, and S. A. K. Muhammad: "Designing Mobile Alumni Tracer Study System Using Waterfall Method: an Android Based." *International Journal of Computer Networks and Communications Security*, vol. 7, no. 9 (2019), pp.196-202.
[29] S. Domazetovska, V. Gavrilovski, M. Anachkova, and Z. Petreski, “Urban sound recognition using different feature extraction techniques”, Facta Universitatis series Automatic Control and Robotics, vol 20, no. 3, pp. 155-165, doi.org/10.22190/FUACR211015012D, 2021.
Republic Hydrometeorological Service of Serbia, „Monitoring of averae monthly air temperatures and precipitations in Serbia (in Serbian)“, available at:https://www.hidmet.gov.rs/data/klimatologija/Monitoringtemperaturaipadavinezaprethodnih6meseci.pdf, accessed on the 5th of October 2022.
[30] L. Anselin, I. Syabri, Y. Kho, GeoDa: An Introduction to Spatial Data Analysis. In: Fischer, M., Getis, A. (eds) Handbook of Applied Spatial Analysis. Springer, Berlin, Heidelberg, doi.org/10.1007/978-3-642-03647-7_5, 2010.
[31] X. Li, L. Anselin, and J. Koschinsky, “GeoDa Web – Enhancing Web-Based Mapping with Spatial Analytics”, SIGSPATIAL '15: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, vol. 94, pp. 1–4, doi.org/10.1145/2820783.2820792, November 2015.
[32] Z. Wu, F. Zhang, D. Di, and H. Wang, “Study of spatial distribution characteristics of river eco-environmental values based on emergy-GeoDa method”, Sci Total Environ. vol. 802, p. 149679, January 2022, doi: 10.1016/j.scitotenv.2021.149679, Epub Aug 19 2021.

[33] L. Anselin, Exploring Spatial Data with GeoDaTM: A Workbook, Center for spatially integrated social science, Urbana, USA, 2005
[34] E. W. Patton, M. Tissenbaum, and F. Harunani, “MIT app inventor: Objectives, design, and development”, *Computational thinking education*, pp. 31-49, Springer, Singapore, 2019.
[35] B. Xie, and H. Abelson, “Skill progression in MIT app inventor”, IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 213-217, September 2016.
[36] Y. A. Badamasi, “The working principle of an Arduino”, 11th international conference on electronics, computer and computation (ICECCO), IEEE, 2014.
[37] J. M. Sáez-López, M. L. Sevillano-García, and E. Vazquez-Cano. “The effect of programming on primary school students’ mathematical and scientific understanding: educational use of mBot”, *Educational Technology Research and Development*, vol. 67, no. 6, pp. 1405-1425, 2019.
[38] J. Pisarov, and G. Mester. “Programming the mBot robot in school”, Proceedings of the International Conference and Workshop Mechatronics in Practice and Education, MechEdu. Subotica, Serbia, December 2019.
[39] M. Resnick, et al. “Scratch: programming for all”, *Communications of the ACM*, vol. 52, no. 11, pp. 60-67, 2009.
[40] J. Maloney, et al. “The scratch programming language and environment”, *ACM Transactions on Computing Education (TOCE)*, vol. 10, no. 4, pp. 1-15, 2010.
[41] D. Nolan, and D. T. Lang. “Keyhole markup language”, XML and Web Technologies for Data Sciences with R. Springer, New York, NY, USA, pp. 581-618, 2014.
[42] D. Li, and L. Ming, “Integrating geometric models, site images and GIS based on Google Earth and Keyhole Markup Language”, *Automation in Construction,* vol. 89, pp. 317-331, 2018.
[43] L. Yu, and G. Peng, “Google Earth as a virtual globe tool for Earth science applications at the global scale: progress and perspectives”, *International Journal of Remote Sensing*, vol. 33, no. 12, pp. 3966-3986, 2012.
[44] O. Mutanga, and K. Lalit, “Google earth engine applications”, Remote Sensing, vol. 11, no. 5, pp. 591, 2019.
[45] N. Gorelick, et al. “Google Earth Engine: Planetary-scale geospatial analysis for everyone*”, Remote sensing of Environment,* vol. 202, pp. 18-27, 2017.
[46] https://earth.google.com/web, accessed on the 5th of October 2022.

FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 21, Nº 3, 2022, pp. 147 - 157
https://doi.org/10.22190/FUACR221120012S

Regular Paper

DESIGNING DISTRIBUTED CONTROLLING TESTBED SYSTEM FOR SUPPLY CHAIN AND LOGISTICS IN AUTOMOTIVE INDUSTRY

UDC (681.58:004.89)

Martin Štufi[1], Aleksandra Panajotović[2], Leonid Stoimenov[3]

[1]Solutia s.r.o., Prague, Czech Republic
[2]University of Niš, Faculty of Electronic Engineering,
Department of Telecommunications, Republic of Serbia
[3]University of Niš, Faculty of Electronic Engineering, Department of Computer Science,
Republic of Serbia

Abstract. *The arrival of the era of autonomous robots is indisputable. In this paper, innovations in the distributed control systems realized by autonomous guided vehicles in the automotive industry are provided as proof of concept. The main goal of the considered distributed control system design is to bring all-in-one dependent and independent VDA 5050 compliant robots that are easily configurable and manageable with the web-based high-quality user interface responsive business-critical application. Special attention is paid to applying a platform to manage all autonomous IoT based robots in one seamless system. In addition, a "single point of truth" as one of the main issues of modern distributed controlled systems has been considered.*

Key words: *Distributed control systems, Autonomous robots, Autonomous guided vehicles, Internet of things*

1. INTRODUCTION

The high demand, rapidly increasing customization, and differentiation of product portfolios in the shorter life cycle of car production in the automotive industry led to the development of flexible manufacturing [1]. Therefore, the automotive corporation strategy is to become very important in logistics within the automotive industry by using a modern distributed control system (DCS). As part of that strategy, many leading worldwide companies

Received November 20, 2022 / Accepted December 14, 2022
Corresponding author: Martin Štufi
Solutia s.r.o., Vrsovicka 1461/64, 101 00 Prague 10, Czech Republic
E-mail: martin.stufi@solutia.cz

recognize that changing global material handling markets relating to productivity requirements, labor cost, and e-commerce create a permanent demand for new logistics solutions [2]. In supplying automotive assembly lines, production automation can be implemented in Autonomous Mobile Robots (AMRs) or Autonomous Guided Vehicles (AGVs) [3].

The rapid growth in IoT and sensor networks, in combination with the challenges in automotive logistics such as operational efficiency, improving customer service level, data-driven demands, and digital transformation of automotive logistics, set to pave the way for new revenue streams. In the industry's evolution, Toyota Industries Corporation has become a leading logistics solutions provider, creating a permanent demand for new solutions. However, selecting appropriate technologies and developing an IoT distributed control system can be challenging for the automotive industry. For example, the need to set up an efficient, effective, and relevant product or service solution for managing AGVs in automotive logistics requires modern technology implementation. AGV is a vehicle with a battery and local computer control that can automatically move in a specific or predefined direction without human intervention. The most specific feature of AGV is that it can move the unit load from one place as a pickup point to another delivery point. In modern automotive logistics systems, DCS frequently manages these operations.

According to a path guidance principle [4], AGVs can be divided into static and dynamic path vehicles [5]. The static path vehicle uses a predefined path from the pickup point to the delivery point. The dynamic path vehicle uses its algorithm and autonomously determines the optimized path by detecting and avoiding obstacles on the floor plan. In the paper [6], the authors examine the limitations and prerequisites for implementing AGV technology for automating specific logistic processes. The case study [7] describes the increase in the productivity of the last workstation of an assembly line by implementing an AGV system to transport finished goods to the warehouse. A comprehensive knowledge base for designing, selecting and implementing the AMR technology, considered as the highest level of AGV, in the form of unit load carriers for supplying assembly lines in the automotive industry is presented in [8]. Another case study [9], answers the following questions: what are the requirements to be considered when using autonomous transport to move materials, and what steps must be taken to introduce autonomous technologies in the industrial environment. Using deep neural networks (DNN) in online monitoring for AGV is presented in [10]. The review of how automation and robots contribute to the increased profit demand by offering lower production costs, shorter delivery times, improved quality and higher customer satisfaction, is presented in [11]. The paper [12] focuses on developing Digital Twins for flexible manufacturing systems with AGVs on the factory floor.

All the above studies propose principal limitations for AGVs/AMRs, their controlling problem, supplying automotive assembly lines, and their relationship with DCS. The contribution of this article can be recognized in the improvements embraced by the Resmanio platform (www.resmanio.com), such as the integration of heterogeneous AGVs that are VDA 5050 compliant and incompliant, supporting different protocols in AGVs' communication and Continuous Integration and Continuous Delivery (CI/CD) readiness.

This paper is arranged as follows: Section 2 presents a global architecture with a high-level DCS system design focusing on modern distributed control. The use of the DCS application is presented in Section 3. Further, Section 4 contains the discussion. Finally, the authors discuss the future trends of DCS development and give concluding remarks in the last section.

2. Material and Methods

Škoda Auto continues its digital transformation journey. AGVs are operated through separate control systems of individual suppliers in their production plants. These systems are not interconnected, i.e., they provide services independently of others and represent separate fleets. After implementing the Unified Control System (UCS), a robust and scalable system will ensure the interconnection of all vehicles and fleets.

The system is built from the start to handle a large number of messages going from AGVs to the system's control section. Furthermore, the system is designed to operate in real-time. From the standpoint of architecture, it may be deployed and operated both in the Cloud environment (public or private) and in the customer's so-called on-premises environment. In this scenario, we propose system implementation in the Škoda Auto environment throughout the locations of the manufacturing plants. The entire solution is designed as an enterprise solution, where a business-critical application built in Java enables Škoda Auto differentiated products and services to their users. It requires only a light client, any web browser such as Google Chrome, Microsoft Edge, Firefox, and Opera. The data created and reported by the AGV software model are included in the data set utilized for this experimental study. The data is produced and sent via a multi-threaded Java application.

The project starts with an identification of challenges for digitalization in the logistics [13], such as (1) excess supplies with materials, spare parts, repairs, and maintenance; (2) overproduction of manufactured goods; (3) waiting for material, spare parts, information, transport; (4) excess manipulation as overlapping, translations, shifts, incorrect organization of warehouses, and material flows in production; (5) search material, documentation, information; (6) unnecessary activities and movements in human work or manipulation in movements, and (7) duplicate information, a selection from IT systems, transcription of reports.

Aligned with the customer objectives, this study focuses on four action points: (1) the testbed platform responsible for managing all AGV devices with in-built IoT sensors in one seamless system; (2) a single point of truth for all AGVs which allows the possibility of connecting these AGVs to the distributed control system; (3) comprehensive real-time notification functionality through the various communication channels as notification messages, e-mails, mobile applications or web applications, and (4) comprehensive support and management of all AGVs through a single VDA 5050 (Verband der Automobilindustrie) compliant platform.

The complexity of enterprise application growth with increases in functionality and features supported. Fig. 1 depicts the overall Microservice architecture and high-level design for the testbed horizontal scalable platform as DSC. The java-based application runs on the application server, and the main concept is built as microservice architecture. A Web Components part provides business logic from the Backend server. It provides features in a single Java application that is served up, operates on the application server, and offers supplementary services to the user.

Nowadays, creating an application with a microservice architecture typically calls for leveraging Java, a lightweight development environment, and widely used global frameworks like Spring Boot or Spring Cloud. This implementation provides enhanced features, such as new classes that provide Stream API or Lambda Expression [14] to support stream data from the sensor network. Moreover, through the API, it can be integrated into our application within external systems.

The DevOps movement has become synonymous with CI/CD, and the Resmanio platform has been designed and developed using the same principle. Continuous integration design practices result in this modular software that can be repeatedly integrated with other components into a single source. The continuous delivery ensures that this software is production ready. This testbed architecture offers a serverless Kubernetes cluster manager for Docker-based microservices with the proposed solution realized in a domain-driven design (DDD) fashion. As a result, Docker has become the standard used to develop and run containerized applications.

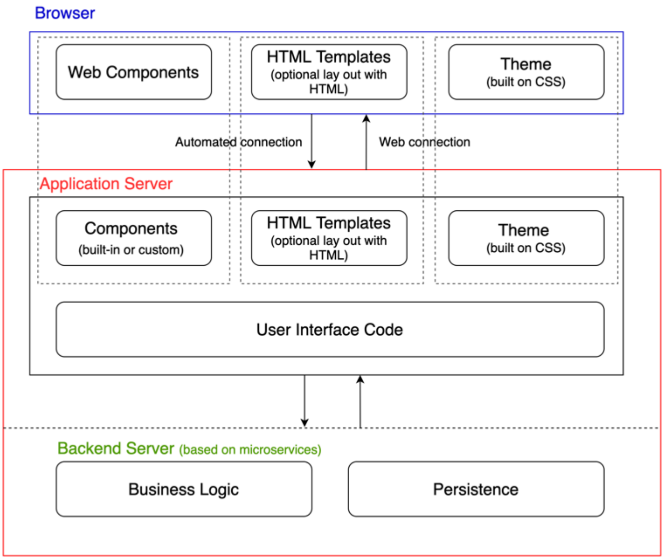


Fig. 1 High-level architecture diagram

3. Implementation of Testbed

Key characteristics of the testbed platform can be identified in Resmanio (Fig. 2), such as the Enterprise Java-based application used to build the mentioned system, which provides

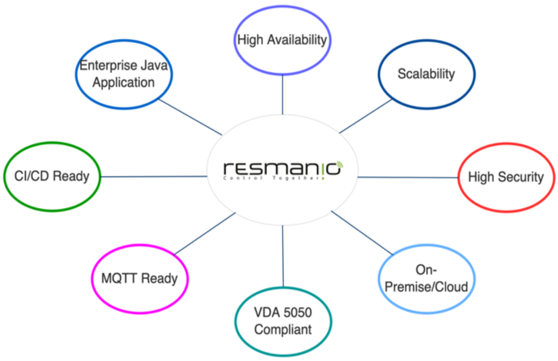


Fig. 2 Resmanio characteristics

high availability, scalability, and security. All components of our solution architecture are scalable to suit many connected devices. Resmanio can also provide on-premises or cloud-based built-in services. It has been designed as VDA 5050 compliant and Message Queuing Telemetry Transport (MQTT) ready. All software components are ready to be deployed in a cloud, such as Google, Amazon, Oracle, and Microsoft Azure.

Implemented testbed platform, shown in Fig. 3, can be recognized in the many production plants in which AGVs can operate. AGV is equipped with IoT sensors and connected to a secured dedicated network. The Data Streaming Sources part provides real-time data to the Real-Time Data Processing Platform and the Application Layer. Using Apache Kafka and Apache Spark, Event Streaming, Transformation, and Processing acquire the stream data and ensures their transformation and processing in near real-time.

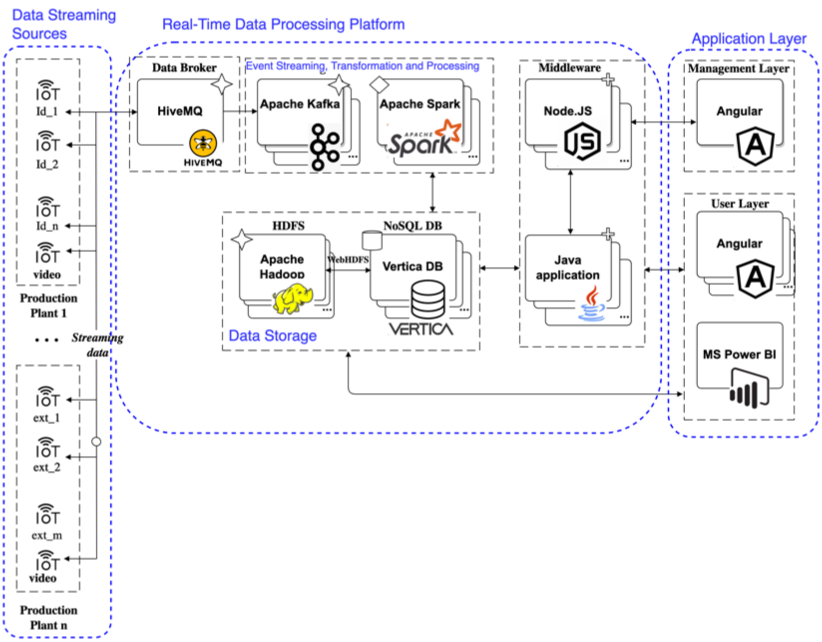


Fig. 3 System architecture design

We identify a layer where individual AGVs are equipped with IoT and location sensors. Information from the sensor is received through a specialized component called a broker, which enables the collection of these events/information in real-time. A Data Broker handles this data via HiveMQ MQTT broker, which connects AGVs and the rest of the implemented platform. It ensures that the information moves between vehicles and testbed platforms implemented on-premises or in a cloud environment. Our solution would allow up to 100,000 events per second within the existing architecture. This component is horizontally scalable. Subsequently, the information from the AGV through the broker passes through a specialized part of the Resmanio system intended for data transformation. Such data is kept in bulk distributed data storage in Apache Hadoop. Vertica is a NoSQL database used for analytics.

Key functionalities and features of this platform are as follows: (1) create standardized real-time communication between different AGVs/AMRs; (2) the unlimited number of

AGV/AMR depending on the size of the floor plan; (3) enable the integration of any AGV/AMR vehicles complying/not complying with the VDA 5050 standard; (4) enable decisions about route selection or intersection behavior using the Resmanio Collision Prevention System; (5) VDA 5050 enabled as a standard interface for all kinds of autonomous robots, not only in the automotive industry; (6) in-built collision prevention system (CPS) enables safe operation in the defined floor plan of the production hall; (7) enables immediate order redirection, order priority and orders reorganization; (8) no predefined path, but the suggested way is the result of the AI/ML system; (9) very high flexibility of route planning and AGV/AMR management; (10) "Plug & Play" functions enable close cooperation with any VDA-compliant vehicle 5050.

Fig. 4 shows communication between UCS Resmanio and AGVs/AMRs to provide a sustainable solution. These AGVs/AMRs can be VDA 5050 compliant or not. This should enable full fledge coverage with legacy AGVs/AMRs, for a guided vehicle. This adapter represents the uniform interface between the USC (Resmanio) and legacy AGVs/AMRs. On this basis, standard communication with AGVs/AMRs integrates transport systems into a continuous process automation. The following features can be provided through the VDA 5050 adapter, which is necessary for managing not compliant AGVs/AMRs: increasing manufacturer independence by using the VDA 5050 adapter, complexity reduction, and increasing vehicle autonomy by consuming existing hardware and software equipment.

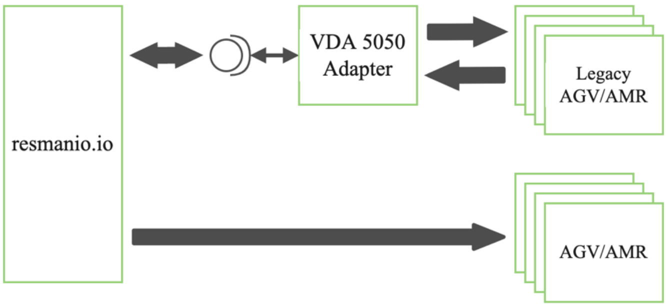


Fig. 4 Resmanio VDA 5050 Adapter

As shown in the sequence diagram for managing all AGVs/AMRs (Fig. 5), there are at least the following actors: (1) USC Resmanio, using resman.io Java Enterprise Application, provides the initial configuration to AGVs/AMRs; (2) VDA 5050 Adapter ensures a seamless communication with non-compliant AGVs/AMRs; (3) MQTT Broker transfer VDA 5050 protocol-based MQTT messages to all AGVSs/AMRs. In the case of VDA 5050 compliant AGVs, instructions or messages generated by resman.io will be broadcasted via MQTT broker to VDA 5050 compliant AGVs/AMRs. Otherwise, these instructions will pass through the VDA 5050 Adapter.

Once the setup of AGVs/AMRs is done, their management is transparent for USC Resmanio in both directions. Some of the key functionalities which Resmanio provides can be recognized as Order planning, Scheduling, Maintenance, Order tracking, Disorders, States of individual devices, AGV report, AGV changes and settings, Delays in orders/processes, Definition of resources or Order Execution which combines action, status, and notification kinds of messages.

As a USC platform, Resmanio provides a feature for integrating with other information systems, such as Enterprise Resource Planning (ERP), Production Service Bus, Messaging-type integration, integration with various identities, Web Sockets (Secured) and Web Services (REST API), application integration, BI and Reporting Server integration, and integration with CAD systems for floor planning design.

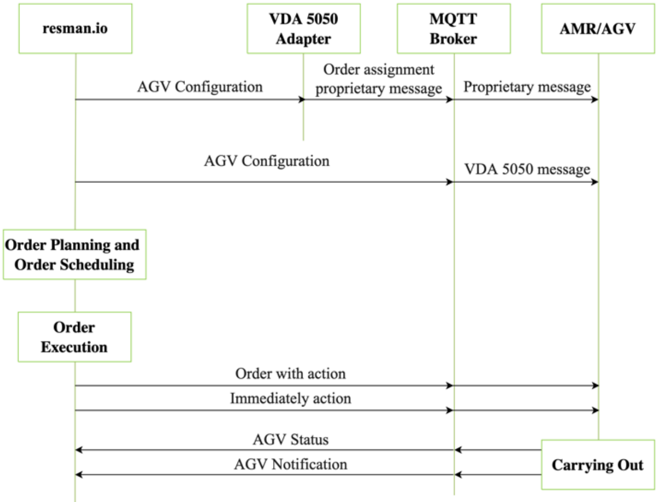


Fig. 5 Time sequence diagram within resman.io and AGV communication using MQTT

The platform's application part (Fig. 6) has the modules realized in a microservice architecture fashion [15]. Distinguished modules are Production Plants, Control Management Centers, Device Settings, Integration, Reporting and Administration. Also, Resmanio provides a central overview within the platform, an overview of all events, order tracking, statuses of individual AGVs/ARMs, and delays or resource definition. We can ensure, for instance, effective AGV planning and control, set special responsibilities within the shifts by fleet, and gather vehicle information or information on progress orders within a production plant. Furthermore, Resmanio offers Floor Plan Designer functionalities inside the modules as Control Management Centers, AGV Management, Order Management, and Fleet Management.

The production line floor layout and various obstacles, such as Barriers, Charge points, Pick points, Delivery points, Emergency and Exits, may be seen in the AGV Management module, shown in Fig. 6. Resmanio provides information about AGVs/AMRs in a real-time fashion. Car production is planned according to shifts based on ordered cars. Using Resmanio as USC makes it possible to plan a shift several weeks in advance. An automated system gives Pick points upon receipt of commodities or specific automobile parts. As a result, AVG is fully automated and capable of supporting the assembly of cars.

The AGV Management module also offers a thorough overview of each AGV/AMR, including identification, battery status, the total distance since the last charge, current status, the dates of the last and subsequent maintenance, and order history with the location of the product factory.

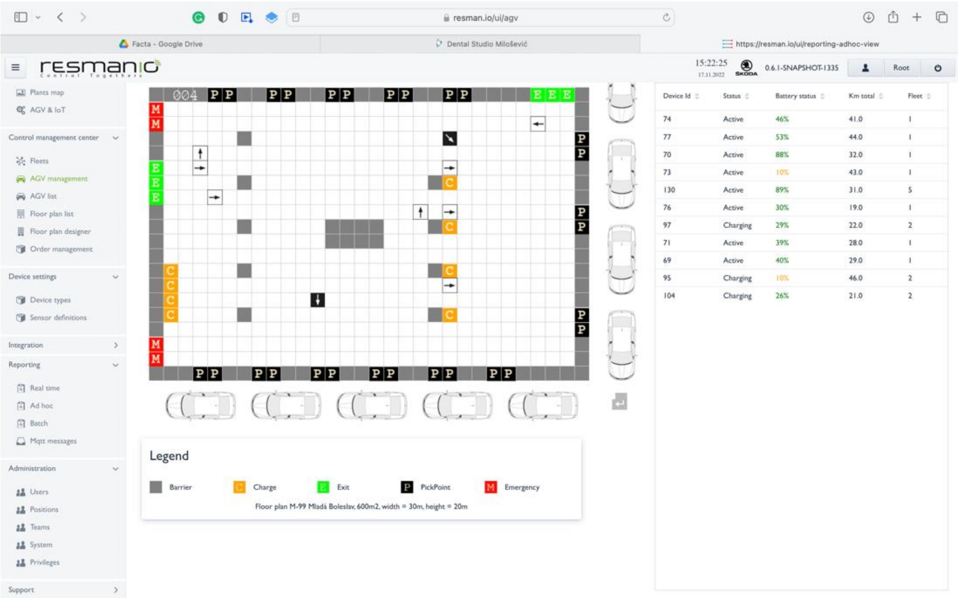


Fig. 6 AGV Management module

Real-time data on orders, users, operating tasks and processes, sensors on AGVs, and monitoring are provided via the Reporting module. There are two report types: batch and online (Fig. 7 and Fig. 8). On a daily, weekly, or monthly basis, these reports offer varied information for all orders or specific orders. Information on VDA messages sent from AGV/AMR to USC or DCS may be found in the online report section (Fig. 8). This report also includes a graph that shows patterns in handling certain orders that have or have had AGVs.

Fig. 7 Reporting Module (batch)

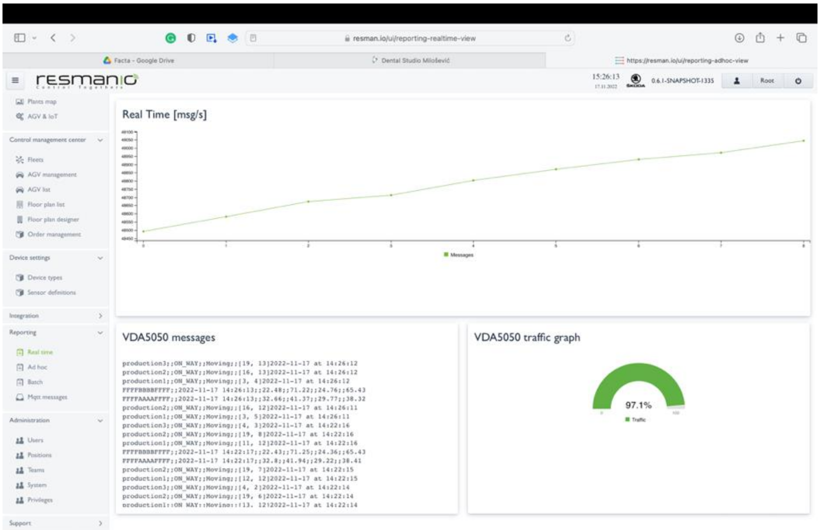


Fig. 8 Reporting Module (online)

4. DISCUSSION

In the case of the testbed platform for Škoda Auto, the proposed platform has to meet various and complex requirements intended to modernize the current system. For example, data streaming and horizontally scalable unified controlling systems intend to advance automotive distributed control systems. The number of conducted AGVs during the test was 150. This is the number of AGVs that Škoda Auto a.s. currently has and we have not recognized performance degradation during the test with the equivalent number of AGV software models.

A Unified Control System as a kind of DSC, branded as Resmanio, can provide AGV status data, reports faults, tracks order material, includes information about order delays, and recalls reports of maintenance intervals. Three modes can be recognized at USC. Emergency mode provides information about process delay and all system data collected through the logs in real-time traffic mode, which allows changing or setting routes. Self-repair mode determines the AGV as a source, determines various navigation procedures for a particular AGV and can be optimized via CI/CD principle. The remaining mode is the Operational mode, which determines the AGV behavior in production plants.

5. CONCLUSION AND FUTURE WORK

In this study, we designed a testbed platform to meet Škoda Auto Requirements and expectations for the horizontal scalable and reliable universal and distributed control system. The presented platform confirms that we can manage heterogenous AGVs by using distributed control system as universal.

The case study reported minimal viable platforms for real or near-real-time stream data processing using HiveMQ messaging broker heavily focused on the consumer-facing web enterprise Java application. All the components used to build this platform are provided as open-source software components except Vertica. Vertica may be swapped out for open-source NoSQL databases like Apache Hive and Casandra for analytics. The Resmanio platform serves as the base for delivering the distinctive value of automotive business.

Future work will provide a solution to support various AGVs. Depending on the Skoda Auto preference, future work is also likely to consider: (1) integration with well-established analytical software used for business intelligence (such as Tableau, Qlik and Microsoft's Power BI); (2) integration with identity management and security enhancement tools (such as CAS Protocol for Single Sign-On and other Network monitoring and Application Performance tools); and (3) distributed processing scenarios with multiple-node performance measures and other technologies such as EMQX broker, RabbitMQ and ActiveMQ messaging systems that could be used for integration via APIs from other components of the Skoda Auto a.s. Universal Control System; (4) the MQTT protocol ensures communication with specific AGVs. The entire solution will be developed with the option of implementing algorithms for artificial intelligence and machine learning needs. Based on these algorithms, it would be possible to use additional functionalities based on a large amount of data. For example, it can predict when any AGV/AMR will be down due to failure or battery life by task.

This business-critical application, built in Java, enables the organization to deliver differentiated products and services inside the automotive industry's production. Examples abound in every industry, including software that manages manufacturing processes, financial transactions, health care delivery, data analytics, and client support.

Acknowledgement: *The Solutia s.r.o. the company from Prague, Czech Republic, has sponsored this study and PoC work. The data collection, benchmarks, and the testbed platform setup required considerable collaborative efforts from Solutia s.r.o. (Prague) employees. We are grateful for their contribution and effort while developing, benchmarking, and setting up infrastructure for these testbed purposes.*
Also, this work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

[1] T. Kalsoom, N. Ramzan, S. Ahmed, and M. Ur-Rehman, "Advances in sensor technologies in the era of smart factory and industry 4.0," *Sensors,* vol. 20, no. 23, p. 6783, 2020, doi: 10.3390/s20236783.
[2] B. Bigliardi, E. Bottani, and G. Casella, "Enabling technologies, application areas and impact of industry 4.0: a bibliographic analysis," *Procedia manufacturing,* vol. 42, pp. 322-326, 2020, doi: 10.1016/j.promfg.2020.02.086.
[3] J. Mehami, M. Nawi, and R. Y. Zhong, "Smart automated guided vehicles for manufacturing in the context of Industry 4.0," *Procedia manufacturing,* vol. 26, pp. 1077-1086, 2018, doi: 10.1016/j.promfg.2018.07.144.
[4] R. Chakma *et al.*, "Navigation and tracking of AGV in ware house via wireless sensor network," in *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC),* 2019, vol. 14, pp. 233-247, doi: 10.1109/CIEEC47146.2019.CIEEC-2019589.
[5] F. Gul and W. Rahiman, "An integrated approach for path planning for mobile robot using bi-RRT," in *IOP Conference Series: Materials Science and Engineering,* 2019, vol. 697, no. 1, doi: 10.1088/1757-899X/697/1/012022.
[6] D. Hrušecká, R. B. Lopes, and E. Juřičková, "Challenges in the introduction of AGVS in production lines: Case studies in the automotive industry," *Serbian Journal of Management,* 2019, doi: 10.5937/sjm14-18064.

[7] N. Correia, L. Teixeira, and A. L. Ramos, "Implementing an AGV system to transport finished goods to the warehouse," *Advances in Science, Technology and Engineering Systems,* vol. 5, no. 2, pp. 241-247, 2020, doi: 10.25046/aj050231.

[8] M. Čech *et al.*, "Autonomous mobile robot technology for supplying assembly lines in the automotive industry," *Acta logistica,* vol. 7, no. 2, pp. 103-109, 2020, doi: 10.22306/al.v7i2.164.

[9] A. F. Costa, M. do Sameiro Carvalho, M. Henriques, and P. V. Ferreira, "Strategy for the introduction of autonomous driving technologies: a case study in the logistics area of an automotive company," *Procedia Computer Science,* vol. 204, pp. 337-345, 2022, doi: 10.1016/j.procs.2022.08.041.

[10] M. Elsisi and M.-Q. Tran, "Development of an IoT architecture based on a deep neural network against cyber attacks for automated guided vehicles," *Sensors,* vol. 21, no. 24, 2021, doi: 10.1016/j.procs.2022.08.041.

[11] R. Patricio and A. Mendes, "Consumption patterns and the advent of automated guided vehicles, and the trends for automated guided vehicles," *Current Robotics Reports,* vol. 1, no. 3, pp. 145-149, 2020, doi: 10.1007/s43154-020-00007-4.

[12] M. Azangoo, A. Taherkordi, J. O. Blech, and V. Vyatkin, "Digital Twin-Assisted Controlling of AGVs in Flexible Manufacturing Environments," in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 2021: IEEE, pp. 1-7, doi: 10.1109/ISIE45552.2021.9576361.

[13] C. L. Garay-Rondero, J. L. Martinez-Flores, N. R. Smith, S. O. C. Morales, and A. Aldrette-Malacara, "Digital supply chain model in Industry 4.0," *Journal of Manufacturing Technology Management,* vol. 31, no. 5, pp. 887-933, 2020, doi: 10.1108/JMTM-08-2018-0280.

[14] K. Gos and W. Zabierowski, "The comparison of microservice and monolithic architecture," in *2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, 2020: IEEE, pp. 150-153, doi: 10.1109/MEMSTECH49584.2020.9109514.

[15] T. Ueda, T. Nakaike, and M. Ohara, "Workload characterization for microservices," in *2016 IEEE international symposium on workload characterization (IISWC)*, 2016: IEEE, pp. 1-10, doi: 10.1109/IISWC.2016.7581269.

FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 21, Nº 3, 2022, pp. 159 - 175
https://doi.org/10.22190/FUACR221118013D

Regular Paper

DATA ANALYSIS OF ENVIRONMENTAL CONDITIONS INFLUENCING THE WORK OF LABORATORY EQUIPMENT AND APPLICATION OF MACHINE LEARNING MODELS FOR CLASSIFICATION OF POOR CONDITIONS

UDC ((004.3/.4:0.034.2)+004.78)

Anđela Đorđević, Miroslav Milovanović, Marko Milojković, Miodrag Spasić

University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Republic of Serbia

Abstract. *Environmental conditions can have a crucial impact on the functioning of laboratory equipment. Electric components are sensitive to the influence of certain environmental factors such as temperature, humidity, vibrations, etc. Environmental factors should, therefore, be monitored to avoid their negative influence on the system and potential faults and failures they could cause. Unlike the traditional approaches which required the presence of special staff to monitor environmental factors and react if they are poor, the rise of the Internet of Things enhanced the application of intelligent solutions where human factor is not necessary. In this paper, research on data analysis, preprocessing and intelligent classification of environmental conditions has been conducted. The data was collected by sensors connected to Raspberry Pi. The applied monitoring system setup enabled long-distance monitoring of laboratory conditions through the internet and full applicability of fundamental IoT concepts. Since data preparation is an important step in the process of designing machine learning models, the collected data was analyzed and preprocessed in Python. Intelligent classification of environmental conditions was performed using machine learning models k-Nearest Neighbors and Random Forest. Grid search was used for model selection, and the performances of k-Nearest Neighbors and Random Forest machine learning models were compared. Experimental results show that these machine learning models can be successfully used for intelligent classification of environmental conditions.*

Key words: *Environmental conditions, Raspberry Pi, Internet of things, Data analysis, Machine learning*

Received November 18, 2022 / Accepted December 21, 2022
Corresponding author: Anđela Đorđević
University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia
E-mail: andjela.djordjevic@elfak.ni.ac.rs

1. INTRODUCTION

Environmental conditions can have a crucial impact on the operation of dynamic systems. Electric components are fundamental units of laboratory equipment, and as such, they can be sensitive to the influence of various external factors. Some of the most important factors to be monitored and controlled are high or low temperatures, humidity, vibrations, air pressure, etc. Beside affecting the functioning of the system, poor environmental factors can cause serious damage on the equipment [1]. For example, high or low temperatures can cause alterations of the electrical component's properties, which could lead to failures and faults of electronic devices. [2]. Also, humidity affects almost every kind of electrical equipment by causing corrosion, changes in electrical resistance, thermal conductivity, capacitance, etc. [2-4]. The effects of vibration should not be neglected, since they can lead to great physical damage on dynamic systems [1,5,6]. Therefore, monitoring and controlling environmental conditions is of crucial importance.

In order to detect undesirable environmental factors, traditional approaches used isolated monitoring units. The main drawback of this approach was the fact that these units were not integrated as a part of a bigger interconnected system. Interconnected intelligent systems of sensors and actuators, which can be applied for monitoring environmental factors, are becoming more and more common with the rise of Internet of Things (IoT) [7-9]. These intelligent approaches give an opportunity to avoid employing special staff to monitor environmental factors and react when the conditions are not satisfying. For example, Usamentiaga et al. proposed a special Industrial Internet of Things architecture for temperature monitoring and fault detection using infrared thermography [7], while Tu et al. proposed using Raspberry Pi to collect data on environmental conditions in a laboratory through various types of sensors and give an adequate response if the conditions are poor [8]. Another interesting research on IoT system applied for environment monitoring and control, can be found in [9]. The authors in [10] conducted research on intelligent classification of environmental conditions. Raspberry Pi was used to collect the data, which was later classified using machine learning models, such as Support Vector Machines, Logistic Regression, *k*-Nearest Neighbors and Decision Tree Classifier [10]. Machine learning models are commonly used as a tool for finding hidden patterns in large datasets, data classification, data mining, etc. Data analysis and preprocessing is an important step in the process of designing machine learning models [11]. Poorly conducted data preparation and analysis can lead to misleading results; thus, this step should be done with special care [11]. In this paper, a detailed analysis of the dataset created using Raspberry Pi was conducted, and the data was later classified using machine learning models Random Forest Classifier and *k*-Nearest Neighbors. In order to find the optimal parameters for each machine learning model, grid search was applied for model selection.

This paper is organized as follows. In Section 2, the description of an IoT system used for monitoring of environmental conditions in a laboratory is given. Also, a short description of Raspberry Pi and the sensors used for data acquisition is presented. Further information on sensing and data acquisition is given in Section 3. Data analysis is described in Section 4 in details, while Section 5 presents approaches used for intelligent classification of optimal and poor conditions. Experimental results are given in Section 6. Accuracy, number of false positives and number of false negatives of machine learning models are compared in this section. Experimental results show that both Random Forest Classifier and *k*-Nearest Neighbors can be successfully used for intelligent classification of environmental conditions. The authors give concluding remarks and discuss the future direction of their research in the last section.

2. IoT System for Detection of Environmental Conditions in a Laboratory

A. Raspberry Pi.

Raspberry Pi is a credit card sized microcomputer. The experiments were conducted on Raspberry Pi 3 B+, which has 1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support (with separate PoE HAT) [12]. Raspberry Pi 3 B+ has Extended 40-pin GPIO header, Full-size HDMI, 4 USB 2.0 ports, CSI camera port for connecting a Raspberry Pi camera, DSI display port for connecting a Raspberry Pi touchscreen display, 4-pole stereo output and composite video port [12]. It has the opportunity of attaching various sensors, collecting data from them and sending it to a server.

Fig. 1 Raspberry Pi 3B+

B. Sensors

In order to monitor environmental factors such as humidity, temperature, vibration, magnetic field, light intensity and fire detection, five sensors were used: Sunfounder Humiture Sensor (humidity and temperature sensor), DFRobot Digital Piezo Disk Vibration Sensor, Sunfounder Analog Hall Sensor, Sunfounder Photoresistor and Sun-founder Flame sensor.

C. Environmental Conditions Monitoring System

The hardware of the environmental conditions monitoring system consists of Raspberry Pi and sensors attached to it to collect data. Raspberry Pi is connected to the local laboratory computer utilized as a mediator between the device and sensors on one side and any computer with access to the local computer on the other. Such setup enables long-distance monitoring of laboratory conditions through the internet and full applicability of fundamental IoT concepts by the realized technical solution. The data analysis and preprocessing were done in Python, and intelligent classification of environmental conditions was performed using machine learning models.

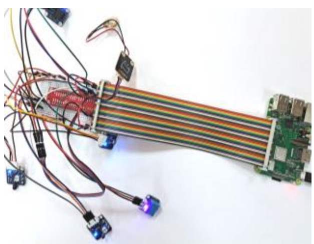


Fig. 2 System used for detection of environmental conditions

3. SENSING AND COLLECTING DATA

In order to create a complete environmental conditions dataset, firstly, information about environmental conditions was recorded when all the conditions were optimal. Secondly, after enough data samples about optimal conditions were collected, the disturbances in the form of poor environmental factors were introduced, concretely in the form of increased humidity and vibration levels. The datasets were then integrated in one unique dataset, which contains information on both optimal and poor conditions. During the recording, sampling time was 0.1 seconds. In total, 12872 samples of data were collected from which 11289 for optimal and 1683 for poor conditions. Fig. 3 shows the information from the humidity sensor throughout time. The blue line in Fig. 3 refers to the period of time when conditions were optimal, while the red line refers to the time period when disturbances were introduced.

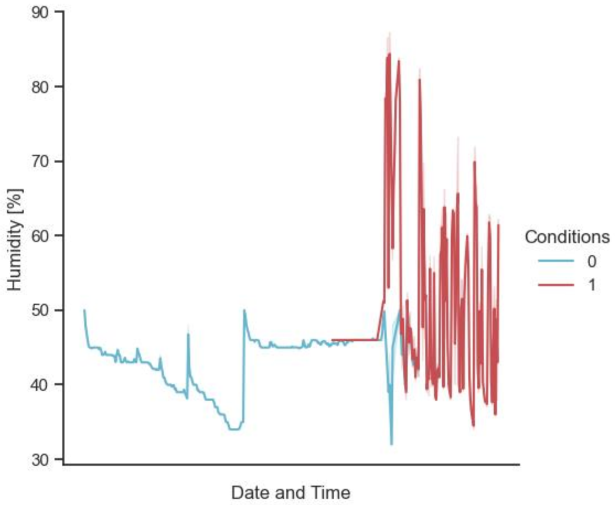


Fig. 3 Information from humidity sensor throughout time

Created dataset included information about the date and time when samples were recorded. Environmental factors are represented through *Humidity* [*%*] (containing information about humidity level), *Temperature* [*'C*] (temperature in degrees Celsius), *Vibration* (strength of vibration), *Hall* (strength of magnetic field), *Photoresistor* (information about light intensity) and *Flame* (existence of fire) columns. The *Conditions* column is of the main interest for the research objectives in this paper and represents the status of environmental conditions in a laboratory. Values in this column were added manually depending on environmental factors. If the values of environmental factors were in optimal range, the conditions were marked as *"optimal"*. On the other hand, if some of the environmental factor's value was not in the optimal range, the conditions were marked as "*poor"*. Values in this column were later mapped in Python with numeric values 0 and 1 respectively. The graphic visualization of the values in *Conditions* column is presented with pie plot in Fig. 4.

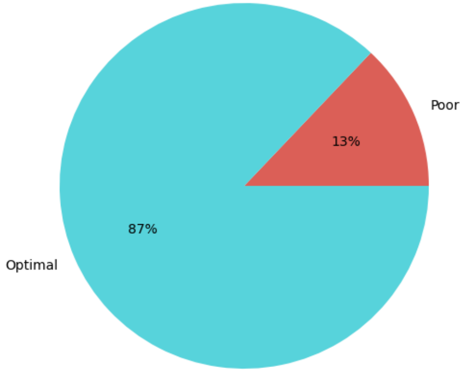


Fig. 4 Pie plot of the decision variable

The pie plot of the decision variable shows that there are much more rows with value *"optimal"* than *"poor"*. This means that significantly more information on factors' values matching optimal than poor environmental conditions were collected. This disproportion could potentially make the classification task harder for some of the machine learning models.

4. Data Analysis in Python

After the data had been collected, the dataset was analyzed and preprocessed in Python. In order to gain an insight into data types and number of rows with Non-Null objects, *Pandas dataframe.info()* function was used. This function was used to find out whether there were some missing values in the dataset. The names of the variables, number of Non-Null objects in them and their data types are presented in Table 1.

Table 1 Names, number of Non-Null objects and data types of the variables

	Feature	Non-Null count	Data type
0	Date and Time	12972	Object
1	Humidity [%]	12972	int64
2	Temperature [‘C]	12972	int64
3	Vibration	12972	float64
4	Hall sensor	12972	int64
5	Photoresistor	12972	int64
6	Flame	12972	int64
7	Conditions	12972	int64
Data types: float 64 (1), int64 (6), object (1)			
Memory usage: 810.9+ KB			

Number of samples, mean value, standard deviation, minimum and maximum value, the median, 25th and 75th percentile for each feature, were obtained using Pandas *dataframe.describe()* function. The results of this function are shown in Table 2.

Table 2 Number of samples, mean value, standard deviation, min. value, 25th percentile, median, 75th percentile and maximum value of the features.

	Number of samples	Mean value	Standard deviation	Min. value	25%	50%	75%	Max. value
Humidity	12972	45.107	8.442719	32	41	45	46	95
Temperature	12972	25.303	1.330145	21	24	25	27	32
Vibration	12972	0.0868	0.451454	0	0	0	0	5
Hall	12972	134.001	0.055518	132	134	134	134	138
Photoresistor	12972	52.0438	30.87137	35	42	44	46	196
Flame	12972	0.00023	0.015206	0	0	0	0	1
Conditions	12972	0.12974	0.336031	0	0	0	0	1

To check the uncertainty of statistics and choose the right representation for the features, bootstrap plots can be used. If the mean value of a feature differs from the median, bootstrap plot could give a clue which estimator of the two would be the superior location estimator. Based on the values in Table 2, all the features had almost the same locations of the mean and the median, except the values of *Photoresistor* features. In the case of this feature, the mean value was 52.0438, while the median value was 44. The bootstrap plots of *Humidity* and *Photoresistor* features are presented in Fig. 5 and Fig. 6 respectively. The bootstrap plot of the *Photoresistor* shows that the median has the smallest variance and would, therefore, be the superior location estimator.

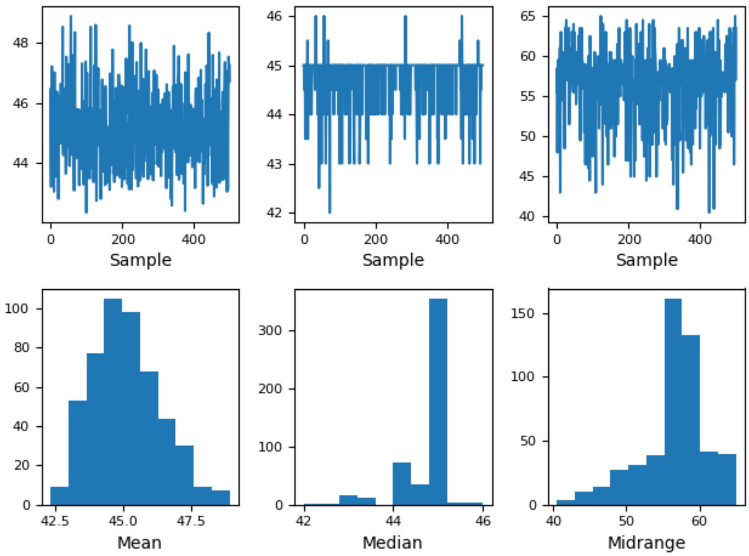


Fig. 5 Bootstrap plot of *Humidity [%]*

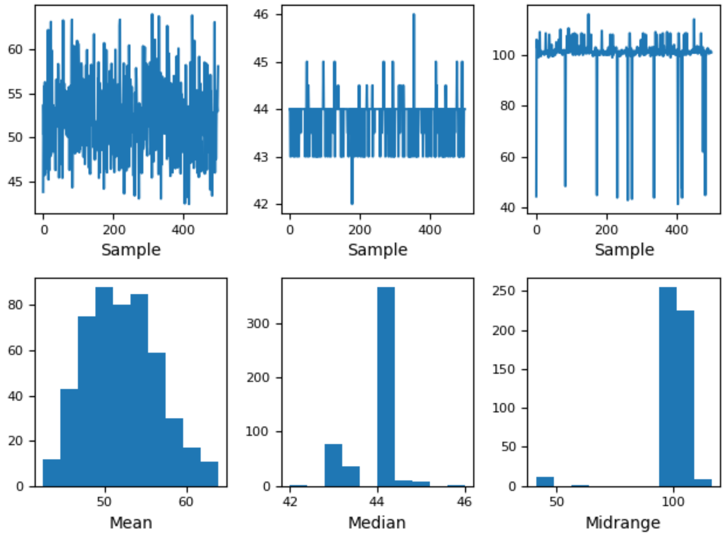


Fig. 6 Bootstrap plot of *Photoresistor*

Data distribution is an important thing to consider when analyzing the dataset. In order to check the distribution of the features, distribution plots were examined for each of the features (for example, Fig. 7 shows the distribution plot of temperature).

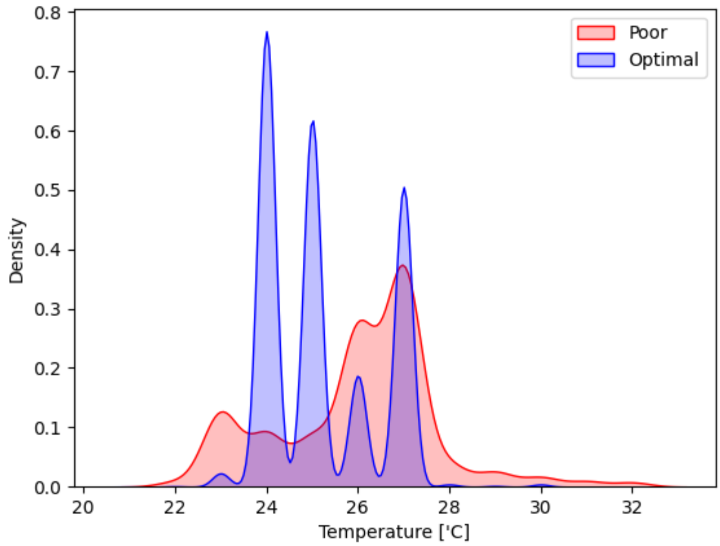


Fig. 7 Distribution plot of *Temperature*: the blue line represents the distribution of optimal conditions, while the red line represents the distribution of poor conditions.

The figure clearly shows that the information from the temperature sensor does not follow normal distribution. Another thing which can be concluded based on the distribution plot of the temperature feature, is the fact that it would be hard to classify environmental conditions simply based on this parameter, since the blue area representing optimal conditions and the red area representing poor conditions are both not following normal distribution, and they overlap. This can also be seen on the Humidity-Temperature joint plot (Fig. 8)

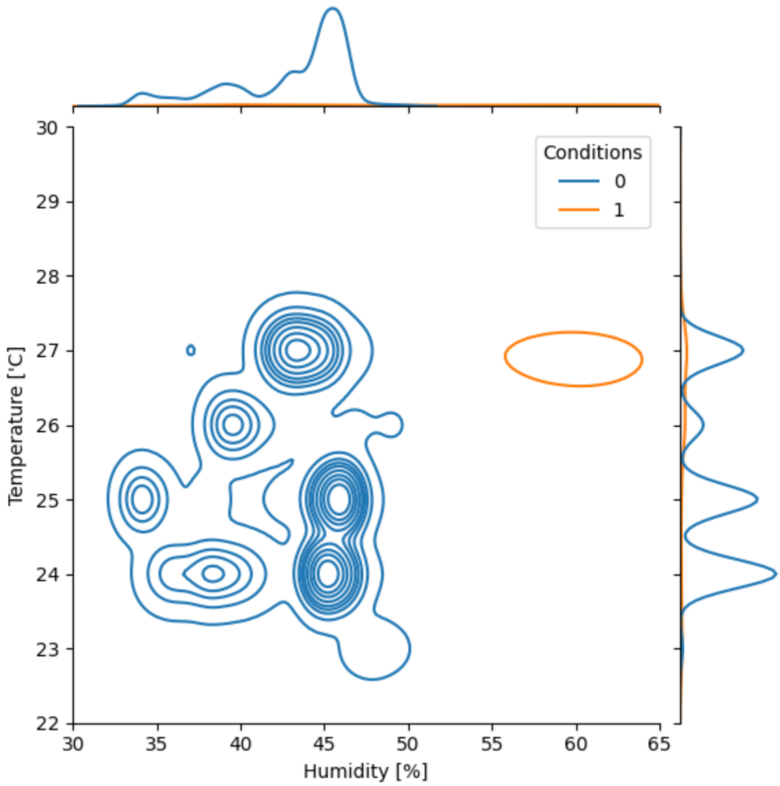


Fig. 8 Joint plot of *Humidity* and *Temperature*

In order to locate outliers, boxplots were used. Boxplots graphically represent the distribution of the data through minimum and maximum value, the first quartile, median and the third quartile. If outliers are present in the data, that implies that some unusual values, which significantly differ from the rest of the dataset are present. In this case, outliers would imply that some environmental factors had unexpected values. Figures 9 and 10 represent boxplots of *Temperature* and *Humidity*.

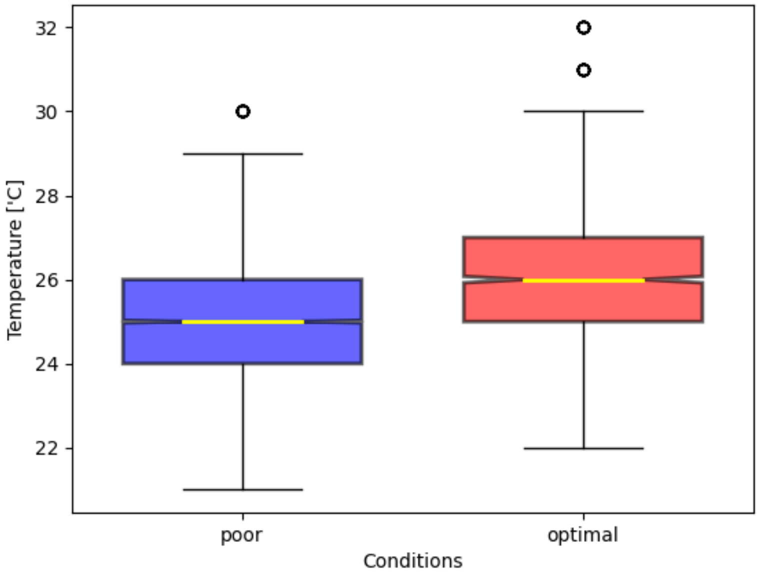


Fig. 9 Boxplot of *Temperature*

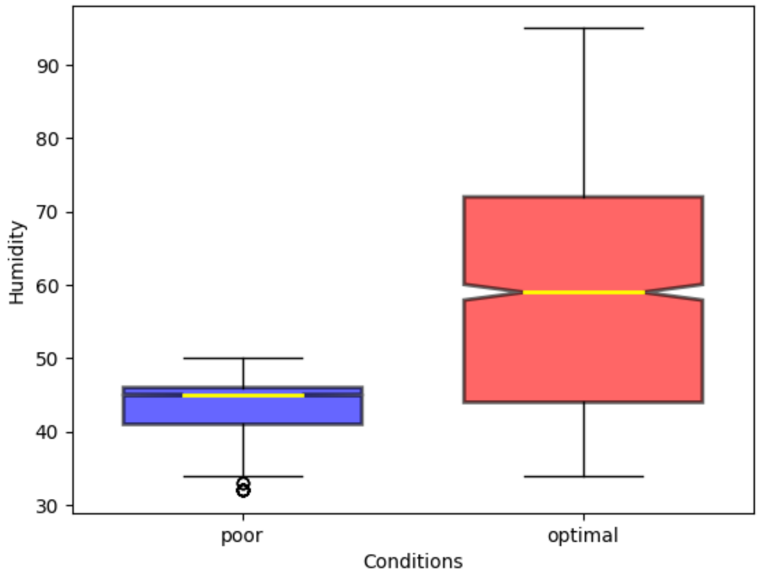


Fig. 10 Boxplot of *Humidity*

Since the distribution of the data was not normal, in order to check the exact values and locations of outliers in the columns, interquartile range method was used [13,14]. As expected, the most outliers were found in humidity and vibration data, while the number of outliers for the rest of the features was insignificant.

In the end, the mutual relations between the variables were analyzed. In order to examine the influence of the parameters on the decision variable, as well as relationships between columns, correlation heatmap function was used to find the correlation index. Correlations between the features and the decision variable are presented in Table 3.

Table 3 Correlation between the decision variable and features.

	Humidity	Vibration	Temperature	Hall	Flame	Photoresistor
Conditions	0.64	0.48	0.2	0.058	0.039	-0.038

The correlation between the data acquiesced from the photoresistor and environmental conditions was insignificant. Also, information from this sensor was irrelevant for the research, and therefore, this feature was dropped from the dataset. Even though the correlation between the data acquiesced from the flame and analog hall sensor and the decision variable was also small, these features were not dropped from the dataset since they can contain useful information about the conditions. For example, flame feature gives an insight on whether a fire exists in a laboratory. If it does, this automatically means that the conditions are poor, which implies that, in reality, the data from flame sensor directly affects the decision variable. Two features with the highest correlation with the decision variable were *Humidity* [%] and *Vibration*.

Finally, autocorrelation of the features was examined. If autocorrelation exists, that means that there is a relationship between the value of a feature at a certain point in time, and its past points. The variables in which autocorrelation was present were *Humidity* and *Temperature* (this was expected because the transition between higher and lower temperatures, for example, cannot be immediate). In order to illustrate this, lag plots of *Humidity*, in which autocorrelation is present, and lag plot of *Vibration* feature, which is much more random, are presented in Fig. 11 and Fig. 12.

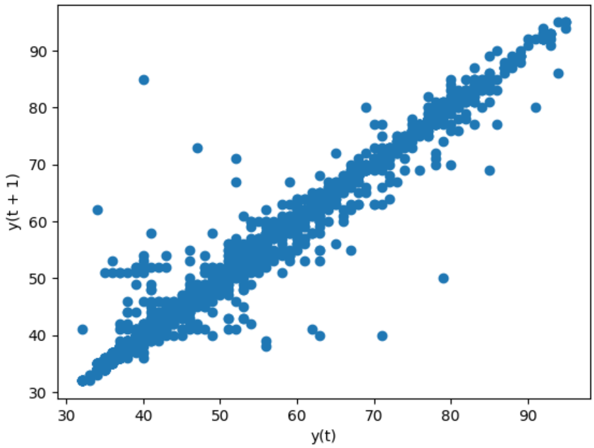


Fig. 11 Lagplot of *Humidity*

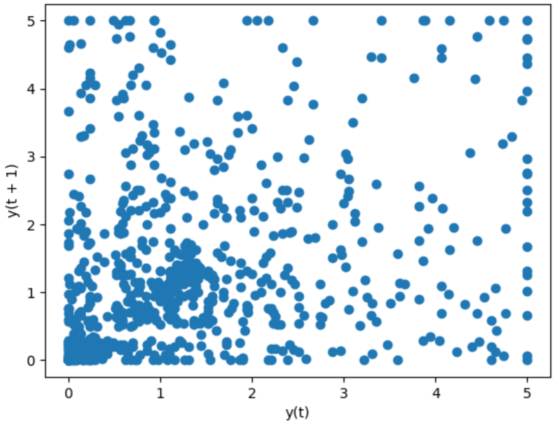


Fig. 12 Lagplot of *Vibration*

5. Intelligent Classification of Environmental Conditions

After the data was analyzed and preprocessed, training and test datasets for machine learning models were formed. Training dataset consisted of 70% of total data samples. After fitting the models, their performance was tested on the remaining 30% of dataset. The machine learning models used in the experiments were k-Nearest Neighbors and Random Forest Classifier. Both *k*-Nearest Neighbors (KNN) and Random Forest Classifier are popular machine learning models which give satisfying results for classification problems [15-18]. In order to select optimal models, grid search was performed. The performance of KNN algorithm depends heavily on the number of neighbors used in the classification process. The models with lower values of *k* used in the classification process tend to lead overfit the data and fail to generalize [18]. Generally, the value of the bias is lower, but the variance is higher than when larger number of neighbors is used. The influence of the number of neighbors on the capability of KNN model to classify the conditions was examined. Grid search was applied to choose the right value of *k*. In order to select Random Forest classifier model, attention was focused on the variations of two parameters: number of estimators and minimum number of samples per leaf node. To choose the optimal values of these parameters, grid search was applied.

6. Results

The prediction of the decision variable "Conditions" was performed based on the values of environmental factors in the test dataset, and the results were compared to the real values. Firstly, grid search was applied in order to set the parameters of the models. Accuracy of the prediction of *k*-Nearest Neighbors for different *k* values were compared in order to choose the best model. Fig. 14 shows the error rate for different values of number of neighbors.

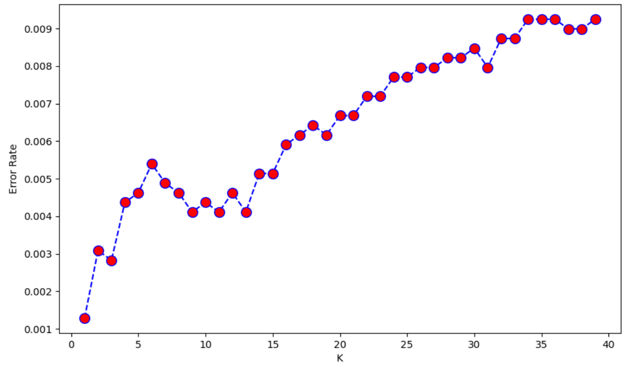


Fig. 13 Error rate for different *k* values

It can be concluded that the error rate of the models becomes higher as *k* increases. In order to investigate this further, the relationship between *k* values and the number of false positives (the number of optimal conditions falsely classified as poor) and the number of false negatives (the number of poor conditions falsely classified as optimal) was examined. Fig. 15 shows the number of false positives (blue line) and the number of false negatives (red line) depending on *k*.

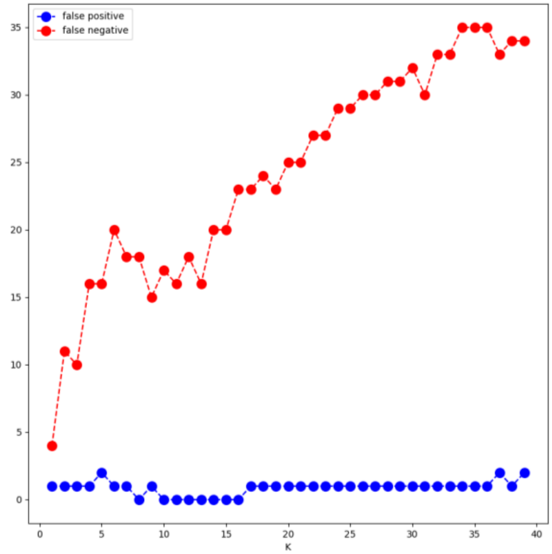


Fig. 14 Number of false positives (blue) and number of false negatives (red) for different *k* values

While the number of false positives is smaller than 3 for all *k* values, the number of false negatives increases. The figure shows that the algorithm had a bigger problem with classifying poor than optimal conditions. This problem occurs due to the fact that the majority of the collected data referred to the optimal conditions.

The accuracy, number of false positives and number of false negatives of *k*-Nearest Neighbors models for *k* = 1, 3, 5, 10, 20, 40 is presented in Table 4. The best accuracy and the smallest number of false negatives were obtained for *k* = 1. Still, *k* = 1 may not be the optimal solution, since it can lead to overfitting and poor generalization of data. A smaller number of neighbors leads to a smaller bias, but the larger variance. On the other hand, for larger values of *k*, variance decreases, but the bias increases. Therefore, a trade-off should be made to avoid overfitting or underfitting the data. Bearing that in mind, k = 10 was chosen as the optimal number of neighbors in this case.

Table 4 Accuracy, number of true negatives, number of false positives, number of true positives and number of false negatives for different values of *k*.

	Accuracy [%]	Number of true negatives	Number of false positives	Number of true positives	Number of false negatives
k = 1	99.8715313464	3395	1	492	4
k = 3	99.7173689620	3395	1	486	10
k = 5	99.5375128469	3394	2	480	16
k = 10	99.5632065776	3396	0	479	17
k = 20	99.3319630010	3395	1	471	25
k = 40	99.0750256937	3394	2	462	34

Another machine learning model used for the prediction was Random Forest Classifier. In this case, grid search was applied to find the optimal number of estimators (*n_estimators*) and the minimum number of samples per leaf (*min_samples_leaf*). Models with *n_estimators* =10, 25, 50, 75, 100, 150, 200, 250 and *min_samples_leaf* = 1, 5, 10, 20 were compared. Fig. 15 and Fig. 16 show the accuracy of different models depending on the number of estimators and minimum samples per leaf, while Fig. 17 shows the mean fitting time.

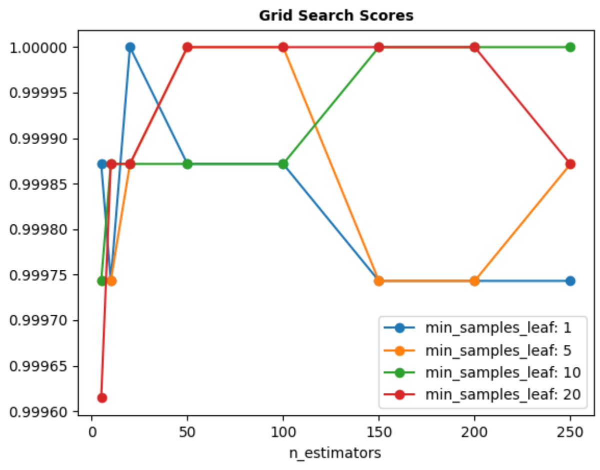


Fig. 15 Accuracy of Random Forest Classifiers depending on the number of estimators and minimum samples per leaf

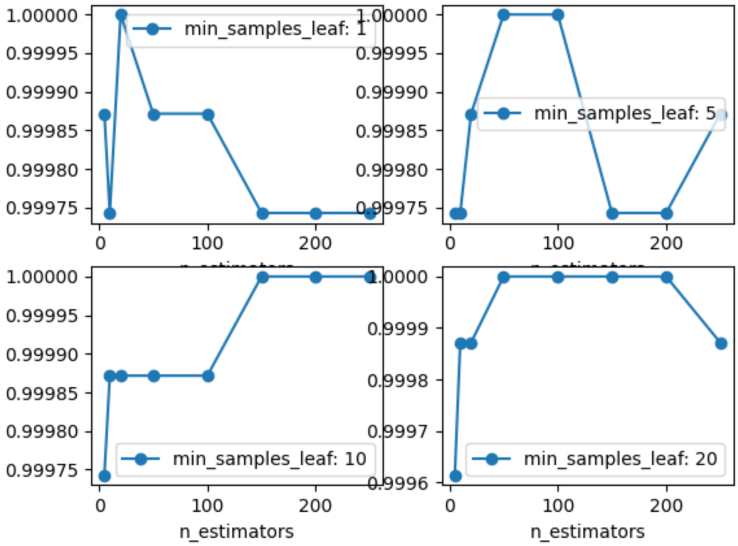


Fig. 16 Accuracy score of Random Forest Classifiers depending on number of estimators for min_samples_leaf={1,5,10,20}

The default value of the number of samples per leaf is one. In general, this number is increased to avoid overfitting the data (tree pruning), and therefore poor prediction. The graph shows a slight decrease of the accuracy of the models with the increase of the number of estimators for *min_samples_leaf* =1 and *min_samples_leaf* =5, while the accuracy of the models with *min_samples_leaf* =10 and *min_samples_leaf* =20 slightly increases.

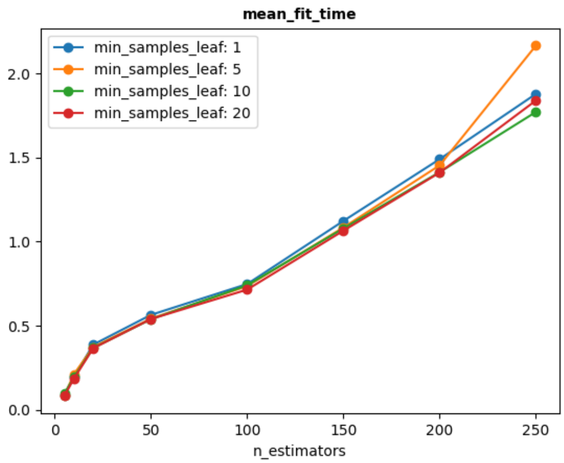


Fig. 17 Mean fitting time for each of the Random Forest Classifiers

As expected, the fitting time increased as the number of estimators increased, while the number of samples per leaf did not affect it as much.

The model which proved to be the best choice was the one with 50 estimators and one sample per leaf. Table 5 shows the accuracy, number of true negatives, number of false positives, number of true positives and number of false negatives for the best *k*-Nearest Neighbors and Random Forest Classifier model.

Table 5 Accuracy, number of true negatives, number of false positives, number of true positives and number of false negatives for *k*-Nearest Neighbors and Random Forest Classifier.

	Accuracy [%]	Number of true negatives	Number of false positives	Number of true positives	Number of false negatives
k Nearest Neighbors (k=10)	99.5632065776	3396	0	479	17
Random Forest Classifier (n_estimators=50, min_samples_leaf=1)	99.9807284641	3396	0	495	1

Even though the accuracy of Random Forest Classifier is only slightly better than the accuracy of *k*-Nearest Neighbors, *k*-Nearest Neighbors model showed a greater tendency to falsely classify *poor* conditions as *optimal*, which could seriously influence the performance of the equipment and electrical components in the laboratory. Because of that tendency and since it falsely classified only one poor condition as optimal, Random Forest Classifier would be the better choice in this case. Still, the difference in classification is small and both models performed with good accuracy.

7. Conclusion and Future Work

This paper investigates data analysis, data preprocessing and intelligent classification of environmental conditions in a laboratory. Laboratory equipment is susceptible to the influence of environmental conditions. Raspberry Pi and several sensors measuring environmental factors were used for monitoring environmental conditions in a laboratory. Firstly, the data recording was done when the conditions were optimal. Afterwards the disturbances in the form of poor environmental factors were introduced. These recordings were used to form a dataset, which was later analyzed and preprocessed in Python. When the data preparation was completed, intelligent classification of environmental conditions was conducted using *k*-Nearest Neighbors and Random Forest Classifier. In order to select optimal models, a grid search was used to set the number of neighbors for the KNN model, and number of estimators and minimum samples per leaf for Random Forest Classifier model. The accuracy of both KNN and Random Forest Classifier models was satisfying, but the results have shown that the KNN algorithm had a bigger problem with classifying poor conditions. This issue occurred because the majority of the collected data referred to the optimal conditions. Therefore, Random Forest Classifier proved to be a better choice in this case, even though the difference in classification accuracy was almost insignificant.

This work is only the beginning of our research. In the future research, we plan on working on a new method for detecting direct influence of environmental conditions on the performance of dynamical systems. Our objective is to propose intelligent approaches to prevent the influence of the disturbances on operation and performance of dynamic systems. In our future research, we will try to analyze and process the detected environmental disturbances which affect the work of the system, using new intelligent algorithms. The results would be used as an input in the control logic and would be employed to minimize the negative influence of environmental conditions.

Acknowledgement: *Funded by the European Union under project Smart Products and Services Engineering (SPaSE) 101047566-ERASMUS-JMO-2021-MODULE. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or EACEA. Neither the European Union nor the granting authority can be held responsible for them. This work has also been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.*

REFERENCES

[1] N. Driendl, F. Pauli and K. Hameyer, "Influence of Ambient Conditions on the Qualification Tests of the Interturn Insulation in Low-Voltage Electrical Machines," in *IEEE Transactions on Industrial Electronics*, vol. 69, no. 8, pp. 7807-7816, Aug. 2022, doi: 10.1109/TIE.2021.3108721.

[2] F. A. L. Souza, P. C. T. Pereira, H. de Paula, B. J. C. Filho and A. V. Rocha, "Motor drive systems reliability: Impact of the environment conditions on the electronic component failure rates," *2014 IEEE Industry Application Society Annual Meeting*, 2014, pp. 1-8, doi: 10.1109/IAS.2014.6978463.

[3] M. S. S. Chani, K. S. Karimov, A. M. Asiri, M. M. Rahman and T. Kamal, "Effect of Vibrations, Displacement, Pressure, Temperature and Humidity on the Resistance and Impedance of the Shockproof Resistors Based on Rubber and Jelly (NiPc-CNT-Oil) Composites," in *Gels*, vol. 8, no. 4, 226, Apr. 2022, doi: 10.3390/gels8040226.

[4] M. Shooshtari, A. Salehi and S. Vollebregt, "Effect of Humidity on Gas Sensing Performance of Carbon Nanotube Gas Sensors Operated at Room Temperature," in *IEEE Sensors Journal*, vol. 21, no. 5, pp. 5763-5770, 1 March1, 2021, doi: 10.1109/JSEN.2020.3038647.

[5] R. A. Amy, G. S. Aglietti and G. Richardson, "Reliability Analysis of Electronic Equipment Subjected to Shock and Vibration – A Review," *Shock and Vibration*, vol. 16, no. 1, 2009, pp. 45-59, doi: 10.3233/SAV-2009-0453.

[6] A. M. Veprik, "Vibration protection of critical components of electronic equipment in harsh environmental conditions," *Journal of Sound and Vibration*, vol. 259, no. 1, 2003, pp. 161-175, doi: 10.1006/jsvi.2002.5164.

[7] R. Usamentiaga, M. A. Fernandez, A. F. Villan and J. L. Carus, "Temperature Monitoring for Electrical Substations Using Infrared Thermography: Architecture for Industrial Internet of Things," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5667-5677, Dec. 2018, doi: 10.1109/TII.2018.2868452.

[8] Z. -x. Tu, C. -c. Hong and H. Feng, "EMACS: Design and implementation of indoor environment monitoring and control system," *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 2017, pp. 305-309, doi: 10.1109/ICIS.2017.7960010.

[9] A. A. Jaber, F. K. I. Al-Mousawi and H. B. Jasem, "Internet of things based industrial environment monitoring and control: a design approach," in *International Journal of Electrical & Computer Engineering (2088-8708)*, pp. 4657-4667, vol. 9. no. 6, Dec. 2019, doi: 10.11591/ijece.v9i6.pp4657-4667.

[10] A. D. Đorđević, M. B. Milovanović, M. T. Milojković, M. D. Spasić, "Intelligent Classification of Environmental Conditions Influencing the Work of Laboratory Equipment", *Proceedings of the XVI International Conference on Systems*, Automatic Control and Measurements, SAUM 2022, Niš, Serbia, November 17.-18., 2022., Publisher: Faculty of Electronic Engineering, Niš, Faculty of Mechanical Engineering, Niš, Serbia (Accepted for publication)

[11] S. Garcia, J. Luengo and F. Herrera, "Data preprocessing in data mining", vol. 72, Cham, Switzerland: Springer International Publishing, 2015.

[12] *Raspberry Pi 3 Model B+*, [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/, visited in June 2022.
[13] D. L. Whaley III, "The interquartile range: Theory and estimation", PhD Thesis, East Tennessee State University, 2005.
[14] H. M. Kaltenbach, "A concise guide to statistics", Springer Science & Business Media, 2011.
[15] L. Jiang, Z. Cai, D. Wang and S. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007, pp. 679-683, doi: 10.1109/FSKD.2007.552.
[16] O. Kramer, "Dimensionality reduction with unsupervised nearest neighbors.", vol. 51. Berlin: Springer, 2013.
[17] S. Sathe and C. C. Aggarwal, "Nearest Neighbor Classifiers Versus Random Forests and Support Vector Machines," 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp. 1300-1305, doi: 10.1109/ICDM.2019.00164.
[18] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," 2017 International Conference on Data and Software Engineering (ICoDSE), 2017, pp. 1-6, doi: 10.1109/ICODSE.2017.8285847.

FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 21, Nº 3, 2022, pp. 177 - 186
https://doi.org/10.22190/FUACR221118014K

Regular Paper

DATA COLLECTION TOOL FOR PROCESS IDENTIFICATION USING PLC AND KEPWARE TOOLS

UDC ((681.5.015:004.65) + 681.3.06)

Igor Kocić[1], Saša S. Nikolić[1], Darko Mitić[1],
Petar Đekić[2], Nikola Danković[1], Nebojša Jotović[1]

[1]University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Republic of Serbia
[2]The Academy of Applied Technical and Preschool Studies-Niš, Republic of Serbia

Abstract. *In this paper, we are using the Kepware tools as a data collection tool to collecting data for identifying the process and making the mathematical models is described in this paper. The connection to the process was made using a programmable logic controller like a real time DAQ (Data Acquisition Board and System) device, database and a client application developed for these purposes. Special attention is paid to the possibility of collecting a large amount of process data as the same moment time of data sampling.*

Key words: *Kepware, PLC, OPC server, MySQL database, Identification*

1. INTRODUCTION

Kepware solutions enabled the connection of various devices for process monitoring and thus enable the concept of the IoT (Internet of Things) [1]. Kepware have a library with multiple device drivers for connection with devices from various hardware manufacturers, like Allen Bradley, Siemens, ABB, Mitsubishi, Omron, Schneider and others. Within the Keepware tools, it is possible to communicate with various types of devices and various network protocols. Programmable logic controllers (PLCs) have been used in the implementation of control tasks. During the realization of regulation tasks, we need to first identify the process as well as possible. For this reason, it is convenient to use Kepware tools [1], PLC and data base for collection of data and after that make a mathematical model of the process. This PLC controller is used to control the same process.

Received November 18, 2022 / Accepted December 19, 2022
Corresponding author: Saša S. Nikolić
University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia
E-mail: sasa.s.nikolic@elfak.ni.ac.rs

Kepware fastest reliable sampling speed is 10 ms, but the real speed of sampling is lower than that.

This paper describes the possibility of collecting data to identify the process on the example of reading the values of specific tags (arrays) of the PLC controller using Kepware tools, MySQL [3] database, plc program and client program for reading, viewing, graphical display, export and printing data from the database that was written in Delphi programming language.

The connection of the PLC controller and Kepware tools is done using its software component Channels/Devices, within which it is easy to configure communication channels and devices from which data are sampled and collected for process identification and making of models. Channel configuration is done by selecting the appropriate communication driver. Kepware tools are used for connecting to the database using its Data Logger software component by selecting the appropriate driver for the ODBC database. The collected data is read from the field of database tables, using Structure Query Language (SQL) or a client application. Free software tools SQLyog [5] is used to write queries on database.

2. Hardware and Software Environment

Kepware product is installed and running on Windows 7 Professional system. OS (Operating systems) Win 7 is general purpose OS system, it does not guarantee the task execution in setting times like a Real time OS [1]. To overcome the problem of real-time operation, a PLC is used that operates and executes instructions in real-time. A MySQL database was installed and setup on the same computer. Client application can be installed and developed on the other Windows OS. The client application is not demanding in terms of the performance of the computer on which it is installed, it can be installed on any computer within the local Ethernet network.

Database for stored of data collection is installed using the multiplatform free open source software XAMPP [3, 4] which is use in the local Ethernet network. XAMPP provides a solution for fast and efficient configuration of the minimum environment that meets all the needs in terms of database application development. In Fig. 1 the basic block diagram of the connection of the Kepware tools with the PLC controller, database and client application with process is shown. Figure 2 shows the layout of the initial screen for setting up the Kepware server communication channel driver and device properties. An example driver was selected to communicate with the Allen Bradley PLC controller using Ethernet [2].

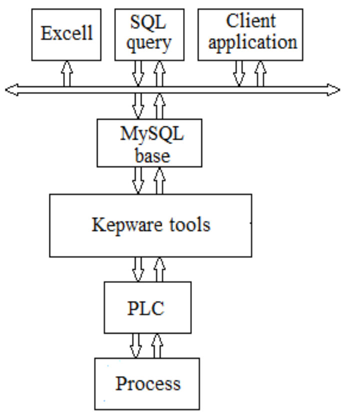


Fig. 1 Block diagram of Kepware tools and PLC connections with process, database and client application

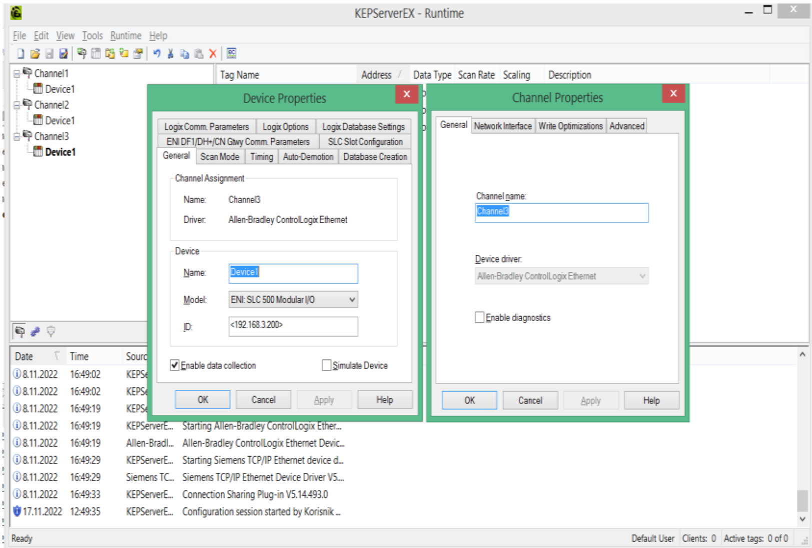


Fig. 2 Selection of communication channel driver and type of PLC device

PLC controller from the SLC 500 series of modular type is connected to the Ethernet network, its IP address, communication port, maximum connection duration, number of connection attempts, and given approval for data collection. Figure 3 shows a window in which the tag from which the data is collected is selected. A tag's array from address N12:160 to N12:163 was entered, which contains the current value of the temperature from zone 1 to zone 4 and others zone of the extruder. The reading (sampling) time is set to 1 s and the access mode is set to read.

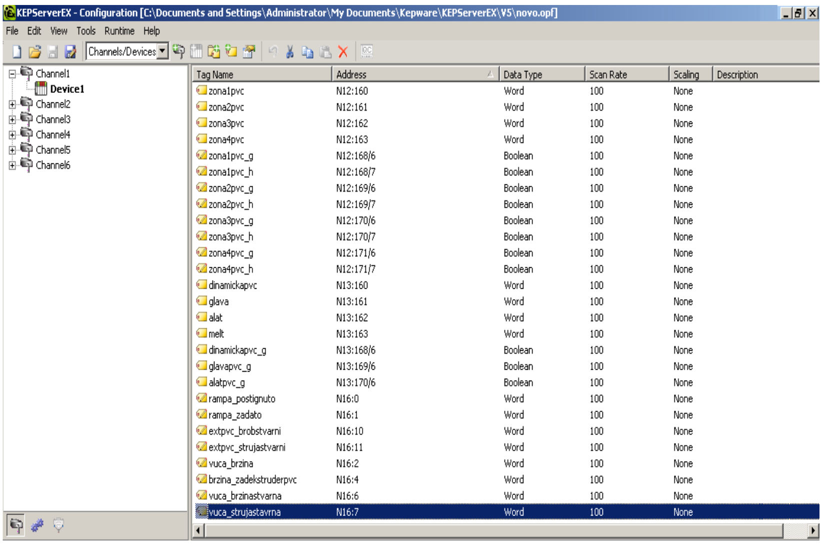


Fig. 3 Defining the parameters of the tag (PLC register)

The SQLyog is an open-source software by which database and appropriate tables can be created. In Fig. 4 the database table and fields in which the current values of temperatures are entered when identifying the temperature model of the extruder are shown. It is easy to see the possibility of simultaneous collection of current values of a large number of parameters important for process identification. The current values of the temperature of the extruder zones, read from the selected PLC controller, are entered in the fields of the table thus created Writing simple SQL queries reviews and exports data and, if necessary, import it into Matlab. Table row in id tags that is set to be the primary key, allows data to be viewed with sorting and time selection. MySQL database and DataLogger connected with the appropriate database driver. The DataLogger is linked to the tag values read by the KEPWareEX server and entered in the appropriate fields in the database table. Linking the tag with the corresponding field of the table is done by mapping and linking the name of the PLC tags and the name of the table field (see Fig. 5). The read value of the selected PLC tag is entered in the selected table field.

When started Kepware begins reading values and writing to the corresponding fields in the table of the selected database. Client application was written in the Delphi programming language. This application simplifies the process of collecting, reading, reviewing and exporting data. Ready-made Delphi components for working with databases, such as TADOConnection, TADOQuery, TADOTable and others, were used to write the program.

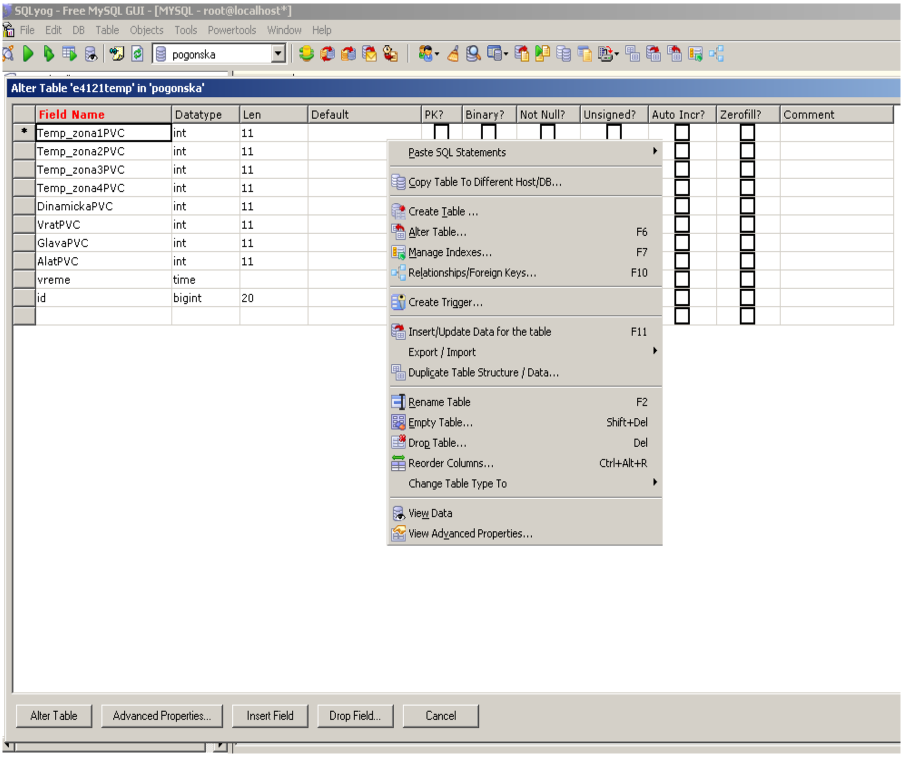


Fig. 4 Defining the database table and field where the values of appropriated PLC tags was stored

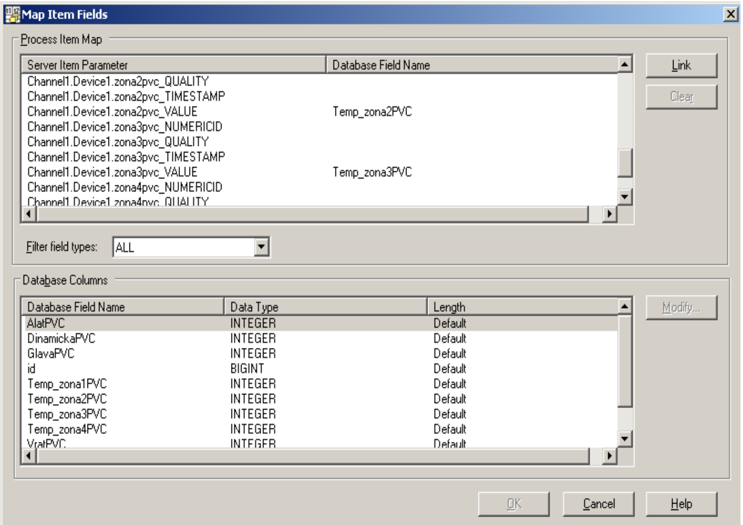


Fig. 5 Associated fields from the PLC tag database using Data Logger

In the PLC controller, a short routine (see Fig. 6) is written that performs the acquisition of the extruder temperatures. A Selectable Timed Interrupt Subroutine (STI) with an execution time of 1s is used. After the execution of the routine, the values of the current temperatures are written in the fields, which are read directly from the registers corresponding to the physical inputs to which the thermocouples are connected (TC type J). The STI program file number must be 8. In this case the enable bit is set to 1. If STI is executing often, the overall scan time may dramatically increase.

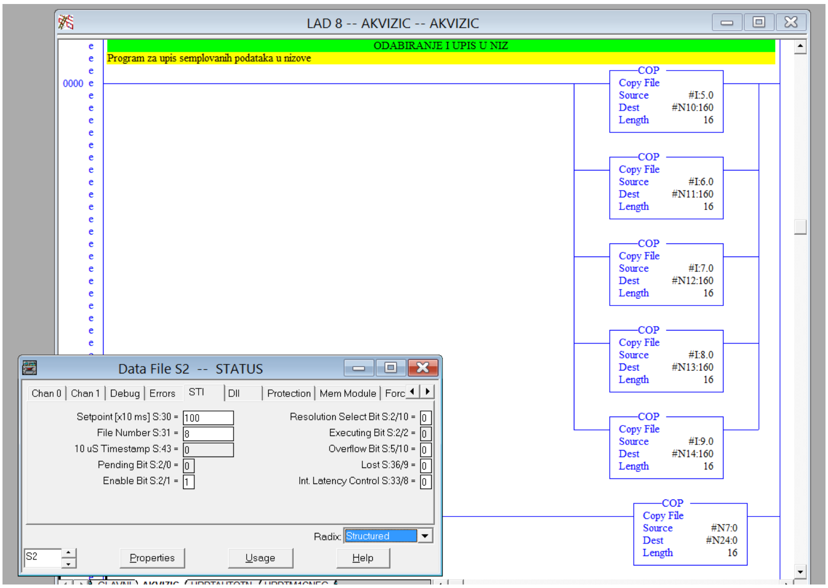


Fig. 6 PLC routine for written current temperatures in arrays

SQL commands are used to create, expand created tables and indexes. The software uses the SQL statements to view data from the database.

Figure 7 shows the layout of the screen of the client application for reading and processing data from the database. Using the client application [6-9], the data from the selected table and column of the table is read, entering the desired time interval in which the review and reading is performed. In a specific example for the identification and development of a mathematical model, the extrusion line is using, and data on the number of turns of the screw extruder that affects the speed and flow of mass are collected, thus the temperature model [6].

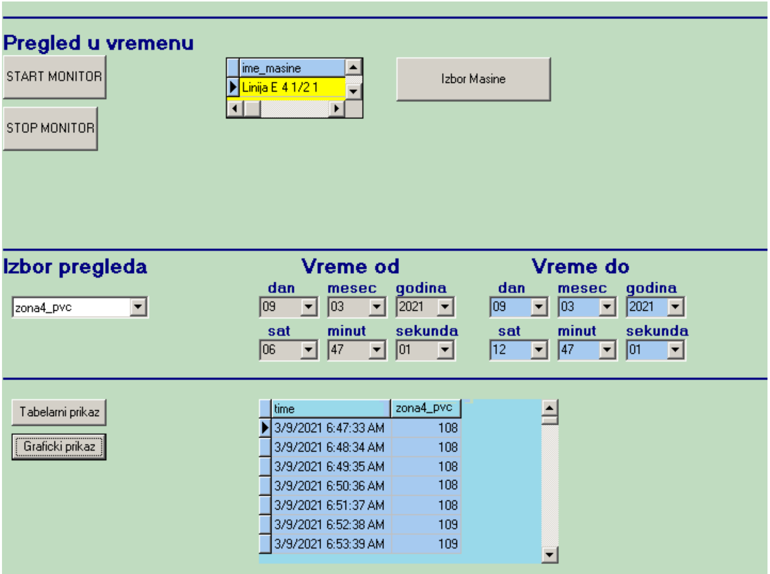


Fig. 7 Part of the application for selection and tabulation of data from the database

The part of the Delphi [10, 11] client application code that reads the selected table field from the database is listed below.

```
procedure TForm1.Button3Click(Sender: TObject);
var Datum1,Vreme1,Datum2,Vreme2,timeer,timeerdo:String;
begin
  if ComboBox13.Text='IZABERI' then
    ShowMessage('Izbor pregleda nije u redu,ponovi izbor')
    else
      begin
Datum1:=ComboBox3.Text+ComboBox2.Text+ComboBox1.Text;
Vreme1:=ComboBox4.Text+ComboBox5.Text+ComboBox6.Text;
Datum2:=ComboBox9.Text+ComboBox8.Text+ComboBox7.Text;
Vreme2:=ComboBox10.Text+ComboBox12.Text+ComboBox11.Text;
  timeer:= Datum1+Vreme1;
  timeerdo:= Datum2+Vreme2;
  kolname:=Edit6.Text;
  ime_tabele:=Edit3.Text;
  Edit1.Text:=timeer;
  Edit2.Text:=timeerdo;
  ADOQuery1.SQL.Clear;
```

```
    ADOQuery1.SQL.Add('select time,'+ kolname + ' from '+ ime_tabele);
    ADOQuery1.SQL.Add('where time >'+timeer);
    ADOQuery1.SQL.Add('and time <'+timeerdo);
    ADOQuery1.SQL.Add('order by vreme');
    ADOQuery1.ExecSQL;
    ADOQuery1.Active:=True;
      end;
    DbGrid3.Visible:=True;
      DbChart1.Title.Text.DelimitedText:=ime_tabele+' '+kolname;
      Form2.DbChart1.Title.Text.DelimitedText:=ime_tabele+' '+kolname;
    Button4.Visible:=True;
  end;
```

By entering the time interval and selecting the tabular display, the query is executed and the table with the values of the zone 4 in the selected time interval is displayed when the model characteristic was recorded. As part of the process identification, in the part of recording the process characteristics, the zone 4 of the extruders was identified, the possibility of identifying only one process parameter is presented.

A table IdentTabela was created on the basis of which a field named TZona4 was created. The identification process consists of several steps. The start time of the experiment is noted. Contactor that turn on the heating are switched on manually, thus simulating a Heaviside step function [7, 8]. The sampling time was set at 1s because it is a slow temperature process and the time for query is set at 1 min. The communication channel of the Kepware is activated, which reads the value of the tag in which the current values of the temperature of zone 4 are located. At the end of the experiment, the time is recorded. In the client application, the IdentTable table, the TZone 4 field and the time from the beginning to the end of the experiment are selected and the tabular value of zone 4 temperature is obtained. By pressing the "Graphic display" button, the temperature characteristic of zone 4 is drawn as a dependence of the temperature on the time on the basis of which the process identification is performed Figure 8.

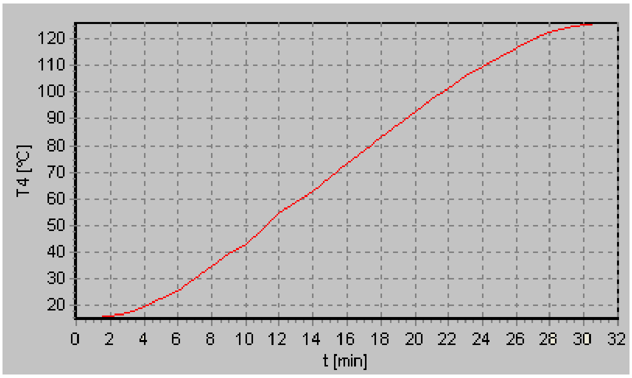


Fig. 8 Temperature characteristic of zone 4

Recorded values using SQL queries or using a client application are transferred to *.txt*, *.csv* format or exported to Matlab or Win Excell. As part of the process of identifying the extruder model, the procedure of recording the characteristics of several zones for the purposes of creating a temperature model of the extruder with the mutual influence of the zones was applied, and a similar procedure was applied. The time of the beginning of the identification is noted. The contactor that a turn on the heating is forced on manually, which simulates the bounce function, the sampling time is set to 1 min. First, zone 2 is heated and its characteristics are recorded. The time of completion of identification is recorded. The extruder is then cooled to ambient temperature, the process is repeated with the difference that zones 1, 2, and 3 are now heated simultaneously and their temperature characteristics are recorded at the same time. It is possible to easily examine the mutual influence of zones and create a model of a multivariable system. Figures 9 and 10 shows the characteristics of zone 2 that is heated by it, and shows the characteristics of zones 1, 2 and 3 that are heated at the same time. Both characteristics were obtained using a client application. The extruder is heated to a temperature of 100 °C, which corresponds to a time of about 25 min with ambient temperature of 17 °C.

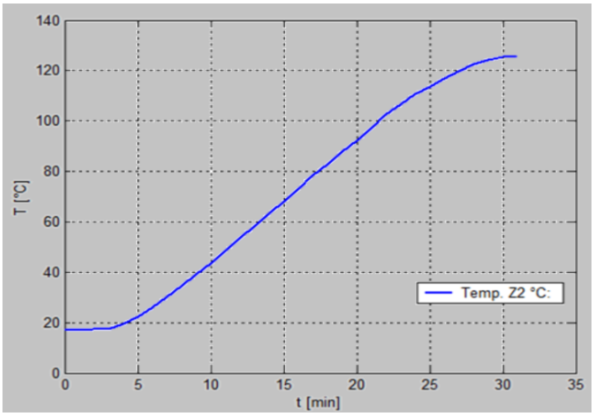


Fig. 9 Characteristic of Zone 2 only, heated only

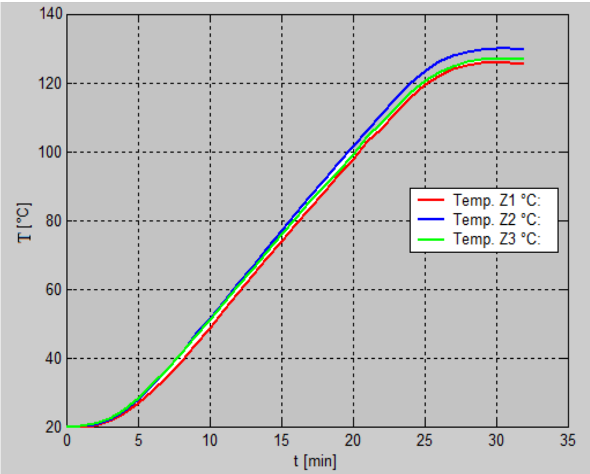


Fig. 10 Temperature characteristic of Zones 1, 2, and 3 obtained using the client application

Using the above identification method, it is easily possible to simultaneously obtain the zone cooling characteristics of an eight -zone extruder (see Fig. 11).

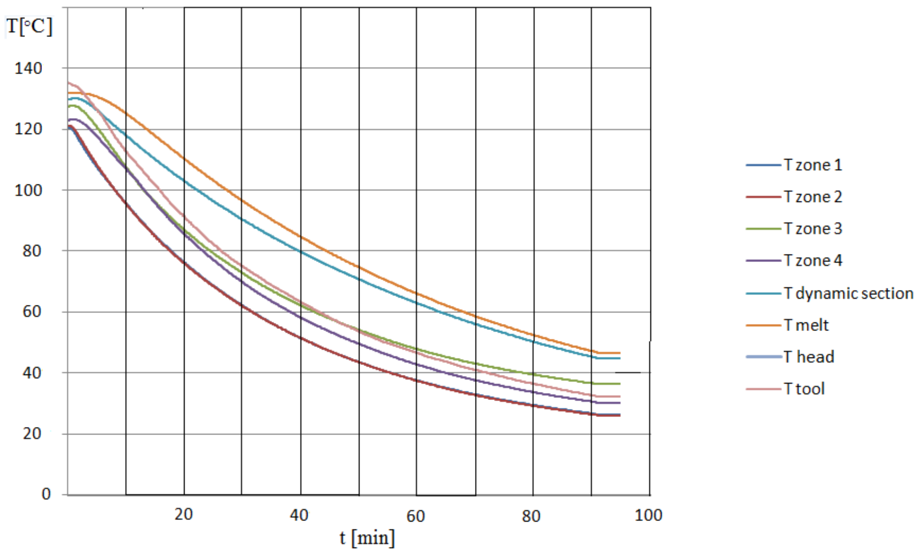


Fig. 11 Temperature characteristic of the extruder zones (cooling)

3. CONCLUSION

Kepware consists of different software tool combined with different PLC controllers can make a powerful tool for data acquisitions. In combination with a database can be used as a data collection tool to identify the process. Any program in which it is possible to write SQL queries can be used to read data from the database. In combination with a PLC, we can read and log data in real time frequency and store it in strings or fields readable by Kepware tools. A more advanced solution consists in developing the client application presented in this paper. The advantage of using this method of identification is reflected in the possibility of simultaneous collection of a large number of data from the process, which is especially convenient when creating models of multivariable systems and their experimental verification. The data collected during the recording of the response of the system can be easily used for obtaining the model in the form of a transfer function and amplitude-frequency characteristics (AFFK).In the client application, it is possible to read the selected data, export them and, then print. Kepware tools give an opportunity for experimental verification for different applied methods of control.

Acknowledgement: *This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.*

REFERENCES

[1] Kepserverex manual, https://www.kepware.com/getattachment/2745a0a9-079a-4630-b15c-8081aba1a91d/kepserverex-manual.pdf

[2] Allen-bradley-controllogix-ethernet-manual, https://www.kepware.com/getattachment/c7cd42b2-e3a8-403e-ac5b-ddff9cb03985/allen-bradley-controllogix-ethernet-manual.pdf

[3] XAMPP FOR MYSQL Amey Laxman Gawde, https://cs.nyu.edu/courses/Fall12/CSCI-GA.2433-001/XAMPP.pdf

[4] WebDevelopmentGuide2013Malveaux, https://viu.eng.rpi.edu/lab/2/WebDevelopmentGuide2013Malveaux.pdf

[5] https://static.webyog.com/pdfs/Using_SQLyog_Enterprise_to_Effectively_Synchronize_MySQL_Databases.pdf

[6] I. Kocić, P. Đekić, A. Milovanović, D. Antić, S. S. Nikolić, N. Danković, "Application of KEPServerEX applications for acquisition and supervision of production processes", *Proceedings of the XV International Conference on Systems, Automatic Control and Measurements, SAUM 2021,* Niš, Serbia, September 09.–10., 2021., pp. 17–20.

[7] I. Kocić, S. S. Nikolić, A. Milovanović, P. Đekić, and N. Danković, “Sequential control of water level in tanks and pumps from remote location using PROFINET network”, *Proceedings of the 9th Small Systems Simulation Symposium, SSSS 2022,* Niš, Serbia, February 28.–March 02., 2022., pp. 90–97.

[8] I. B. Kocić, D. B. Mitić, N. B. Danković, S. S. Nikolić, P. S. Đekić, “KEPServerEX as a data collection tool for process identification”, *Proceedings of the XVI International Conference on Systems, Automatic Control and Measurements, SAUM 2022,* Niš, Serbia, November 17.–18., 2022.

[9] S. S. Nikolić, I. Kocić, D. Antić, D. Mitić, N. Danković A. Milovanović, P. Đekić, “The winder dancer position control model using different PID control structures and micrologix PLC”, *FACTA UNIVERSITATIS Series: Automatic Control and Robotics*, vol. 21, no. 2, 2022, pp. 77–93. https://doi.org/10.22190/FUACR220409007N

[10] M. Cantu, *Delphi 7*, Kompjuter biblioteka Čačak, 2003.

[11] N. Hodges, *Coding in Delphi*, Nepeta Enterprises, 2014.

FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 21, Nº 3, 2022, pp. 187 - 199
https://doi.org/10.22190/FUACR221121015B

Survey Paper

DIGITAL TWIN BASED LIGHTWEIGHTING OF ROBOT UNMANNED GROUND VEHICLES

UDC (621.391:004.934)

Milan Banić[1], Miloš Simonović[2], Lazar Stojanović[2], Damjan Rangelov[1], Aleksandar Miltenović[1], Marko Perić[1]

[1]University of Niš, Faculty of Mechanical Engineering, Department of Machine Design, Development and Engineering, Republic of Serbia
[2]University of Niš, Faculty of Mechanical Engineering, Department of Mechatronics and Control, Republic of Serbia

Abstract. *Battery powered outdoor robot Unmanned Ground Vehicles (UGV) should have minimal mass for designed payload capacity to facilitate their movement on rough terrain and increase their autonomy. On the other hand, stiffness and strength of their mechanical components should be sufficient to sustain operational loads. The paper presents the case study of robot UGV lightweighting based on UGV digital twin data. The process of creation of robot UGV digital twin is described, as well as the type and quantity of data acquired from robot sensors. Based on the data from the digital twin of the robot about operational loads, the worst operational load cases were identified and the structural analysis of the robot chassis for noted load cases was performed. The results of the analysis were used in the topology optimization of the robot UGV mechanical components with a goal to satisfy design requirements and reduce the robot mass. By application of noted procedure the mass reduction of approximately 17 % was achieved.*

Key words: *Robot unmanned ground vehicle, Digital twin, Lightweight design, Static structural analysis, Topology optimization*

1. INTRODUCTION

Outdoor robot UGV move over the rough terrain, and it is necessary to ensure satisfactory strength and stiffness of the robot mechanical components against impacts, bending, twisting during movement and changes in the direction of movement. The geometry of the robot

---

Received November 21, 2022 / Accepted December 26, 2022
Corresponding author: Milan Banić
University of Niš, Faculty of Mechanical Engineering, Department of Control Systems, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia
E-mail: milan.banic@masfak.ni.ac.rs

during operation must not be compromised, thus deformation of the chassis must be as low as possible to ensure the robot functionality, as well as the sensors/actuators positioning and accuracy of sensor readings. Furthermore, the robot mass must be as low as possible to improve the robot autonomy, lower the dynamic forces and decrease the ground pressure in order to increase maneuverability. Lowering of robot mass for a same strength and stiffness also allows to increase the usable payload. The conflict of requirements regarding strength and stiffness and keeping the robot mass as low as possible can be solved by principles of lightweight design. The lightweight design is a development strategy, which aims to realize a required function with a system minimized mass under predefined boundary conditions [1]. The lightweight design can be achieved in many ways that include material substitution, design elements thickness reduction, and change of shape of the parts, or elimination of parts that do not contribute significantly to stiffness, without sacrificing the overall structural integrity or functionality [2]. But to perform the lightweighting procedure, operational loads must be known. Determining the operational loads that an outdoor robot UGV may experience can be a complex and challenging task, as it requires considering a wide range of factors that can impact the loads experienced by the robot. As by its nature the digital twin contains data about operational loads, the noted fact can be used in the system lightweighting [3].

The Digital Twin (DT) is the virtual representation of a physical object/system across its full life-cycle using real-time operational data and other sources to enable understanding, learning, reasoning, and dynamical recalibration for improved decision making [4]. Although DT was introduced as a concept during the 1960s by NASA, it is officially recognized in 2002 at the University of Michigan. Since then, DT is gaining in popularity as one of the most cost-effective technologies in real applications [5]. Using data derived from Internet of Things (IoT) sensor technologies that are attached or embedded in a physical object the digital twin provides an opportunity to transfer the physical objects/system to a virtual objects/system, so that there is a possibility of creating virtual representations of products, or simulate of various operational processes, manipulate and monitor them [6]. DT can have multiple maturity levels based on the degree of communication between digital and the physical domain, or according to the communication direction and intelligent data processing [7].

Goraj [8] predicted a fatigue lifetime based on an one-step load spectrum. Time-dependent normal and shear stress components were estimated using a high-fidelity digital twin built in Siemens PLM Nx Nastran. The author considered all structural and thermal loads identified by sensor data and estimated the equivalent cyclic degree of utilization and a safety margin against the slip of a press-fitted shaft to rotor hub connection. Kokonen et all [9] used the digital twin for collecting information of manufacturing the quality deviation in the case of welded structures and for the prediction of the individual remaining fatigue lives. The authors considered that design goal was minimizing the mass of the structure as a process of optimization, while the required fatigue life is defined as a constraint. Ryll et all [10] demonstrate the potential of the digital twin technology in a hybrid lightweight structure by realizing the demonstrator for creating a table size of aluminum profiles. Rupfle el all [11] used a digital twin for real-time analysis of load and displacement calculation of large-scale onshore wind turbines. Based on a digital twin data the “Message Queuing Telemetry Transport” (MQTT) transmission protocol was chosen for the lightweight design of constrained services. Zhang et all [12] optimized the wall-climbing robot system by implementing multiple sensors integration for magnetic particle testing. They established a high-quality detection that is based on the collected sensor information and a multi-degree of freedom component collaborative flexible detection.

Digital twin approach thus is providing a good basis of data for the lightweight design of products. Combining real-data information with detailed data about the flight missions of the whole aircraft fleet gives better picture about the real use and requirements for lightweight structure [13, 14, 15]. Herlitzius [16] estimated that tractors and self-propelled machines today realize a power-to-weight ratio of 45 kg/kW (±15 %). Potential of lightweight design in these cases is almost halved for the same power.

The paper presents the lightweighting procedure of robot UGV based on robot digital twin data about operational loads. The process of creation of robot digital twin is described, as well as the data acquisition from robot sensors. Based on the data from the digital twin of the robot, the worst load cases were identified and the structural analysis of the robot mechanical components for noted load cases was performed. The results of the analysis were used in the topology optimization of the robot mechanical components with a goal to satisfy design requirements and reduce the robot mass.

2. The Lightweighting Study Object

The object of the lightweighting case study is a TRL6 (TRL – Technology Readiness Level) prototype of a battery powered robot UGV developed by the company Coming Computer Engineering and the Faculty of Mechanical Engineering of University of Niš. The robot, branded as Agriculture Autonomous Robot (AgAR), is developed for indoor and outdoor tasks and designed with precision agriculture in mind. Equipped with ROS based software that enables remote control, moving along predefined paths, and autonomous movement, AgAR represents a multipurpose robotic UGV platform which can accommodate extensive variety of payloads customized to meet precision agriculture needs. Due to the rugged design and high torque drive train AgAR can cover rough terrain with high gradient slopes even with large payloads.

AgAR robotic platform consists of four main subassemblies, subassembly of the main body, main body enclosure, mechanism for the change of clearance from the ground, and a subassembly of the drive train as shown on Figure 1.

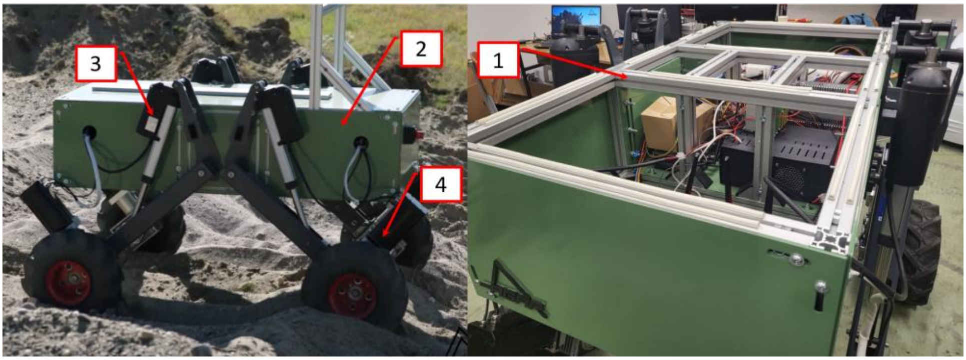
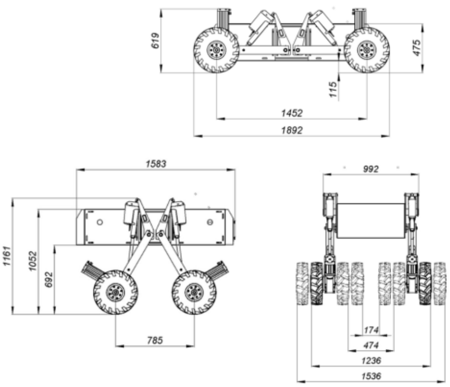


Fig. 1 3D model of AgAR with marked subassemblies

The subassembly (1, Figure 1) of the main body is made of standard aluminum profiles for easier assembly and disassembly and protected with a sheet metal enclosure (2, Figure 1). To increase the stiffness of the main body, in every corner of the main body there is a reinforcement in the form of steel plates. The subassembly of the mechanism for changing the clearance from the ground of the main body (3, Figure 1) is designed as a four bar mechanism where the connecting rod is expandable. The mechanism rocker is a square tube, and the mechanism position change is achieved by the extension of the rod of electric linear actuator which acts as a connecting rod. As there are four mechanisms for changing the clearance from the ground of the main body, it is possible to change the height of the main body from the ground i.e. clearance of the robot, as well to adjust the inclination of the main body in lateral and longitudinal directions. The subassembly of the drive train (4, Figure 1) consists of 1 kW BLDC motor, worm gearbox with reduction ratio 1:40 and a wheel with a pneumatic tire connected to the worm gearbox output shaft via a wheel hub. The motor and worm gearbox are connected by an elastic coupling which is transferring the torque from the motor to the worm gearbox. The connection between subassemblies is designed as separable, with bolted connections. Such robot design allows for easy changes to the universal robotic platform, satisfying the requirements related to modularity and universality. The overall AgAR overall dimensions and its technical specification are shown on Figure 2. The total mass of the robot with all components is 420 kg, while the mass of the robot mechanical components is 340 kg.



SIZE AND WEIGHT		SPEED AND PERFORMANCE	
External dimensions	1.9 x 1.2 x 0.36 m	**Max speed** Wheel diameter dependant (260/420/480)	1.1 / 1.75 / 2 m/s
Weight	420 kg	Max slope	40°
Max payload	300 kg	**Run time** (typical use)	5 h
Ground clearance	115 - 692 mm (adjustable)	Onboard sensors	RTK GNSS, AHRS IMU, ultrasonic sensors, stereo camera, ...
Change of the slope of the platform	±30° in lateral and transverse direction	User power	5V, 12V, 24V, 48V
Wheel diameter	260 mm, 420 mm, 480 mm	Drivers and API	ROS, LabVIEW

Fig. 2 AgAR dimensions and technical specification

The AgAR is equipped with an extensive set of sensors which continuously acquire data during its operation as shown on Figure 3. The robot position is determined by the fusion of data from the Real Time Kinematic (RTK) GNSS system and Attitude and Heading Reference system (AHRS) Inertial Measurement Unit (IMU). The AHRS IMU also acquires data about the acceleration in all three directions, as well as roll, pitch, yaw and angular acceleration of the robot main body. The environment perception is enabled by 3D LiDAR, stereo camera, RGB camera and four Ultrasonic Sensors (US). The BLDC motor controllers also gather data about the motor RPM, motor voltage (*U*), current (*I*) and motor and controller temperature (*T*), as well as other data significant for the robot operation. The data about battery operational parameters (*U*, *I*, *T* and State of Charge – SoC) are also acquired from the smart Battery Management System (BMS).

Fig. 3 AgAR sensor set and data gathered by a digital twin

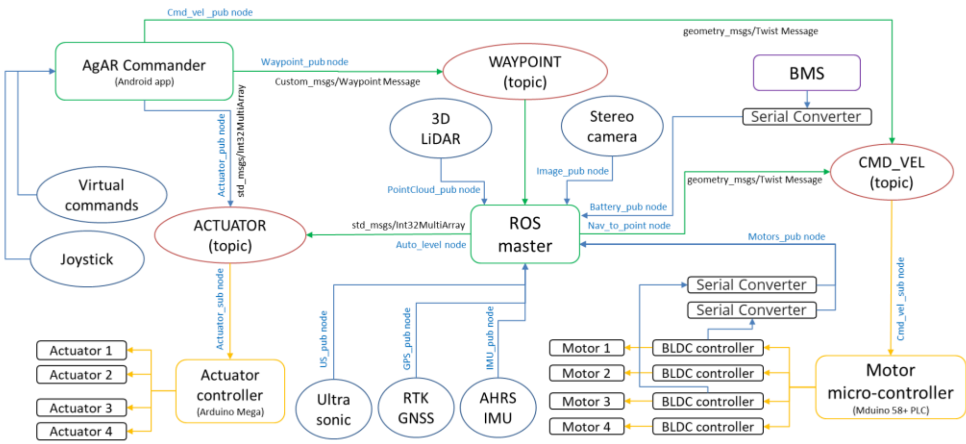


Fig. 4 AgAR sensor set and data gathered by a digital twin

All of above-mentioned sensor data are acquired stored both locally and in a cloud to form an AgAR smart monitor digital twin according to the control architecture shown on Figure 4. The sensor data are acquired locally, stored, organized and structured on ROS master (Industrial PC), where they are processed and control algorithms are applied. The robot has three operational modes – teleoperation, GPS navigation and autonomous movement. In each of the operational modes, the sensor data is acquired at 25 Hz. Furthermore, the most significant data is visualized on the AgAR Android app – AgAR Commander in real-time.

3. LIGHTWEIGHTING METHOD

The lightweighting procedure was performed based on the assumption that the robot mechanical components must have sufficient strength and stiffness to sustain the worst-case operational loads. The worst-case operational loads were identified by the analysis of digital twin data, which was gathered during the validation of the TRL6 AgAR prototype against all the functional and technical requirements for all use cases defined and for a maximal AgAR payload. Two worst-case operational load cases were identified:

- during the change of direction of movement by a rotation around the central vertical axis i.e. during "tank" in place turning with maximal payload and with minimal clearance, and
- while climbing the curb on direct path of moving at full speed.

Fig. 5 Identification of worst-case load case scenario by motor current

The motor current during the movement of the robot can be seen on Figure 5. The peaks in the motor current diagram correspond to the "tank" turning. It was observed during testing that the maximal peaks occur while the robot is in minimal clearance from the ground configuration i.e. when the wheels have a maximal distance from vertical axis of rotation. Furthermore, from Figure 5, one can observe that the motor current while robot performs the "tank" turning is not the same on the motors on the left and right robot side – the motors on the left robot side during counter clock rotation have up to three times larger current than the motors on the right robot side. The same applies for clockwise rotation, as in that case the motors on the right have approximately three times larger current than the motors on the left robot side. As the motor current is directly proportional to the generated torque by a motor constant, it is straightforward to calculate the torque on wheels during the "tank" turning according to:

$$T = K \cdot I \cdot i \cdot \eta \tag{1}$$

where: T - torque on wheel, K - motor constant (0.142 Nm/A), I - motor current, i - worm gear transmission ratio (40) and η worm gear efficiency (0.7).

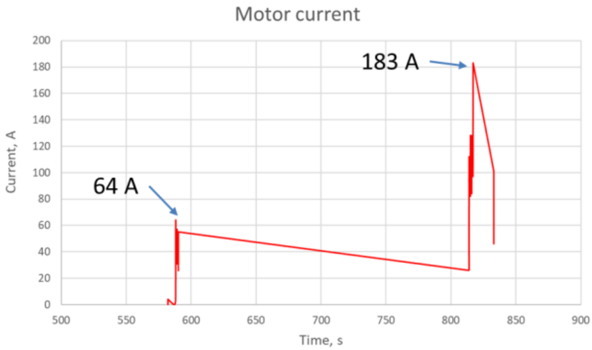


Fig. 6 Maximal values of motor current during “tank” robot rotation

Based on maximal values of motor current for the right and left robot side (183 and 64 A), as shown in Figure 6, by substituting the maximal values into equation 1, the maximal torques were calculated T_{right} = 728 Nm and T_{left} = 254.5 Nm.

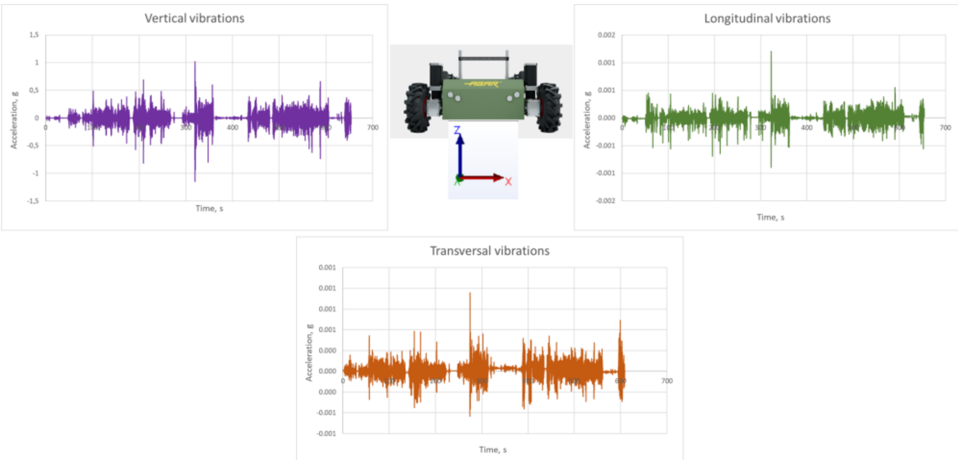


Fig. 7 Identification of worst-case load scenario by robot main body acceleration

In the second worst-case operational scenario it is determined that crossing the elevated static objects, such as rocks or curbs, along the movement path introduces an additional vibration load to the robot mechanical components. The movement over the rough terrain generates a lower vibration load than the crossing of elevated static objects; the contact of robot wheels with noted objects generates an impact which additionally stresses the mechanical components. The maximal vibration load during TRL6 prototype validation was observed while climbing the curb at full robot speed as shown on Figure 7. The vibration load is the largest in the vertical direction and significantly lower in lateral and longitudinal directions. As the vertical vibrations are two orders of magnitude larger than the lateral and longitudinal vibrations, the latter two were neglected in further analysis. The maximal acceleration in the vertical direction is 2.1 g, as shown in Figure 7.

The method used to lightweight the robot mechanical components consists of a sequential combination of the static structural analysis by the finite element method followed by the topology optimization. The analysis was performed in Ansys Workbench 19.2 software package.

The static analysis was used to calculate the effects of static loading of the mechanical components of the robot, while an inertial force of gravity was also considered. The static structural analysis is divided into two load cases:

- first analysis (load case 1) is the analysis of the structural integrity of the robot mechanical components due to the "tank" rotation during the change of the direction of moment, while considering the friction between the wheels and the surface base. In the noted load case it is considered that one of the wheels is fixed, as it introduces the maximal load for the robot mechanical components;
- second analysis (load case 2) is the analysis of the robot mechanical components structural integrity for the second worst-case operational scenario i.e. due to the vertical acceleration load of 2.1 g.

The results of the static structural analysis were then used in the topology optimization analysis with a goal to identify which geometry volumes which can be removed to reduce mass of the robot mechanical components. Based on results of topology optimization, as well as strain energy distribution determined in the static structural analysis, the 3D model of the robot was revised to reduce the mass. The new static structural analysis was then performed with a revised 3D robot geometry to ensure that the revised design has satisfactory strength and stiffness.

4. Static Structural Analysis

For this analysis purposes geometrical model of the robot are simplified in order to reduce time and resources needed for the analysis without tradeoffs in simulation accuracy. The linear electric actuators were replaced by a steel rod which is locking the clearance adjustment mechanism at the lowest position, as the noted position was identified as a worst-case load in load case 1. Fasteners were removed from the analysis as they do not contribute significantly to the robot stiffness. The pneumatic tires were replaced with a simplified geometry and it was considered that they were made from ABS plastics as their stresses and deformations were not a subject of static structural analysis. The simplified geometric model of the robot can be seen in Figure 8.

Materials which are used in robot design were:

- Aluminum alloy 6061 for all main body profile components,
- structural steel S355 J2GR for the main body enclosure and reinforcements, components of the clearance change mechanism and wheel hub and rim,
- steel C45 for shafts, pins and spacers, and
- ABS plastics for tires, as the pneumatic tires were replaced by ABS plastics for analysis purposes.
- Table 1 defines the mechanical characteristics of the materials used in the analysis.

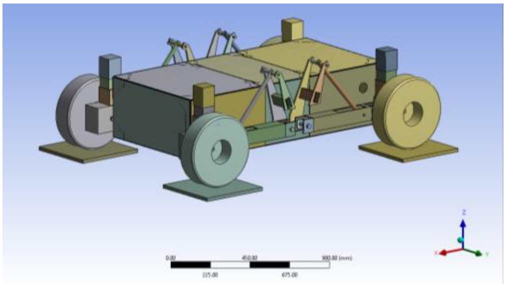


Fig. 8 Simplified geometrical model of the robot

Table 1 Material properties

Aluminium Alloy 6061	Value	Structural Steel S355 J2GR	Value
Tensile Yield Strength	259 MPa	Tensile Yield Strength	355 MPa
Tensile Ultimate Strength	313 MPa	Tensile Ultimate Strength	510 MPa
Density	2710 kg/m3	Density	7850 kg/m3
Young's Modulus	68300 MPa	Young's Modulus	200000 MPa
Poisson's Ratio	0.33	Poisson's Ratio	0.3
Structural Steel C45E	Value	ABS plastics	Value
Tensile Yield Strength	490 MPa	Tensile Yield Strength	41.4 MPa
Tensile Ultimate Strength	700 MPa	Tensile Ultimate Strength	44.3 MPa
Density	7850 kg/m3	Density	1040 kg/m3
Young's Modulus	200000 MPa	Young's Modulus	2390 MPa
Poisson's Ratio	0.3	Poisson's Ratio	0.399

Meshing was performed with 3D [17] higher order elements to exactly capture the geometry of some of the design elements, as well to increase the accuracy of simulations. Maximum mesh skewness of the model was below 0.85 which satisfy the mesh quality criterion and thus provides confidence in simulation results. The model consists of 1178893 nodes which form 328824 finite elements. A mesh size sensitivity test was performed to obtain confidence in accuracy of finite element simulations. It was assumed that the simulation results are insensitive to the mesh size if the difference in equivalent stress and total deformation in two adjunct meshes is below 3%. The model finite element mesh is shown on Figure 9. Face sizing was done on the contact surfaces where the frictional contact between the elements exists.

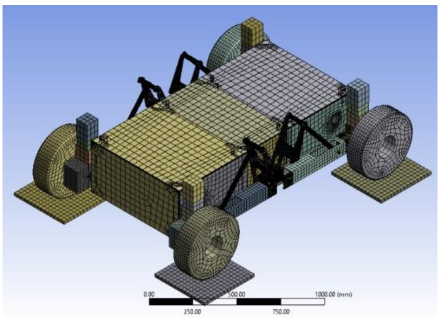
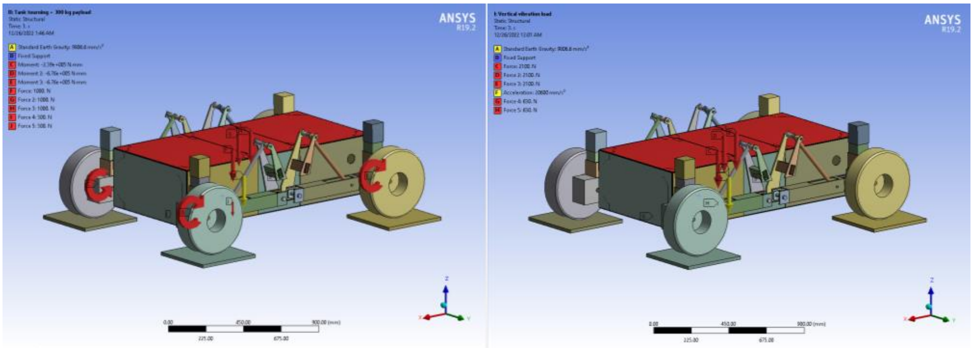


Fig. 9 FEM mesh of the AgAR robot

Figure 10 shows the loads and boundary conditions for the two worst-case load scenario – “tank” turning with one wheel fixed (Load case 1) and increased load due to vertical vibrations (Load case 2). In both load cases, the battery weight (600 N) and robot payload (3000 N) were introduced as force loads acting on corresponding surfaces.



a) b)

Fig. 10 Boundary conditions and loads for Load case 1 (a) and Load case 2 (b)

For the Load case 1, the friction coefficient is considered as 0.6 for the contact of wheels with the ground.

The performed static structural analysis showed that the maximal stresses and deformations occur for the Load case 1. The analysis showed that the robot chassis has a good structural integrity. The maximum equivalent stress (Figure 11) occurs on the connecting rod pivot pin and it amounts to approximately 320 MPa which is bellow the yield strength of the material of the pivot pins (steel C45E).

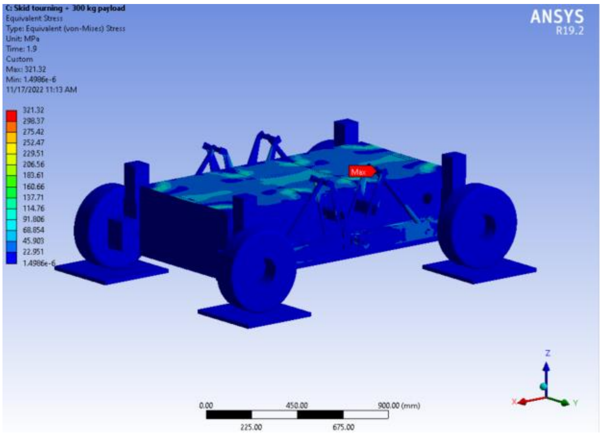


Fig. 11 Stress distribution of robot mechanical components for Load case 1

5. Topology Optimization

Topology optimization was performed for the critical load case – Load case 1, where equivalent stresses had the highest values. The first step in the topology optimization was to define an optimization region i.e. the 3D region where the change of material layout is possible. All surfaces where loads or boundary conditions were applied were excluded from the optimization region. Furthermore, all other functional surfaces were also excluded from the optimization region. The second step in the topology optimization procedure was to define an optimization goal. As already noted, the primary goal of the study was to decrease the mass of the AgAR mechanical components, so a 20% reduction of mass was adopted as a goal of analysis. The results of topology optimization are shown in Figure 12.

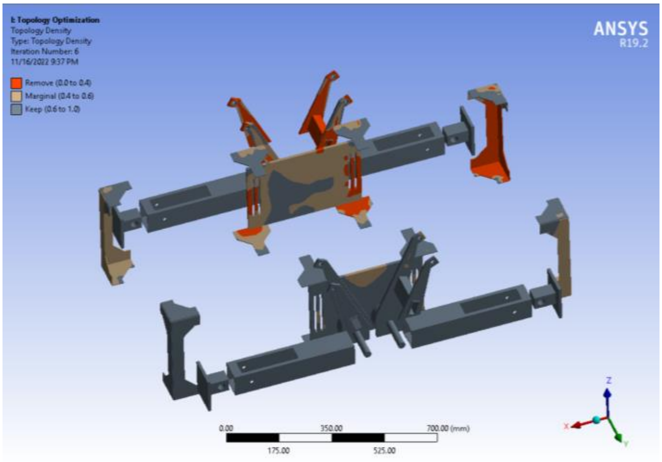


Fig. 12 Results of topology optimization for Load case 1

Based on results of the topology optimization a redesign of AgAR robot was performed, keeping in mind that during operation the robot can rotate around the vertical axis in tank turning on both sides – clockwise and counterclockwise. The redesign was performed also in respect to results of the strain energy distribution for Load case 1 shown on Figure 13.

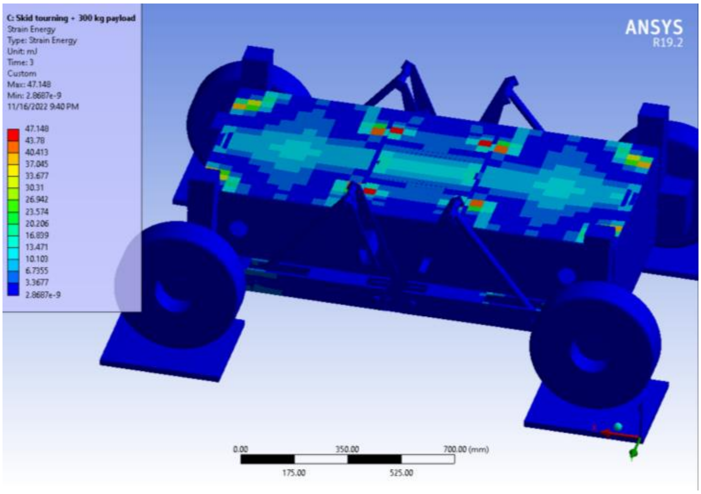
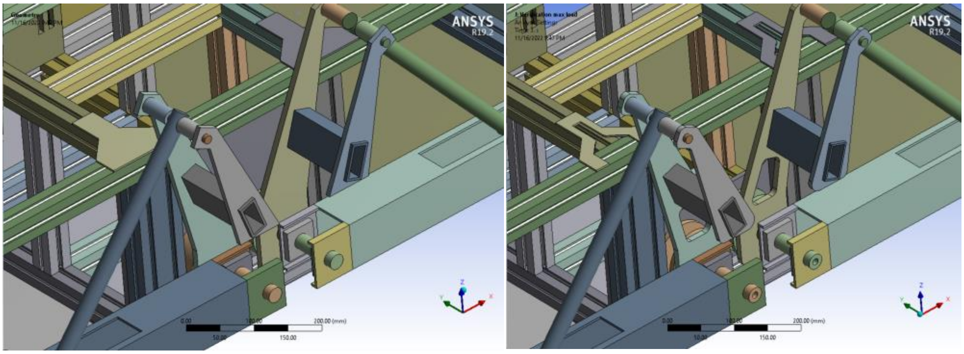


Fig. 13 Strain energy distribution for Load case 1

Figure 14 shows the comparison of the preliminary and new 3D design of robot mechanical components. The redesigned AgAR mechanical components mass is 285 kg which is a reduction of approximately 17% compared to the starting robot mass of 340 kg.



a) b)

Fig. 14 Results of topology optimization for Load case 1

The structural analysis was again performed for a redesigned robot for a Load case 1 in order to perform the verification of the new design. Figure 15 shows the results of stress distribution of the redesigned AgAR mechanical components. The maximum equivalent stress again occurs on the connecting rod pivot pin, and it is increased to 345 MPa. Although the maximal stress is increased, it is bellow the yield strength of the steel C45E as the material of the pivot pins.

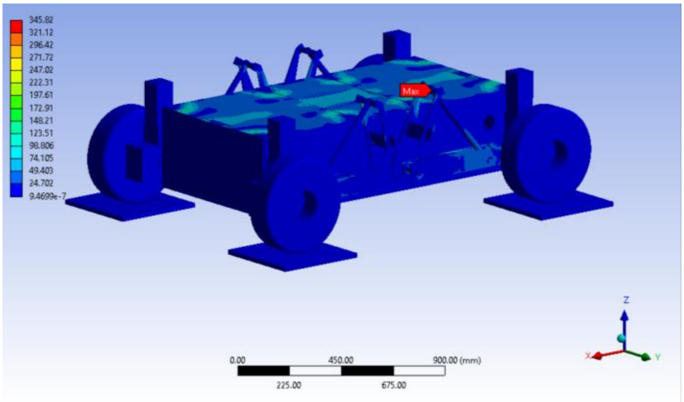


Fig. 15 Stress distribution of the redesigned robot mechanical components for Load case 1

7. Conclusion

The paper presents the case study of the robot UGV lightweighting based on digital twin data about operational loads. The process of creation of the robot UGV digital twin is described, as well as the identification of the worst-case operational load cases. The static structural analysis and topology optimization were performed for the worst-case operational load.

Based on the obtained results, it can be concluded that digital twin data can be successfully used to optimize the robot UGV. By application of real-world operational load data, the mass reduction of approximately 17 % was achieved. The future direction of research would be directed towards the further optimization of the AgAR design with a goal of mass reduction by material substitution and using sheet metal parts to substitute the parts made by machining. Furthermore, the AgAR digital twin will be progressed to higher maturity level with the inclusion of the Artificial Intelligence for maintenance and control purposes.

Acknowledgement: *This research work was supported by the Innovation Fund of Republic of Serbia (IF 50471) and co-funded by Coming Computer Engineering.*

REFERENCES

[1] Henning, F., Moeller, E., Handbuch Leichtbau, Hanser Verlag, (2011).
[2] M. Banić, M. Perić, L. Stojanović, Structural Analysis of Robot for Train Undercarriage Visual Inspection, Proceedings of XX Scientific-Expert Conference on Railways RAILCON '22, 978-86-6055-160-5, Niš, Serbia, 13. - 14. Oct, 2022, pp. 21 – 24.
[3] M. Banić, M. Simonović, L. Stojanović, D. Rangelov, A. Miltenović, M. Perić, Digital Twin-Based Unmanned Outdoor Field Robots Lightweighting, XVI International Conference SAUM 2022, 978-86-6125-258-7, Niš, Serbia, 17. - 18. Nov, 2022, pp. 9 – 14.
[4] M. Mikell, Immersive analytics: the reality of IoT and digital twin, IBM Business Operations Blog, (2017).
[5] M. Crawford, 7 Digital Twin Applications for Manufacturing, ASME, (2021).
[6] C. Miskinis, The history and creation of the digital twin concept, Challenge Advisory, (2019).
[7] Lockheed Martin Corporation, Visualizing the Digital Thread = Digital Twins at Lockheed Martin, (2021).
[8] R. Goraj, Digital twin of the rotor-shaft of a lightweight electric motor during aerobatics loads, Aircraft Engineering and Aerospace Technology, 92(9), 2020, pp.1319-1326.
[9] P. Kokkonen, B. Hemming, E. Mikkola, J. Junttila, V. Lämsä, Robust lightweight design and digital twins considering manufacturing quality variation and sustainability requirements, XIV Finnish Mechanics Days 2022, May 18–20, 2022, Lappeenranta, Finland
[10] T. Ryll, M. Kuhtz, L. Quirin, M. Nguyen, M. Gude, A. Filippatos, Design approach for the development of a digital twin of a generic hybrid lightweight structure, MATEC Web of Conference 349, 03004 (2021), ttps://doi.org/10.1051/matecconf /202134903004
[11] J. Rupfle, C.U. Grosse, Soft real-time framework for Structural Health Monitoring of wind turbine supporting structures, International Conference on NDE 4.0 – 2022
[12] Zhang, X.; Zhang, X.; Zhang, M.; Sun, L.; Li, M. Optimization Design and Flexible Detection Method of Wall-Climbing Robot System with Multiple Sensors Integration for Magnetic Particle Testing. Sensors 2020, 20, 4582. https://doi.org/10.3390/s20164582
[13] Laukotka, F.; Seiler, F.; Krause, D.: MBSE als Datenbasis zur Unterstützung von Konfiguratoren und Digitalen Zwillingen modularer Produktfamilien, Proceedings of the 31th Symposium Design for X (DFX2020), Bamberg, Germany, 2020, pp. 61-70. DOI: 10.35199/dfx2020.7
[14] Tuegel, E.J.: The Airframe Digital Twin: Some Challenges to Realization, 53rd AIAA/ASME/ASCE/ AHS/ASC Struct. Struct. Dyn. Mater. Conf, pp. 1812, 2012. DOI: 10.2514/6.2012-1812
[15] Laukotka, F.; Hanna, M.; Schwede, L.-N.; Krause, D.: Lebensphasen-übergreifende Nutzung Digitaler Zwillinge – Modellbasierte Produktfamilienentwicklung am Beispiel der Flugzeugkabine, Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF), Vol. 115, 2020, pp. 101-104. DOI: 10.3139/104.112332
[16] Herlitzius, T. How Much Lightweight Design Does Agricultural Technology Need?. ATZ Heavy Duty worldw 15, 54 (2022). https://doi.org/10.1007/s41321-021-0466-7
[17] ANSYS Release 19.2 documentation, (2020).

FACTA UNIVERSITATIS Series: Automatic Control and Robotics would like to acknowledge the expertise of the following reviewers, which has helped to set the standard of the journal in 2022

Aleksandar Milenković/Dr./University of Niš, Faculty of Electronic Engineering, Republic of Serbia/Teaching assistant

Ana Gavrovska/Dr./University of Belgrade, School of Electrical Engineering, Republic of Serbia/Assistant professor

Dragan Stojanović/Dr./University of Niš, Faculty of Electronic Engineering, Republic of Serbia/Associate professor

Emilie Avignon Meseldzija/Dr./Universite Paris-Saclay, Centrale Supelec, France/Assistant professor

Irfan Fetahović/Dr./State University of Novi Pazar, Technical Sciences, Republic of Serbia/Assistant professor

Ivan Vulić/Dr./University of Defence, Military Academy, Republic of Serbia/Associate professor

Milan Savić/Dr./University of Priština in Kosovska Mitrovica, Faculty of Sciences and Mathematics, Republic of Serbia/Associate professor

Milica Petković/Dr./University of Novi Sad, Faculty of Technical Sciences, Republic of Serbia/Assistant professor

Miloš Bogdanović/Dr./University of Niš, Faculty of Electronic Engineering, Republic of Serbia/Assistant professor

Miloš Stojanović/Dr./College of Applied Technical Sciences Nis, Republic of Serbia/Assistant professor

Mladen Tomić/Dr./University of Novi Sad, Faculty of Technical Sciences, Republic of Serbia/Associate professor

Nikola Simić/Dr./University of Novi Sad, Faculty of Technical Sciences, Republic of Serbia/Teaching assistant

Petar Spalević/Dr./University of Priština, Faculty of Technical Sciences, Republic of Serbia/Full professor

Siniša Ješić/Dr./University of Belgrade, School of Electrical Engineering, Republic of Serbia/Full professor

Siniša Suzić/Dr./University of Novi Sad, Faculty of Technical Sciences, Republic of Serbia/Teaching assistant

Slađana Dimitrijević/Prof. Dr./University of Kragujevac, Faculty of Science, Republic of Serbia/Assistant professor

Sreten Stojanović/Prof. Dr./University of Niš, Faculty of Technology, Republic of Serbia/Full professor

Stefan Panić/Prof. Dr./University of Priština, Faculty of Natural Science and Mathematics, Republic of Serbia/Full professor

Stefan Tomić/Prof. Dr./ALDAR University College | University in Dubai, UAE/Assistant professor

Vesna Golubović/Dr./University of Belgrade, School of Electrical Engineering, Republic of Serbia/Full professor

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

62

FACTA Universitatis. Series, Automatic Control and Robotics /
University of Niš ; editor-in-chief Marko Milojković. - Vol. 7, no. 1
(2008)- . - Niš : University of Niš, 2008- (Niš : Atlantis). - 24 cm

Dostupno i na: http://casopisi.junis.ni.ac.rs/index.php/FUAutContRob. -
Tri puta godišnje. - Delimično je nastavak: Facta Universitatis. Series: Automatic
Control and Robotics = ISSN 0354-2009. - Drugo izdanje na drugom medijumu:
Facta Universitatis. Series: Automatic Control and Robotics (Online) = ISSN 1820-6425
ISSN 1820-6417 = Facta Universitatis. Series: Automatic Control and Robotics
COBISS.SR-ID 158108940