International Journal
Information Technology Applications (ITA)
Volume 10, Number 1, September 2021

AN APPLICATION FOR OPTIMAL MATERIAL FLOW IN INDUSTRIAL WAREHOUSES

Filip Žemla, Ján Cigánek

Abstract:

The aim of the paper is to design and to create an application in the selected visualization environment, which will be used to simulate the movement of material in industrial warehouses. After entering the input data, the application automatically suggests the optimal way to store packages in the warehouse. The paper will describe the storage systems, database systems and relation database model used in the application. The visualization systems are introduced here, where the Promotic system will be presented and the development of practical result. The last part of paper describes the developed application and its functions.

Keywords:

SCADA systems, optimization, material movement, dynamic programming.

ACM Computing Classification System:

Information systems, data management systems, database management system engines, database views.

Introduction

A SCADA system (Supervisory Control And Data Acquisition) is an architecture of control systems, consisting of computers, a communication infrastructure and a graphical interface. Their functions are to supervise management, data collection and the provision of a clear user environment. SCADA systems are used mainly in industrial production, but also in distribution and transport networks.

In storage systems, the material flow is increased; therefore it is necessary to ensure its fluidity in the shortest possible time. Today, there are many methods to increase efficiency, but their implementation is technically and financially demanding and is therefore only required in larger warehouses. In this work we therefore deal with the optimization of material movement in the SCADA system environment.

The aim of this work is to design an application in a SCADA system with decentralized control to display the optimal path of material movement. The optimal path will be designed for loading and unloading packages in a simulation warehouse with six racks. If a suitable location is found, its weight, dimensions and the current position of the user will be taken into account. If the packages are removed from storage, the closest package from the current position of the user will be found.

1 Storage systems

One of the main components of the modern logistics center infrastructure is the warehouse. In the logistics center, the warehouse serves as a distribution center, so its task is to receive and dispense pallets. The warehouse in the logistics center, as opposed to the warehouse in the production process, is focused on maximizing profit by satisfying customers' transport requirements. There are many storage systems, the use of which depends on the method of storage and the equipment used in the warehouse. We also select the storage system according to the type of stored material, its dimensions, weight and storage time. In our case we will use Pick-to-Light and Put-to-Light systems [7, 8].

The Pick-to-Light system is one of the modern material retrieval systems, significantly increasing the productivity of order fulfillment and at the same time reducing operator errors. It eliminates the need for paper expenses, removal orders and assembly sheets. This system also uses light sensors located in the racks to navigate the operator. After entering the shipment number into the system, the sensor lights up above the package, signaling the storage location with the specified quantity for the operator. **The Put-to-Light system** is based on the same principle, with the difference that the entered quantity is subtracted (selected) from the entered storage [5, 9, 10].

Fig.1. Warehouse using Pick-to-Light and Put-to-Light systems [4].

2 Database systems

The program system can be called a database system if two criteria are met. The first criterion is the ability to manage persistent data and the second is the ability to process large amounts of data efficiently. From these requirements, we can define a database system as a software system that is adapted to the efficient storage, selection and manipulation of large amounts of persistent data. Persistent data persists in the system for a long time, regardless of the programs used. Database systems work with data stored in a database, so its main functions are [2]:

- inserting new, empty files into the database,
- inserting data into an existing file,
- selection of data from an existing file,
- correction of data in an existing file,
- deleting an existing file from the database,
- deleting data from an existing file.

The database is an organized collection of data in electronic form. It is a set of structured, interrelated data grouped together, without unnecessary redundancy. A database consists of schemas that represent a set of data describing a database data model, which is a table. The tables represent an array of *n x m*, where *n* is the number of columns and *m* is the number of rows. The table column defines the data name and data type. The most used data types in databases are numeric, date or logical [11].

The relational database model is currently a unified way of representing data at the logical level due to the simplicity that facilitates data representation. This database model is used in most popular database applications, such as Microsoft SQL Server, Oracle Database, MySQL, IBM DB2 or Microsoft Access [3].

Microsoft Access is a database management system from Microsoft. It offers a combination of graphical user interface and application development tools with the Microsoft Jet relational database engine. Microsoft Access can create tables and forms, as well as create complex queries without requiring any programming skills of users. Thanks to this feature, it deserved a place in the Microsoft Office package. Microsoft Access works with relational databases, saving them in its own file with the *.accdb* extension. In older versions, the *.mdb* extension was used [1, 14].

Microsoft Access is complemented by other services from Microsoft that allow the database to be stored in the cloud, backed up and encrypted. An interesting feature is that Access is so compatible with competing databases that it can work with data from Oracle databases. From the information obtained so far, it may be obvious that it can also communicate with other software from Microsoft, Microsoft SQL Server. As this application is available for students in the student version and sufficient to work with data in our work, it will be used to create a database and its subsequent connection with the visualization program Promotic [1, 14, 18].

3 Visualisation Systems

Visualization systems are responsible for consistent data display. All data must be sufficiently visible in compliance with ergonomic rules and psychological requirements. This means that the work environment in which the data is displayed must be configured and customized [15]. The ideal interface is clear, transparent and unobtrusive as little as possible. It allows the user to focus on process control without drainage attention to its very appearance [16].

SCADA systems consist of three components, namely a **PLC**, a **main station** with HMI and a **communication structure**. The PLC reads the values, which are then sent to the main station via the communication structure. The main station displays the received values to the operator in graphical and numerical form via the HMI. It is also possible to control these values after entering a value by the operator. The station sends a signal to the PLC, which sets the process to the desired behavior. Such systems are used mainly in industrial production halls, but nowadays they are also used in Smart home technology or Web technologies due to the possibility of remote access [15].

Promotic is SCADA / HMI system for creating applications in industrial automation. The Czech company MICROSYS created Promotic system with its operations and functions in 1991. Today, Promotic is one of the most widely used SCADA systems in the world. It deserved this award thanks to Promotic team's direct attention to the quality of services they provide to the customer. This means that they are constantly developing their product in the form of new versions, the current version is 9.0.8. It also contains rich documentation and provides top technical support.

As Promotic is constantly evolving, the features it provides are also being added. Promotic can access databases from the simplest ones such as Microsoft Excel or Microsoft Access to Oracle, MySQL and Post-greSQL databases [17, 18].

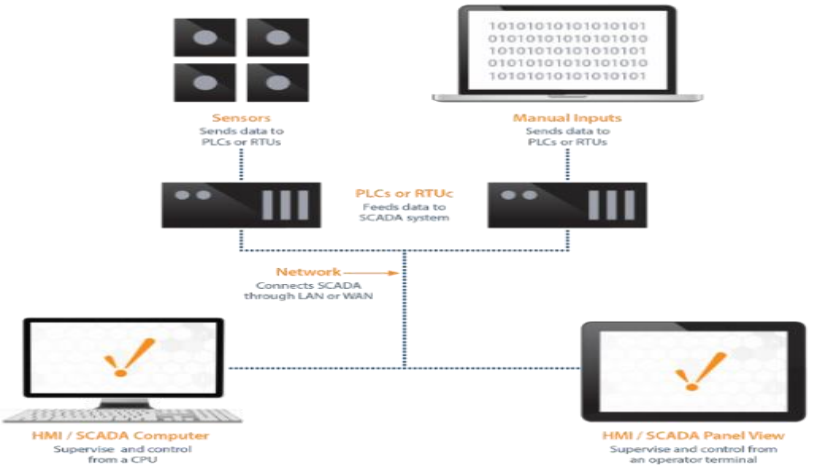

Fig.2. Diagram of SCADA systems [13].

It has also adapted to the needs of users for remote access by adding a Web server feature. The web server provides application conversion to HTML, XML, PHP and JS files for Internet browsers. Data transfer is based on the standard HTTP protocol and access to the application is handled by a secure login. Promotic system supports work with two scripting languages, VBScript and JavaScript. We use VBScript for local applications and JavaScript for the mentioned Web applications. In this work, we decided to use Promotic application using both programming languages [6, 17].

4 Case Study

The aim of the application is to streamline the storage and retrieval of packages in the warehouse. This is achieved by using dynamic programming, looking for the shortest way to save the package. In the application, we used an imaginary, presentation warehouse consisting of six shelves. Each of the shelves is divided into two sides, each side having two shelves and 20 segments, places where packages can be stored.

The application consists of two main parts, according to which the job description will be divided. The first part is a description of the tables used in the database and their connection with the visualisation application Promotic. The second part is a description of the HMI application, in which the functionalities of the elements are described. The HMI application was created in accordance with the IEC 73 standard.

a) Database

An important part of the application is the database, which is divided according to the use of variables into tables. All tables were created and managed in the MS Access environment.

The Segments table contains the properties of all segments in the warehouse, where the segment number (attribute "Segment" in the database) is the primary key with unique values. The second table is **the package table**, in which the values entered by the user in the application are written. Each entity corresponds to one package and attributes.

The next **table** is for **user management**, each user having the attributes of his ID, which serves as the primary key, login name, password, access level, and default user language. The access level corresponds to the function of the user in the warehouse. Users with an access level corresponding to 2 are allowed to load and unload packages from the warehouse with the shortest route displayed. Users with access level value 1 do not have permission to insert and unload packages, and the shortest path will be displayed after entering input values to other users with access to package manipulation in the database. The Language attribute numerically represents one of the seven languages that can be selected in the application.

The distance table and **the neighbour table** serve as two-dimensional arrays of the properties of individual nodes. Both tables have 21 attributes, with the first attribute indicating information for which node the entity is. Other attributes indicate how it relates to a given node. In the case of a distance table, this is the distance of the node to which the entity belongs to the given node to which the attribute corresponds. And in the neighbors table as in the previous table, but instead of the distance, the cell informs about all the neighbors of the entity node.

b) Dynamic Programming

Dynamic programming is one of the most commonly used methods for effective process optimization solutions. The principle of dynamic programming is to evaluate the smallest subproblems until the largest. The algorithm based on the principle of dynamic programming is described in 4 points [19]:

- creating characteristics of the optimal solution,
- defining a recursive optimal solution,
- solving the task from the smallest problems,
- constructing a solution from calculations.

In the case of the shortest path, we will use the theory of graphs and networks, and therefore it is necessary to introduce the terminology used. The first and main term is an acyclic network, which can be represented by a graph having a nonempty finite set of nodes $V = \{v_i\}$ and a set of edges $H = \{h_{i,j}\}$. The acyclic network can be described by a subset $V \times V$, which determines the topology of the graph.

The elements of the set $h_{i,j} = \{v_i, v_j\}$ are thus the edges of the network, which express the existence of the transition from node v_i to node v_j. A node from which at least two edges protrude is called a branching node. The result we need to get from the solution is a path or route $S \{v_1, v_2, \ldots v_n\}$ with a finite sequence corresponding to the node for the shortest path [4].

In our case, it will be an acyclic network with 20 points, which is shown in (Fig.3). Nodes are represented by an empty circle in which the own sequence number is given. The edges represent the lines that connect the nodes, and the numbers above the edges are the distance between the nodes. The symbol C_i denotes the distance to the target point.

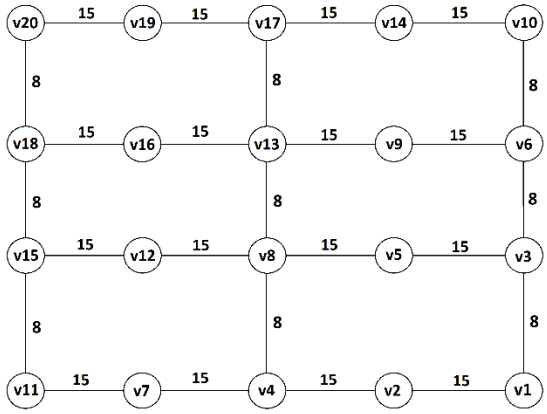

Fig.3. Acyclic network of application.

We will clearly explain the whole principle on the example with the start node v_1 and the end node v_9 in three steps. In the first step, starting from node v_1, whose adjacent nodes are v_2 with a distance to the end point of 46 meters and v_3 of 23 meters, we find that the node v_3 is closer to the end node and we move to it (Fig.4a). In the second step, we have a choice of three adjacent nodes v_1, v_5 and v_6, again we compare their distances to the end point and move to the nearest, which corresponds to node v_6 (Fig.4b). In the third step, the source node v_6 also has three adjacent nodes v_3, v_9 and v_{10}. It can be seen (Fig.4c) that the node v_9 has a distance of 0, as it is the end point. We move to it and find the shortest route between nodes v_1 and v_9.

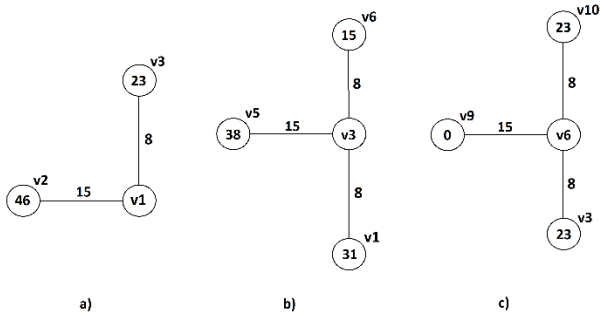

Fig.4. The procedure for finding the shortest path.

c) Application Design

The application is created in Promotic environment, version 8.3.11, using a developer license. The license allows the use of an unlimited number of variables (329 variables were used in the work) with a limited runtime for one hour. The application consists of HMI and the structure of global objects. The structure of objects is divided according to their type. A separate folder with a corresponding name is created for each object type. We also tried to name each object authentically according to its function for better consistency of application scripts.

Fig.5. The main panel: Roofs, Colors, Insulation, Flooring, Cement, Pavement.

In the following section, we will describe all the objects that make up HMI applications and we will take a closer look at all the elements in the panel and their functionality. As mentioned above, the main panel is the warehouse panel, which is displayed to the user after entering the login details correctly. Therefore, we will clarify this panel first.

The main panel consists of two parts, the control panel and the display panel. The display panel provides a top view of the warehouse with information about the category of shelves and to display the filling of the warehouse after selecting a certain side of the shelf, it opens a modal rack window, the functionality of which we will explain in the next section. It also provides a display of the shortest route when instructed by the storekeeper. The control panel, in turn, provides a change in the color of the environment, the display of the legend, the display of the next or previous route, the logout of the user and the loading or unloading of goods from the shelf. We will describe all these functionalities in more detail below.

The application contains three modal windows, which can be accessed from the main window. These modal windows are the main functions of the application. Next, we will describe all modal windows found in the application.

The modal import window shows us the environment in which the user can insert the parameters of the package and then write them to the database. The main parameters are the six-digit number of the package in the range from 000001 - 999999, the category to which the package belongs, its weight in kilograms, the number of pieces and the name of the goods. The application is created for packages on euro pallets, so the X and Y dimensions are predefined to the dimensions of the euro pallet in centimeters.

The modal export window allows you to search for packages in the database after entering the name in the Product field and the number of pieces in the Number of pieces field. After entering the values and pressing the Load button, the package corresponding to the number of pieces will be searched, or several packages, if the selected number of pieces is not contained in one package. If there are not enough pieces of the selected product in the database, the text will appear in the notification window: “There are not enough pieces of goods in stock!”.

Alternatively, if there is no package with the specified name in the warehouse, the notification window will display: “The given goods are not in the warehouse!”. However, if a package with a sufficient number of pieces is in the database, a notification window will display the message: “Package found” and a table drawn by the Canvas method in JavaScript will be displayed with the number and storage of the found package.

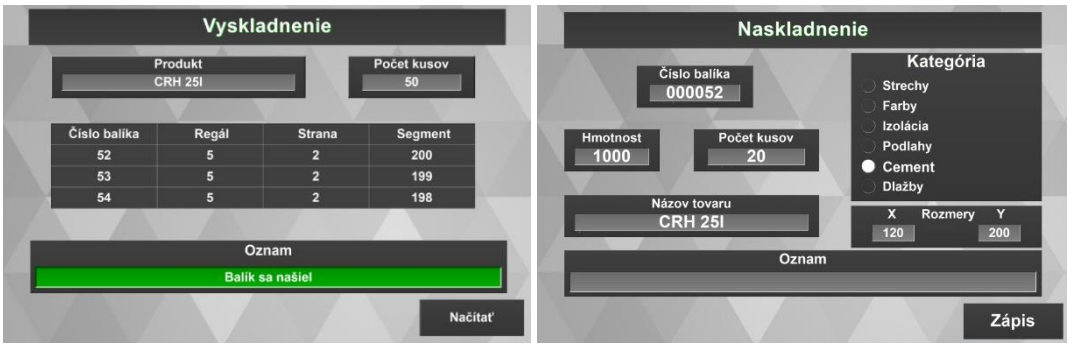

Fig.6. The modal Export Window (left): **Stock-out**, 50 pieces, Number of package, Numbers of position, green text: Package is found, Read button. The modal Import Window (right): **Stock-in**, Package number weight pieces, Name of item, Category, Dimensions XY, Write button.

The modal rack window displays all full segments on a given side of the rack with the name of the package that is in the segment. This allows the user to see which packages are physically in stock. There is also a Back button in this modal window that allows you to close the modal window.

Fig.7. The modal rack window: names and flask volumes of chemicals in stock.

Finding the shortest path consists of three main steps. In the first step of inserting and unloading packages from the warehouse, according to the entered data, suitable storages will be found, which will be written to a text file in a common directory for both applications. In this way, all entered data is written with the found segments into which packages can be inserted or taken. This process is repeated until the modal import or export window is closed. This will allow the application to decide on the optimal sequence of the warehouse keeper’s interaction with the packages on the road.

The second step is to get the values of the distance from the current point where we are to the given segment from the Distance table for each segment from the text file. For each segment, we compare its distance value with previous values of segments that have already been compared. If the value of the segment being compared is less than the previous one, its distance value is entered in the variable Shortest Distance and in the appropriate field the number of the rack and the number of the page in which the segment is located. After comparing all the segments from the text file, we delete the package, which we insert first into the warehouse and all its suitable segments. This will get the package with the next save and perform the third step for the obtained save.

In the third step, we will use the knowledge of dynamic programming, while we have the current node in which we are and the node in which we need to get. First, we read the values from the database of dynamic programming values, specifically from the neighbor table for the current point. This will get all the nodes connected to the current point and for each of them we will get the value of the distance to the end node. Evaluate the node that has the shortest distance to the end node as the current node and set the visibility of the respective edge to true in the visibility field. We perform this procedure until the current point is the end point. When we arrive at the end node, we load all suitable segments on the given side of the rack, select the nearest segment and set the end node as the current one. The visibility field is then written to a text file, from which the paths to the packages are later read. When specifying multiple packages for picking or stocking, steps two and three are repeated for all packages entered.

The application also contains additional functions for better work with the application, increased security of the application and faster acquaintance of the user with the application. Next, we exchange and describe all additional functions that the application contains.

As with all systems, our application requires credentials to run. The registration is performed by the administrator, as the application is intended for a narrow working team. Login is not case sensitive. After pressing the Login button, the entered data are compared with the data from the database and, if they match, the main panel is opened. When logging in, the user can choose one of the seven languages in which the application interface will be displayed. The default language is Slovak, after pressing one of the buttons in the language selection section, the texts will be overwritten in the selected language and the value of the selected language will be entered into the database. This outlines another function of the application, where after logging in, the language of the application changes to the last language selected by the user.

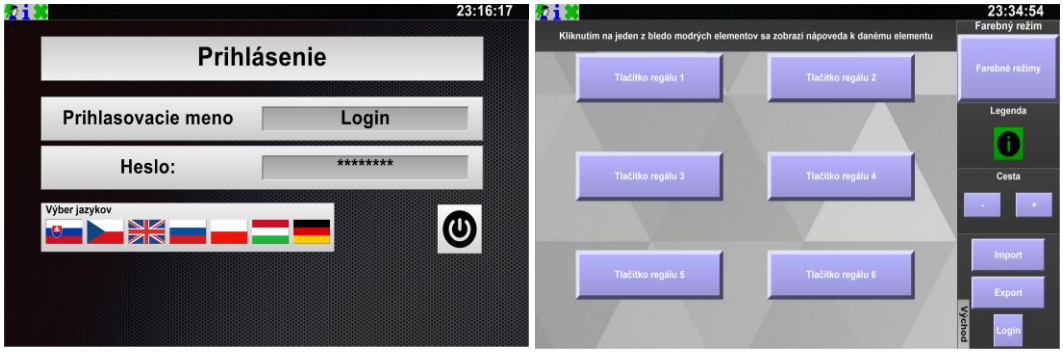

Fig.8. Login+Password+Language panel (left), Help panel (right): shelf number 1-6.

Another feature of the application is help. Help provides information about all the elements that are on the taskbar. When the help is activated, all buttons are highlighted in pale blue and the information panel is displayed at the top of the window. When you press one of the buttons, its function appears in the information bar.

The application includes the ability to switch between day and night mode. The difference between these two modes is the color of the application interface. In day mode, the elements are brighter with black text to achieve optimal visibility in daylight. In night mode, the elements darken, the background is highlighted, and the text turns white. This maintains the same visibility of the elements, but with less contrast of the colors of the elements at night. Mode switching is set to automatic switching when the application is opened, day mode is active from 6:00 to 18:00 and night mode at night from 18:00. The user can also choose to turn off automatic switching and set the mode manually.

Conclusion

The aim of the paper was to develop and to create an application with optimization of material movement in the warehouse. We dealt with the calculation and subsequent display of the shortest path to the packages that needed to be stored in or removed from storage.

The developed application could be used in warehouses of various sizes and employee structures. The application would run separately on multiple devices, connecting to a database stored in the cloud that would be common to all devices.

Two situations can occur in a company. The first one is that one employee would be in charge of recording the packages received in the warehouse or removed from storage to hand over the packages from the warehouse. The second employee would being shown the shortest way to remove or store packages. The second situation is that one employee performs both options, in this case he would enter the packages into the application himself, to which he would then see the shortest route.

Acknowledgement

This paper was supported by the Scientific Grant APVV-17-0190 and by Scientific Grants the Slovak Grant Agency KEGA 016STU-4/2020 and 039STU-4/2021.

References

[1] Poatsy M. A., Williams J., Rutledge A. (2019), *Exploring Microsoft Office Access 2019 Comprehensive Spiral-bound*, ISBN 978-01-354-3581-6

[2] Koreň, M. (2009). *Databázové systémy*, ISBN 978-80-228-2084-4

[3] Delikát, Tomáš; Základy databázových systémov. 2006. ISBN 809694844X

[4] Hudzovič, Peter: Optimalizácia. 2001. ISBN 85-259-2001

[5] Poorvika Prakash (2019), *Pick to Light, Light-directed picking technology in Warehouse | Addverb*, https://medium.com/@prakashpoorvika/pick-to-light-voice-picking-technology-in-warehouse-addverb-59eba48a8cde

[6] Promotic, *Přehled Web technologie v systému PROMOTIC*, https://www.promotic.eu/cz/pmdoc/Subsystems/Web/Web.htm

[7] SvetDopravy. Systémy skladovania v logistických centrách, http://www.svetdopravy.sk/systemy-skladovania-v-logistickych-centrach/

[8] Transport Geography. Cross-Docking Distribution Center, https://transportgeography.org/?page_id=4453

[9] LightningPick. Pick-to-Light, https://lightningpick.com/products/pick-to-light/

[10] PuttoLight. Put-to-light Overview, https://puttolight.com/overview/

[11] W3school. DBMS Introduction, https://www.w3schools.in/dbms/intro/

[12] Worldvector. Microsoft Access 2013 logo, https://worldvectorlogo.com/logo/microsoft-access-2013

[13] Oshiro. Scada - SCADA Remote Terminal Unit Modbus Programmable Logic Controllers Diagram, https://favpng.com/png_view/scada-scada-remote-terminal-unit-modbus-programmable-logic-controllers-diagram-png/SkP7eHt1

[14] Tutorialspoint. MS ACCESS – Overview, https://www.tutorialspoint.com/ms_access/ms_access_overview.htm

[15] SCADA Systems. SCADA systems, http://www.scadasystems.net/

[16] Ing. Ján Cigánek PhD., Database and visualization systems – lectures.

[17] Promotic. Co je Promotic, https://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsPromotic.htm

[18] Promotic. Databázové možnosti v systému Promotic, https://www.promotic.eu/cz/pmdoc/Subsystems/Db/Database.htm

[19] Input. Dynamické programovanie, http://input.sk/struct2017/08.html

Authors

Ing. Filip Žemla
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
filip.zemla@icloud.com

Currently a student of doctoral studies at Slovak University of Technology in Bratislava. The main focus of his studies is oriented to virtualization and optimalisation of modern manufacture processes. His main skills are SCADA systems, database systems and front-end programming.

Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ciganek@stuba.sk

He was born in 1981 in Malacky, Slovakia. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in Bratislava, in 2005 and 2010, respectively. He is now Assistant Professor at Institute of Automotive Mechatronics FEI STU in Bratislava. His research interests include optimization, robust control design, computational tools, SCADA systems, big data, and hybrid systems.