



УДК 681.5.004.94

ПРИКЛАД МОДЕЛЮВАННЯ СИСТЕМИ ЛОГІЧНОГО КЕРУВАННЯ В ПРОГРАМІ STATEFLOW

AN EXAMPLE OF A LOGICAL CONTROL SYSTEM SIMULATION IN THE STATEFLOW

Левінський М.В.¹, Левінський В.М.²
Levinskyi M.V.¹, Levinskyi V.M.²

¹ Національний університет «Одеська морська академія»,

² Одеський національний технологічний університет, м. Одеса, Україна

¹ National University «Odessa Maritime Academy»,

² Odessa National Technological University, Odessa, Ukraine

ORCID: ¹ <https://orcid.org/0000-0002-6544-5110>, ² <https://orcid.org/0000-0002-3563-528X>

E-mail: ¹ MaxLevinskyi@gmail.com, ² ValeryLevinskyi@gmail.com

Copyright © 2024 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v16i2.2838>

Анотація: Актуальність. Складність розробки алгоритмів і програм логічного керування, наприклад, транспортними ділянками переміщення сипучих вантажів, зростає із збільшенням числа механізмів та датчиків, які забезпечують їх функціонування. При цьому використання лише евристичних методів розробки алгоритмів і програм створює загрозу виникнення помилок та збільшує час, необхідний для тестування. В даному випадку раціонально використати програми, які засновані на теорії кінцевих автоматів і дають можливість моделювати процеси, що управляються подіями.

Мета. Розглянути на прикладах доцільність використання таких програм для побудови алгоритмів і програм системи логічного керування транспортною ділянкою переміщення сипучих вантажів.

Метод. В якості методу дослідження обране моделювання системи керування із застосуванням програм Simulink та Stateflow середовища Matlab.

Результати. Наведені приклади моделювання системи логічного керування із застосуванням програм Simulink та Stateflow. Отримані тексти програм керування на мові SCL для подальшого програмування промислових контролерів Simatic в середовищі TIA Portal.

Висновки. Наведені приклади підтвердили доцільність використання програми Stateflow для побудови та тестування алгоритмів системи логічного керування і автоматичного отримання текстів програм для промислових контролерів. Програма Stateflow може бути рекомендована для підготовки спеціалістів з автоматизації виробничих процесів в отриманні навичок по аналізу і синтезу систем логічного керування.

Abstract: Topicality. The complexity of developing algorithms and programs for logical control, for example, of transport sections for the movement of bulk goods, increases with the increase in the number of mechanisms and sensors that ensure their functioning. At the same time, the use of only heuristic methods of developing algorithms and programs creates a threat of errors and increases the time required for testing. In this case, it is rational to use programs that are based on the theory of finite state machines and provide an opportunity to model event-driven processes.

Goal. To consider the expediency of using such programs for building algorithms and programs of the logical control system for the transport section of the movement of bulk cargoes.

Method. Simulation of the control system using Simulink and Stateflow programs in the Matlab environment was chosen as the research method.

The results. Examples of logical control system simulation using Simulink and Stateflow programs are given. Received control program codes in the SCL language for further programming of Simatic industrial controllers in the TIA Portal environment.

Conclusions. The given examples confirmed the expediency of using the Stateflow program for building and testing algorithms of the logical control system and automatically obtaining program texts for industrial controllers. The Stateflow program can be recommended for the training of production process automation specialists in the acquisition of skills in the analysis and synthesis of logical control systems.

Keywords: Stateflow MATLAB program, synthesis of logical control systems, TIA Portal.



Ключові слова: програма Stateflow MATLAB, синтез систем логічного керування, TIA Portal.

1. Вступ

При керуванні технологічними процесами система автоматичного керування (САК) реалізує функції передпускової сигналізації, блокування від переростання передаварійних ситуацій в аварію, автоматичного пуску та зупинки механізмів в певній послідовності та інші, які можна віднести до функцій логічного керування, а також функції стабілізації змінних процесу на заданому значенні, оптимізацію цього значення, налаштування параметрів регулятора та інші, які можна віднести до функцій регулювання. Відповідно САК умовно поділяють на підсистеми логічного керування та регулювання. Методи аналізу і синтезу підсистеми регулювання складають основу теорії автоматичного керування, широко відомі і достатньо розвинені для вирішення практичних задач. В цій статті розглядаються методи аналізу і синтезу підсистеми логічного керування на простому прикладі транспортної ділянки переміщення сипучих матеріалів.

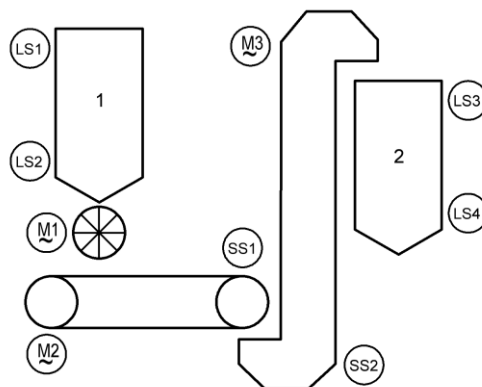


Рис. 1 Приклад транспортної ділянки переміщення сипучих матеріалів
Fig. 1 Example of a transport section for moving bulk materials

Сипучий матеріал із ємності 1 за допомогою живильника з електроприводом M1, транспортера з електроприводом M2 та норії з електроприводом M3 необхідно перемістити в ємність 2. Алгоритм керування цим процесом повинен передбачати послідовність вмикання електроприводів проти руху матеріалу, вимкання навпаки - за рухом матеріалу із затримками у часі, достатніми для видалення залишків матеріалу з транспортера і норії, та враховувати сигнали від датчиків-реле рівня матеріалу в ємностях LS1, ...LS4, а також сигнали про недопустимі рівні швидкості руху стрічок транспортера і норії від датчиків-реле SS1, SS2.

Ставиться задача розробити цей алгоритм керування та відповідну програму для програмованого логічного контролера (ПЛК) і перевірити їх роботоспроможність.

2. Аналіз літературних даних і постановка проблеми

Алгоритми, подібні описаному у вступі, є складовою частиною більш складних алгоритмів вибору оптимального маршруту руху матеріалу, наприклад, на елеваторі [1]. Алгоритми маршрутизації доволі часто описують із застосуванням теорії графів, коли розглядається орієнтований граф із ваговими коефіцієнтами в кожній гілці.

В [2] на основі огляду методів алгоритмізації та програмування для систем логічного керування сформульовано основні положення технології алгоритмізації та програмування для цього класу систем, у рамках якої алгоритми та програми розглядаються як кінцеві автомати.

Аналогічний підхід реалізувала в своєму програмному продукті Stateflow відома американська фірма MathWorks Inc [3, 4, 5].

Проблема розробки алгоритмів і програм логічного керування, наприклад, транспортними ділянками переміщення сипучих вантажів, полягає в зростанні складності розробки із збільшенням числа механізмів та датчиків, які забезпечують функціонування цих ділянок. При цьому використання лише евристичних методів розробки алгоритмів і програм створює загрозу виникнення помилок та збільшую час, необхідний для тестування. Тому в даному випадку раціонально використати програми, які засновані на теорії кінцевих автоматів і дають можливість моделювати процеси, що управляються подіями.

3. Мета і завдання дослідження

Розглянути на прикладах доцільність використання таких програм для побудови алгоритмів і програм системи логічного керування транспортною ділянкою переміщення сипучих вантажів.

4. Методи і матеріали досліджень

В якості методу дослідження обране моделювання системи логічного керування (СЛК) із застосуванням програм Simulink [6] та Stateflow [7] середовища Matlab.

В моделі Simulink (рис. 1) СЛК транспортної ділянки імітується блоком Chart, вхідні сигнали від датчиків-реле і сигнал початку роботи Start – блоками Step та Compare, а команди про вмикання електроприводів з виходів



Chart подаються на багатокомпонентний осцилограф Scope. Імена і тип сигналів налаштовуються в підпрограмі Model Explorer, яку можна викликати з меню View Simulink.

В ході досліджень розглядалися два варіанти побудови блоку Chart Motor. Перший (рис.2) містить три паралельно працюючі підчати M1, M2, M3, стан яких запам'ятовується між циклами моделювання завдяки встановленню опції History. В свою чергу кожен із цих підчатів містить два внутрішніх підчати StopMn, StartMn, які поєднуються між собою стрілками-подіями, умови виконання яких записані поруч з ними. Цей варіант виник першим як результат аналізу технічної структури СЛК.

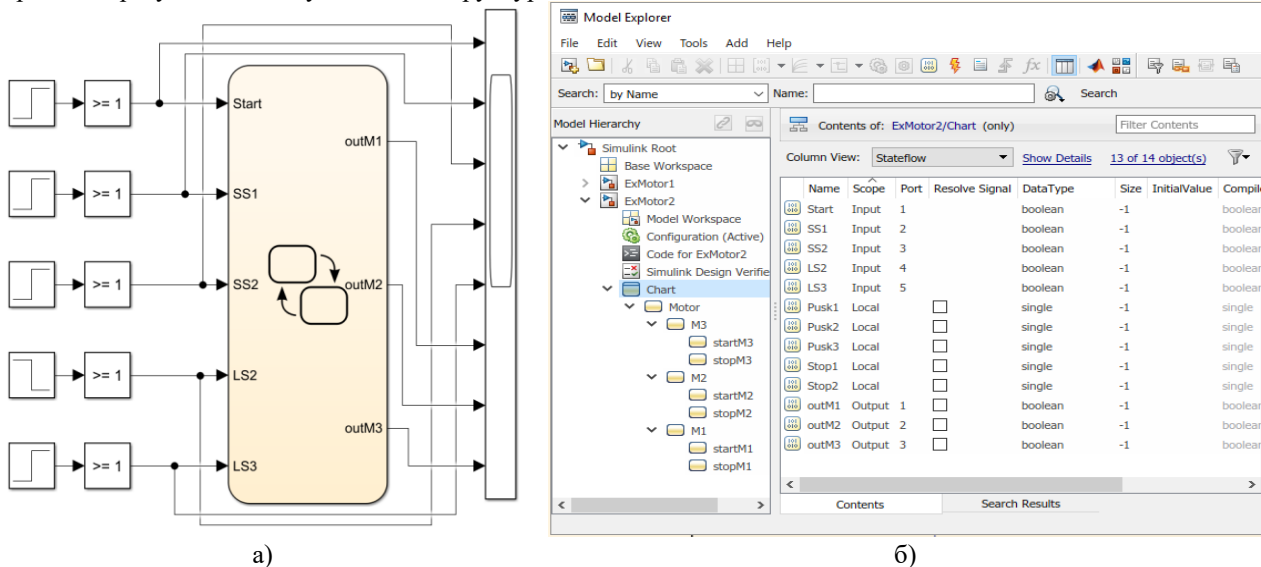


Рис. 1 Моделювання СЛК в програмі Simulink:

- a) схема моделювання; б) налаштування імен і типів входів та виходів моделі
 Fig. 1 Simulation of SLK in the Simulink program:
 a) modeling scheme; b) setting the names and types of inputs and outputs of the model

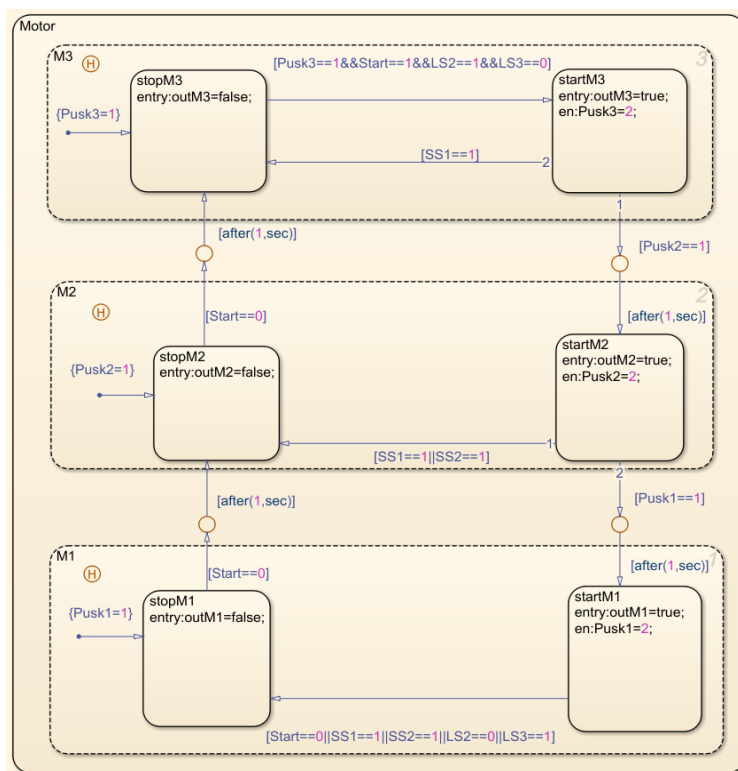


Рис. 2 Перший варіант блоку Chart Motor
 Fig. 2 The first version of the Chart Motor block

При побудові другого варіанту блоку Chart Motor (рис. 3) розглядалися стани не окремих електроприводів, а стани транспортної ділянки в цілому в підчатах Init, Push, Ostanov, Avaria1, Avaria2. Підчат Init виконується першим, оскільки має на вході подію по замовчуванню Default Transition. Перехід від нього до підчату Push здійснюється при співпаданні одразу трьох умов $Start==1 \& \& LS2==1 \& \& LS3==0$, тобто коли надійшла команда



від оператора Start, є матеріал в бункері 1 та є місце для його вивантаження в бункері 2. Перехід від підчату Pusk до підчату Ostanov можливий при виконанні однієї із умов $Start==0 \parallel LS2==0 \parallel LS3==1$.

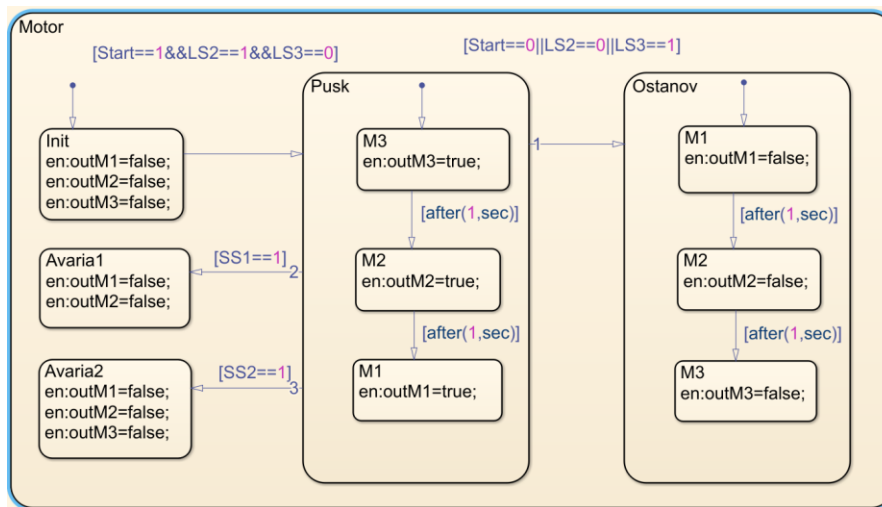


Рис. 3 Другий варіант блоку Chart Motor
Fig. 3 The second version of the Chart Motor block

Команди на аварійні зупинки електроприводів формуються при переході від підчату Pusk до підчатів Avaria1, Avaria2 при виникненні подій $SS1==1$, $SS2==1$, тобто при спрацюванні датчиків реле швидкості транспортера або норії. В підчатах Pusk і Ostanov стан внутрішніх підчатів M1, M2, M3 перемикається в заданій послідовності із необхідними затримками у часі, які були реалізовані завдяки застосуванню функції $after(n, sec)$. При вході в кожен із цих підчатів виконується команда $en:outMn=true$ або $en:outMn=false$, яка змінює значення вихідних командних змінних блоку Chart Motor.

5. Результати досліджень

При моделюванні СЛК в програмі Simulink з першим варіантом блоку Chart Motor (рис. 2) спочатку були виявлені перемикання між станами внутрішніх підчатів stopM3 та startM3, для усунення яких довелося ввести до розгляду додаткові локальні змінні Pusk1, 2, 3 та ускладнити умови переходу між підчатами startM3, startM2, startM1 з використанням перехрестів Junction. Однак, як свідчать результати моделювання, представлені на рис. 4а, так і не вдалося досягнути коректної роботи СЛК, наприклад, при зменшенні рівня матеріалу в ємності 1 і спрацюванні датчика-реле рівня LS2 (перехід $1 \rightarrow 0$) змінилася лише команда outM1 на вимкання електроприводу M1, а команди outM2, outM3 залишились без змін, що протирічить алгоритму функціонування транспортної ділянки.

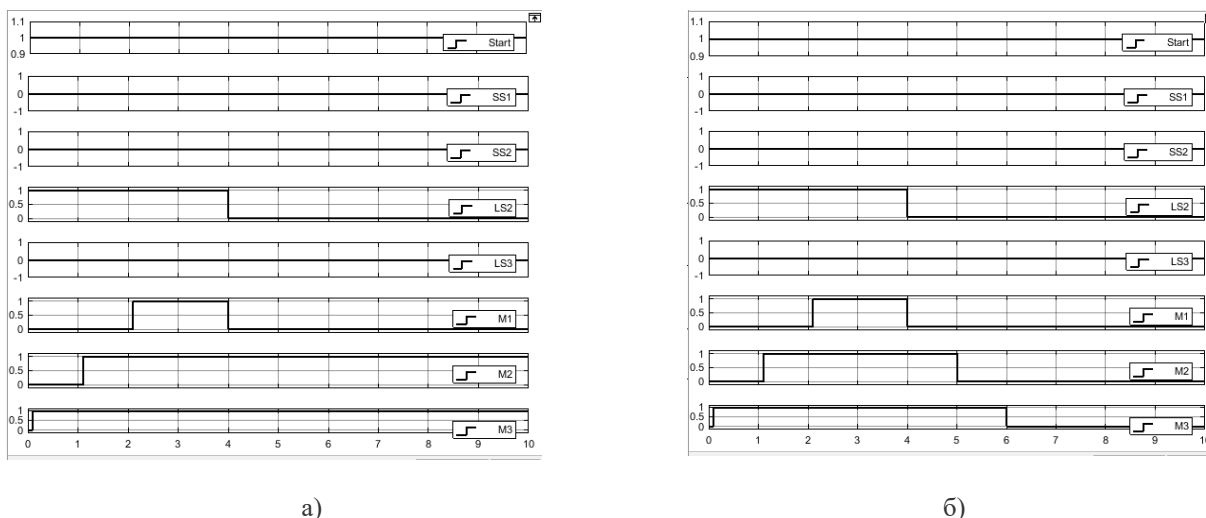


Рис. 4. Результати моделювання СЛК при використанні:
а) першого варіанту блоку Chart Motor; б) другого варіанту блоку Chart Motor

Fig. 4. Results of SLK simulation when using:
a) the first version of the Chart Motor unit; b) the second version of the Chart Motor block

Результати моделювання СЛК в програмі Simulink, представлені на рис. 4б, з другим варіантом блоку Chart Motor (рис. 3), показують, що система керування працює коректно. У відповідності з алгоритмом функціонування транспортної ділянки спочатку проходить послідовний пуск всіх електроприводів, а при зменшенні рівня



матеріалу в ємності 1 і спрацюванні датчика-реле рівня LS2 проходить послідовна зупинка електроприводів із затримками у часі, які потрібні для сходу залишків матеріалу з транспортера і норії.

При подальшій перевірці всіх можливих комбінацій вхідних сигналів було зроблено остаточний висновок, що СЛК з другим варіантом блоку Chart Motor працює коректно і тому можна перейти до автоматичного формування коду програми для ПЛК. Для цього слід виконати ланцюг команд: меню Code→PLC Code→Generate Code for Subsystem, в результаті яких отримана програма на мові SCL.

Перенесення отриманого текстового коду програми, яка в даному прикладі складається з двох функціональних блоків FUNCTION_BLOCK PLC_CODER_TIMER та FUNCTION_BLOCK CHART, в середовище програмування TIA Portal труднощів не викликає, лише треба зважати на відмінність в описі інтерфейсу цих блоків – автоматично згенерованого в MATLAB і прийнятого виду в TIA Portal (рис. 5)

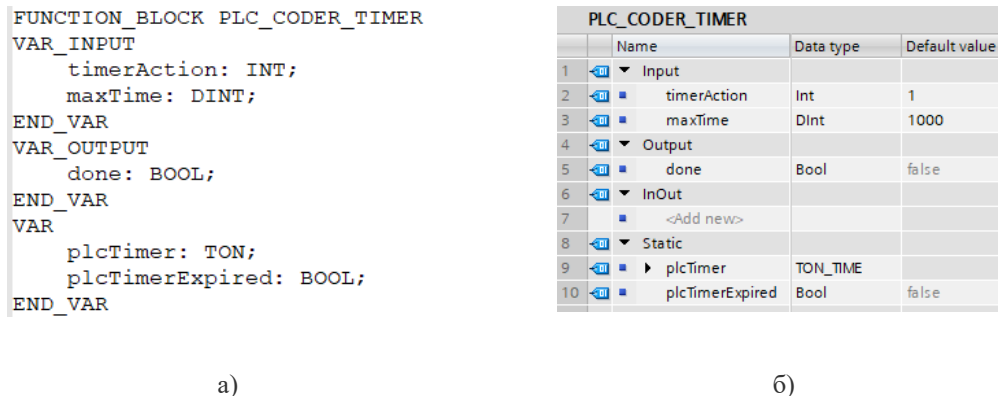


Рис. 5 Відмінність в описі інтерфейсу функціональних блоків:
а) автоматично згенерованого в MATLAB; б) прийнятого в TIA Portal
Fig. 5 The difference in the description of the interface of functional blocks:
a) automatically generated in MATLAB; b) accepted in the TIA Portal

Перевірка роботоспроможності автоматично згенерованого коду при його застосуванні в TIA Portal показала (рис. 6), що алгоритм роботи СЛК транспортною ділянкою виконується коректно.

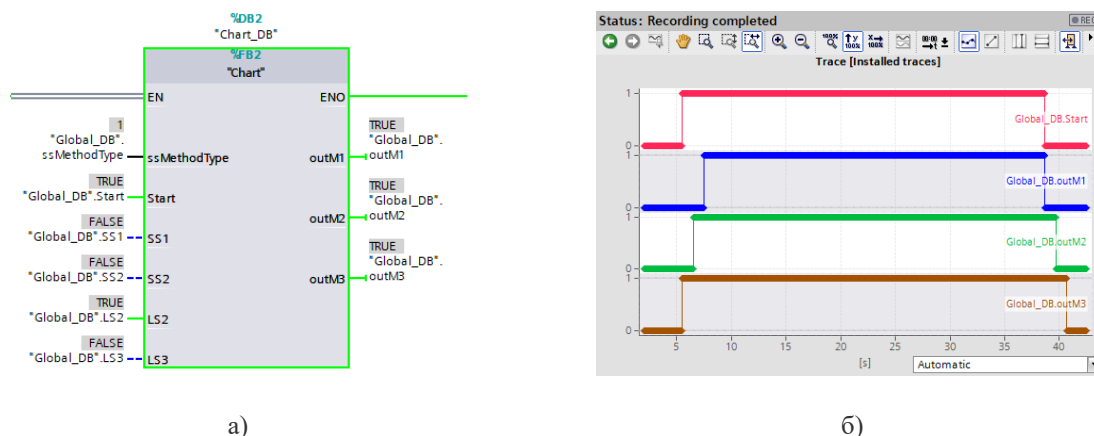


Рис. 6 Результати моделювання СЛК в середовищі TIA Portal:
а) вигляд функціонального блоку Chat в режимі online;
б) трасування вхідної змінної Start і вихідних змінних outM1, outM2, outM3
Fig. 6 Results of SLK modeling in the TIA Portal environment:
a) view of the Chat functional block in online mode;
b) trace input variable Start and output variables outM1, outM2, outM3

6. Обговорення результатів

Практика моделювання СЛК в Simulink/Stateflow та реалізація автоматично згенерованих кодів програми в TIA Portal значно скорочує строки розробки алгоритмів і програм, зменшує ризик виникнення помилок для більш складних об'єктів керування.

Розглянутий у статті підхід дозволяє згенерувати коди програм і для інших сімейств ПЛК.

7. Висновки

Наведені приклади підтвердили доцільність використання програми Stateflow для побудови та тестування алгоритмів системи логічного керування і автоматичного отримання текстів програм для промислових контролерів. Програма Stateflow може бути рекомендована для підготовки спеціалістів з автоматизації виробничих процесів в отриманні навичок по аналізу і синтезу систем логічного керування.

**Список використаних джерел**

- [1]. Т. Фесун. Елеваторна промисловість: традиції та інновації. Вітчизняний та світовий досвід. Київ: Національний університет харчових технологій, 2021. [Електронний ресурс]. https://dspace.nuft.edu.ua/jspui/bitstream/123456789/34263/3/Elevator_industry_traditions_and_innovations._Domestic_and_world_experience.pdf
- [2] Y.Bagljk, and A. Shalyto. "SWITCH-technology. Algorithmic and programming methods in solution the logic control problems of shipping equipment". Intern. conference on informatics and control. ICI & C'97. Proceedings. Vol.1, 1997.
- [3]. Stateflow and Stateflow CoderUser's Guide. USA:The MathWorks Inc, 2009.
- [4] D. Shames, and D. Rosner. "Stateflow: A simulation environment for embedded systems". IEEE Control Systems Magazine, 19(4), 49-57, 1999.
- [5] M. Burke. Stateflow Best Practices. USA:The MathWorks Inc, 2012.
- [6] <https://uk.mathworks.com/products/simulink.html>
- [7] <https://uk.mathworks.com/products/stateflow.html>

References

- [1]. Т. Фесун. Elevator industry: traditions and innovations. Domestic and international experience. Kyiv: National University of Food Technologies, 2021. [Electronic resource]. https://dspace.nuft.edu.ua/jspui/bitstream/123456789/34263/3/Elevator_industry_traditions_and_innovations._Domestic_and_world_experience.pdf
- [2] Y. Bagljk, and A. Shalyto. "SWITCH-technology. Algorithmic and programming methods in solving the logic control problems of shipping equipment". Intern. conference on informatics and control. ICI & C'97. Proceedings. Vol. 1, 1997.
- [3]. Stateflow and Stateflow CoderUsers Guide. USA: The MathWorks Inc, 2009.
- [4] D. Shames, and D. Rosner. "Stateflow: A simulation environment for embedded systems". IEEE Control Systems Magazine, 19(4), 49-57, 1999.
- [5] M. Burke. Stateflow Best Practices. USA: The MathWorks Inc, 2012.
- [6] <https://uk.mathworks.com/products/simulink.html>
- [7] <https://uk.mathworks.com/products/stateflow.html>

Отримана в редакції 26.04.2024. Прийнята до друку 20.05.2024. Received 26 April 2024. Approved 14 May 2024. Available in Internet 23 July 2024

УДК 004.934.2

DEVELOPMENT OF VOICE ASSISTANT BASED ON NEURAL SYSTEMS

РОЗРОБКА ГОЛОСОВОГО ПОМІЧНИКА НА ОСНОВІ НЕЙРОСИСТЕМ

Svitlana Tyshchenko¹, Oleksandr Zhebko², Yevhen Zakharchenko³, Yan Balyuk⁴
Світлана Тищенко¹, Олександр Жебко², Євген Захарченко³, Ян Балуєк⁴

^{1,2,3,4} Mykolayiv National Agrarian University, Mykolayiv, Ukraine

ORCID: ¹<https://orcid.org/0000-0001-7881-8740>, ²<https://orcid.org/0009-0009-1604-5952>

E-mail: ¹tyschenko@mna.u.edu.ua, ²zhebko@mna.u.edu.ua, ³jekazinferno@gmail.com, ⁴balyuk.yangare@gmail.com

Copyright © 2024 by author and the journal "Automation of technological and business – processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v16i2.2854>

Abstract. The development of a voice assistant based on neural systems is an actual line of research in the field of artificial intelligence. Existing voice assistants were analyzed and their advantages and disadvantages were identified in this paper. The following concepts were considered: natural language processing, and neural networks. Different methods of natural language processing significantly increase the quality of work of many programs, because they can