Розроблено методику аналізу логіко-динамічних процесів перетворення аргументів в арифметичних пристроях цифрових систем управління. Описано недоліки і обмеження використовуваних формальних методів опису процесів в системах управління, запропонований графо-аналітичний метод опису процесів перетворення аргументів. Виконано аналіз логіко-динамічних процесів перетворення інформаційних аргументів на суматорах і помножувачах, які використовуються в цифрових системах керування

Ключові слова: логіко-динамічний процес, перетворення аргументів, частковий добуток, графо-аналітична модель

Разработана методика анализа логико-динамических процессов преобразования аргументов в арифметических устройствах цифровых систем управления. Описаны недостатки и ограничения используемых формальных методов описания процессов в системах управления, предложен графо-аналитический метод описания процессов преобразования аргументов. Выполнен анализ логико-динамических процессов преобразования информационных аргументов в сумматорах и умножителях, которые используются в цифровых системах управления

Ключевые слова: логико-динамический процесс, преобразование аргументов, частичное произведение, графо-аналитическая модель

UDC 004.62 : 681.5.015
DOI: 10.15587/1729-4061.2017.118167

ANALYSIS OF LOGICAL-DYNAMIC ARGUMENT CONVERSION PROCESSES IN ARITHMETIC DEVICES OF DIGITAL CONTROL SYSTEMS

Mahmoud M. S. Al-Suod
PhD, Assistant Professor
Department of Electrical Power Engineering and Mechatronics
Tafila Technical University
New Hauway str., 179, Tafila, Jordan, 66110
E-mail: m.alsoud@ttu.edu.jo
A. Ushkarenko
PhD, Associate Professor*
E-mail: maestrotees@gmail.com
L. Petrenko
Engineer*
E-mail: levpetrovich1991@gmail.com
*Department of theoretical electrotechnics and electronic systems
Admiral Makarov National University of Shipbuilding
Heroiv Ukrainy str., 9, Mykolaiv, Ukraine, 54025

1. Introduction

One of the problems in the present-day theory of automatic control is development of new methods for formalized recording of various logical-dynamic processes of analog and digital signal conversion. This formalized recording of the signal conversion processes should be performed in a form of analytical symbols sequenced to form a functionally completed mathematical model. This model, in turn, must ensure accessibility of its informational content. At the same time, the main quality of a functionally complete mathematical model of the logical-dynamic process of conversion of signals should consist in minimization of verbal description of their content.

The current theory of automatic control can analyze control systems and synthesize laws for them. However, this is not enough because of diversity of the control processes. Therefore, a large number of scientific trends dealing with the control processes in their areas have emerged. System analysis points to the fact that such processes should be developed in accordance with certain principles. Therefore, the problem of creating a mathematical apparatus that would allow unification of the control processes of diverse physical, organizational and target nature is relevant at present.

Any formalized process includes in its recording diverse signal conversion processes, such as optical, electrochemical, and electromechanical ones. It can also be electronic processes presented as functional blocks of the control and management systems, for example, Matlab Simulink [1]. Therefore, its own formal description method has been developed for each conversion process, e. g. chemical formulas in chemistry, analytical formulas in electrical engineering or block diagrams in electronics. To solve such problems, there should be a mathematical model applicable in all control processes.

The relevance of present work was called forth by the necessity of forming scientifically substantiated analytical rules for conversion of logical arguments and the functional structures through which they are implemented. This is connected with development of a positional-sign notation allowing one to significantly speed-up adders and multipliers in digital control systems. Moreover, development of graph-analytical solutions applied to data processing in digital control systems will make it possible to analyze logical-dynamic processes of argument conversion at a due data-quality level.

Also, it is urgent to improve quality of formalized analysis of the logical-dynamic processes of argument conversion with increased information content. Evaluation of speed of arithmetic operations when using diverse digital codes can serve as the result of analysis fulfillment. This opens up the possibility of improving the methods and algorithms for data processing in digital control systems.

2. Literature review and problem statement

The problem of creating mathematical models of diverse conversion processes arises in the comprehensive study and optimization of control systems. An overview of the methods of system analysis and advantages and disadvantages of each of them in solving various optimization problems are considered in [2]. In particular, it was noted that one of the problems in optimizing systems or algorithms is the limitation of applicability of a particular method because of peculiarities of the processes occurring in the systems of diverse physical nature. Absence of a single mathematical apparatus for describing processes in the systems of diverse physical nature necessitates studies in this direction. This is also confirmed by the study [3] which noted that the analysis of the system at different levels of decomposition requires the use of different methods.

It should be noted that each of the methods of formal presentation has its advantages, or rather its structural qualities, which must be preserved and strengthened by structural qualities of other methods of formal presentation.

When designing digital devices, schematic and block diagrams are most widely used. In this case, timing charts are used for analysis of such systems as shown in [4]. Works [5, 6] pointed out necessity of speeding-up data processing. To this end, the use of formalized methods for analyzing the data conversion processes is required which would make it possible to evaluate speed of algorithms.

One of the methods for analyzing and synthesizing the processes for converting arguments in control systems is the method of forming algorithms and block diagrams. But these methods do not enable performing of these procedures at a formalized level. The process of synthesis of high-speed elements performing arithmetic operations is considered in [7]. Block diagrams are used to describe the system. In this case, there is no formal description of the processes of data conversion which makes it difficult to understand the principles of data conversion performed by the system. This is because they are not functionally completed mathematical models having an analytical form of record. In addition, they also are not graph-analytical expressions of the analyzed logical-dynamic process of converting input arguments.

Analytical and graph-analytical methods are the methods for improving quality of analysis of logical-dynamic processes with increased data content. As is shown in [8], such methods are successfully used in the design and analysis of hierarchical software systems. With their help, it is possible to analyze correctness of the processes of signal conversion in various control systems. However, the methods discussed in [8] are inapplicable to the analysis of hardware nodes in computing devices.

The problem of necessity of combining two analytically incompatible logical-dynamic processes is called in higher mathematics the boundary-value problem [9]. The boundary-value problem can be solved by a method involving development of a common structural and functional language for describing conversion processes. At the same time, recording of the conversion processes should be performed at the analytical level in a form of a uniform mathematical model.

Since the analytical record form is necessary for the subsequent formalized optimization of a concrete process, it is necessary to create the mathematical model easy to write and provide it with a maximum informational content. A method for synthesis of mathematical models of logical-dynamic process of control and the control applied to electric power industry is proposed in [10]. Study results presented in [11, 12] confirm advantage of using the developed formal methods and models in analyzing and optimizing the control systems. However, these formal methods require their further development taking into account the peculiarities of functioning and the ways of presenting information in computing devices.

Analysis of literary sources has shown an insufficient development of the methods for analyzing data conversion processes in the present-day computing devices of digital control systems. In particular, the problem of analytical description of the argument conversion processes taking place in various algorithms of programs with the aim of their optimization, in particular, speeding-up arithmetic operations remains unresolved. This is especially true for real-time control systems.

With development of microprocessor technology and its widespread use in control systems, it is necessary to use analytical information technologies for analysis and synthesis of various control systems. To do this, it is necessary to present the processes of argument conversion in the control systems in the analytical form of record and establish formalized methods of their adjustment taking into account the emerging problems.

3. The aim and objectives of the study

This study objective was analysis of the logical-dynamic processes of argument conversion using graph-analytic models, determination of their properties and the logic of argument conversion for application in development and optimization of digital control systems.

To achieve this objective, the following tasks were solved:

– perform analysis of the methods for presentation of analog and digital signals in microprocessor control systems; develop a method of graph-analytical form of recording digital codes representing information arguments of voltage;

– develop functionally completed models of logical-dynamic processes of conversion of information arguments and determine their properties;

– perform analysis of logical-dynamic processes of conversion of information arguments in adders and multipliers of digital control systems and determine logic of the argument conversion.

4. Development of the procedure for the logical-dynamic process of argument conversion in digital control systems

4. 1. Transition from the graphical form of signal recording to their analytical representation

A symbol must be selected as a generalized functional structure for the mathematical model of any process of the

logical-dynamic signal conversion. A fundamental expression of mathematical analysis [9] should be used as such symbol for synthesis of specific functional structures:

$$y = f(x). \quad (1)$$

If expression (1) is considered as a process of argument conversion, then y is the product of certain operations $f(\rightarrow)$ with the input argument x, then this is the final result of these operations. Accordingly, taking into account the elements of the functional structure which are an integral temporal parameter, there are reasons to write this expression in the form:

$$f(x) \rightarrow y. \quad (2)$$

The corrected generalized process of argument conversion (2) can be analyzed taking into account coherence of recording of the block diagrams of the computing devices and the control systems. As a whole, this expression can be represented as a process of conversion of the argument x to the argument y.

The proposed method of analytical recording of the argument conversion process is more in line with the block diagrams in which functions are represented as logical elements that are electronic systems.

Therefore, if a mathematical sign of the system (}) is used instead of the geometric-figure logical element, then, e. g. analytical record for the element AND can be presented as shown in Fig. 1.

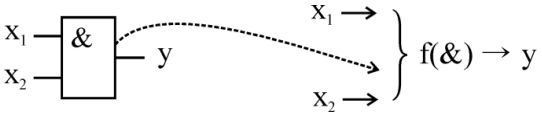

Fig. 1. Analytical presentation of the logical element AND: x_1 and x_2 are input arguments, $f(\&)$ is the logical function AND

The method of analytical recording (2) of the argument conversion process can be used for analytical recording of the functionally completed structures in which functionally completed elements are represented as functional blocks.

For example, analytical record of the functional blocks of an analog-to-digital and digital-to-analog conversion has the form:

$$U_{in} \rightarrow U_{in}, [m_i] \rightarrow [m_i],$$

$$[m_i] \rightarrow [m_i], U_j \rightarrow U_{out}. \quad (3)$$

An example of a procedure for argument conversion for an analog harmonic signal is given in Fig. 2.

The advantage of the graph-analytic solutions is that they enable analysis of the logical-dynamic conversion processes at an information-quality level.

This is true for both the voltage arguments $^{\pm}\mathbf{U}_m\sin(\omega t+\varphi 0)$ and information voltage arguments $^{\pm}\mathbf{U}_m f((t) \rightarrow {}^{+}[n_i] f(2^n))$.

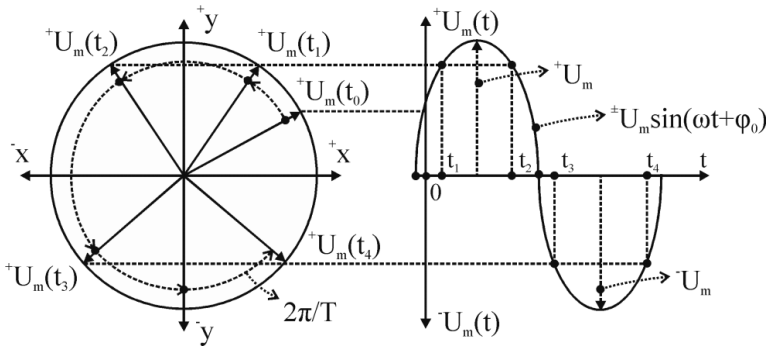

Fig. 2. The argument conversion procedure

4.2. The forms of representation of numbers in digital control systems and the procedure for conversion of arguments

In the present-day microprocessor control systems, numbers written in the binary code can be represented in a graph-analytical form (Fig. 3):

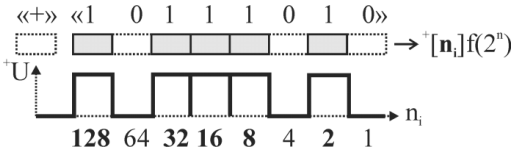

Fig. 3. Representation of numbers in the binary code

The graph-analytical form of recording arguments of voltage $^{\pm}\mathbf{U}_m\sin(\omega t+\varphi_0)$ is widely used due to the high information content. The graph-analytical form for recording information arguments of voltage $^{\pm}\mathbf{U}_m f((t) \rightarrow {}^{+}[\mathbf{n}_i] f(2^n))$ shown in Fig. 3 has no such spread.

At the same time, such a record has great potential. This can be demonstrated by analysis of the logical-dynamic process of conversion of information arguments "–/+"$[\mathbf{m}_j]f(+/-)$: – "Additional code" in adders $f(\Sigma)$ and multipliers $f_{\Sigma}(\Sigma)$. It is these elements that are the computing core of the regulators in the current digital control systems.

If we write down the logical-dynamic process of converting arguments of partial products, for example, for positive arguments of multipliers "–/+"$[\mathbf{m}_j]f(+/-)\rightarrow\rightarrow$"+1""1011"$\rightarrow$"+**5"**$f(10)$ and "–/+"$[\mathbf{m}_j]f(+/-)\rightarrow\ \rightarrow$"+1""1101"$\rightarrow$"+**3"**f(10) in the form presented in Fig. 4, then the logic of formation of partial products is arithmetically correct.

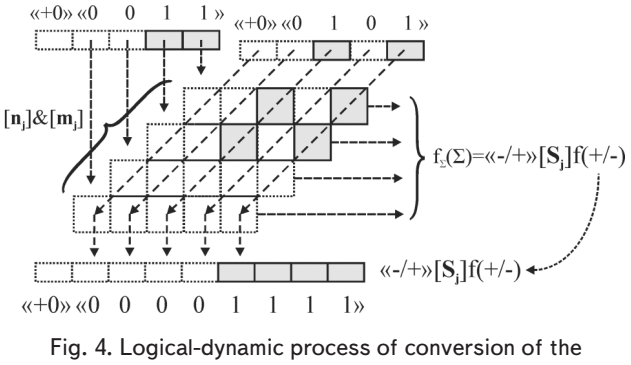

Fig. 4. Logical-dynamic process of conversion of the arguments of partial products for positive arguments

However, if we record the logical-dynamic process of conversion of the arguments of partial products, e. g. for conditionally negative arguments of the multiplicand "–/+"$[\mathbf{m}_j]f(+/-)$→"–1""1011"→"–**5"***f*(10) and the multiplier "–/+"$[\mathbf{m}_j]f(+/-)$→"–1""1101"→"–**3"***f*(10) in (3) as shown in Fig. 5, then, without applying arithmetic axioms of the ternary notation *f*(+1, 0, –1), it is not clear how the resulting sums $[\mathbf{S}_j]_1$, $[\mathbf{S}_j]_1$, $[\mathbf{S}_j]_1$ and "–/+"$[\mathbf{S}_j]f(+/-)$ were formed in the functional structures of the adders $f_1(\Sigma) - f_4(\Sigma)$.

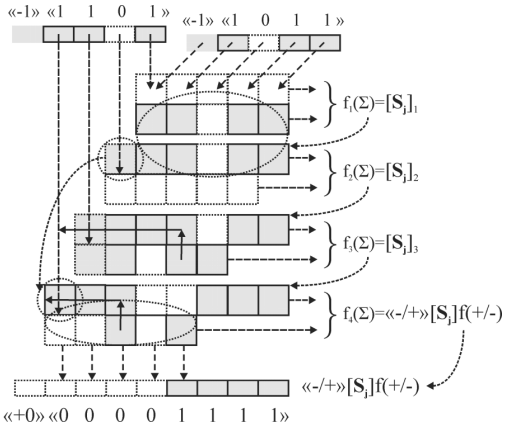

Fig. 5. Logical-dynamic process of conversion of the arguments of partial products for negative arguments

The procedure for converting the structure of the arguments of multiplicand "–/+"$[\mathbf{m}_j]f(+/-)$→"–1""1011"→"–**5"***f*(10) in the functional structure of the adder $f_1(\Sigma)$ can be written as graph-analytical expressions presented in Fig. 5, 6. It follows from their analysis that the resulting sum $^{\pm}[\mathbf{S}_j]_1$ was formed as a result of either application of the arithmetical axiom "±0"→"+1/–1" or by means of the arithmetic axiom "–1"→"–2""+1". This axiom is not only the arithmetic basis of forming the structure of arguments of "–/+"$[\mathbf{m}_j]f(+/-)$ – "Additional code" but also the procedure of summation of conditionally negative arguments.

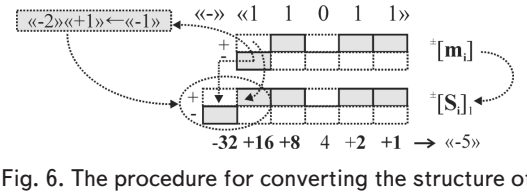

Fig. 6. The procedure for converting the structure of arguments

It is necessary to perform a logical-dynamic process of summation of the conditionally negative argument of the summand "–/+"$[\mathbf{m}_j]f(+/-)$→"–1""10001101"→"–**69"** *f*(10) of expression (Fig. 7) and the conditionally negative argument of the summand "–/+"$[\mathbf{m}_j]f(+/-)$→"–1""1011"→"–**5"***f*(10) of expression (Fig. 8). At the preliminary stage, a functional positional superposition of sign bits of the summands of expression (Fig. 7) and expression (Fig. 8) is performed. This is done using the arithmetic axiom "–1"→"–2|""+1" which can be written as expression (Fig. 9).

Only after executing the procedure of functional superposition of the sign bit $\mathbf{m}_{\pm}$, the process of summing the arguments of the summands "–/+"$[\mathbf{m}_j]f(+/-)$→"–1""10001101""→"–**69"** *f*(10) and "–/+"$[\mathbf{n}_j]f(+/-)$→"–1""1011"→"–**5"***f*(10) can be executed in the "Additional code". It should be noted that such an action only leads to an increase in the technological cycle $\Delta\mathbf{t}_{\Sigma}$ of conversion of the arguments of summands. If we form a procedure for conversion of the argument of summand "–/+"$[\mathbf{m}_j]f(+/-)$→"–**69"** «–/+» [mj] f (+/–) → → –69 and the argument of summand "–/+"$[\mathbf{n}_j]f(+/-)$→"–**5"**, then the process of their summation can be written in the form of a graph-analytic expression (Fig. 10). After forming the first and the second intermediate sums $^{\pm}[S_j^1]f(\})$–OR and $^{\pm}[S_j^2]f(\&)$–AND, it is necessary to perform the procedure for logical differentiation of the structure of the arguments of the first intermediate sum $^{\pm}[S_j^1]f(\})$–OR. After this, the corrected structure of the derivative arguments $^{\pm}[S_j^1]d/dn^{+}$ can be formed.

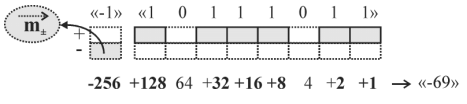

Fig. 7. The negative argument of the summand «–69»

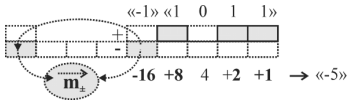

Fig. 8. The negative argument of the summand «–5»

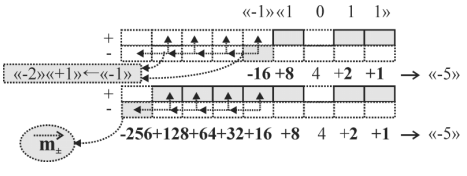

Fig. 9. Functional positional superposition of sign bits of the summands

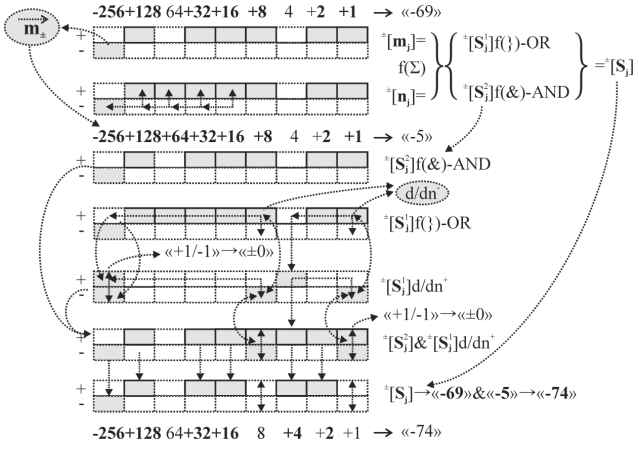

Fig. 10. The process of summing the arguments

Then, after the functional unification $^{\pm}[S_j^2]\&^{\pm}[S_j^1]d/dn^{+}$ of the structure of the arguments of the second intermediate sum $[S_j^2]^{\pm}f(\&)$–AND and the corrected structure of the arguments of the first intermediate sum $^{\pm}[S_j^1]d/dn^{+}$, the resultant structure of the arguments of the sum $^{\pm}[S_j]$→»–/+»$[S_j]f(+/-)$ – «Additional code» is formed by changing activity of the logical zeros "+1/–1"→"±0". However, the correct finite resultant structure of the arguments of the sum $^{\pm}[\mathbf{S}_j]$→"–**74"** can also be formed without the positional superposition of the active sign argument $\mathbf{m}_{\pm}$ of two structures of the arguments of summands "–/+"$[\mathbf{m}_j]f(+/-)$→"–**69"** and "–/+"$[\mathbf{n}_j]f(+/-)$→"–**5"**. To do this, it is necessary to write down the logical-dynamic process of argument conversion

in the form of a graph-analytic expression (Fig. 11). The first intermediate sum $^{\pm}[S_j^1]f(\})$–OR includes two conditionally negative arguments of the sign bit of two summands with an information content **m±→"–256"** and $\mathbf{m}_{\pm}$**→"–16"**.

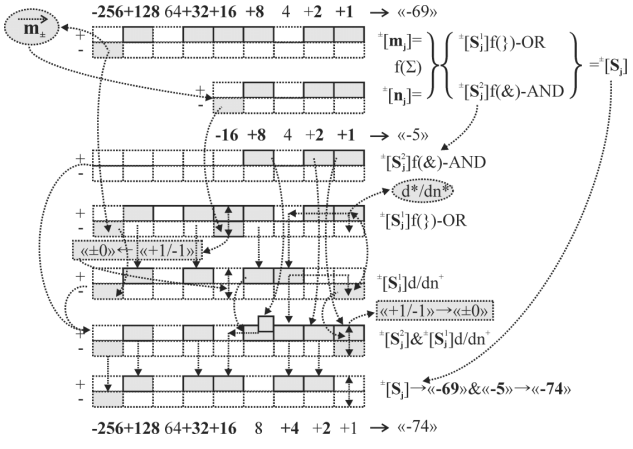

Fig. 11. The logical-dynamic process of argument conversion

In this case, the argument of the sign bit $\mathbf{m}_{\pm}$**→"–16"** in the structure of the first intermediate sum $^{\pm}[S_j^1]f(\})$–OR forms an active logical zero "+1/–1"→"±0" which is converted to a non-active argument «±**0**». Formation of the resulting sum $^{\pm}[\mathbf{S}_j]$**→"–74"** can be realized by means of selective logical differentiation d^*/dn^+ of the structure of positive arguments of the first intermediate sum $^{\pm}[S_j^1]f(\})$–OR. It should be noted that the number of possible variants of the summand argument conversion is not limited to the graph-analytical expression (Fig. 11). If we take in account the possible structures of the positionally-sign structures of the arguments of the first variant "–/+"$[\mathbf{m}_j]f(+/-)$ – «Additional code» and its second variant, then we can assert that the notation «Additional code» is not perfect and requires additional analysis and subsequent adjustment. If we return to the graph-analytic expression of the procedure for formation of partial products in the multiplier $f_{\Sigma}(\Sigma)$, then it can be written as a corrected graph-analytical expression (Fig. 12). The first partial product $[\mathbf{S}_j]_1$ is formed with a preliminary direct positional shift of the sign argument $\mathbf{m}_{\pm}$ of the structure of the multiplicand $^{\pm}[\mathbf{m}_j]$ in accordance with the direct arithmetic axiom «-1»→»–2»«+1».

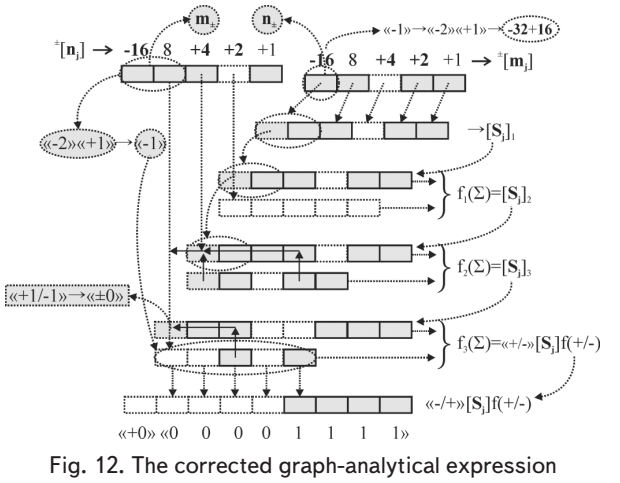

Fig. 12. The corrected graph-analytical expression

The last partial product in the system of the functional structure of the adder $f_3(\Sigma)$ is formed with a preliminary positional inverse shift of the sign argument $\mathbf{n}_{\pm}$ of the structure of the multiplier $^{\pm}[\mathbf{n}_j]$. The mentioned procedure is performed in accordance with the inverse arithmetic axiom «–1»«+1»→»–2». This leads to a minimization of the multiplier structure in accordance with the graph-analytical expression (Fig. 13).

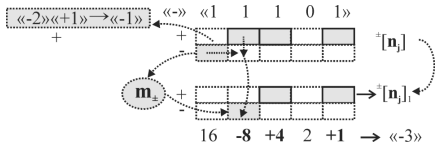

Fig. 13. Minimization of the multiplier structure

In the structure of the multiplier $^{\pm}[\mathbf{n}_j]$, the argument of the sign $\mathbf{m}_{\pm}$ is formed with a conditionally negative content. In this situation, the multiplicand $^{\pm}[\mathbf{m}_j]$, like the structure of the partial product $^{\pm}[\underline{\mathbf{m}}_j]$, must be written as a structure with altered levels of analog signals. This is done in accordance with the procedure (Fig. 14) in which the structure of arguments $^{\pm}[\underline{\mathbf{m}}_j]$ – «Additional code» is formed with a positive informational content.

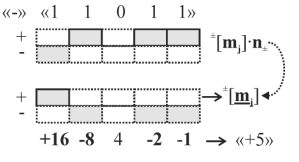

Fig. 14. Record of a structure with altered levels of analog signals

However, since the adder $f_3(\Sigma)$ implements transformations of only positive arguments, therefore the structure of arguments $^{\pm}[\underline{\mathbf{m}}_j]$ of the expression (Fig. 14) needs to be corrected in accordance with the procedure of inverse conversion (Fig. 15).

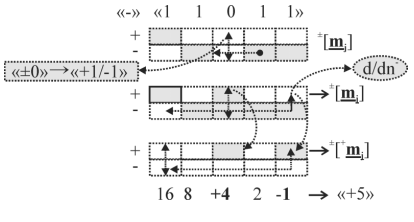

Fig. 15. The procedure of inverse conversion

The inverse conversion procedure (Fig. 15) is performed by activating the logical zero arguments "±0"→»+1/–1». Next, the logical differentiation procedure d/dn^- of the structure of the conditionally negative arguments is performed with removal of the active logical zero in the high-order bit. The process of forming a positive structure of arguments is shown in Fig. 16.

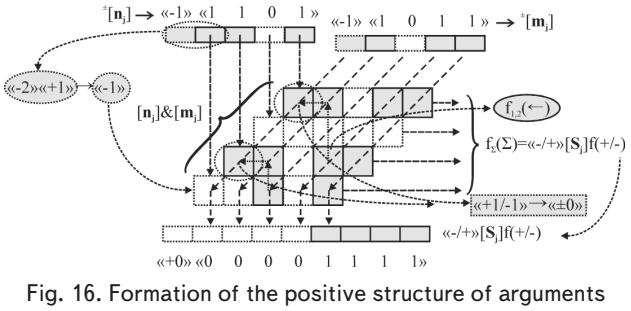

Fig. 16. Formation of the positive structure of arguments

As a result of above conversions, the positive structure of the arguments $^{\pm}[^{+}\underline{\mathbf{m}}_j]\rightarrow$"+**5**" of the last partial product in the system of the functional structure of the adder $f_3(\Sigma)$ of the expression is formed.

5. The result of the logical-dynamic process of converting arguments in a digital control system

From the analysis of the graph-analytical process of argument conversion in the functional structure of the combinatorial multiplier $f_\Sigma(\Sigma)$ (Fig. 12), the following conclusions can be drawn. Multiplication involves a procedure for increasing the number of active arguments in the higher bits of the first partial products. At the final stage of formation of the structure of the arguments of the last partial product, a procedure is performed to minimize them. If the technological cycle $\Delta\mathbf{t}_\Sigma$ of conversion of partial products increases in the first situation, it decreases in the second situation. The minimized technological cycle $\Delta\mathbf{t}_\Sigma$ of conversion of partial products is the basic parameter of the functional structure of the multiplier $f_\Sigma(\Sigma)$. Therefore, increase in capacity of the partial products at the first stages of their formation is inappropriate. The procedure for forming arguments of partial products can be corrected and written without increasing capacity in a form of a graph-analytical expression (Fig. 16). To do this, it is necessary to enter the procedure for deleting active logical zeros «+1/–1»→»±0» into the functional structure of the multiplier $f_\Sigma(\Sigma)$ when summing them. Active logical zeros are formed as a result of local transfers $f_{1,2}(\leftarrow)$ of a positive argument from the previous bit. A similar minimization of the structure of the argument of the multiplicand $^{\pm}[\mathbf{m}_j]$ and the procedure for formation of partial products in a form of a graph-analytical expression are presented in Fig. 17.

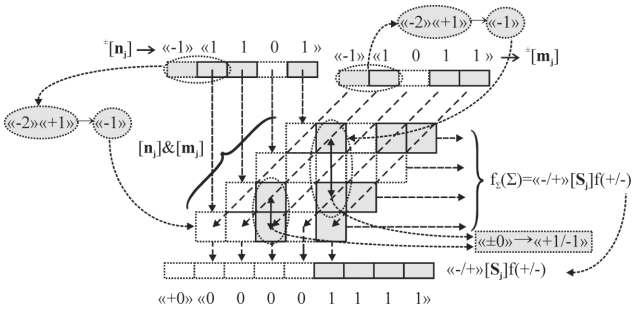

Fig. 17. Formation of partial products in a form of a graph-analytical expression

In this case, only the procedure for removing active logical zeros in the structure of partial products of the multiplier $f_\Sigma(\Sigma)$ must be performed.

6. Discussion of the results of the logical-dynamic process of argument conversion in digital control systems

The work presented logical-dynamic processes of forming the structure of the arguments of partial products and their conversion in expressions. As it follows from the comparative analysis, the structures of the arguments of multipliers «–/+»$[\mathbf{m}_j]f(+/-)$ and «–/+»$[\mathbf{n}_j]f(+/-)$ – «Additional code» in their original state are not optimal to perform the procedure for their conversion in the functional structure of the multiplier $f_\Sigma(\Sigma)$.

The considered graph-analytical method for analyzing logical-dynamic processes of argument conversion supplements the studies carried out earlier by the authors in the field of modeling, analysis and structural optimization of automation systems [10–12]. The proposed approach is applicable to analysis of the functional structure of the computing microcontroller core (Core^{MK}) of digital control systems. It was implemented in a form of computational mathematical models with input and converted arguments. This allows one to perform system analysis at so many decomposition levels as one needs to generate an idea of its basic properties.

Introduction of an analytical form for writing logical functions with input and converted arguments greatly simplifies analysis and synthesis of any logical-dynamic processes of argument conversion. The advantage of the proposed approach is that when the analytical form of the record of the functional structure of the adder $f(\Sigma)$ is introduced, it is possible to write down the logical-dynamic process of argument conversion in a form of a generalized expression. The graph-analytical form of the record makes it possible to supplement it with logical content by means of directed vectors. On the one hand, it enables displaying of the logical-dynamic process of argument conversion at an extremely minimized level of formalization. On the other hand, this form of record enables comparative analysis of various variants of logical procedures for a subsequent selection of the most optimal one and formation of mathematical models at an analytical level.

Thus, analysis of existing procedures for converting arguments has shown that they are not minimized and are not arithmetic axioms of conversion of logical arguments. Therefore, these rules must be corrected and reduced to arithmetic axioms, both for summing arguments and their subtraction. It also follows from the obtained results that the ternary state of the arguments (+1, 0, –1) allows one to formulate a procedure for conversion of arguments in the structure of the adder $f(\Sigma)$ at an arithmetically correct level.

7. Conclusions

1. A method of recording information voltage arguments and the processes of argument conversion in a graph-analytical form was proposed. The distinctive feature of this method is minimization of the verbal description.

2. Mathematical models of the process of argument conversion which represent formalized recording of the processes of signal conversion were developed. The main quality of a functionally complete mathematical model of the logical-dynamic process of signal conversion is its enhanced information content. It was shown that the arithmetical axioms «+1»→«+2»+«–1» or «–1»→«–2»+«+1» and «+1»+«–1»→«0» form the logical basis of conversion of the arguments of the ternary notation $f(+1, 0, -1)$. The arithmetical axioms of the ternary notation form the theoretical basis for the process of summation of arguments implemented in the binary notation format.

3. Analysis of the logical-dynamic processes of argument conversion that occur in adders and multipliers of the digital control systems was performed with application of the developed mathematical models. It was established that the process of summing logical arguments in arithmetic

devices, regardless of the numerical system, is implemented in accordance with the logic of conversion of the arguments of the ternary notation $f(+1, 0, -1)$. The proposed approach makes it possible to evaluate speed of performed arithmetic operations with the use of various digital codes and choose the most optimal one. It also opens up the possibility of improving the methods and algorithms for data processing in digital control systems.

References

1. Le-Huy, H. Modeling and simulation of electrical drives using MATLAB/Simulink and Power System Blockset [Text] / H. Le-Huy // IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.37243). – 2001. doi: 10.1109/iecon.2001.975530
2. Liu, Y. A Survey on Particle Swarm Optimization Algorithms for Multimodal Function Optimization [Text] / Y. Liu, X. Ling, Z. Shi, M. Lv, J. Fang, L. Zhang // Journal of Software. – 2011. – Vol. 6, Issue 12. doi: 10.4304/jsw.6.12.2449-2455
3. Zhang, Y. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications [Text] / Y. Zhang, S. Wang, G. Ji // Mathematical Problems in Engineering. – 2015. – Vol. 2015. – P. 1–38. doi: 10.1155/2015/931256
4. Rajasekhar, K. Design and Analysis of Comparator Using Different Logic Style of Full Adder [Text] / K. Rajasekhar, P. Sowjanya, V. Umakiranmai, R. Harish, M. Krishna // Int. Journal of Engineering Research and Applications. – 2014. – Vol. 4, Issue 4. – P. 389–393.
5. Asaduzzaman, A. GPU Computing to Improve Game Engine Performance [Text] / A. Asaduzzaman, H. Y. Lee // Journal of Engineering and Technological Sciences. – 2014. – Vol. 46, Issue 2. – P. 226–243. doi: 10.5614/j.eng.technol.sci.2014.46.2.8
6. Igić, T. S. Design of a System for Monitoring Reliability of Structures and Constructions in Civil Engineering [Text] / T. S. Igić, N. Ž. Veljković // International Journal of Engineering Pedagogy (iJEP). – 2011. – Vol. 1, Issue 2. doi: 10.3991/ijep.v1i2.1634
7. Murthy, G. R. A Novel Design of Multiplexer Based Full-Adder Cell for Power and Propagation Delay Optimizations [Text] / G. R. Murthy, C. Senthilpari, P. Velrajkumar, L. T. Sze // Journal of Engineering Science and Technology. – 2013. – Vol. 8, Issue 6. – P. 764–777.
8. Aminof, B. Synthesis of hierarchical systems [Text] / B. Aminof, F. Mogavero, A. Murano // Formal Aspects of Component Software. – 2011. – P. 42–60. doi: 10.1007/978-3-642-35743-5_4
9. Wang, Y. Concept Algebra: A Denotational Mathematics for Formal Knowledge Representation and Cognitive Robot Learning [Text] / Y. Wang // Journal of Advanced Mathematics and Applications. – 2015. – Vol. 4, Issue 1. – P. 61–86. doi: 10.1166/jama.2015.1074
10. Ryabenkiy, V. M. Metod sintezu matematychnih modelei logiko-dynamichnyh protsesiv kontrolyu i keruvanya [Text] / V. M. Ryabenkiy, A. O. Ushkarenko // Tehnichna elektrodynamika. – 2011. – Issue 2 (1). – P. 121–125.
11. Ryabenkiy, V. M. Formalnoye opisanie elementov avtomatizirovannogo rabochego mesta operatora electroenergeticheskoi sistemy [Text] / V. M. Ryabenkiy, A. O. Ushkarenko // Naukivyi vistnyk Khersonskiy derzhavniy morskiy akademiyi. – 2014. – Issue 1 (1). – P. 43–50.
12. Mahmoud, M. A. S. Analytical Representation of Control Processes of Induction Motor and Synchronous Generator in Power Plants [Text] / A. S. M. Mahmoud, O. O. Ushkarenko // Jordan Journal of Electrical Engineering. – 2016. – Vol. 2, Issue 4. – P. 278–288.