Mech. Adv. Technol., Vol. 9, No. 4, 2025, pp. 432–441

DOI: 10.20535/2521-1943.2025.9.4(107).345026

УДК 625.25

# Mechatronic module with alternative-probability control

**Oleksandr Gubarev**[1] - ORCID https://orcid.org/0000-0003-1059-5768
**Konstantin Belikov**[1] - ORCID https://orcid.org/0000-0002-0924-4103
**Alona Murashchenko**[1] - ORCID https://orcid.org/0000-0003-1059-5768

Received: 2 November 2025 / Revised: 5 December 2025 / Accepted: 24 December 2025

***Abstract.*** *This article presents control algorithms for adaptive mechanical systems. Object of research: processes occurring in industrial hydraulic drive systems during their operation. Subject of research: dependence of operational efficiency of industrial hydraulic drive systems on the term and operating modes, and applied fundamental circuit solutions and technical means that affect the functional, energy, and cost indicators of the system.The problems solved in the presented article are actual tasks. They involve the use of previous experience in the operation of a specific mechatronic system in specific conditions, for the formation of an adaptive control algorithm. The study is based on the creation of logical interpretations in the algorithm for decision-making. The simultaneous consideration of logical connections and the probability of the component "success of system actions" is considered. On the basis of which a two-component structure of logical expressions of control commands is formed. Accordingly, examples are given for evaluating each of the operation options. The article presents the scope and conditions of practical use of the obtained results. They are based on several examples, namely, the technical implementation of the manipulator macromodule with alternative probabilistic control is considered. It is possible to use the success of the system's actions and the choice of the option for distributing attempts by digits or/and the basis of logarithmic weight. The peculiarity of the obtained results lies in the use of the criterion of "volume of involved memory" in the control system. This makes it possible to prioritize one of the alternative reactions of the system to external excitation, which provides a basis for the formation of control commands.*

***Keywords:*** *mechatronic module, automation, alternative probabilistic control, adaptive algorithms, system operating modes.*

## Introduction

The spread of mechatronic automation tools with pneumatic and hydraulic actuators, which are sensitive to changes in conditions and modes of use, is impossible without the creation of adaptive control algorithms. Control systems respond in different ways to external and internal factors, which is one of the hallmarks of the Industry 4.0 platform in creating adaptive mechanical systems with an open architecture [1]–[3], [5]–[8].

Unlike parametric adaptation problems, with the search for rational parameter values, a certain range of problems requires a dynamic change in the control algorithm. This may involve not only adjusting the algorithm, but also changing the criteria for finding a rational solution. This type includes control tasks with the development of a conditioned reflex [11], [14].

A system that contains such an algorithm adjusts the sequence of its actions according to the criterion of increasing the positive effect, choosing a rational action in partially uncertain conditions.

Examples of such control systems are systems that automatically adjust the speed of a car taking into account its current maneuverability or controllability, Examples of such control systems are systems that automatically adjust the speed of a car taking into account its current maneuverability or controllability. assembly links with adjustable part positioning speed, flexible automated production systems, energy systems with combined use of multiple energy sources, etc [4], [9], [10], [17], [19], [20]. Thus, an urgent practical task is to use previous successful experience of operating a specific mechatronic system in specific conditions to form an adaptive control algorithm.

The proposed approach is based on the conditional complexity of the system, on the basis of which the logical inertia of decision-making is determined [14], [16].

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*
*https://ror.org/00syn5v21*

✉ *O. P. Gubarev*
*o.gubarev@kpi.ua*

The purpose of the research is to develop the principles of the structure of adaptive control algorithms for mechatronic systems using a cause-and-effect model of discrete systems, which takes into account the positive and negative experience of previous system actions. The approach is based on creating logical inertia in the decision-making algorithm.

Basic material the logical inertia of decision-making regarding the choice of the module's next action should be distributed according to its typical alternative reactions (for example, A and B). That is, the system can produce a command "by inertia" based on previous successful actions. The relative fraction of the reaction inertia forms the inertial component of the control command expression. The logical expression of the control command has a two-component deterministic-probabilistic logical form.

The deterministic component of control commands corresponds to the selected discrete-logic model of the system and can be obtained using known methods [15], [16], [21]. The probabilistic component is most often defined using fuzzy logic, neural networks, fuzzy sets, and other approaches to building intelligent systems [2], [4], [9], [16], [17]. In the proposed approach, the determination of the probabilistic component is based on the calculation of logical inertia based on the current state of the system's memory and the amount of information about the previous functioning of the system [13], [16].

The process of system functioning has alternative sub-processes that are part of a closed structure. Such a system uses previous operating experience when choosing an alternative solution. That is, the control system must determine the priority alternative for performing the next action.

Formalization of the description of probabilistic-alternative systems

The functioning process has probabilistic-alternative subprocesses combined into a closed structure. The system, by calculation of decision-making inertia, selects the desired alternative for performing the following actions according to the criterion of the most probable result. The number of alternative choice nodes is unlimited. According to formal requirements, each alternative subprocess must have at least 3 consecutive transitions. The overall structure of the process is closed (the beginning coincides with the end).

The existence of a fragment common to probabilistic-alternative regimes is mandatory. After this fragment, selection is taking place of a rational alternative on the value of the probabilistic indicator and the transition to executing the corresponding mode of system actions of the system occurs.

The systems incorporate conditioned reflex development function in the control algorithm, which accounts for the results of both effective and unproductive actions of the system or an individual module.

Probabilistic-alternative choice module.

The use of probabilistic-alternative modules allows the system to automatically adjust to achieving a successful result in partially uncertain environments.

In the capacity of a theoretical basis for the conditioned reflex development function Structured memory volume was used, which reflects the quantitative and qualitative correlation of positive and negative results in the system's or module's previous actions [12], [13], [16]. To simultaneously account for logical connections between module actions and the probabilistic component of system action success, a two-component structure of control command logical expressions was formed.

Each command of the primary action for the probabilistic-alternative choice modules combines a deterministic component (obtained as a result of logical synthesis) and a probabilistic component (constructed based on the result of memory processing):

$$y_i = yi\_iner_{i\,ip\,pi}\,\overline{x_{ip}}_{\,i\_det}\,, \tag{1}$$

where $yi_{i\_det}$ – logical condition for deterministic activation, $y_{i\_iner} = (\{x_{ip}\} \wedge x_p)$ – condition for activation based on the inertial component, $x_p$ – binary variable – an indicator of probabilistic choice, defined by taking into account the probability distribution of positive and negative experience, $\{x_{ip}\}$ – logical condition for enabling/disabling probabilistic control.

To implement the probabilistic selection of an alternative action, a methodology was developed for calculating the probabilistic component using the system's logical inertia based on a structured memory in a countdown mode. In determining the probabilistic indicator of alternatives, a number generator function was applied, which is synchronized with the system's inherent state processing frequency.

For each alternative, the logical inertia term forms the inertial component of the control signal (1). The ratio of logical inertia between two alternative options determines the probability of applying one or the other alternative [16]. That is, if the inertial components are equal, the probability of choosing each alternative is 0.5. If the inertial components have different values, but one of them must necessarily be executed, the sum of the inertial components is taken as 1. Then the probability of choosing each of the alternatives will be proportional to its inertial component.

An important feature of the probabilistic-alternative choice modules is that they are combined into pairs. One module in the pair ensures the transition to one alternative ("A"), and the second module in such a pair ensures the transition to the second alternative ("B"). The modules carry a specific information load. The number of probabilistic-alternative modules is equal to the number of alternative modes in the system's operation.

The Probabilistic-Alternative Choice Module is purely informational, functioning as a memory element with an added probabilistic component. Its purpose is to confirm or refute the results of logical synthesis that are sufficient to initiate an alternative sub-process.

The alternative choice module takes the form of a set of two binary variables that change their values in antiphase, for example $X_{A1}$ i $X_{\overline{A1}}$.

Module Activation.

Module activation occurs when there is a simultaneous existence of both the logical expression of the deterministic part of the control command expression and a positive value or state of the probabilistic indicator:

$X_p$ :

$$y_{A1} = yA1\_iner_{A1\_Logic\,pA1A1\overline{A1}A1\_det}.$$

<u>Module deactivation</u>: deactivation occurs upon the existence of the control command signal for deactivation, with no additional conditions:

$$y_{\overline{A1}} = y\overline{A1}\_Logic_{A1\overline{A1}\overline{A1}\_det}.$$

The content of the main actions for each of the pair of probabilistic-alternative choice modules $y_{A1}$ and $y_{B1}$ must be mutually exclusive. That is, the true value of the probabilistic indicator for enabling the main action of one module pair in the alternative $X_{pA1}$ is equivalent to the negation (or NOT) of the true value of the probabilistic indicator of the other module in the same pair: $X_{pA1} = \overline{X}_{pB1}$.

<u>The </u>content<u> of the main action</u> for each of the pair of alternative modules begins simultaneously, since they share a single immediate cause. The content of the main action of the first probabilistic-alternative module can be formulated in a generalized form as follows:

– Initiate the calculation of the inertial component regarding the system's entry into alternative sub-process "A";
– Obtain the calculation result (a quantitative value);
– Determine the value of the probabilistic indicator.

<u>The content of the main action</u> of the second probabilistic-alternative module can be formulated in a generalized form as follows:

– Initiate the calculation of the inertial component regarding the system's entry into alternative sub-process "B";
– Obtain the calculation result (a quantitative value);
– Determine the value of the probabilistic indicator.

Next, based on the obtained values of the inertial components, the execution probability for each main action is calculated.

After the probability values are obtained, the state of the probabilistic indicator (a binary variable) for the alternative actions of each module in the pair is determined. The indicator is determined using a synchronous random number generator. The generator operates in synchronization with the signal processing rhythm of the system modules' state, which prevents its value from changing during the execution of alternative actions. Random numbers are selected from an interval equal to the total number of system activations. The selection of a number that falls within the range corresponding to variant "A" actuations or variant "B" actuations determines the positive value of the probabilistic choice indicator.

<u>The content of the return actions</u> for each of the pair of alternative choice modules can be formulated in a generalized form as follows:

– stop the calculation of the inertial component regarding the conditions for the system's entry into alternative sub-process "A";
– set the initial values of the module state indicators, for example:

$$X_{pA1} = 0; X_{\overline{pA1}} = 1.$$

The object of control is a manipulator whose task is to remove a part–which is being pushed out by a loader from the previous stage into the working zone–as quickly as possible and to transfer this part to the next production stage.

As a variant of the technical implementation, a macro-module consisting of three executive devices is considered: a pneumatic manipulator of the loading/unloading mechanism (Fig. 1). The macro-module executes a substantive action $Z_1$ – removing the part from the working area and moving it to the next step, and after its execution it must return to the original state, i.e. perform the reverse action $Z_{\overline{1}}$. The successful execution of the main action is accompanied by a unit signal $\phi(Z_1)$ of the external signal, which confirms that the next stage of the production line has received the part.

<u>Note</u>: The external signal $\phi(Z_1)$ is not a signal from a position or pressure sensor that controls the movement of the output stage. It is a signal from the next stage of automated production. The next stage confirms, via a unitary signal value, that it has accepted the transferred part for processing.

The macro-module begins to execute the main action $Z_1$ based on the truth value of the logical condition for starting the removal of the part from the working zone. Following this signal, the manipulator enters the working zone (command Y1), lowers the gripper (command Y2), grips the part (command Y3), raises the gripper (command YN2), moves the part to the next stage (command YN1), lowers the gripper (command Y2), and releases the part (command YN3).

At this point, the main action is completed.

Analogously, the return action $Z_{\overline{1}}$ (or reversal action) involves the macro-module returning to its initial state, meaning the manipulator raises the empty gripper (command YN2)

Additional Condition: In practice, various macro-module action variants exist, which are caused by the different timing of external signals. The impetus for the external signal to have a unitary value for the macro-module is a change in the state of the surrounding environment, which the system perceives as changes in the values of the binary signals $x_{s1}$ and $x_{s2}$.

Signals $x_{s2}$ is faster, and the signal $x_{s1}$ is slower. For example, signal $x_{s2}$ – is the start of movement of the part loader, which moves the part toward the working zone. Signal $x_{s1}$ is the product logical AND of signals from an optical sensor and a pressure sensor. Their product (logical

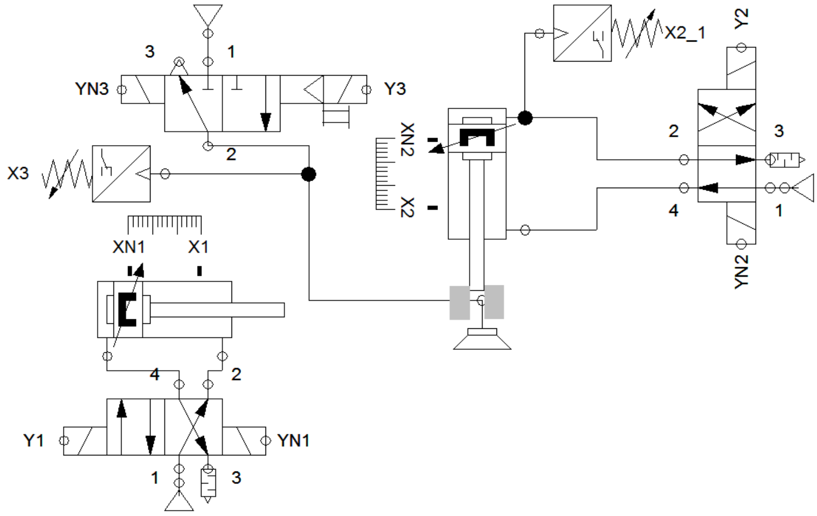


**Fig. 1.** Example diagram of a pneumatic manipulator: Y1, YN1 – control signals for the horizontal movement actuator X1, XN1 – signals of position control for the horizontal movement actuator; Y2, YN2 – signals of control for the vertical movement actuator; X2, XN2 – signal of position control for the vertical movement; X2_1 – pressure control signal; Y3, YN3 – control signals for the gripper; X3 – control signal for part retention by the gripper

AND) confirms that the part has already entered the working zone and is waiting to be transferred to the next stage.

The executive device (the manipulator) must remove the part from the working zone as quickly as possible and transfer it to the next operation.

If one were to rely only on the start of the loader's movement, the part would, most likely, still be in the process of being loaded.

If one were to rely only on the control for the part's presence in the working zone, the time taken to bring the manipulator into position and unload the part increases. As a result, controlling based on the confirmed signal leads the system to lose productivity.

When using signal $\boldsymbol{x}_{s2}$, the system operates "proactively" (or "anticipatorily"). If the result of the action based on signal $\boldsymbol{x}_{s2}$ turns out to be false (incorrect), the system has performed an unnecessary action (redundant actuation), meaning it wasted energy for nothing. This confirms the absence of the transferred part.

The adaptive algorithm of the control system must take into account possible variants of task execution that correspond to different combinations of external influence signal values $\boldsymbol{x}_{s1} \in \{0,1\}$ and $\boldsymbol{x}_{s2} \in \{0,1\}$ and the different times at which they are received.

That is, we have several possible combinations of signals. The option of triggering a signal about the presence of parts without the loader starting to move is marked as impossible or purely virtual for a mechanical system. The option with zero values of both signals corresponds to the system waiting mode.

We have the following possible options for the macromodule:

a) $\boldsymbol{N}_b$ successful work on the start of movement and cargo control $\boldsymbol{x}_{s2} = 1, \boldsymbol{x}_{s1} = 1$,

b) $\boldsymbol{N}_a$ $\boldsymbol{x}_{s2} = 1, \boldsymbol{x}_{s1} = 0$,

c) $\boldsymbol{N}_c$ false triggering when working after starting the movement $\boldsymbol{x}_{s2} = 1, \boldsymbol{x}_{s1} = 0$.

In the first variant, the system, based on the product of the signals of the optical and weight sensors and the signal about the start of the loader movement, removes the part from the working area. That is, it executes a sequence of commands for the main and reverse actions (Fig. 2).

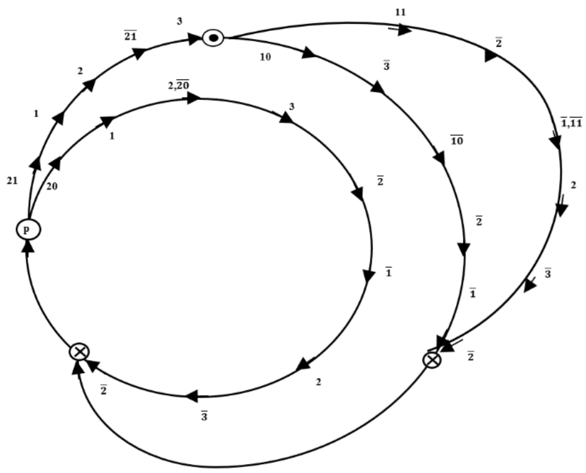

**Fig. 2.** Initial version of the functional graph of the system with probabilistic-alternative (beginning – arcs *20*, *21*) and alternative (beginning – arcs *10*, *11*) subprocesses for the pneumatic manipulator: *1* – horizontal movement, *2* – vertical movement, *3* – gripper drive; two arcs are reserved for memory elements of non-alternative parts of the process

Y1 – Y2 – Y3 – YN2 – YN1 – Y2 – YN3 – YN2.

In the second variant, the system, upon the signal to start the movement, unloads the part from the working area. That is, it executes the same sequence of commands, but checks the presence of the part in the gripper:

Y1 – Y2 – Y3 – ? "+" – YN2 – YN1 – Y2 – YN3 – YN2.

In the third variant, the system, upon the signal to start the movement, tries to unload the part from the working area, but the control of the part's presence when the gripper is triggered gives a negative result and the manipulator returns to its initial state:

Y1 – Y2 – Y3 – ? "–" – YN3 – YN2 – YN1.

In fact, the beginning of the actions in the second and third options coincide, and only after checking the presence or absence of the part, the option becomes the second (part is present) or the third (part is absent). That is why a check with a positive result was added to the second option: ? "+" – YN2 – YN1 – Y2 – YN3 – YN2, and in the third option, a check with a negative result? "–" – YN3 – YN2 – YN1.

Structuring the system memory. The system memory accumulates information about the previous operation of the module in at least two dimensions:

– The first dimension is the number of times the system has executed each variant.

– The second dimension is the proximity of each previous execution of the system to the current moment.

– The context for evaluating both dimensions is the total number of system activations.

Calculation of inertia. Structuring memory and calculating the probability of response options is the main action of module 7 (Fig. 2). After the calculation is completed and the random number generator is triggered, the value of the probability indicator for each of the alternative actions is determined.

Let us assume, as a first approximation, that the logical inertia of a certain system response option is a consequence of the amount of memory occupied by the information dedicated to this particular option. To store the number of attempts N in binary code, the required number of bits or memory elements will be $\log_2 N + 1$.

The total number of attempts N is equal to the sum of the attempts of all options a), b) and c). Each of the attempts of the listed options ($N_a$, $N_b$, $N_c$) has its own ordered place in the number N.

Conditionally, the history of attempts can be represented by the following sequence (Table 1), where (*) indicates the current waiting state.

In a second approximation, let us assume that the current influence of the external environment should be determined by the system more by the results of current attempts than by the results of past periods. We ensure this by counting down the attempts (line 2, table 1).

The next step is the transition from the number of different attempt results to the decision-making algorithm. To do this, we adapt Newton's 2nd Law for a logical-binary transition in the following form:

$$F_i = m_i \cdot \ddot{x}_i = J_i \cdot y_i \,, \tag{2}$$

where $J_i$ – measure of system inertia with respect to a destabilizing influence, $F_i$ – destabilizing influence, $y_i$ – system reaction to a destabilizing influence.

If the influence can differ by its weight value, then the left side of the equation must also have a variable component. Since the logical reaction can only take two fixed values (act – 1, or not act – 0), the variable component must be the internal system inertia with respect to the exciting factor (or perturbation), that is $J_i$. In the "logical inertia", both measurement components must be taken into account: the number of actuations (or attempts) and their proximity to the current moment (recency).

It is proposed, for the assessment of the quantitative component recorded in binary code, to measure inertia by the required number of memory elements (bits) needed to store the count of actuations with a specific result. Thus, each actuation variant can be evaluated:

$$J_a = INT\left(\log_2 n_{1,2}\right) + 1 \tag{3}$$

evaluation of fully successful trials,

$$J_b = INT\left(\log_2 n_1\right) + 1 \tag{4}$$

evaluation of conditionally successful trials,

$$J_c = INT\left(\log_2 n_{1,\bar{2}}\right) + 1 \tag{5}$$

evaluation of failed trials,

where $n_{1,2}$ – number of entirely successful attempts, $n_1$ – number of conditionally successful attempts, $n_{1,\bar{2}}$ – number of unsuccessful attempts.

**Table 1.** History of changes

| $N$ Count Up | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\bar{N}$ Count Down | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | * |
| $N_a$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | |
| $N_b$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | |
| $N_c$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |

Each of the three inertia assessments by the quantitative component, on its own, does not provide justification for considering the influence of the surrounding environment when selecting the variant for the next action. For example, at the beginning of the system's operation, the signal to start moving was almost not perceived due to vibrations. As a result, there were no conditionally successful attempts over a large number of actuations, for example, 2000. During the device adjustment period, the influence of vibrations gradually weakened compared to the start-of-movement signal. Thus, in the 500 attempts following the 2000th attempt, there were 176 conditionally successful attempts. Furthermore, in the last 200 attempts of these 500, there were 81 conditionally successful attempts, and in the last 100 attempts of those 200, there were 49 conditionally successful attempts. If one were to evaluate the total number of system actuations and the conditionally successful actuations, their ratio would be 0,088.

If we consider only the last 100 attempts, the same ratio takes the value 0,49.

An intermediate analysis of the number of attempts over the last 500 yields a value of 0,352. That is, the number of conditionally successful actuations in relative terms is gradually increasing, and this growth can be considered an indication of a trend in the surrounding environment's influence.

The assessments provided do not take into account the common context of all variants. The context must, at the very least, take into account the total number of attempts for a fixed exciting factor. $\boldsymbol{n} = \boldsymbol{n}_{1,2} + \boldsymbol{n}_1 + \boldsymbol{n}_{1,\bar{2}}$ :

$$\boldsymbol{INT}\left(\log_2(\boldsymbol{n}_{1,2} + \boldsymbol{n}_1 + \boldsymbol{n}_{1,\bar{2}}\right) + 1 . \tag{6}$$

The components from expressions (3), (4), and (5), in relation to (6), approximately indicate the relative weight of each attempt variant over the entire history of the system's actions for the chosen exciting factor. However, this assessment does not take into account the different weights of recent attempts and attempts from past periods. This distinction provides the system with information regarding current trends in the influence of the surrounding environment.

The assessment of the memory used based on (expressions 3 to 6) only takes into account the number of different attempt results, but neglects the remoteness of their execution time from the current moment. That is, the memory engaged for information about a specific attempt result is dispersed among the memory values for other attempt results over the entire duration of the system's functioning. But if the memory of the considered result is concentrated in a small current segment of the total memory, then this indicates the relevance of exactly that result at the present time.

That is, with an equal number of the same attempt result, if they fall, for example, into the first half of the memory or the second half of the memory, their inertial influence should differ by approximately a factor of two.

Accordingly, the probability of a given system reaction to an exciting influence can be calculated precisely at the current moment based on the memory's weight.

The proposed variant for utilizing logical inertia is as follows: The probability function of a specific result takes into account the memory occupancy by actuations corresponding exactly to that result. The actuation results are denoted by binary variables.

The actuations are taken into account during their countdown (reverse count). That is, each subsequent bit (in the reverse count) accounts for half the weight of the previous bit.

The weight of the first bit is 0,5, the second is 0,25, and in the general case $2^{-k}$, where $k$ – the ordinal number of the bit. However, in each bit, a specific attempt result is "dispersed" and occupies only a "portion" of it. Let us assume that Table 1 provides the following context for the distribution of attempt variants across the bits (Table 2).

**Table 2.** History of Attempts by Digit

| $\boldsymbol{N}$ Count Up | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\bar{\boldsymbol{N}}$ Count Down | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | * |
| Digit | 4 (One trial is missing) | | | | | | | 3 | | | | 2 | | 1 | * |
| $\boldsymbol{\omega_j N_a}$ Unconditional | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | |
| $\boldsymbol{\omega_j N_b}$ Conditional "+" | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | |
| $\boldsymbol{\omega_j N_c}$ Conditional "–" | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |

The distributed memory for the results of attempt variant "a" according to (Table 2) provides the following context:

The first digit $p_{1/a}$ = 0,5;

The second digit $p_{2/a}$ = 0,25/2;

The third digit $p_{3/a}$ = 0,125/4;

The fourth digit $p_{4/a}$ = 2·(0,0625/8).

Total probability assessment of variant "a" for the next attempt: $p_{\Sigma/a}$ = 0,6875.

Similarly, the distribution of trial results for the variant "*b"*:

The first digit $p_{1/b}$ = 0,0;

The second digit $p_{2/b}$ = 0,0;

The third digit $p_{3/b}$ = 3·(0,125/4);

The fourth digit $p_{4/b}$ = 3·(0,0625/8).

The total probability estimate of the outcome for option "b" for the next attempt is:

$p_{\Sigma/b}$ = 0,1171875.

Similarly, the distribution of trial results for the variant "c":

The first digit $p_{1/c}$ = 0,0;

The second digit $p_{2/c}$ = 0,25/2;

The third digit $p_{3/c}$ = 0,0;

The fourth digit $p_{4/c}$ = 2·(0,0625/8).

Total probability estimate of option "c" for the next attempt: $p_{\Sigma/c}$ = 0,140625.

In the general case, with the number of all attempts *n*, for option *j* we get:

$$p_i = \sum_{k=1}^{\log_2 n} p_{k/i} = \sum_{k=1}^{\log_2 n}\left(\frac{1}{k+1}\cdot\sum_{j=2^{k-1}}^{2^k}\omega_i\right), \quad (7)$$

where $\omega_j$ – the effectiveness of the $j$ – th attempt of the variant $i$, $\log_2 n$ – number of memory bits involved.

Since the number of completed attempts is not infinite, the sum of the calculated probabilities is less than one. That is, the probability distribution between possible outcomes must take into account the probability deficit, namely $p_{\Sigma/abc}$ = 0,9453125. When proportionally distributing the deficit of attempts across all options, we obtain the clarifying coefficient:

$$\chi_{pr} = \left(\sum_{i=1}^{N} p_i\right)^{-1} = \left(\sum_{i=1}^{N}\sum_{k=1}^{\log_2 n} p_{k/i}\right)^{-1}, \quad (8)$$

where $N$ – the number of options for the system's reaction to a fixed external destabilizing influence.

According to (7) and (8), we obtain:

$$p_i = \left(\sum_{i=1}^{N}\sum_{k=1}^{\log_2 n} p_{k/i}\right)^{-1}\cdot\sum_{k=1}^{\log_2 n}\left(\frac{1}{k+1}\cdot\sum_{j=2^{k-1}}^{2^k}\omega_i\right). \quad (9)$$

For the considered example, taking into account (9), the adjusted probability values distributed according to the results of the options are *a, b* and *c*, will consist of:

$p'_{\Sigma/a}$ = 0,727273; $p'_{\Sigma/b}$ = 0,12397; $p'_{\Sigma/c}$ = 0,14876. The sum of the probabilities of all options is 1.

The next step is to use the probability estimate of each outcome option in the system control algorithm. That is, the control algorithm must determine whether the system should react to the start of the loader movement or wait for signals from the optical and weight sensors at the working position.

Since the system can work either in a deterministic way, when the input data is sufficient to make a clear decision, or according to the option with a deficit of external information, then the expression of the control commands for conditional selection of an option must take into account both options (1).

Next, according to the number of triggers for each outcome option and the total number of triggers and the proximity of the results to the current time, the value of the probability indicator is determined.

To implement the probabilistic component of expression (1), a T/C random number generator was used. If the value obtained using the generator, in percentage terms, exceeds the calculated probability (9), then the binary indicator is equal to "0", otherwise $x_p = 1$:

$$x_p = \begin{cases} 1: p_i = \left(\sum_{i=1}^{N}\sum_{k=1}^{\log_2 n} p_{k/i}\right)^{-1}\times \\ \times\sum_{k=1}^{\log_2 n}\left(\frac{1}{k+1}\cdot\sum_{j=2^{k-1}}^{2^k}\omega_i\right) \le CW0 \\ 0: p_i = \left(\sum_{i=1}^{N}\sum_{k=1}^{\log_2 n} p_{k/i}\right)^{-1}\times \\ \times\sum_{k=1}^{\log_2 n}\left(\frac{1}{k+1}\cdot\sum_{j=2^{k-1}}^{2^k}\omega_i\right) > CW0 \end{cases}. \quad (10)$$

The control algorithm on the arc preceding the vertex of the probabilistic alternative choice chooses between two options:

b) – react to the start of the loader movement (module 21, Fig. 2),

c) – do not react to the start of the movement (wait for confirmation of the presence of the part in the working position) and use option a) (module 20, Fig. 2).

The adjusted probability values of different outcomes, taking into account the limited number of trials, are $p'_{\Sigma/b}$ = 0,12397; and $p'_{\Sigma/c}$ = 0,14876 and $p'_{\Sigma/a}$ = 0,727273 in accordance.

That is, if the T/C generator provides a value greater than 0,12397, then the system waits for confirmation of the presence of the part (command 20 is triggered. Fig. 2). If the value from the T/C generator is less than or equal to 0,12397, then the system works ahead of schedule, that is, the manipulator tries to grab the part (command module 21 is triggered).

Next, the part is controlled to be captured (command 11), or the absence of the part in the grip is controlled (command 10). Depending on the activation of the part presence control sensor, either the manipulator returns to its initial state (after alternative selection command 10) or the part is shipped to the next production link (after alternative selection command 11).

For example, if the T/C generator provided a value of 9/100, then the multiplier and the system will issue a command to advance. But if the T/C generator provided a value of 32/100, then то $x_p = 0$ and the system will execute command 20 and wait for confirmation of the presence of the part in the working position.

For a system that satisfies the criterion of logical determinism, a logical synthesis of control command expressions can be performed, including alternative modules (10 and 11) and probabilistic-alternative choice modules (20 and 21). The exceptions are the actions of alternative and alternative-probabilistic choice and the function of determining a probabilistic indicator.

Supplementing systems with redundant memory elements removes excessive complexity from logical command expressions. For example, for the pneumatic manipulator system under consideration, 4 memory elements are sufficient, but some control command expressions then consist of 10 or more signals. This complicates the execution of command synthesis and creates inconveniences for further modernization of the system, for example, adding new functions and modules that execute them. By adding two more memory elements, the functional graph obtains a segmented structure and simplified expressions of control commands.

The list of logical expressions of control commands is given below.

$$y_1 \Leftarrow x_{21} + x_{20}; \qquad y_{\bar{1}} \Leftarrow x_{\bar{2}} \cdot x_{\bar{4}} \cdot x_5 + x_5 \cdot x_{\bar{2}} + x_{\bar{2}} \cdot x_{\bar{4}};$$

$$y_2 \Leftarrow x_1 \cdot x_{21} + x_{\bar{1}} \cdot x_{\overline{11}} \cdot x_5 \cdot x_4 + x_{\bar{1}} \cdot x_9 \cdot x_3;$$

$$y_{\bar{2}} \Leftarrow x_3 \cdot x_1 \cdot x_9 + x_{\bar{9}} \cdot x_{\bar{3}} \cdot x_{\bar{4}} \cdot x_{\bar{8}} + x_{11} + x_{\bar{3}} \cdot x_{\bar{4}} + x_{\overline{10}} \cdot x_{12};$$

$$y_3 \Leftarrow x_2 \cdot x_{\overline{20}} \cdot x_9 \cdot x_4 + x_{\overline{21}} \cdot x_8 \cdot x_4 \cdot x_{\bar{5}} \cdot x_{\overline{10}};$$

$$y_{\bar{3}} \Leftarrow x_2 \cdot x_{\bar{4}} \cdot x_{\bar{1}} + x_{10} + x_2 \cdot x_{\bar{4}} \cdot x_5;$$

$$y_4 \Leftarrow x_{\bar{7}}; \qquad y_{\bar{4}} \Leftarrow x_3 \cdot x_9 + x_{\overline{10}} \cdot x_5 \cdot x_8 + x_{\bar{1}} \cdot x_{\overline{11}} \cdot x_5 \cdot x_8;$$

$$y_5 \Leftarrow x_{11} + x_{10}; \qquad y_{\bar{5}} \Leftarrow x_{\bar{8}};$$

$$y_6 \Leftarrow x_{\bar{1}} \cdot x_{\overline{12}} \cdot x_{\bar{2}} \cdot x_{\bar{4}} \cdot x_8; \qquad y_{\bar{6}} \Leftarrow x_{\bar{5}};$$

$$y_7 \Leftarrow x_4 \cdot x_{s2}; \qquad y_{\bar{7}} \Leftarrow x_{\bar{6}} \cdot x_{\bar{2}} \cdot x_{\bar{9}} \cdot x_{\bar{4}} \cdot x_{\bar{8}};$$

$$y_8 \Leftarrow x_1 \cdot x_{21}; \qquad y_{\bar{8}} \Leftarrow x_6;$$

$$y_9 \Leftarrow x_{20}; \qquad y_{\bar{9}} \Leftarrow x_2 \cdot x_{\bar{1}};$$

$$y_{10} \Leftarrow x_3; \qquad y_{\overline{10}} \Leftarrow x_{\bar{3}} \cdot x_5 \cdot x_{12};$$

$$y_{11} \Leftarrow x_3 \cdot x_{\bar{5}} \cdot x_8; \qquad y_{\overline{11}} \Leftarrow x_{\bar{2}} \cdot x_5;$$

$$y_{12} \Leftarrow x_{10}; \qquad y_{\overline{12}} \Leftarrow x_{\bar{2}} \cdot x_{\bar{4}};$$

$$y_{20} \Leftarrow x_7 \cdot x_4 \cdot x_{\bar{9}} \cdot \bar{x}_p \cdot B_{ip} + x_7 \cdot x_4 \cdot x_{\bar{9}} \cdot \bar{B}_{ip};$$

$$y_{\overline{20}} \Leftarrow x_1 \cdot x_9;$$

$$y_{21} \Leftarrow x_7 \cdot x_4 \cdot x_8 \cdot x_p \cdot B_{ip} + x_7 \cdot x_4 \cdot x_8 \cdot \bar{B}_{ip};$$

$$y_{\overline{21}} \Leftarrow x_2 \cdot x_8.$$

The bootloader start condition has been added to the logical expression of the 7th module command $x_{s2}$. In logical expressions of commands of the 20th and 21st modules have been added systems in probabilistic-alternative mode $B_{ip}$ and the probabilistic indicator $x_p$

Module 7 is informational (based on a memory element with two binary variables). Its main function is to calculate a probabilistic indicator based on the current state of the attempt history table. An additional external signal to start the main action of module 7 is the signal to start the movement of the loader $x_{s2}$.

The reverse action of module 7 is turning off the memory element.

The attempt history table is supplemented (memory expansion, Table 2) upon activation of alternative choice modules.

When module 20 is triggered, i.e. the manipulator is triggered by signals of the start of the loader movement $x_{s2}$ and the presence of the part $x_{s1}$, variant “a” gets “1”.

When module 21, that is, upon the signal of the start of the loader movement, and the module activation 11, that is, successful operation of the manipulator in the presence of a part, variant “b” gets “1”.

When module 20 is triggered, i.e. upon the signal of the start of the loader movement, and the activation of module 10, i.e. false activation of the manipulator in the absence of a part, variant “c” gets “1”.

## Discussion

The example above uses the “shock” perception of the result of the last attempt. That is, the weight of this result in the formation of the inertial component is equal to the weight of all previous attempts. Such a scheme is focused only on the current external influence on the operation of the system. There is almost no consideration of certain trends in changes in the inertia of the system over a period of time.

A more balanced option can be obtained by shifting the start of the countdown in the digits of the structured history of attempts. That is, the first category may have not one attempt, but two. Then the first 0,5 of the inertial probability will be distributed over the first two attempts. The last attempt will have a weight of 0,25 and can be balanced by the previous attempt, which also has a weight of 0,25. The weight distribution for the remaining attempts is the same as in the example given. Each subsequent discharge accounts for half of the unused weight. That is, the second digit has 0,25 weight. The number of attempts in each subsequent level is twice the number of attempts in the current level.

An even more restrained version of the reaction to the last attempt arises, if you concentrate four attempts in the first category.

The weight of the last attempt will be 0,125, the same as the three previous attempts. That is, the system’s reaction to the last result will be 4 times weaker than in the example given.

The second option for influencing the formation of logical inertia is to withdraw from the sheer number of binary memory elements involved, that is, a departure from logarithmic weight modulo 2, for example, to modulo 3. Instead $\boldsymbol{J_a} = \boldsymbol{INT}(\log_2 \boldsymbol{n}_{1,2}) + 1$ we get $\boldsymbol{J_a} = \boldsymbol{INT}(\log_3 \boldsymbol{n}_{1,2}) + 1$. Similarly, not 0,5 of the inertia weight can be attributed to the first category, and the corrected value is, for example, 0,25. Then the next digit will account for 25 % of the remainder, i.e. 0,75·0,25 = 0,1875. The third digit will include $0,25 \cdot (1 - 0,25 - 0,1875) = = 0{,}1406$. Thus, the total weight of third-rate attempts is 56,3 % of the total weight of first-rate attempts. While in the first variant this percentage was 25 %, that is, older attempts had half the influence on the formation of the choice.

It is possible to use the success of the system's actions and the choice of the option for distributing attempts by digits or/and the basis of logarithmic weight, the basis for which will be the actual results of the operation of a particular system.

## Conclusions

Increasing the efficiency of automated mechatronic systems can be ensured by considering previous experience of their operation, which can be used in control algorithms by developing a conditioned reflex to positive results, and using this reflex when forming commands for further system actions.

Using the criterion of the amount of memory involved in the control system as a criterion allows you to prioritize one of the alternative system reactions to external excitation, which provides a basis for forming control commands.

Structuring system memory using a logarithmic estimate of the amount of information, other than a binary base and using countdown when analyzing memory allocated by results gives priority to the system's response to results that are closest in time.

## Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, including financial, personal, authorship, or any other nature that could affect the research and its results presented in this article.

## Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

## References


[1] A. Rojko, "Industry 4.0 Concept: Background and Overview," *iJIM*, Vol. 11, No. 5, pp. 77–90, 2017, doi: https://doi.org/10.3991/ijim.v11i5.7072.

[2] L. Kozlov and Y. Burennikov, "Mechatronic Hydraulic System with Adaptive Controller on the Basis of Neural Networks," *Universitatea Tehnica "Gheorghe Asachi" din Iasi Tomul LXI (LXV)*, Fasc. 1–2, pp. 132–151, 2015. Available: https://sim.tuiasi.ro/wp-con- tent/uploads/2015/03/Stiinta-si-Ingineria-Materialelor-1-2-pe-2015.pdf.

[3] J. Lee, H.-A. Kao and S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, Vol. 16, pp. 3–8, 2014, doi: https://doi.org/10.1016/j.procir.2014.02.001.

[4] Y. Li, H. Wang and J. Zhang, "Precise control of hydraulic actuators using adaptive backstepping and neural networks", *Computers, Materials & Continua*, 141(2), pp. 1235–1250, 2024.

[5] L. Knapčíková, Industry 4.0: Trends in Management of Intellegent Manufacturing Systems, Springer, 146 p. ISBN 3030140113, 9783030140113, 2019, doi: https://doi.org/10.1007/978-3-030-14011-3.

[6] M. Brettel, N. Friederichsen, M. Keller and M. Rosenberg, "How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective", *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, Vol. 8, No. 1, pp. 37–44, 2014.

[7] L. Schmidt and K. Hansen, "Electro-Hydraulic Variable-Speed Drive Networks–Idea, Perspectives, and Energy Saving Potentials", *Energies*, Vol. 15(3), 1228, 2022, doi: https://doi.org/10.3390/en15031228.

[8] T. Bauernhansl, M. ten Hompel and B. Vogel-Heuser, Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung Technologien Migration, Springer-Verlag, 2014, doi: https://doi.org/10.1007/978-3-658-04682-8.

[9] S.-C. Vanegas-Ayala, J. Baro´n-Velandia and D.-D. Leal-Lara, "A systematic review of greenhouse humidity prediction and control models using fuzzy inference systems," *Advancesin Human-Computer Interaction*, 2022, doi: https://doi.org/10.1155/2022/8483003.

[10] S. Zhao et al., "A High-Order Load Model and the Control Algorithm for an Aerospace Electro-Hydraulic Actuator", *Actuators*, Vol. 10(3), p. 53, 2021, doi: https://doi.org/10.3390/act10030053.

[11] P. K. Anokhyn, Byolohyia y neirofyzyolohyia uslovnoho refleksa, Moscow: Medytsyna, 1968.

[12] K. O. Belikov and O. P. Gubarev, "Adaptation of control in electropneumatic systems with discrete software control", *Bulletin of the National Technical University "KhPI", Series: Hydraulic machines and hydraulic units*, No. 1, pp. 18–22, 2020.

[13] O. S. Hanpantsurova and O. P. Gubarev, "Lohiko-inertsiina skladova komand keruvannia vykonavchym modulem mekhatronnoi systemy," in *Proc. XVII MNTK AS PHP "Promyslova hidravlika i pnevmatyka"*, Kharkiv, 2016.

[14] M. V. Hlushkov, Yu. V. Kapytonova and A. T. Myshchenko, *Lohichne proektyvannya dyskretnukh prystroiv*, Kyiv: Nauko-vadumka, 1987.

[15] A.P. Gubarev, *Dyskretno-lohychne upravlinnya v systemakh hydropnevmoavtomatyky*, Kyiv: YSMO, 1997, ISBN 5-7763-8725-6.

[16] A. P. Gubarev, *Do putannya adaptacyy lohychnoho keruvannya*, Deponent: UkrNYYNTY, N282-Uk86, 1986.

[17] L. H. Kozlov, "Using neural network for regulation time reduction in the mechatronic hydraulic system," *Visnyk Sumskoho derzhavnoho universytetu. Seriia "Tekhnichni nauky*, No. 4, pp. 165–174, 2013, [Online]. Available: https://essuir.sumdu.edu.ua/handle/123456789/33794.

[18] T. Petrakis et al., "Neural Network Model for Greenhouse Microclimate Predictions", *Agriculture*, Vol. 12(6), 2022, doi: https://doi.org/10.3390/agriculture12060780.

[19] V. P. Tarasyk and S. A. Runkevych, *Intellektualny systemy upravlinnya avtotransportnymy zasobamy*, Monohrafyia, Mn.:UP "Tekhnoprynt", 2004, 512 p. ISBN 985-464-664-5.

[20] M. V. Cherkashenko, *Avtomatyzatsiya proektuvannya system hidro- i pnevmopryvodiv z dyskretnym upravlinnyam*, Kharkiv: NTU "KhPY", 2007, 210 p. ISBN 5-217-01882-8.

## Мехатронний модуль з альтернативно-ймовірнісним керуванням

**О. П. Губарев[1] • О. П. Бєліков[1] • А. М. Муращенко[1]**

[1] *КПІ імені Ігоря Сікорського, Київ, Україна*

***Анотація.*** *В цій статті представлено алгоритми керування адаптивними механічними системами. Об'єкт дослідження: процеси, що відбуваються в системах промислового гідроприводу при їх експлуатації. Предмет дослідження: залежність експлуатаційної ефективності систем промислового гідроприводу від терміну і режимів роботи, та застосованих принципових схемних рішень і технічних засобів, які впливають на функціональні, енергетичні, та вартісні показники системи. Проблеми, що вирішуються в представленій статті є актуальним задачами з використанням попереднього досвіду експлуатації конкретної мехатронної системи в конкретних умовах для формування адаптивного алгоритму керування. Дослідження базується на створенні логічних інтерпретацій в алгоритмі для прийняття рішень. Розглянуто одночасне врахування логічних зав'язків та ймовірність складової «успішності дій системи», на основі чого сформовано двокомпонентну структуру логічних виразів команд керування. Відповідно, наведено приклади для оцінки кожного з варіантів спрацювання. В статті наведено сферу та умови практичного використання отриманих результатів на основі декількох прикладів, а саме технічної реалізації розглянуто макромодуль маніпулятора з альтернативно-ймовірнісним керуванням. Можливо використати успішність дій системи і вибору варіанту розподілу спроб по розрядах або/та основі логарифмічної ваги. Особливість отриманих результатів полягає у використанню критерію «об'єму задіяної пам'яті» в системі керування, що дає можливість пріоритетно обирати одну з альтернативних реакцій системи на зовнішнє збудження, що надає підґрунтя для формування команд керування.*

***Ключові слова:*** *мехатронний модуль, автоматизації, альтернативно-ймовірнісне керування, адаптивні алгоритми, режими роботи системи.*