Machinery & Energetics
Vol. 17, No. 2. 2026

Journal homepage:
https://technicalscience.com.ua/en

UDC 681.587.3
DOI: 10.31548/machinery/2.2026.5

Oksana Hanpantsurova
PhD in Technical Sciences
Bosch Rexroth AG
97816, 1 Zum Eisengießer, Lohr am Main, Germany
https://orcid.org/0000-0002-4709-0336

Oleksandr Hubarev
Doctor of Technical Sciences, Professor
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
03056, 37 Beresteyskyi Ave., Kyiv, Ukraine
https://orcid.org/0000-0002-0924-4103

Kostyantyn Bielikov
PhD in Technical Sciences, Associated Professor
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
03056, 37 Beresteyskyi Ave., Kyiv, Ukraine
https://orcid.org/0000-0002-7393-1848

Alyona Murashchenko*
PhD in Technical Sciences, Associated Professor
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
03056, 37 Beresteyskyi Ave., Kyiv, Ukraine
https://orcid.org/0000-0003-1059-5768

Oleh Levchenko
PhD in Technical Sciences, Associated Professor
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
03056, 37 Beresteyskyi Ave., Kyiv, Ukraine
https://orcid.org/0000-0002-7620-9009

Synthesis and decomposition of a mechatronic system with hydro-pneumatic automation devices

Abstract. The relevance of this study is driven by the increasing complexity of mechatronic systems with hydropneumatic automation devices and the need to improve the efficiency of their design in the context of Industry 4.0 development. The aim of the study was to substantiate the suitability of the cyclic-modular approach for decomposing a mechatronic system into macromodules, synthesising these modules, and subsequently integrating them without compromising the system’s properties. The research methodology was based on the application of systems analysis, decomposition, and logical synthesis principles for the design and investigation of mechatronic systems with hydropneumatic automation devices. Within the synthesis object, the electronic component was represented by the control algorithm of the system’s actuating devices. The authors consider efficient and competitive technical solutions aimed at introducing new functions and increasing the level of automation in accordance with Industry 4.0 trends. An abstract model of a mechatronic system element, referred to as a macromodule, is proposed. The advantages of autonomous configuration and testing of macromodules were demonstrated, showing a significant reduction in development time and simplification of system design. The consequences of deep

Article’s History: Received: 13.01.2026; Revised: 17.04.2026; Accepted: 21.05.2026; Published: 29.05.2026.

Suggested Citation:
Hanpantsurova, O., Hubarev, O., Bielikov, K., Murashchenko, A., & Levchenko, O. (2026). Synthesis and decomposition of a mechatronic system with hydro-pneumatic automation devices. *Machinery & Energetics,* 17(2), 65-77. doi: 10.31548/machinery/2.2026.5.

*Corresponding author (a0976478579@gmail.com)

fragmentation of a mechatronic system were analysed, and the relationship between module complexity and control architecture is established. The results of the control model analysis demonstrated that excessive decomposition leads to the emergence of a separate hierarchical level responsible for coordinating interactions among simplified elements. It was substantiated that the division of a system into macromodules cannot be arbitrary; rather, it was strictly constrained by the physical properties of the equipment and the functions of the design object. The paper proposed evaluating the quality of decomposition not by the number of elements but by the practical reduction in system development time before and after scaling. The optimal size of macromodules is unique to each specific project because it was determined by the functional and physical content of the components. Effective decomposition achieves a balance between macromodule autonomy and the complexity of coordination links, thereby enabling the rapid implementation of sophisticated automation systems

Keywords: control architecture; macro-module; hydraulic devices; functional decomposition; logical control; modular synthesis; control architecture; Industry 4.0

INTRODUCTION

Efficient and competitive technical solutions for any industry are impossible without adding new functions and increasing the degree of automation, which is in line with the trends of Industry 4.0. Authors such as E.H. Aboul *et al*. (2022) presented AI applications in industry, including Internet of Things (IoT) and Industrial Internet of Things (IIoT) technologies. H. Sikandar *et al*. (2021) found that Industry 4.0 is primarily focused on the fields of engineering and computer science. These changes lead to the scaling up of mechatronic and automation systems, as noted by M. Becker *et al*. (2025). A consequence of the increased number of controlled devices in a technical system, from 2-5 to 30-100, is system segmentation. The segments are combined into a larger system that forms a common operating space for all segments. System composition and decomposition are recognised as key approaches in the development, testing, and modernisation of automated technical systems with a significant number of managed and controlled components and multiple operating modes (Mesarovic & Takahara, 1975; Li *et al*., 2025). Similar tasks have also arisen in computing and information technology. As programmable controllers (Programmable Logic Controllers, Smart Relays and Industrial PCs) have spread to other technology fields, developers of traditional hydropneumatic and mechatronic systems have faced similar challenges (Barbanera, 2021; Webert *et al*., 2022; Belgacem *et al*., 2026). If different areas of technology are considered from the point of view of dividing a larger technical system into smaller subsystems, followed by the development of these subsystems and their subsequent integration into a unified system, many common principles and approaches can be identified.

The peculiarities of certain automatic and mechatronic technical systems can be imparted into several aspects that do not allow the direct application of informatics and computer science approaches and methods for the development of automated systems, including the variety of devices and functions both within a single system and among the available technical means used for design (Lambert & Herder, 2016; Cherkashenko, 2023), the natural asynchronicity of pneumatic and hydraulic devices, the non-deterministic timing of repetitive actions in pneumatic and hydraulic processes, the dependence of the duration of hydraulic and pneumatic processes on environmental conditions and the system state, and differences in the representation of safety-related pneumatic, hydraulic, and mechanical elements compared with components used in computer science (Vieira *et al*., 2021).

The aim of the research was to substantiate the suitability of the cyclic-modular approach for dividing a mechatronic system into parts and combining the parts into a mechatronic system without violating the system properties. To achieve this aim, the following tasks were addressed: the formation of a generalised structure of a macromodule for hydraulic and pneumatic automation and mechatronic systems; the determination of the relationships between the macromodule and the system; the determination of the connections between the system and the macromodule; and the verification of the proposed approach through a practical example involving the development of a hydraulic and pneumatic automation system.

MATERIALS AND METHODS

The number of different means, for example, control and monitoring in pneumatic automation, is measured in thousands, and in hydraulic automation and mechatronics even more. The diversity of the element base of such models as finite state machines or EF networks, or Petri nets, which are the basis of the language for compiling control algorithms, is measured in tens and differs thousands of times from the diversity of mechatronics technical means (Findeisen & Helduser, 2015; Albrecht & Taylor, 2020; Lu *et al*., 2022).

Secondly, it is the natural asynchrony of processes in pneumatic and hydraulic devices (the time of their state transformation). For these devices, the time difference between the operation periods differs from tens to hundreds of thousands of times. The third is the unregulated time of the same hydraulic, mechanical or pneumatic transformations when they are repeated (Subramanya, 2010; Khond *et al*., 2019; Zhang *et al*., 2024). That is, a single transformation of the state of a hydraulic or pneumatic device, depending on the state of the environment and system parameters, may have a repeated operating time with a difference of several dozen times (Peterson, 1981). The fourth difference relates to the prevention of physical destruction of system elements and is focused on the features of mechanical, pneumatic or hydraulic devices that differ significantly from the representation of elements in information technology (Parr, 2011; Adedigba *et al.*, 2016). In the case of simultaneous issuance of two commands to transform the state of a mechanical device in opposite directions, its physical destruction or uncontrollable actions occur (Polishchuk *et al.*, 2018; Kozlov *et al.*, 2023). Also, peculiarities can be extended by differences in Power from a tenth of watts to tens of kW and oscillations of variables in transition processes.

The listed features are sufficient to define the basic requirements for the components attribute and its representation in tasks of decomposition and composition of hydraulic automation and mechatronic systems (Tiboni, 2023; Nazarova *et al.*, 2024; López *et al.*, 2026). The universal form of a system element should be suitable for the variety of technical means used in the system. This does not mean that each means is an element or component; conversely, each element has a certain technical device as its prototype (Mesarovic & Takahara, 1975; Subramanya, 2010; Cherkashenko, 2023). In a formal representation, a system element must have the properties of controllability and self-control of transformations of its state, regardless of the physical nature of the structure and complexity of the technical devices used (Fig. 1).

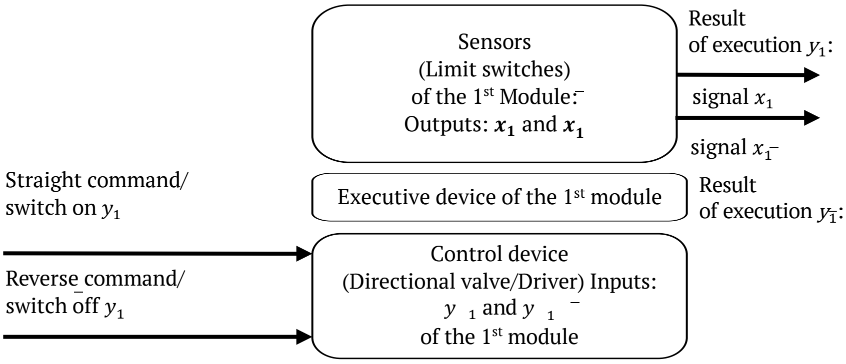


Figure 1. Example of the 1st module structure

Source: compiled by authors

In other words, an element of the system, in terms of information, includes a command to change its state and a signal that confirms the expected change:

$$X_i = \phi(Y_i, t_i), \tag{1}$$

where X_i – signal of changed state, $\phi(Y_i)$ – transition function of changing state by command, t_i – time of transition. This option takes into account the first feature and largely covers the second and third features of the class of systems under consideration. As for the fourth feature, all components have a long-term unregulated transient state, which requires the use of three-digit logic (0 – ready to operate, * – transient state or uncertainty, 1 – testing is completed): $X_i \in \{1, 0, *\}$. Thus, the form of a system element will have meaningful content (based on a physical prototype) beyond its formal description. The physical content of the system element (involved devices) will not affect the system properties of the element. Only the function of changing the state and the function of self-monitoring the state of the element are taken into account, not time of transition.

Prevention of mechanical destruction, as a result of applying opposite commands to an element, module or system, can be ensured by taking into account a cycling feature (Lishchenko *et al.*, 2024; Fu *et al.*, 2026). Every element has two opposite actions ("switch on + action" and "switch off + action") – execution and control of result:

$$\begin{cases} X_i = \phi(Y_i, t_i) \\ X_{\bar{i}} = \phi(Y_i, t_i)' \end{cases} \tag{2}$$

where $X_{\bar{i}}$ – signal of returning to initial state, result of reverse command $Y_{\bar{i}}$, $t_{\bar{i}}$ – time of transition. Two opposite functions (main and reverse) of one element form a sequential cycle (Fig. 2).

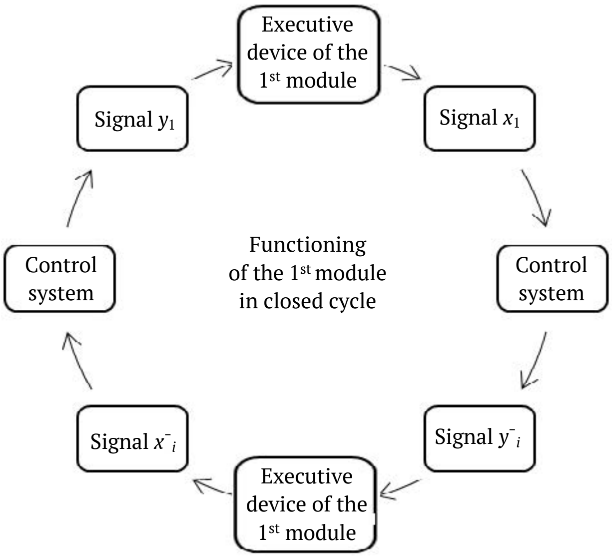


Figure 2. Example of function diagram of the 1st module

Source: compiled by authors

In this form, any separate part of the system, if it meets the listed requirements, can obtain the status of a system element with standard functions for the developer such as connection to the system using the main and reverse control commands and interaction with other elements via state control signals.

A separate system that has "switch-on" functions with control of the result obtained after switching on, and "switch-off" functions also with control of the result obtained after switching off, can be used as a system element and, therefore, is suitable for the synthesis of an automated system of a technical object (Fig. 3). The main and reverse functions performed by such a system element or macro module form their technological cycle, divided into smaller functions and modules. But for the main system, such a macro module is an ordinary element with a three-digit cyclic logic of state changes (1, 0, *) based on the issuance of "switch on" and "switch off" commands:

$$\begin{aligned} MM_{Mi} &= \{Y_{Mi}, X_{Mi}, t_{Mi}, Y_{M\bar{i}}, X_{M\bar{i}}, t_{M\bar{i}}, \} = \\ &= \{\{Y_j, X_j, t_j\}_{YMi}, \{Y_j, X_j, t_j\}_{YM\bar{i}}\}\Big|_{j=1}^{j=k}, \end{aligned} \tag{3}$$

where *i* – serial number of macro module, *j* – serial number of unit/module of macro module, *k* – quantity of units/modules of macro model.

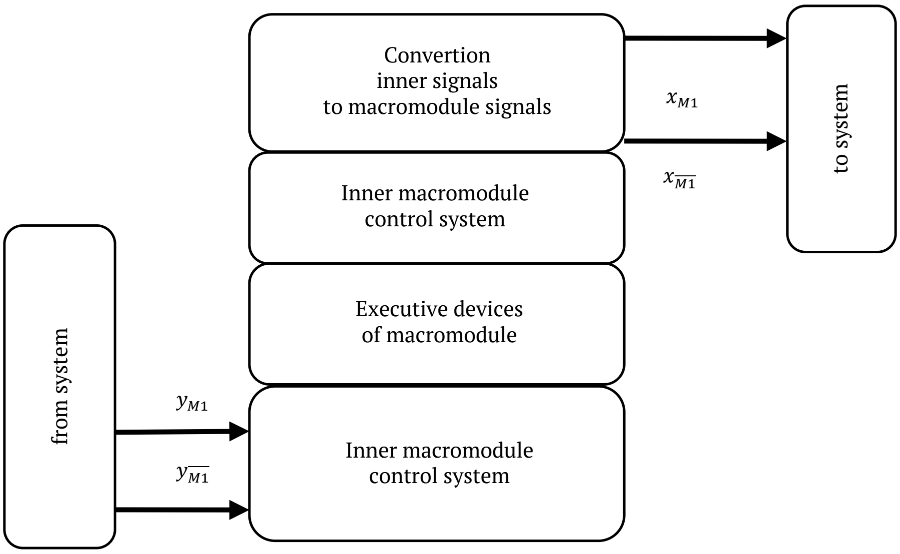


Figure 3. Structure diagram of macro module *M*1

Source: compiled by authors

The issue is rules for adding macro module in to the system and for separating part of the system as macro module. The function of the macro module for the main command Y_{Mi} includes several actions of system units *k*(*Mi*). This means the macro module is a system of units/modules, which include actuators with control signals and commands. There are no differences in using of inputs $\{Y_{Mi}, Y_{M\bar{i}}\}$ and outputs $\{X_{Mi}, X_{M\bar{i}}\}$ of a macro module and unit. But they can be different, in the physical meaning, of the main and reverse action of the macro module. It means inputs are not connected with one specific unit and outputs – give information about state of the macro module. Output signal is a combination of control signals of units in the macromodule. An additional advantage of using macro modules is the removal of complex functions from the main system and their transfer to macro modules. The functions may be more complex than the main system, but their provisioning, testing, and debugging can be performed simultaneously with the development of the entire facility. This significantly speeds up the development of a new system.

Application of the method on the example of a system with pneumatic actuators Example. The technical solution is considered using pneumatic automation. Pneumatic grippers are widely used in assembly plant manipulators to perform several actions. The gripper combines several actuators with monitoring and control devices. A pneumatic workpiece holding mechanism, horizontal and vertical movement drives (Fig. 4). Before starting work, the workpiece holding mechanism is in the open position. At the command to perform the main function of the macro module: the vertical feed drive lowers the holding mechanism to the workpiece (position control), then – if the workpiece is present – the mechanism grabs the workpiece (pressure control X''_{2M1}, signal from detector of workpiece X_{2M1}'), Next, the vertical feed drive lifts the workpiece (position control). If necessary, the horizontal feed drive moves the workpiece to the shipping position. On a command to perform the reverse function of the macro module: the vertical feed drive lowers the mechanism with the held part (position control), then the mechanism releases the part (inverse pressure control $X_{\overline{2M1}}$, no signal from detector of workpiece $\overline{X}'_{2M1}$). The vertical feed drive then lifts the mechanism without the part (position control). The macro module returns to its original state.

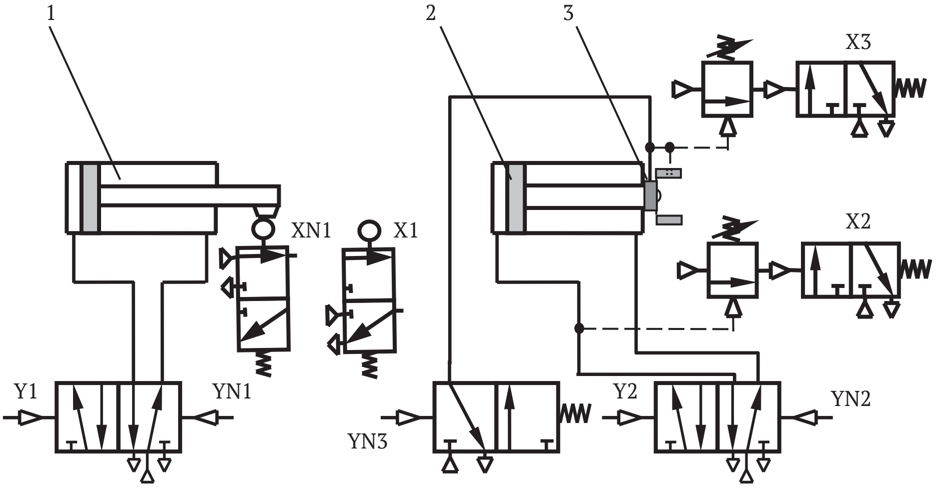


Figure 4. Macro module "Manipulator" (pneumatics)

Note: actuators: 1 – pneumatic cylinder (horizontal axis), 2 – pneumatic cylinder (vertical axis), 3 – gripper; signals: Y1, YN1– commands for horizontal displacement, X1, XN1 – end position switches for the horizontal axis, Y2, YN2 – commands for vertical displacement, X2 – end position of the vertical axis (pressure sequence valve), YN3 – gripper (command "release"), X3 – signal from gripper

Source: compiled by authors

The first step is to identify the macro module and the modules that make up the macro module, and to name the macro module, as well as the modules and signals that make up the macro module. Consider the name of the macro module. Suppose that the grab macro module has the name "M1" in the system, then the commands to turn it on and off will receive the designation: Y_{M1}, $Y_{\bar{M}i}$. Macromodule components: 1M1 vertical feed drive (1st module of Macromodule 1), 2M1 holding mechanism module (2nd module of Macromodule 1). Command marked as *Y* with index, control signals marked as *X* with the same index. The second step is to build detailed descriptions of the main and return functions of the macro module. The execution of the main function "part capture" will have the following detailed form:

$$Y_{M1} = Y_{1M1} \to Y_{2M1} \to Y_{\overline{1M1}}. \tag{4}$$

The opposite function of "part capture" will look like this:

$$Y_{\overline{M1}} = Y_{1M1} \to Y_{\overline{2M1}} \to Y_{\overline{1M1}}. \tag{5}$$

The third step is the formation of the content of the logical conditions for controlling the performance of macro module functions. For the example under consideration, t he logical condition that states the fulfilment of the main function of the macro module "approval" is as follows:

$$X_{M1} = X_{\overline{1M1}} * X_{2M1}, \tag{6}$$

(top position, workpiece is held). The logical condition that states the execution of the reverse function of the macro module is as follows:

$$X_{\overline{M1}} = X_{\overline{1M1}} * X_{\overline{2M1}}, \tag{7}$$

(top position, no workpiece in gripper). In general, logical conditions are drawn up by means of logical synthesis, which will be discussed in the example.

The fourth step is to formalise the description of the macro module. By its actions and their ordering, the macro module is a cyclic system. The difference lies in the division of the cycle into two segments. The first segment contains the part of the cycle that corresponds to the main function. To execute it, you need an enable signal from the main command of the macro module. Similarly, the second segment contains the rest of the cycle, which is responsible for executing the return function of the macro module. To execute it, you need the enable signal of the return command of the macro module. That is, for a macro module, there is a functional graph built according to the rules of cyclic systems and the methods of the cyclic-modular approach (Peterson, 1981; Wu *et al.*, 2014; Vanegas-Ayala *et al.*, 2022) can be applied to it. The macromodule graph has two externally controlled gaps such as the "RUN" signal. They separate the segments of the main and return functions (Fig. 5). If necessary, the system of the macro module can be supplemented with memory elements, and logical expressions of control commands for individual modules that make up its structure can be compiled (Wu *et al.*, 2014; Zhao *et al.*, 2021).

Fifth step is establishing a logical connection between the macro module and the external system. According to the diagram (Fig. 3), the macro module should supply the system with signals to control the performance of its main and reverse functions, the logical expressions of which were built in the third step. X_{M1} and $X_{\overline{M1}}$ will be used in control commands for the logical synthesis of the external system to which the macro module belongs. Two commands that will be formed during the logical synthesis for the macro module Y_{M1}, $Y_{\overline{M1}}$, will be used as external signals when performing the main and return functions of the macro module. Logical commands for controlling the macro module Y_{M1}, $Y_{\overline{M1}}$, will be multipliers in the expressions of commands that start the execution of the main function segment and in the expressions of commands that start the execution of the inverse function segment.

RESULTS AND DISCUSSION

The system acquires a homogeneous structure. This is because the macromodule takes the form of a regular module. That is, a complex subsystem is represented as a memory element with turn-on and turn-off time delays. This allows for synthesising the interaction logic of all system modules using a single set of rules. Consequently, this enables the investigation of such system properties as reachability, safeness, liveness, and boundedness, which are commonly applied in Petri nets. Formally, the macro module does not exist in the system. As an example, the gripper macromodule (Fig. 5) is involved in an automated system that provides the following actions:

1. Loading the part into the system – the main action of module 1;
2. Picking up the part by the macromodule – the main action of the macromodule *M*2;
3. Transferring the gripper with the part to the working position – the main action of module 3;
4. Returning the loader to its initial state – the reverse action of module 1;
5. Release of the part by the macro module – the reverse action of the macro module *M*2;
6. Clamping the part – the main action of module 4;
7. Performing technological processing of the part – the main action of module 5;
8. Returning the technological equipment to its original state – the reverse action of module 5;
9. Releasing the part – the reverse action of module 4;
10. Picking up the processed part by the macromodule – the main action of the macromodule *M*2;
11. Returning the gripper with the workpiece to its original position – the reverse action of module 3;
12. Release of the workpiece by the macromodule – the reverse action of the M2 macromodule;
13. Removal of the part by the extractor – the main action of module 6;
14. Returning the extractor to its original position – reverse action of module 6. Then the cycle is repeated.

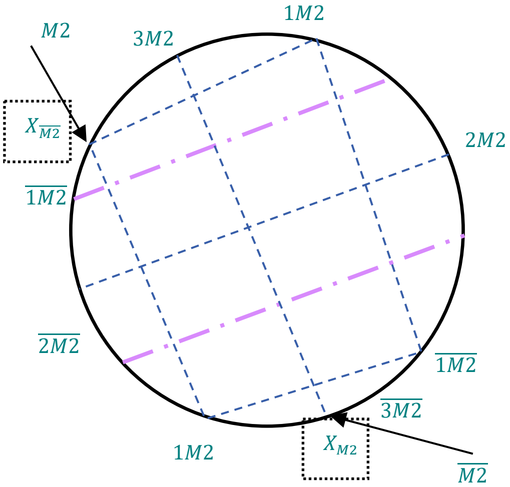


Figure 5. Functional diagram of macro module *M*2: dotted – connection lines, dash-dotted – uncertainty lines, 3*M*2 – memory element

Source: compiled by authors

The first step is to identify the macro module and its components. The grip macro module is named "*M*2" in the system, and the commands will be designated: main Y_{M2}, reverse . Components of the macro module: vertical feed drive 1*M*2 (1st module of Macromodule 2), module of the holding mechanism 2*M*2 (2nd module of Macromodule 2). Second step is detailed descriptions of the main and reverse functions of the macro module. Performing the main function of capturing a part: $Y_{M2} = Y_{1M2} \rightarrow Y_{2M2} \rightarrow Y_{\overline{1M2}}$. Performs a reverse "part release" function: $Y_{\overline{M2}} = Y_{1M2} \rightarrow Y_{\overline{2M2}} \rightarrow Y_{\overline{1M2}}$. Third step is the formation of logical conditions for controlling the execution of macro module functions. The logical condition for capturing a part: $X_{M2} = X_{\overline{1M2}} * X_{2M2}$ (top position, workpiece is held). A logical condition for releasing a workpiece: $X_{\overline{M2}} = X_{\overline{1M2}} * X_{\overline{2M2}}$ (upper position, no workpiece). The fourth step is formalisation and logical synthesis of the macro module. By the actions of the macro module components and their ordering

$$Y_{M2} = Y_{1M2} \rightarrow Y_{2M2} \rightarrow Y_{\overline{1M2}} \text{ and } Y_{\overline{M2}} = Y_{1M2} \rightarrow Y_{\overline{2M2}} \rightarrow Y_{\overline{1M2}}$$

a functional graph of the macromodule was constructed (Fig. 5). According to the results of the memory test, two uncertainty lines (dashed and dotted lines) were obtained.

To get rid of the memory deficit, a module (memory element, $3M2$ and $\overline{3M2}$). Specify external commands to the main function ($M2$) and reverse function ($\overline{M2}$) of the macro module (marked with arrows on Fig. 5). Based on the created graph, the module control commands and logical expressions of the macro module status signals were compiled. Control commands:

$$Y_{1M2} = X_{3M2} * X_{\overline{2M2}} + X_{\overline{3M2}} * X_{2M2} * Y_{\overline{M2}};$$
$$Y_{2M2} = X_{1M2} * X_{3M2}; Y_{3M2} = X_{\overline{1M2}} * X_{\overline{2M2}} * Y_{M2};$$
$$Y_{\overline{1M2}} = X_{2M2} * X_{3M2} + X_{\overline{2M2}} * X_{\overline{3M2}}; Y_{\overline{2M2}} = X_{1M2} * X_{\overline{3M2}};$$
$$Y_{\overline{3M2}} = X_{\overline{1M2}} * X_{2M2}.$$

The main command of the third module contains an external signal to turn on the macro module, and the second addition of the 1st module command contains an external command signal to turn off the macro module. The expressions of these commands (Y_{M2} and $Y_{\overline{M2}}$) should be built by logical synthesis of the main system. Composing logical expressions of the macro module state (X_{M2} and $X_{\overline{M2}}$, shown with the frames, Fig. 5) corresponds to the conditions of termination of the main function and the inverse function of the macro module. On the graph, the vertex of the end of the last action of the main function has the logical expression $X_{M2} = X_{\overline{3M2}} * X_{\overline{1M2}} * X_{2M2}$. As soon as the reverse function starts, the signal will disappear (state $X_{\overline{1M2}}$) and the macro module will lose its state X_{M2}. Similarly, for the vertex of the end of the last action of the inverse function: $Y_{M2} = X_1 + X_{\overline{4}} * X_{\overline{7}} * X_3$. Compared to the meaningful conditions of the macro module state, multipliers have been added to take into account the addition of a memory element.

Fifth step is ensuring the interaction between the macro module and the external system. The logical connection between the macromodule and the external system is provided by the expressions of the commands to turn the macromodule on and off (Y_{M2} and $Y_{\overline{M2}}$). These expressions are built through the state signals of the external system. The logical connection between the external system and the macro module is provided by the expressions of the macro module status signals (X_{M2} and $X_{\overline{M2}}$), built through the state signals of the macro module components. Based on the results of the logical synthesis of the external system, the logical expressions of all commands, including those of the macro module (Fig. 6), are obtained.

$$Y_1 = X_{\overline{6}} * X_7 * X_{\overline{3}} * X_{\overline{M2}}; Y_{\overline{1}} = X_{M2};$$
$$Y_{M2} = X_1 + X_{\overline{4}} * X_{\overline{7}} * X_3; Y_{\overline{M2}} = X_3 * X_{\overline{1}} * X_7 + X_{\overline{3}} * X_{\overline{7}};$$
$$Y_3 = X_{M2} * X_7; Y_{\overline{3}} = X_{M2} * X_{\overline{7}}; Y_4 = X_{\overline{M2}} * X_7 * X_3;$$
$$Y_{\overline{4}} = X_{\overline{5}} * X_{\overline{7}}; Y_5 = X_4 * X_7; Y_{\overline{5}} = X_{\overline{7}};$$
$$Y_6 = X_{\overline{M2}} * X_{\overline{7}} * X_{\overline{3}}; Y_{\overline{6}} = X_7; Y_7 = X_6; Y_{\overline{7}} = X_5 \quad .$$

Macro module state signals X_{M2} and $X_{\overline{M2}}$ should be replaced with their expressions obtained in step 3 or during the logical synthesis of the macro module: $X_{M2} = X_{\overline{3M2}} * X_{\overline{1M2}} * X_{2M2}$ and $Y_{M2} = X_1 + X_{\overline{4}} * X_{\overline{7}} * X_3$;.

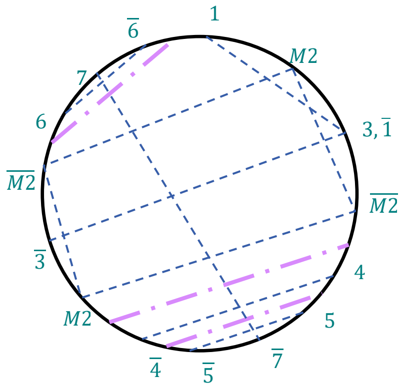


Figure 6. Functional diagram of system with the 2nd Macro Module: dashed – connection lines; dotdash – uncertainty; 7 and $\overline{7}$ – memory

Source: compiled by authors

The command expressions of the external system after replacement are as follows:

$$Y_1 = X_{\overline{6}} * X_7 * X_{\overline{3}} * X_{\overline{1M2}} * X_{\overline{2M2}} * X_{\overline{3M2}},$$
$$Y_{\overline{1}} = X_{\overline{3M2}} * X_{\overline{1M2}} * X_{2M2},$$
$$Y_3 = X_{\overline{3M2}} * X_{\overline{1M2}} * X_{2M2} * X_7,$$
$$Y_{\overline{3}} = X_{\overline{3M2}} * X_{\overline{1M2}} * X_{2M2} * X_{\overline{7}},$$
$$Y_4 = X_{\overline{1M2}} * X_{\overline{2M2}} * X_{\overline{3M2}} * X_7 * X_3,$$
$$Y_6 = X_{\overline{1M2}} * X_{\overline{2M2}} * X_{\overline{3M2}} * X_{\overline{7}} * X_{\overline{3}}.$$

This ensures that the external system is connected to the state of the macro module components.

The logical connection of the macro module with the external system is made through the macro module control commands from the external system. The signals of these commands are present as multipliers in the expressions of the control commands of the macro module components. Instead Y_{M2} and $Y_{\overline{M2}}$ in the expressions of the commands of the macro module components must be:

$$Y_{M2} = X_1 + X_{\overline{4}} * X_{\overline{7}} * X_3 ;$$
$$Y_{\overline{M2}} = X_3 * X_{\overline{1}} * X_7 + X_{\overline{3}} * X_{\overline{7}}.$$

This replacement ensures that the macro module switches from its own event counting to the event counting of the external system. The command expressions for the components of the macro module are determined as follows:

$$Y_{1M2} = X_{3M2} * X_{\overline{2M2}} +$$
$$+X_{\overline{3M2}} * X_{2M2} * (X_3 * X_{\overline{1}} * X_7 + X_{\overline{3}} * X_{\overline{7}});$$
$$Y_{2M2} = X_{1M2} * X_{3M2};$$
$$Y_{3M2} = X_{\overline{1M2}} * X_{\overline{2M2}} * (X_1 + X_{\overline{4}} * X_{\overline{7}} * X_3);$$
$$Y_{\overline{1M2}} = X_{2M2} * X_{3M2} + X_{\overline{2M2}} * X_{\overline{3M2}};$$
$$Y_{\overline{2M2}} = X_{1M2} * X_{\overline{3M2}}; Y_{\overline{3M2}} = X_{\overline{1M2}} * X_{2M2}.$$

The construction of the “gripper” macromodule for the system and the system with the macromodule is complete. The resulting expressions allow us to develop a control system: to design electrical relay circuits, to design pneumatic automation circuits, to design a control algorithm (Zhao *et al*., 2021; Vanegas-Ayala *et al*., 2022).

The use of a macromodule as a tool by a developer when creating systems provides another opportunity – to use the principle of reversibility of the system and its parts. According to formal features, an external system that includes a macro module that operates once per cycle meets all the requirements for building a separate “macro module”. Main and reverse functions Y_{M1} and $Y_{\overline{M1}}$ of the macro module that has already been built and is named “*M*1”, divides the system’s action cycle into two segments (Busi, 2002; Gomes & Barros, 2005; Reisig, 2013). In comparison, a common feature is the logical structure of the control systems. In other models, the system architecture is fixed. In the proposed model, within its foundational premises, the system architecture is open. That is, cyclic modules have specific logical interaction conditions among themselves. The operational process is a consequence of fulfilling (executing) these conditions.

The first segment between switching on the macromodule *M*1 and switching off the macromodule *M*1. The second segment from the switching off of macromodule *M*1 to the next switching on of macromodule *M*1 (Fig. 7). Since the system’s development of the first process segment leads to the state of the beginning of the second segment, and the development of the second segment leads to the state of the beginning of the first segment, it can be said that the segments are mutually reversible. That is, by analogy with the cyclic modules of the system, these segments can be designated as follows: the main action of the external system Y_{Syst}, reverse action of the external system $Y_{\overline{Syst}}$. According to the graph (Fig. 7), to start the main action Y_{Syst} the reverse function of the macro module must be completed $Y_{\overline{M1}}$. To start performing the reverse action $Y_{\overline{Syst}}$ the main function of the macro module must be completed Y_{M1}. That is, logical commands are enough for the macro module and the external system to interact:

$$Y_{M1} = X_{\overline{Syst}}; Y_{\overline{M1}} = X_{Syst};$$
$$Y_{Syst} = X_{M1}; Y_{\overline{Syst}} = X_{\overline{M1}}.$$

At the same time, the system state signals X_{Syst} and $X_{\overline{Syst}}$ will have a structure similar to the status signals of the of the macro module, i.e., they are logical expressions for a single vertex of the expanded graph of the system. will have a structure similar to the status signals of the of the macro module, i.e., they are logical expressions for a single vertex of the expanded graph of the system.

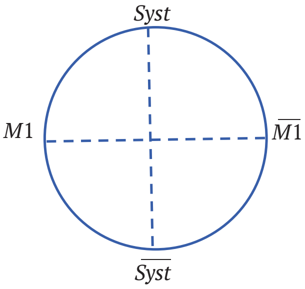
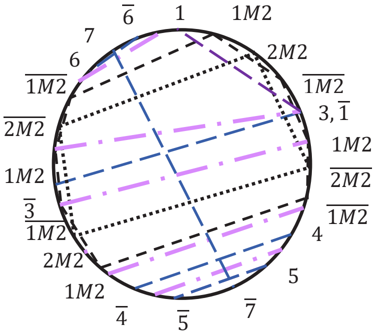


Figure 7. Functional graph with the inverse image of the system and macromodule *M*1
Note: dotted lines are communication lines, *Syst* – main action of the system, $\overline{Syst}$ – reverse action of the system
Source: compiled by authors

The number of actuating and controlling devices and monitoring devices is the same for the homogeneous system and the system with macromodules. This number is due to the same set of functions and control devices for both system variants. Quantitative indicators for the homogeneous system (Fig 9) and the system after decomposition have approximately the same values (Table 1).



$$1M2 \quad 3M2 \quad \overline{2M2} \quad \overline{3M2} \quad 2M2$$
$$* (X_3 * X_{\overline{1}} * X_7 + X_{\overline{3}} * X_{\overline{7}});$$
$$Y^*{}_{1M2} = X_1 * X_{2M2} + X_3 * X_{\overline{1}} * X_{2M2} * X_7 +$$
$$+X_{\overline{4}} * X_{\overline{7}} * X_{\overline{2M2}} * X_3 + X_{\overline{3}} * X_{\overline{7}} * X_{2M2}$$
$$Y_{2M2} = X_{1M2} * X_{3M2};$$
$$Y^*{}_{2M2} = X_{1M2} * X_1 + X_{1M2} * X_{\overline{7}} * X_3$$
$$Y_1 = X_{\overline{6}} * X_7 * X_{\overline{3}} * X_{\overline{1M2}} * X_{\overline{2M2}} * X_{\overline{3M2}},$$
$$Y^*{}_1 = X_{\overline{6}} * X_7 * X_{\overline{2M2}} * X_{\overline{3}}$$
$$Y_{\overline{1}} = X_{\overline{3M2}} * X_{\overline{1M2}} * X_{2M2},$$
$$Y^*{}_{\overline{1}} =* X_{\overline{1M2}} * X_{2M2} * X_7.$$

Figure 9. Functional graph of the homogeneous system of the first example, comparison of logical expressions of commands
Note: * commands are marked in the case of a homogeneous system, without marking – command expressions for a system with a macromodule *M*2
Source: compiled by authors

Table 1. Comparative indicators of the example

Indicator	Homogeneous	2 macro modules
Quantity of transitions	23	15 + 9 = 24
Logical uncertainty	6	3 + 2 = 5
Elements of memory	1	1 + 1 = 2
Quantity of modules	8	7 + 3 = 10
Quantity of logical connections (modules 1*M*1, 2*M*2 and 1)	10 + 4 + 5 = 19	8 + 1 + 7 = 16
Relative dimensionality	23 * 8 = 184	15 * 7 + 9 * 3 = 132

Source: compiled by authors

The main difference in the approach is related to the relative dimensionality of the system $\tilde{N}$, which is calculated as the product of the number of transitions in the process z and number of modules in the system m: $N = z * m$. In terms of content, this is the maximum number of logical conditions that determine a certain action of a certain module in a certain state of the system, applied for each action.

These conditions are the formal basis for the content of the system's algorithm. For the above example of a homogeneous system, this figure is 184. For a distributed system, the developer must synthesise the main system and the macro module subsystem. For the main system, the relative dimensionality is 105, and for the macromodule 27. For the developer, this means that the required number of synthesised and minimised logical conditions that he must check is reduced by 1.4 times.

$$\tilde{t}_{sin\,t}\frac{t^{sin\,t}_{cont}}{t^{sin\,t}_{mod\,ul}} = \tilde{t}_{test}\frac{t^{test}_{cont}}{t^{test}_{mod\,ul}} = 1.4.$$

This will result in a proportional reduction in development time, a reduction in testing time, a proportional reduction in the number of technical errors in the development, testing, and design of the control algorithm, and the time required to correct them. The total reduction in development time using the macromodular approach will be:

$$\tilde{t} = {}^{t_{cont}}/_{t_{mod\,ul}} = \tilde{t}_{sin\,t}\frac{t^{sin\,t}_{cont}}{t^{sin\,t}_{mod\,ul}} + \tilde{t}_{test}\frac{t^{test}_{cont}}{t^{test}_{mod\,ul}} + \tilde{t}_{mistake}\frac{t^{mistake}_{cont}}{t^{mistake}_{mod\,ul}},$$

where $\tilde{t}_{sin}\ t$, $\tilde{t}_{test}$, $\tilde{t}_{mistake}$ – percentage of development time spent on synthesis $\tilde{t}_{sin\,t}$ (5 – 15%), testing $\tilde{t}_{test}$ (60-90%) and fixing errors $\tilde{t}_{mistake} = 100\% - (\tilde{t}_{sin}\,t + \tilde{t}_{test})$ (up to 30%). The time to correct errors is further reduced by reducing the dimensionality of the objects (system or macro module) in which the correction is performed. For the example under consideration, the parameter falls within the range from 1.4 to 2.1. The second value corresponds to the absence of errors (qualified developer), and the first value corresponds to the presence of typical technical errors in the construction of the circuit and the design of the control algorithm.

The effectiveness of the approach depends on the relative dimensionality of both the system and the macromodules: the number of modules in the system m, number of transitions in the process z, quantity of macro modules K_{mod}, uniformity of distribution of functional modules across macromodules. Clarification: the total number of transitions when adding each macro module can increase by 1, due to the addition of memory elements that compensate for the absence of a macro module in the system: $z_{mod\,u\,l} = z_{sist} + K_{mod}$. For a system consisting of three macromodules, each of which has the conditional dimension of the entire system of the previous example, the following indicators are obtained (Table 2).

Table 2. Comparative indicators of the 2nd example

Quantity of transitions	Homogeneous	3 macro modules
Logical uncertainty	69	23 + 23 + 23 + 3 = 72
Elements of memory	18	$nD_1 + nD_2 + nD_3 \leqslant 18$
Quantity of modules	nM	nM + 3
Quantity of logical connections	24	24 + 3 = 27
Relative dimensionality	69 * 24 = 1656	3 * (24 * 8) = 576

Source: compiled by authors

For the developer, this means that the number of synthesised and minimised logical conditions is reduced by 2.8 times. Accordingly, the reduction in the development time, depending on the ratio of the stages' terms, will range from 2.15 (in the absence of errors) to 7.5. The division of the same system into 2 macromodules of larger dimension (12 functional modules each) leads to the following indicators (Table 3).

Table 3. Comparative indicators of the 3rd example

Quantity of transitions	Homogeneous	3 macro modules
Quantity of transitions	69	35 + 35 + 2 = 72
Logical uncertainty	18	$nD_1 + nD_2 \leqslant 18$

Continued Table 3.

Quantity of transitions	Homogeneous	3 macro modules
Elements of memory	nM	nM + 2
Quantity of modules	24	24 + 2 = 26
Quantity of logical connections (modules 1*M*1, 2*M*2 and 1)	69*24=1656	2*(35*12)=840

Source: compiled by authors

Separating one macromodule in the system (decomposition into 2 components: 2 to 5) reduced the relative dimensionality by 1.4 times. Dividing the system into 2 macromodules of the same dimension reduced the relative dimension by 2 times. Dividing the system into 3 macromodules of the same dimensionality reduced the relative dimensionality by 2.8 times. Further fragmentation of the system will lead to simplification of macromodules and complication of connections between them. In fact, a second layer of the control system is created – a coordinator of macromodules. In the ultimate variant, an initial system will be obtained that will serve as a coordinator of the initial functional modules identified with macromodules. These data provide a quantitative characterisation of the simplification of an abstract system when it is decomposed into macromodules. Since mechatronic systems are subject to the properties of equipment and technological process, an arbitrary division of the system into a given number of macromodules and the desired dimension of macromodules is not realistic. The separation of macromodules, as a rule, has a substantive basis, which determines the number of macromodules and their dimensions, as was the case in the example with the manipulator, and the qualifications of the developer. In other words, the decomposition efficiency indicator will be the reduction of development time and the ability to automate larger-scale processes. This indicator will be formed for each practical system based on its output data and the equipment involved.

CONCLUSIONS

The cyclic-modular methodology for decomposing systems into macromodules provides a structured and efficient framework for the development of hydropneumatic automation and mechatronic systems. By introducing macromodules, the influence of scale factors on development time is reduced, and conditional dimensionality is minimised through the elimination of unnecessary logical connections between system components. This simplification accelerates design processes and enhances the transparency and manageability of complex architectures.

A prerequisite for effective application is the ability to isolate distinct subsystems. Such isolation is feasible when sequences of actions involve equipment not shared with other functions, or when complex functions exist independently of the subsequent technological cycle. Under these conditions, macromodules can be treated as autonomous units, facilitating both circuit design and algorithmic control. Quantitative analysis confirms the efficiency of this approach: the use of N macromodules in a system of n modules can reduce development time by approximately N times, provided that the system contains more than 3N modules and each macromodule incorporates at least three modules. This ensures meaningful decomposition and maximises the benefits of modularisation.

Future development should focus on the integration of simulation tools to validate system behaviour, reliability, and fault tolerance prior to physical implementation. Embedding macromodule decomposition into simulation environments will allow predictive testing and optimisation of performance. Additionally, the methodology can be aligned with Industry 4.0 principles, enabling flexible, reconfigurable production systems that leverage modular control units within smart manufacturing frameworks. In conclusion, the cyclic-modular methodology for macromodule decomposition offers significant advantages in reducing complexity, shortening development cycles, and promoting scalability. Its extension through simulation-based validation and integration with Industry 4.0 paradigms positions it as a universal design strategy capable of transforming system development into a more efficient, modular, and future-ready process.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Aboul, E.H., Jyotir, M.C., & Vishal, J. (2022). *Artificial intelligence and industry 4.0.* Amsterdam: Elsevier Science.

[2] Adedigba, S.A., Khan, F., & Yang, M. (2016). Dynamic safety analysis of process systems using nonlinear and non-sequential accident model. *Chemical Engineering Research and Design*, 111, 169-183. doi: 10.1016/j.cherd.2016.04.013.

[3] Albrecht, O., & Taylor, C.J. (2020). Unknown and time-varying time delays in the modelling and control of hydraulic actuators. In *2020 Australian and New Zealand control conference (ANZCC)* (pp. 232-237). Gold Coast: Institute of Electrical and Electronics Engineers (IEEE). doi: 10.1109/ANZCC50923.2020.9318375.

[4] Barbanera, F., Dezani-Ciancaglini, M., Lanese, I., & Tuosto, E. (2021). Composition and decomposition of multiparty sessions. *Journal of Logical and Algebraic Methods in Programming*, 119, article number 100620. doi: 10.1016/j.jlamp.2020.100620.

[5] Becker, M., Karaoglu, S., Makansi, F., Radtke, J., & Schmitz, K. (2025). Industry 4.0: Review of the state of the art in fluid power research and industry. *International Journal of Fluid Power*, 26(4), 573-624. doi: 10.13052/ijfp1439-9776.2643.

[6] Belgacem, H., Abuabiah, M., Boulescu, A., & Chihi, I. (2026). A three-level hierarchical fault diagnosis framework for APS mechatronic systems with adaptive Bayesian root-cause analysis. *Results in Engineering*, 30, article number 111163. doi: 10.1016/j.rineng.2026.111163.

[7] Busi, N. (2002). Analysis issues in Petri nets with inhibitor arcs. *Theoretical Computer Science*, 275(1-2), 127-177. doi: 10.1016/S0304-3975(01)00127-X.

[8] Cherkashenko, M. (2023). Synthesis of discrete drives control systems. *Bulletin of the National Technical University "KhPI". Series: "Hydraulic machines and hydraulic units"*, 1, 12-17. doi: 10.20998/2411-3441.2023.1.02.

[9] Findeisen, D., & Helduser, S. (2015). *Oil hydraulics. Handbook of hydraulic drives and controls*. Berlin: Springer Vieweg. doi: 10.1007/978-3-642-54909-0.

[10] Fu, Z., Zhang, H., Zhu, S., Jiang, Z., & Zhang, L. (2026). A diagnosis and classification method for energy efficiency anomalies in mechatronic equipment based on multi-condition inherent energy efficiency benchmark. *Energy*, 342, article number 139671. doi: 10.1016/j.energy.2025.139671.

[11] Gomes, L., & Barros, J.P. (2005). Structuring and composability issues in Petri net modeling. *IEEE Transactions on Industrial Informatics*, 1(2), 112-123.

[12] Khond, V.V., Kavale, P.K., & Dixit, N.S. (2019). An approach to mechatronics system design. *International Journal of Engineering Development and Research*, 7(2), 360-366.

[13] Kozlov, L., Poliakov, A., Yakobinchuk, O., Gubarev, O., & Makarova, T. (2023). Mechatronic hydraulic system with adaptive regulator for a manipulator of the mobile working machine. In V. Ivanov, I. Pavlenko, O. Liaposhchenko, J. Machado & M. Edl (Eds.) *Advances in design, simulation and manufacturing VI. DSMIE 2023. Lecture notes in mechanical engineering* (pp. 64-73). High Tatras: Sumy State University. doi: 10.1007/978-3-031-32774-2_7.

[14] Lambert, P., & Herder, J.L. (2016). Parallel robots with configurable platforms: Fundamental aspects of a new class of robotic architectures. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(3), 463-472. doi: 10.1177/09544062156025.

[15] Li, J., Lu, Y., He, Y., Zhou, G., & Miao, L. (2022). Analysis and compensation control of engine valve response delay based on the electro-hydraulic variable valve actuator. *Machines*, 10(8), article number 701. doi: 10.3390/machines10080701.

[16] Li, J., Sun, Y., Wang, Y., & Su, Q. (2025). Identifying critical nodes in cyber-physical power systems based on an improved mixed degree decomposition method. *Advanced Engineering Informatics*, 68(Part B), article number 103630. doi: 10.1016/j.aei.2025.103630.

[17] Lishchenko, N., & Gushchin, A., & Larshin, V. (2024). Hierarchical control in mechatronic technological systems. *Machines*, 12(10), article number 697. doi: 10.3390/machines12100697.

[18] López, E.J., Ruiz, J.E.P., Chávez, O.L., Muñoz, F., Velásquez, L.A.G., & Lugo, J.G.C. (2026). The evolution of mechatronics engineering and its relationship with industry 3.0, 4.0, and 5.0. *Technologies*, 14(2), article number 81. doi: 10.3390/technologies14020081.

[19] Mesarovic, M.D., & Takahara, Y. (1975). *General systems theory: Mathematical foundations*. London: Academic Press.

[20] Nazarova, O., Osadchyy, V., Hutsol, T., Glowacki, S., Nurek, T., Hulevskyi, V., & Horetska I. (2024). Mechatronic automatic control system of electropneumatic manipulator. *Scientific Reports*, 14, article number 6970. doi: 10.1038/s41598-024-56672-4.

[21] Parr, A. (2011). *Hydraulics and pneumatics: A technician's and enginee's guide*. Oxford: Butterworth-Heinemann.

[22] Peterson, J.L. (1981). *Petri net theory and the modeling of systems*. Englewood Cliffs: Prentice-Hall.

[23] Polishchuk, L.K., Kozlov, L.G., Piontkevych, O.V., Gromaszek, K., & Mussabekova, A. (2018). Study of the dynamic stability of the conveyor belt adaptive drive. In *Proceedings volume 10808 photonics applications in astronomy, communications, industry, and high-energy physics experiments 2018*. Wilga: Warsaw University of Technology. doi: 10.1117/12.2501535.

[24] Reisig, W. (2013). *Understanding Petri nets. Modeling techniques, analysis methods, case studies*. Berlin Heidelberg: Springer. doi: 10.1007/978-3-642-33278-4.

[25] Sikandar, H., Vaicondam, Y., Khan, N., Qureshi, M.I., & Ullah, A. (2021). Scientific mapping of industry 4.0 research: A bibliometric analysis. *International Journal of Interactive Mobile Technologies (iJIM)*, 15(18), 129-147. doi: 10.3991/ijim.v15i18.25535.

[26] Subramanya, K. (2010). *Fluid mechanics and hydraulic machines: Problems and solutions*. New Delhi: McGraw Hill Education.

[27] Tiboni, M. (2023). Power drive architectures for industrial hydraulic axes: Energy-efficiency-based comparative analysis. *Applied Sciences*, 13(18), article number 10066. doi: 10.3390/app131810066.

[28] Vanegas-Ayala, S.-C., Barón-Velandia, J., & Leal-Lara, D.-D. (2022). A systematic review of greenhouse humidity prediction and control models using fuzzy inference systems. *Advances in Human-Computer Interaction*, article number 8483003. doi: 10.1155/2022/8483003.

[29] Vieira, M., *et al*. (2021). Towards an integrated decision-support framework for the new generation of manufacturing systems. In S. Relvas., J.P. Almeida, J.F. Oliveira & A.A. Pinto (Eds.) *Springer Proceedings in Mathematics & Statistics* (Vol. 374). Cham: Springer. doi: 10.1007/978-3-030-85476-8_14.

[30] Webert, H., Döß, T., Kaupp, L., & Simons, S. (2022). Fault handling in Industry 4.0: Definition, process and applications. *Sensors*, 22(6), article number 2205. doi: 10.3390/s22062205.

[31] Wu, P., Lai, Z., Wu, D., & Wang, L. (2014). Optimization research of parallel pump system for improving energy efficiency. *Journal of Water Resources Planning and Management*, 141(8). doi: 10.1061/(ASCE)WR.1943-5452.0000493.

[32] Zhang, Y., Xie, Z., Yue, Y., & Qi, L. (2024). Automatic refactoring approach for asynchronous mechanisms with completable future. *Applied Sciences*, 14(19), article number 8866. doi: 10.3390/app14198866.

[33] Zhao, S., Chen, K., Zhang, X., Zhao, Y., Jing, G., Yin, C., & Xiao, X. (2021). A high-order load model and the control algorithm for an aerospace electro-hydraulic actuator. *Actuators*, 10(3), article number 53. doi: 10.3390/act10030053.

Оксана Ганпанцурова
Кандидат технічних наук
Bosch Rexroth AG
97816, вул. Цум Айзенгісер, 1, м. Лор-на-Майні, Німеччина
https://orcid.org/0000-0002-4709-0336

Олександр Губарев
Доктор технічних наук, професор
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0000-0002-0924-4103

Костянтин Бєліков
Кандидат технічних наук, доцент
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0000-0002-7393-1848

Альона Муращенко
Кандидат технічних наук, доцент
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0000-0003-1059-5768

Олег Левченко
Кандидат технічних наук
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0000-0002-7620-9009

Синтез та декомпозиція мехатронної системи з гідропневматичними пристроями автоматизації

Анотація. Актуальність дослідження зумовлена зростанням складності мехатронних систем із гідропневматичними пристроями автоматизації та необхідністю підвищення ефективності їх проєктування в умовах розвитку концепції Industry 4.0. Метою дослідження було обґрунтування придатності циклічно-модульного підходу для декомпозиції мехатронної системи на макромодулі, їх синтезу та подальшого об'єднання без порушення системних властивостей. Методологія дослідження базувалась на застосуванні принципів системного аналізу, декомпозиції та логічного синтезу для побудови й дослідження мехатронних систем з гідропневматичними пристроями автоматизації. В об'єкті синтезу з боку електронної складової взято алгоритм керування виконавчими пристроями системи. Автори розглядають ефективні та конкурентоспроможні технічні рішення з оглядом додавання нових функцій та підвищення ступеня автоматизації, що відповідає тенденціям Індустрії 4.0. Запропоновано абстрактну модель елемента мехатронної системи – макромодуля. Представлено перевагу автономного налаштування і тестування макромодулів, що значно скорочує термін і спрощує проєктування об'єкту. Розглянуто наслідки глибокої фрагментації мехатронної системи та встановлено взаємозв'язок між складністю модулів і архітектурою керування. Наведено результати проведеного аналізу, щодо моделі керування, а саме доведено, що надмірне декомпозиція призводить до появи окремого ієрархічного рівня, що забезпечує зв'язки між спрощеними елементами. Обґрунтовано, що поділ системи на макромодулі не може бути випадковим; він жорстко обмежений фізичними властивостями обладнання і функціями об'єкту проєктування. У статті запропоновано оцінювати якість декомпозиції не за кількістю елементів, а за практичним скороченням часу розробки системи до і після масштабування. Оптимальна розмірність макромодулів є унікальною для кожного конкретного проекту, оскільки базується на змістовному (фізичному) наповненні компонентів. Ефективна декомпозиція знаходить баланс між автономністю макромодуля та складністю координаційних зв'язків, забезпечуючи швидке впровадження складних засобів автоматизації

Ключові слова: архітектура керування; макромодуль; гідропневлічні приводи; декомпозиція функцій; логічне керування; архітектура керування; Industry 4.0