

© А.В. Бубліков¹, О.В. Репич¹, С.Є. Зубенко¹, Д.В. Славінський¹, І.Ю. Колисниченко¹

¹ Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

СПОСІБ ДИСТАНЦІЙНОЇ ЗМІНИ СХЕМИ АЛГОРИТМУ В КОНТРОЛЕРІ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ КЕРУВАННЯ ТЕХНОЛОГІЧНИМИ ОБ'ЄКТАМИ

© A. Bublikov¹, O. Repych¹, S. Zubenko¹, D. Slavinskyi¹, I. Kolysnychenko¹

¹ Dnipro University of Technology, Dnipro, Ukraine

METHOD OF REMOTE CHANGE OF THE ALGORITHM SCHEME IN THE CONTROLLER FOR AUTOMATION OF CONTROL PROCESSES OF TECHNOLOGICAL OBJECTS

Мета. Підвищити ефективність процесу автоматизації дистанційного програмування контролерів за рахунок створення способу передачі алгоритмів автоматичного керування технологічними об'єктами та процесами через мережу Інтернет або локальну мережу за допомогою посилки, яка активує певні фрагменти програмного коду.

Методика. Для завдання алгоритмів автоматичного керування технологічними процесами використані теорія графів та булева алгебра. Окрім того, як одна з форм представлення знань використані продукційні правила для опису алгоритму керування. Для перетворення системи продукційних правил в сукупність матриць використані елементи теорії матричного представлення. Також для підтвердження ефективності запропонованого рішення застосовані методи комп'ютерного моделювання.

Результати. Створений спосіб перетворення алгоритмів автоматичного керування, що задані у формі системи продукційних правил, на сукупність чисельних матриць для дистанційної зміни алгоритму автоматичного керування технологічним процесом, що програмно реалізується в контролері.

Наукова новизна. Встановлені нові закономірності матричного перетворення правил продукцій як форми представлення знань про алгоритм автоматичного керування технологічними процесами для подальшої передачі по мережі у вигляді посилки в контролер. Для цього окремо створюються матриці вхідних величин, вихідних величин, переходу до логічних величин та виконання над ними логічних операцій з урахуванням послідовності їх виконання, що задається умовами правил продукцій. Заповнення матриць відбувається за спеціально створеними правилами, які дозволяють відтворювати на програмному рівні контролера продукційні правила будь якої структури зі змінними вхідними та вихідними величинами алгоритму керування. Запропоновані закономірності набули подальшого розвитку за умови створення способу дистанційного програмування контролерів.

Практична значимість. Запропонований спосіб є необхідною умовою для створення складних багаторівневих систем автоматизації, на верхньому рівні яких відбувається аналіз зібраної інформації та формування в результаті цього керуючих впливів у вигляді певних тактик керування, що задані системами продукційних правил. Також цей спосіб може бути використаний при застосуванні технології Інтернет речей та в кіберфізичних системах, коли "розумні" компоненти складних систем автоматизації обмінюються між собою не тільки зібраними даними, а й фрагментами бази знань з метою координації дій та виконання сумісної роботи в автономному режимі.

Ключові слова: алгоритм автоматичного керування, продукційні правила, граф переходу, дистанційна зміна алгоритму керування в контролері.

Вступ. У напрямі автоматизації процесів керування рівень scada-систем набуває все більшої функціональності та автономності з позиції керування технологічними процесами. Інтелектуалізація процесів керування технологічними об'єктами за умови реалізації на рівні scada-систем передбачає зміну алгоритмів керування на рівні контролерів автоматично без участі людини в режимі реального часу. Йдеться про, умовно кажучи, формування на рівні scada-систем певних тактик автоматичного керування технологічними процесами, які змінюються в залежності від результатів аналізу зібраних даних з датчиків, та мають виконуватись в контролерах.

Наразі існують різні рішення щодо перепрограмування контролерів в дистанційному режимі.

Одним з прикладів є AnyViz – хмарна платформа, яка дозволяє моніторити, керувати та аналізувати роботу контролерів віддалено [1]. За допомогою безкоштовного програмного забезпечення CloudAdapter, яке підтримує протоколи OPC UA, Modbus, Siemens TCP, Beckhoff ADS та інші, можна підключити контролер до хмари без додаткового обладнання. Платформа підтримує зміну значень у контролері, що дозволяє дистанційно змінювати параметри керування.

Іншим прикладом є CODESYS – інтегроване середовище розробки для програмування контролерів відповідно до стандарту IEC 61131-3 [2]. Воно підтримує віддалене оновлення програмного забезпечення контролерів через мережу, що дозволяє змінювати програми без фізичного доступу до пристрою. Крім того, CODESYS підтримує різні протоколи зв'язку, такі як OPC UA, що забезпечує гнучкість у виборі методів віддаленого доступу.

Ще одним прикладом є FlexiHub – програмне забезпечення, яке дозволяє віддалено підключатися до ПЛК через USB або послідовні порти (RS-232/485/422) [3]. Підтримує Windows, macOS, Linux та Android. Забезпечує безпечне з'єднання та просту інтеграцію з HMI та іншими пристроями, що дозволяє дистанційно змінювати програми керування контролера.

Всі наведені вище рішення дозволяють або змінювати тільки параметри алгоритмів керування, що реалізуються в контролерах, або отримати до них дистанційний доступ для оновлення програмного забезпечення. Тому актуальним є створення способу, що дозволяє дистанційно змінювати алгоритм автоматичного керування, що програмно реалізуються в контролерах, без зміни самої програми.

Основна частина. В основу запропонованого способу дистанційної зміни схеми алгоритму в контролері для автоматизації процесів керування технологічними об'єктами закладений алгоритм, що перетворює систему продукційних правил на цифровий код в матричній формі. Цей код формується як посилка, що передається по комп'ютерній мережі від верхнього рівня системи автоматизації (наприклад, рівень scada-системи автоматизованої системи керування технологічним процесом) на контролер. На контролері ця посилка ідентифікує структуру, певні змінні та логічні операції правил більш складної системи продукцій, в якій кожне правило задане в наступній загальній формі:

ЯКЩО $F(\text{Вхідна змінна №1, Логічна операція №1, Вхідна змінна №2, Логічна операція №2, ..., Вхідна змінна №n, Логічна операція №n})$ ТО $[\text{Вихідна змінна №1} = \text{Значення №1, Вихідна змінна №2} = \text{Значення №2, ... Вихідна змінна №m} = \text{Значення №m}]$

В наведеному вище рівнянні умовою правила фактично є функція F , аргументами якої є вхідні змінні та логічні операції, що можуть об'єднуватись в будь яку структуру, визначаючи таким чином задану на верхньому рівні логіку.

Таким чином, описана загальна форма продукційного правила дозволяє змінювати у часі в автономному режимі як структуру умови правила з вхідними змінними та логічними операціями, так і дискретні значення вихідних змінних, не підключаючись до контролера з метою зміни його програмного забезпечення. Це є умовою реалізації більш складних багаторівневих систем автоматизації, на верхньому рівні яких відбувається аналіз зібраної інформації та формування в результаті цього керуючих впливів у вигляді певних тактик керування, що задані системами продукційних правил.

Також цей спосіб може бути використаний при застосуванні технології Інтернет речей та в кіберфізичних системах, коли “розумні” компоненти складних систем автоматизації обмінюються між собою не тільки зібраними даними, а й фрагментами бази знань з метою координації дій та виконання сумісної роботи в автономному режимі.

Далі наведемо опис запропонованого способу дистанційної зміни схеми алгоритму в контролері для автоматизації процесів керування технологічними об'єктами. При цьому схему алгоритму керування представимо для спрощення сприйняття у вигляді графу переходів (рис. 1), який є однією з поширених графічних форм завдання алгоритмів автоматичного керування [4, 5]:

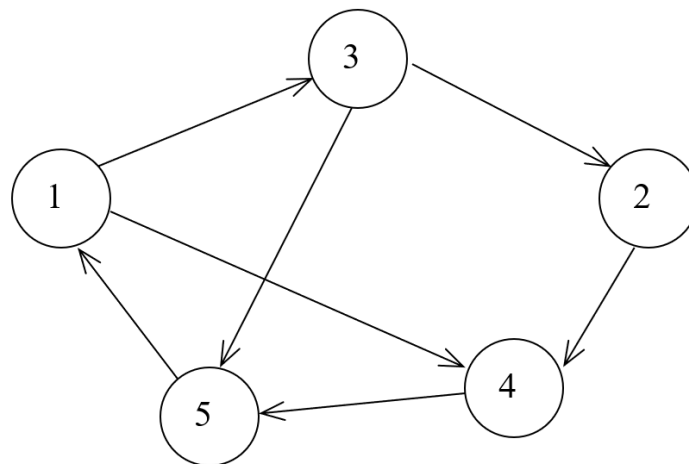


Рис. 1. Граф стану для графічного представлення алгоритму автоматичного керування технологічним об'єктом

За умови опису запропонованого способу будемо розглядати абстрактний алгоритм автоматичного керування без прив'язування до конкретного технологічного об'єкта. Приймемо, що є сім вхідних величин (чотири дискретні, наприклад кнопки на пульті керування (Button), та три неперервні, наприклад, якісь заміряні за допомогою аналогових датчиків фізичні величини (Sensor)). Також маємо три вихідні дискретні величини, що є керуючими впливами на виконавчі пристрої технологічного об'єкту (наприклад, на реле (Rele)). Окрім цього, є два програмно реалізовані таймери (Timer), кожний з одним виходом, що є

неперервною величиною (кількість відрахованих таймером секунд), та одним дискретним входом, який запускає та зупиняє таймер зі скиданням в нуль.

Нижче, як приклад, наведена система продукційних правил, що відповідає графу на рис.1 (описані тільки ті переходи, що є активними):

Перехід 1-3: ((Sensor1 < Sensor2) TA (Sensor1 > Sensor3)) TA Button1=1 /
Rele1=1, Timer2=1;

Перехід 1-4: Button3=1 / Rele2=1;

Перехід 2-4: (Timer1 > 7 TA Button3=1) АБО (Sensor3 > 30 TA Button2=0) /
Rele2=1;

Перехід 3-2: (Sensor3 < 30 TA Button2=1) АБО Sensor1 < Sensor3 / Timer1=1,
Rele1=0;

Перехід 3-5: (Timer2 > 5 TA Button2=1) / Rele3=1;

Перехід 4-5: (Sensor3 > 30 TA Button2=1) / Rele3=1;

Перехід 5-1: Button4=1 / Rele1=0, Rele2=0, Rele3=0, Timer1=0, Timer2=0.

На початку задається матриця вхідних змінних X , в якій для кожного переходу, якому відповідає певний рядок (перехід 1-1 – перший рядок, 1-2 – другий тощо), перелічуються вхідні змінні. При цьому матриця заповнюється на основі таких правил:

1. Кожна пара елементів в рядку виділяється для подальшого формування логічної вхідної змінної першого порядку.

2. Якщо в умові переходу на графі (продукційного правила) вхідна змінна є дискретною, то другий елемент в парі буде або 1 (перехід до логічної змінної першого порядку без змін), або 0 (відбувається інверсія вхідної дискретної змінної).

3. Якщо в умові переходу на графі (продукційного правила) вхідна змінна є неперервною, то другий елемент в парі буде або числом, або іншою неперервною змінною, з якими порівнюється неперервна змінна, що є першим елементом в парі.

Матриця X для прикладу вище, що заповнена за наведеними правилами, буде виглядати таким чином:

$$X = \begin{matrix} & \begin{matrix} Sensor1 & Sensor2 & Sensor1 & Sensor3 & Button1 & 1 \\ & & Button3 & 1 & & \\ Timer1 & 7 & Button3 & 1 & Sensor3 & 30 & Button2 & 0 \\ Sensor3 & 30 & Button2 & 1 & Sensor1 & Sensor3 & & \\ & & Timer2 & 5 & Button2 & 1 & & \\ & & Sensor3 & 30 & Button2 & 1 & & \\ & & & & Button4 & 1 & & \end{matrix} \\ \end{matrix} .$$

Далі задається матриця вихідних змінних Y , в якій для кожного переходу, якому відповідає певний рядок (перехід 1-1 – перший рядок, 1-2 – другий тощо), перелічуються вихідні змінні, які є в діях відповідного продукційного правила. При цьому матриця заповнюється на основі таких правил:

1. Кожна пара елементів в рядку виділяється для подальшого формування вихідної дискретної змінної.

2. В якості першого елемента пари в рядку матриці вказується ім'я дискретної вихідної змінної, а в якості другого елемента – значення цієї змінної.

Матриця Y для прикладу вище, що заповнена за наведеними правилами, буде виглядати таким чином:

$$Y = \begin{matrix} & \begin{matrix} Rele1 & 1 & Timer2 & 1 \\ & Rele2 & 1 \\ & Rele2 & 1 \\ Timer1 & 1 & Rele1 & 0 \\ & Rele3 & 1 \\ & Rele3 & 1 \\ Timer1 & 0 & Timer2 & 0 & Rele1 & 0 & Rele2 & 0 & Rele3 & 0 \end{matrix} \end{matrix}$$

Наступна матриця C (comparison) задає тип логічних операцій порівняння, які мають бути виконані по відношенню до вхідних змінних, що задані в матриці X . Ця матриця заповнюється за наступними правилами:

1. В першому елементі рядка задається тип операції порівняння по відношенню до першої пари елементів того самого рядка в матриці X , в другому – для другої пари тощо.

2. Тип операції порівняння визначається цифрою (1 – це менше, 2 – менше або дорівнює, 3 – дорівнює, 4 – більше або дорівнює, 5 – більше, 6 – не дорівнює).

Матриця C для прикладу вище, що заповнена за наведеними правилами, буде виглядати таким чином:

$$C = \begin{matrix} & 1 & 5 & 3 \\ & & 3 & \\ 5 & 3 & 5 & 3 \\ 1 & 3 & 1 & . \\ & 5 & 3 \\ & 5 & 3 \\ & 3 \end{matrix}$$

Таким чином, матриці X та C дозволяють для кожного переходу в графі на рис. 1 отримати сукупність логічних вхідних змінних першого порядку, по відношенню до яких далі потрібно здійснити логічні операції в певній послідовності.

Далі формується сукупність матриць O_1 (operation), O_2 , ..., O_n , які визначають тип логічних операцій над логічними вхідними змінними першого порядку з урахуванням послідовності їх виконання. Так, в матриці O_1 вказуються типи логічних операцій, які мають бути виконані в першу чергу. Як підсумок, отримуємо сукупність логічних вхідних змінних другого порядку. Матриця O_1 формується за наступними правилами:

1. Перший елемент рядка задає тип логічної операції, яку потрібно виконати по відношенню до першої та другої логічних змінних першого порядку, другий – по відношенню до другої та третьої тощо.

2. Якщо по відношенню до певних логічних змінних першого порядку не виконується логічна операція, то вводиться нуль.

3. Для переведення логічної змінної першого порядку на другий рівень без змін (з точки зору послідовності виконання логічних операцій) введена спеціальна операція, яка має в нашому випадку код 3.

4. Тип логічної операції визначається цифрою (1 – логічна операція ТА, 2 – логічна операція АБО, 3 – підвищується рівень логічної змінної без її зміни, 4 – інверсія).

Матриця O_1 для прикладу вище, що заповнена за наведеними правилами, буде виглядати таким чином:

$$O_1 = \begin{matrix} & 1 & 0 & 3 \\ & 3 & & \\ 1 & 0 & 1 & \\ 1 & 0 & 3 & \\ & 1 & & \\ & 1 & & \\ & 3 & & \end{matrix} .$$

Таким чином, матриця O_1 дозволяє для кожного переходу в графі на рис. 1 отримати сукупність логічних вхідних змінних другого порядку, частина з яких є результатом виконання логічних операцій на першому етапі з точки зору дотримання заданої послідовності, а інша частина – перенесені логічні змінні з першого рівня без змін.

Матриця O_2 , що задає тип логічних операцій для логічних змінних другого порядку, формується за аналогічними правилами. Для прикладу, що розглядається, вона буде такою:

$$O_2 = \begin{matrix} 1 \\ 3 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{matrix} .$$

Отже, описані матриці містять інформацію як про структуру продукційних правил, так і про вхідні й вихідні змінні та їх значення, які в цих правилах використовуються.

Перевірка дієздатності запропонованого способу дистанційної зміни схеми алгоритму в контролері для автоматизації процесів керування технологічними об'єктами здійснена для наведеного вище прикладу в режимі симуляції в додатку Simulink програми Matlab. Схема моделі, що дозволяє виконати перевірку, наведена на рис. 2.

На схемі на рис.2 значення вхідних змінних задаються за допомогою блоків Constant, а їх зміна за допомогою блоків Manual Switch. Як бачимо на рис. 2, в лівій частині задані блоки, що імітують Кнопки та Датчики.

В правій частині рис. 2 за допомогою чисельних індикаторів відбувається відстеження зміни вихідних змінних – стану реле та таймерів.

В блоці S-Function на рис. 2 знаходиться програма, що реалізує спосіб зворотного перетворення матриць на алгоритм керування з його подальшою програмною реалізацією. Цей спосіб буде розглянутий в наступних роботах. В якості вхідних аргументів програмі в блоці S-Function передаються матриці X , Y , C та O , які наведені вище.

Для перевірки працездатності запропонованого рішення порівняємо між собою рис. 2 (перший стан системи керування на рис. 1) та рис. 3 (третій стан системи керування на рис. 1).

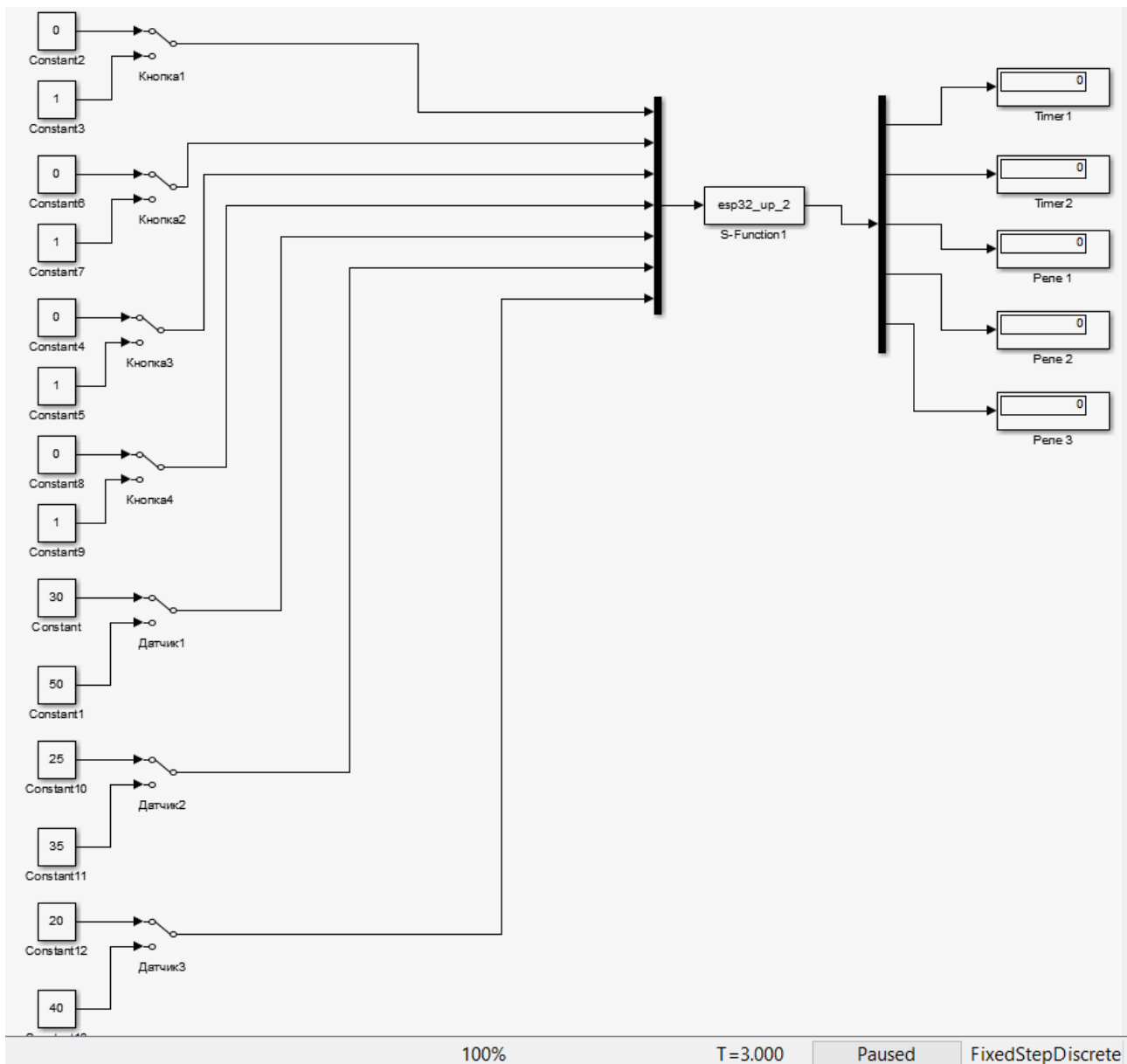


Рис. 2. Схема імітаційної моделі системи дистанційної передачі алгоритму керування на контролер (перший стан системи керування)

Як було описано вище, перехід зі стану 1 в стан 3 системи керування відбувається за умови натиснення Кнопки 1, а також за умови, що вихідний сигнал з Датчика 1 буде менше за вихідний сигнал з Датчика 2 та більше за вихідний сигнал з Датчика 3.

Як бачимо з рис. 3, комбінація перемикачів відповідає умові переходу з першого стану системи в третій. З іншої сторони, в правій частині рис.3 ми бачимо, що значення Реле 1 наразі дорівнює одиниці, а також, що відбувається відлік часу на виході Таймеру 2. Це як раз відповідає діям при переході з першого стану системи керування в третій. Аналогічним чином проведена успішна перевірка коректності всіх інших переходів графу на рис. 1.

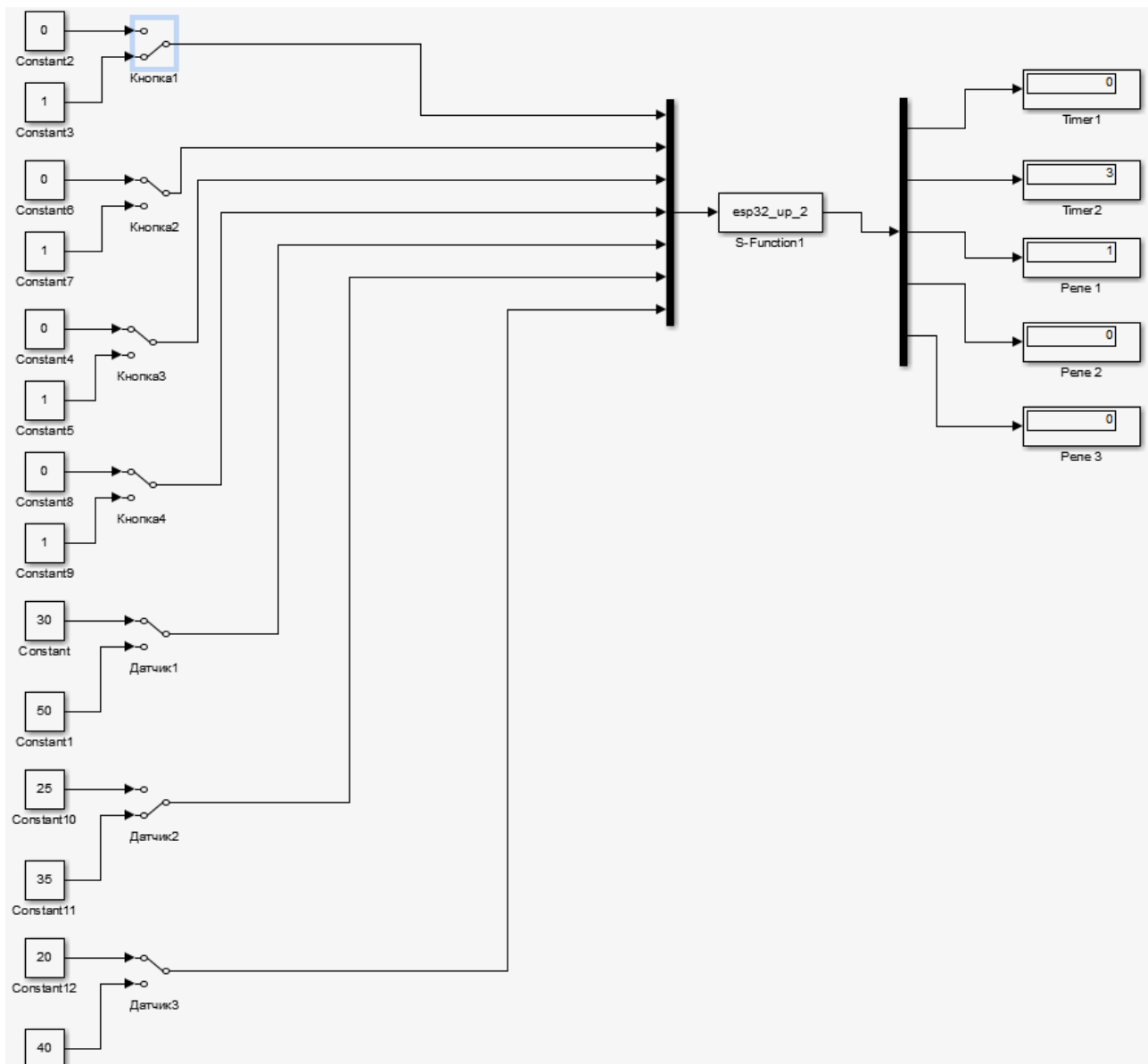


Рис. 3. Схема імітаційної моделі системи дистанційної передачі алгоритму керування на контролер (третій стан системи керування)

Висновки. Для дистанційної зміни алгоритму автоматичного керування в контролері за умови автоматизації процесів керування технологічними об'єктами запропонований спосіб, що дозволяє схему алгоритму трансформувати в сукупність матриць, які заповнюються з певними правилами в залежності від умов та дій переходів на графі стану, який є графічним відображенням алгоритму. В такому вигляді алгоритм керування може без проблем передаватися по мережі за умови подальшого зворотного перетворення в контролері. Таким чином, запропоноване рішення дозволяє в режимі реального часу дистанційно змінювати алгоритми керування, які програмно реалізуються в контролері, без виконання процедури зміни його прошивки. Запропонований спосіб може бути використаний в технології Інтернет речей, в кіберфізичних системах, а також в багаторівневих системах автоматизації, де застосовується інтелектуальне керування технологічними процесами та об'єктами.

Перелік посилань

1. Network King. (n.d.). *Top 10 Programmable Logic Controller (PLC) Monitoring Tools*. <https://network-king.net/top-10-programmable-logic-controller-monitoring-tools/>
2. Wikipedia contributors. (2024, May 10). *CODESYS*. Wikipedia. <https://en.wikipedia.org/wiki/CODESYS>
3. Bastani, O., Duggirala, P., & Natarajan, S. (2024). Agents4PLC: Automating Closed-loop PLC Programming with Large Language Models [Preprint]. arXiv. <https://arxiv.org/abs/2410.14209>
4. Бойко, О.О., Воскобойник, Є.К., Проценко, С.М., & Славінський, Д.В. (2021). Проектування та розробка програмного забезпечення промислових контролерів на базі графів станів. *Збірник наукових праць національного гірничого університету*, 64. 229–238. <https://doi.org/10.33271/crpnmu/64.229>
5. Bublikov, A., & Boyko, O. (2022). Methodology of teaching software development for the systems of automatic discrete control of technological processes. *Collection of theses of scientific and methodical reports of scientific and pedagogical internship : “Vasile Goldis” Western University of Arad (Romani), North University Centre of Baia Mare (Romani), Institute of professional development (Slovakia)*, 109–115.

ABSTRACT

Purpose. To increase the efficiency of the process of automation of remote programming of controllers by creating a method of transmitting algorithms for automatic control of technological objects and processes via the Internet or local network using a message that activates certain fragments of the program code.

The methods. Graph theory and Boolean algebra were used to set algorithms for automatic control of technological processes. In addition, production rules were used as one of the forms of knowledge representation to describe the control algorithm. Elements of the theory of matrix representation were used to transform the system of production rules into a set of matrices. Also, computer modeling methods were used to confirm the effectiveness of the proposed solution.

Findings. A method was created for converting automatic control algorithms, given in the form of a system of production rules, into a set of numerical matrices for remote change of the algorithm for automatic control of the technological process, which is implemented in the controller programmatically.

The originality. New regularities of matrix transformation of production rules as a form of knowledge representation about the algorithm of automatic control of technological processes for further transmission over the network in the form of a package to the controller are established. For this purpose, matrices of input values, output values, transition to logical values and execution of logical operations on them are separately created, taking into account the sequence of their execution, which is given by the conditions of the production rules. The filling of the matrices occurs according to specially created rules, which allow to reproduce at the controller program level production rules of any structure with variable input and output values of the control algorithm. The proposed regularities have been further developed provided that a method of remote programming of controllers is created.

Practical implementation. The proposed method is a necessary condition for creating complex multi-level automation systems, at the upper level of which the analysis of the collected information takes place and the formation of control influences as a result in the form of certain control tactics, which are given by the production rule systems. This method can also be used in the application of Internet of Things technology and in cyber-physical systems, when “smart” components of complex automation systems exchange not only collected data, but also fragments of the knowledge base with the aim of coordinating actions and performing joint work in an autonomous mode.

Keywords: automatic control algorithm, production rules, transition graph, remote change of the control algorithm in the controller.