



УДК 004.383.8.032.26

© 2015 г. **В.А. Романчук**, канд. техн. наук
(Рязанский государственный университет им. С.А. Есенина)

РАЗРАБОТКА АЛГОРИТМИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НЕЙРОКОМПЬЮТЕРНОГО МОДУЛЯ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ ЭЛЕКТРОМЕХАНИЧЕСКИМИ СИСТЕМАМИ

Рассматриваются вопросы разработки алгоритмического и программного обеспечения специализированного вычислительного устройства на базе нейропроцессоров для автоматического управления модулями электромеханических систем на примере гексапода в режиме, приближенном к реальному времени. Практические исследования рассматриваются на примере использования нейропроцессора NM6406 на базе инструментального модуля МС 51.03 и микрокомпьютера МВ 77.07, разрабатываемых НТЦ "Модуль".

Ключевые слова: нейропроцессорная вычислительная система, гексапод, управление, матричные вычисления.

Введение

Задача управления динамическим объектом предполагает наличие оперативного и неоперативного режимов, обусловленных жесткими требованиями обработки большого количества информации в высоком темпе. По существу, ни одна из современных управляющих вычислительных машин не может обеспечить обработку информации, поступающей с датчиков объекта управления в заданном темпе, поэтому приходится распараллеливать вычислительные операции в подключаемых дополнительных специализированных вычислительных устройствах (СВУ). Такое разделение функций позволяет обеспечить нестационарный режим без нарушения темпа основного вычислительного процесса. В реальном масштабе времени должны решаться две основные задачи: вычисление управляющего воздействия и интегрирование системы дифференциальных уравнений. Сведение системы дифференциальных уравнений к системе алгебраических уравнений в векторно-матричной форме удобно для ее реализации средствами цифровой вычислительной техники в СВУ, в процессорном блоке которого должны выполняться операции «умножения с накоплением» одновременно над несколькими операндами [1].

Поэтому для реализации математического, алгоритмического и программного обеспечения предлагается использовать специализированную аппаратную

базу: нейропроцессорные устройства на базе процессоров особого класса, 6-го поколения, широко используемые в задачах обработки информации. Выбор нейропроцессорных устройств предопределен рядом преимуществ использования, таких как скорость параллельной обработки, наличие операции "сложение с накоплением", высокая энергоэффективность [2]. В качестве примера было выбрано семейство процессоров NeuroMatrix 640x, нейропроцессоры которого выпускаются научно-техническим центром "Модуль" (Россия, Москва) [3].

Исходными данными для разработки алгоритмического и программного обеспечения являются:

описание цифровых сигналов датчиков;

описание алгоритмов формирования управляющих сигналов;

математическая модель, реализованная в виде модели модуля SIMULINK программного комплекса MATLAB.

Исходными данными разработанных алгоритмов являются:

описание сигналов датчиков, полученных посредством преобразования аналоговых сигналов с гексапода в цифровой вид с помощью аналогово-цифрового преобразователя.

Выходными данными разработанных алгоритмов являются:

управляющие сигналы, преобразуемые в аналоговое представление посредством использования цифро-аналогового преобразователя;

информационные сигналы вспомогательного и служебного характера.

Математическая формализация задачи работы СВУ

Рассмотрим схему работы СВУ, которая представлена на рис. 1, где $\hat{\Phi}$, $\tilde{\Phi}$, \hat{S} , \tilde{S} , \tilde{G} , $\tilde{\Phi}$ – рабочие матрицы, полученные из матриц параметров гексапода и матриц параметров корректирующих фильтров; P' , P – матрицы параметров корректирующих связей (6x100); $\alpha^*(t)$ – вектор-функция программы (18x1); $Y(t)$ – вектор-функция наблюдения (сигналы, поступающие от измерительной системы) (30x1); $X(t)$, $X^*(t)$ – вектор-функция текущего и программного состояния системы на i -м шаге управления (100x1); $X(t+h)$, $X^*(t+h)$ – то же на $(i+1)$ -м шаге (100x1); $U(t)$ и $U(t+h)$ – вектор управляющих воздействий на i -м и $(i+1)$ -м шагах управления (30x1); h – шаг интегрирования (~0.001); θ – значение отклонения фокальной оси гексапода от линии визирования объекта (3x1); C^* , R^* , R – постоянные матрицы размерностью (3x3, 3x100, 3x100).

Целью функционирования блока является вычисление оптимальных параметров группового регулятора и вычисления коэффициентов (k_1, k_2, k_3) ПИД регуляторов двигателей ног гексапода, исключая заклинивания и обеспечивающих плавность перемещения платформы гексапода.

Пусть $A^{(j)} \in A$ – алгоритм функционирования СВУ, который относится в классу алгоритмов $A = \langle A_1, A_2, \dots, A_n, \dots, A_N \rangle$, предназначенных для управления сложными объектами. Введем множество операций $O = \{O_1, O_2, \dots, O_p, \dots, O_P\}$. С учетом введенного множества операций этот алгоритм обработки информации

представляет собой кортеж, состоящий из операций $O_1, O_2, \dots, O_m, \dots, O_M$ длиной $l_j = |A^{(j)}|; j = \overline{1, N}$:

$$A^{(j)} = \langle O_1, O_2, \dots, O_m, \dots, O_M \rangle; j = \overline{1, N}. \quad (1)$$

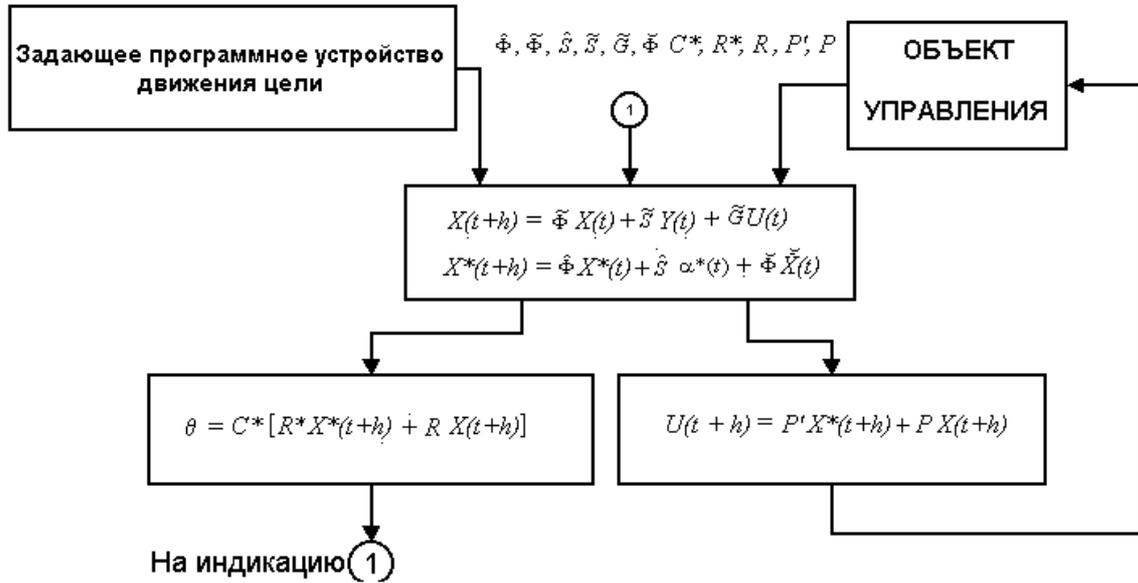


Рис. 1. Алгоритм функционирования СВУ.

Каждая m -я операция нейропроцессора, за исключением команд управления O_m , представляет собой математическую модель формального нейрона:

$$O_m = f\left(\sum_{m=1}^n a_m x_m + a_0\right). \quad (2)$$

На этапе разработки программы каждой операции ставится в соответствие микрокоманда процессора $MK_p; \forall p = \overline{1, K_i}$:

$$\forall O_l \in O: O_m \rightarrow \{MK_p\}; \forall p = \overline{1, K_i}; \forall m = \overline{1, M}, \quad (3)$$

где $MK = \{MK_1, MK_2, \dots, MK_i, \dots, MK_I\}$ – множество микрокоманд, написанных на внутреннем языке процессора; K_i – минимальное количество микрокоманд, необходимое для реализации операции O_m .

В зависимости от решения задачи (3) на следующем этапе необходимо каждому j -му алгоритму обработки поставить в соответствие некоторую программу $PR^{(j)}$, т.е. требуется определить некоторое отображение φ :

$$\varphi: A^{(j)} \rightarrow PR^{(j)}, j = \overline{1, N}. \quad (4)$$

Под программой $PR^{(j)}$ понимается кортеж микрокоманд:

$$PR^{(j)} = \langle MK_1, MK_2, \dots, MK_i, \dots, MK_I \rangle. \quad (5)$$

Таким образом, исходя из формулы (3), можно сказать, что в основе нейропроцессорного блока лежит эмуляция формального нейрона, который математически представляет собой совокупность операций сложения и умножения, т.е. "умножение с накоплением".

Из алгоритма управления гексаподом следует, что основной в СВУ является

процедура умножения матрицы на вектор:

$$\Phi \times X = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \dots & \Phi_{1n} \\ \Phi_{21} & \Phi_{22} & \dots & \Phi_{2n} \\ - & - & - & - \\ \Phi_{m1} & \Phi_{m2} & \dots & \Phi_{mn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \Phi_{11}x_1 + \Phi_{12}x_2 + \dots + \Phi_{1n}x_n \\ \Phi_{21}x_1 + \Phi_{22}x_2 + \dots + \Phi_{2n}x_n \\ - & - & - & - \\ \Phi_{m1}x_1 + \Phi_{m2}x_2 + \dots + \Phi_{mn}x_n \end{bmatrix}. \quad (6)$$

Одна компонента вектора результата получается суммированием произведения элементов строки матрицы на соответствующие элементы вектора-столбца. Эта последовательность операции повторяется столько раз, сколько строк в матрице. Для вычисления произведения матрицы на вектор необходимо выполнять $T_{\text{УВМ}} = m \times n$ операций умножения, где m – число строк матрицы; n – число столбцов и $N_{\text{СМВ}} = m \times (n - 1)$ операций сложения, или около 3000 пар операций "умножение-сложение". Таким образом, модель формального нейрона (2) может быть использована для реализации задачи управления, в частности операции умножения матрицы на столбец (6).

Разработка алгоритмического обеспечения работы СВУ

Так как практические исследования производились с использованием нейропроцессора NM6406 на базе инструментального модуля МС 51.03 и микрокомпьютера МВ 77.07, то необходимо сказать, что эффективное распараллеливание вычислений достигается за счет аппаратной поддержки операции векторно-матричного умножения на нейропроцессоре NeuroMatrix NM6406. Все арифметические вычисления, относящиеся непосредственно к вычислению управляющих воздействий, производятся на векторном сопроцессоре. Так как векторный узел позволяет оперировать данными переменной разрядности, то для хранения входных данных и результатов вычислений удобно отводить по 32 разряда на целую и вещественную часть.

Для разработки алгоритмов и программного кода был реализован ряд подпрограмм функциональных элементов математической модели специализированного вычислительного устройства (СВУ) управления гексаподом [7].

1. Матричный умножитель и матричный умножитель с накоплением.

Дифференциальные уравнения, описывающие динамику гексапода, могут быть представлены в виде разностных в векторно-матричной форме, что удобно для реализации средствами вычислительной техники. Поэтому основная операция, которую должен выполнять СВУ, – перемножение матриц, которые могут быть разбиты на пару элементарных операций «умножение-сложение» (умножение с накоплением). Для этой операции используется векторный сопроцессор, представляющий собой общий матричный узел для выполнения операций умножения с накоплением арифметических и логических операций, маскирования, функций активации над векторами и матрицами.

Это основной функциональный элемент с набором регистров общего назначения. Представляет собой матричную структуру 64x64, произвольно разделенную на столбцы и строки.

Структурная схема векторного сопроцессора показана на рис. 2.

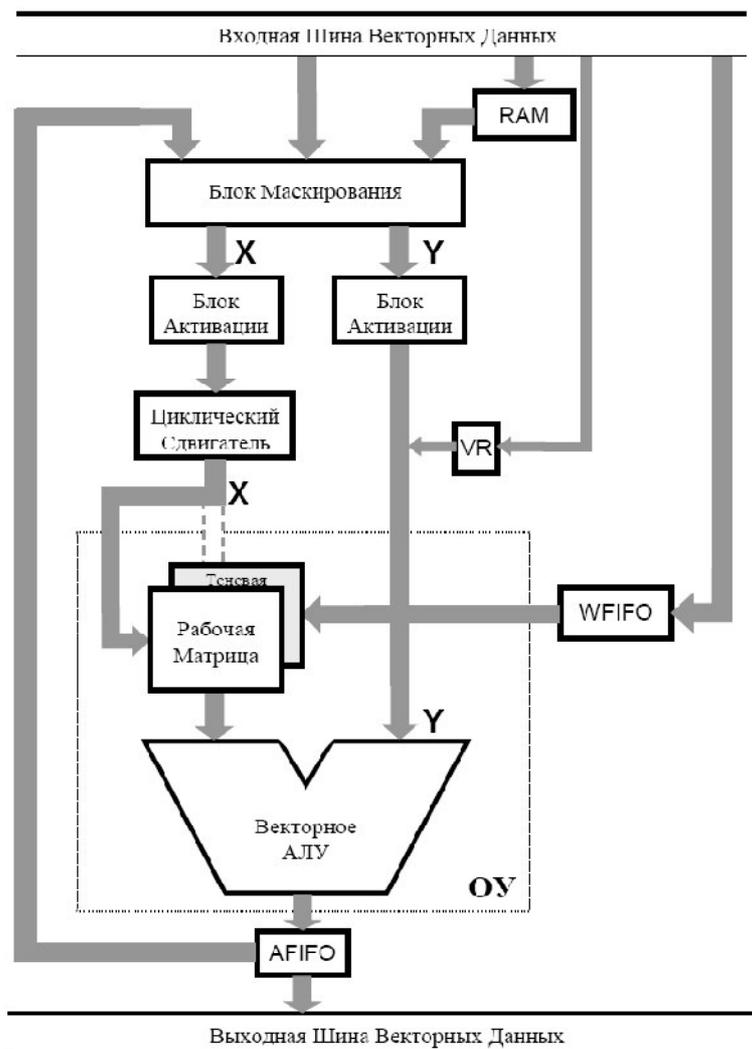


Рис. 2. Структура векторного сопроцессора.

Операция умножения с накоплением выполняется в рабочей матрице нейропроцессора: $Z_i = Y_i + \sum(X_i W_{ij})$; $i = \overline{1, m}$; $j = \overline{1, n}$, где X_i – элемент данных, поступающих на вход векторного сопроцессора; Y_j – частичная сумма, накопленная на предыдущем этапе взвешенного суммирования или остаток после предыдущей операции; Y_0 – элемент входного вектора данных; W_{ij} – весовой коэффициент, расположенный в соответствующей ячейке рабочей матрицы процессора; m – количество столбцов рабочей матрицы процессора; n – количество строк рабочей матрицы процессора. Рабочая матрица имеет входы: $X = \{x_1, x_2, \dots, x_n\}$ – 64 разряда и $Y = \{y_1, y_2, \dots, y_m\}$.

Данные, полученные на входе X , умножаются на ячейки матрицы, после чего суммируются в пределах столбца, т.е. в случае переполнения происходит потеря знаковых разрядов. Данные со входа Y поэлементно складываются с результатами умножения, выполненным над данными входа X .

Для выполнения операции взвешенного суммирования в рабочую матрицу предварительно загружаются весовые коэффициенты W_{ij} . Это возможно двумя способами – либо непосредственно в рабочую матрицу, либо через теневую матрицу. Загрузка весовых коэффициентов через теневую матрицу наиболее эффек-

тивна, так как реализуется за один такт нейропроцессора. Теневая матрица существует в векторном узле нейропроцессора для ускорения работы, так как весовые коэффициенты загружаются в фоновом режиме за 32 такта. Перегрузка весовых коэффициентов из теневой в рабочую матрицу происходит за один такт.

Разбиение матрицы на строки определяется рабочим регистром $sb2$. Он определяет предварительное разбиение 64-разрядных слов входных данных, поступающих на вход Y . Разбиение рабочей матрицы на столбцы задается рабочим регистром $nb2$. Этот же регистр определяет разбиение 64-разрядных данных на входе Y и также предварительно заносится в виде слова данных в регистр. Этот же регистр определяет разрядность результатов вычисления, которые в итоге будут расположены в буферном регистре AFIFO.

Таким образом, входные данные (операнды) и выходные значения упаковываются путем разбиения в 64-разрядные слова. Все операции в матрице производятся параллельно за один такт.

2. Сумматор двух и более операндов.

Сумматор двух и более операндов реализован с использованием векторного блока процессора и операции взвешенного суммирования (рис. 1) с пустым оператором умножения, то есть с "единичной" теневой и рабочей матрицей (все биты 64×32 заполнены единичными битами). Эффективность использования этой операции зависит от разрядности входных данных, так как за счет использования разбиения матрицы на строки и столбцы увеличивается скорость вычислений в связи с одновременной работой с несколькими операндами. Важной особенностью векторного процессора является реализация работы с операндами различной длины.

3. Функции активации (насыщения).

Функции активации реализованы с использованием векторного блока процессора и операции взвешенного суммирования (рис. 1). Реализованы подпрограммы двух типов функций активации: пороговая функция и функция насыщения. В блоках активации осуществляются вычисления над упакованными словами данных. Блоки активации позволяют применять функции активации ко всем элементам упакованного слова одновременно. Основную роль в управлении функциями активации играют регистры $f1cr$ и $f2cr$. Блоки активации размещаются между устройством маскирования и рабочей матрицей или векторным вычислителем. Функции активации могут быть подвергнуты либо данным, поступающим на вход X , либо на Y , либо на оба входа сразу.

Все элементарные базовые операторы реализованы в виде макросов на языке нейроассемблера.

Разработка программного обеспечения работы СВУ

Для удобной работы с различными исходными данными была реализована модель задачи автоматического управления модулями электромеханических систем в разработанной подсистеме "Визуальное программирование" программной платформы "NP Studio" [4 – 8] (рис. 3).

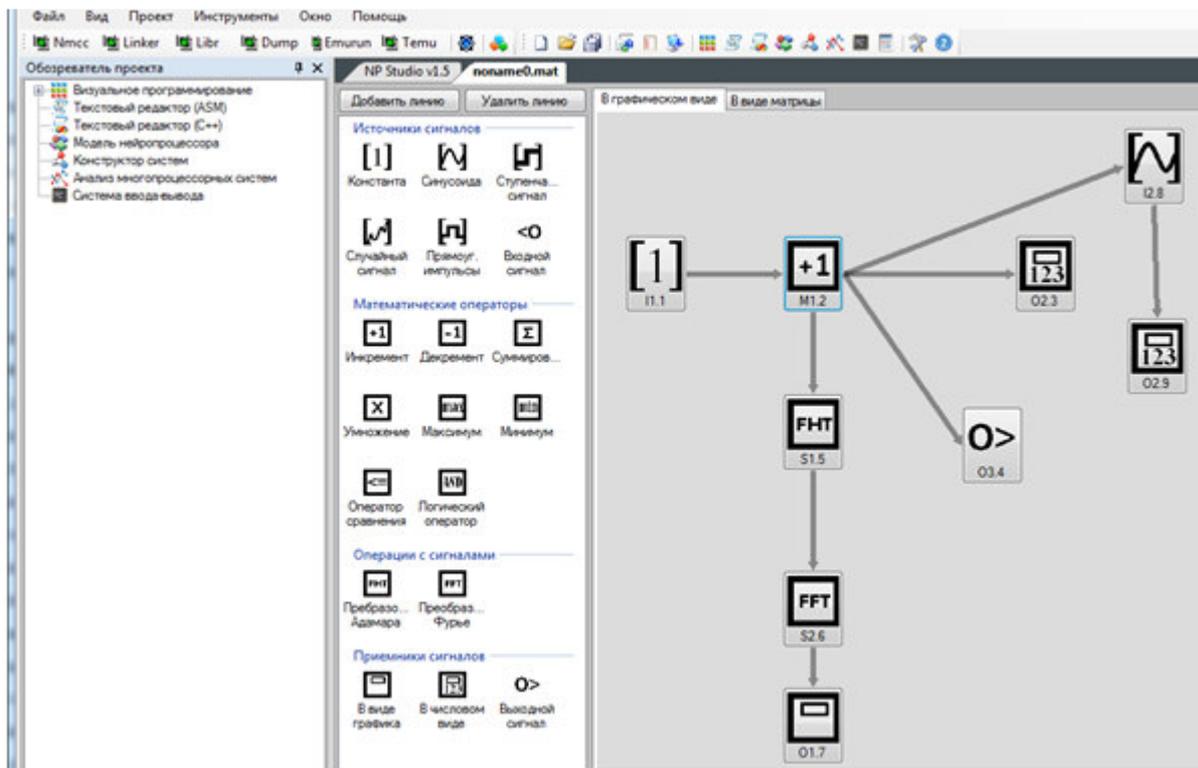


Рис. 3. Модель задачи автоматического управления модулями электромеханических систем в разработанной подсистеме "Визуальное программирование" программной платформы "NP Studio".

Главным элементом подсистемы является понятие "функциональный блок", который может быть связан с другими функциональными блоками и выполняет какие-либо функции.

На первом этапе пользователь описывает функциональные блоки (ФБ) и связи между ними с использованием их графического представления в программе.

Рабочая область программы разделена на функциональные части.

1. Область функциональных блоков, экземпляры которых могут быть перенесены в рабочую область моделирования методом Drag and Drop. В этой же области расположены управляющие элементы для визуализации и удаления связей между элементами. Каждый элемент при добавлении в рабочую область получает уникальное обозначение, представляющее собой конкатенацию категории элемента, номера элемента в этой категории и порядкового номера элемента, – например, "O2.9" определяет элемент из категории "Выходные сигналы", элемент "В числовом виде", с уникальным номером "9".

2. Рабочая область, которая в свою очередь может быть представлена в виде визуального представления элементов и связей между ними и в виде матрицы связей между элементами (рис. 4).

Заключение

Предложены алгоритмы и программное обеспечение специализированного вычислительного устройства на базе нейропроцессоров для автоматического управления модулями электромеханических систем на примере гексапода в ре-

жиме, приближенном к реальному времени с использованием концептуально новой вычислительной техники – нейропроцессора NM6406 на базе инструментального модуля МС 51.03 и микрокомпьютера МВ 77.07, разрабатываемых НТЦ "Модуль". Показана эффективность применения нейропроцессора для решения задачи автоматического управления модулями электромеханических систем за счет повышения скорости параллельной обработки, наличия операции "сложение с накоплением" и высокой энергоэффективности.

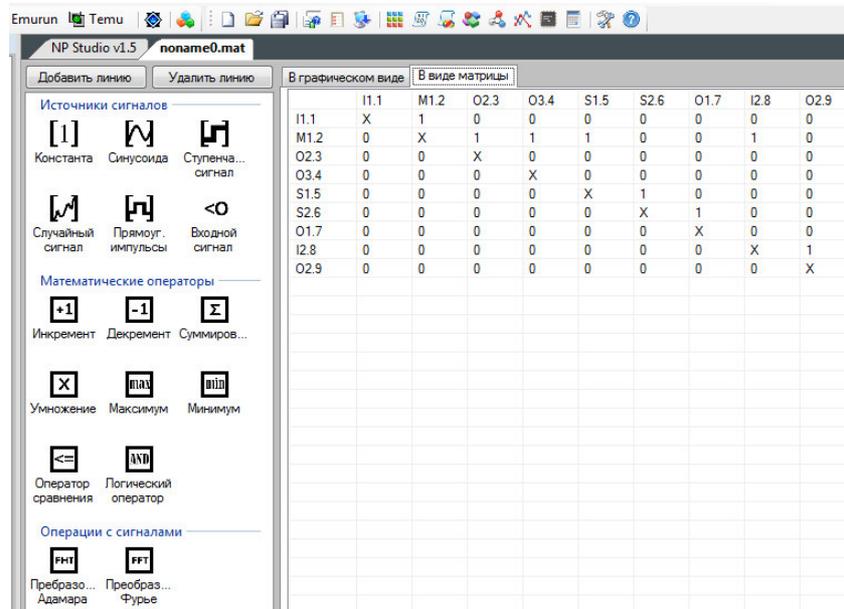


Рис. 4. Модель задачи автоматического управления модулями электромеханических систем в разработанной подсистеме "Визуальное программирование" в виде матрицы связей.

ЛИТЕРАТУРА

1. *Vladimir Ruchkin, Vitaliy Romanchuk, Roman Sulitsa.* Clustering, Restorability and Designing Of Embedded Computer System Based On Neuroprocessors // Proceedings of the 2nd Mediterranean Conference on Embedded Computing (MECO). – 2013. – С.58-62.
2. *Viksne P.E., Fomin D.V., Chernikov V.M.* The single-crystal digital neuroprocessor with variable digit capacity of operands // News of Higher education institutions. – 1996. – Vol.39, № 7.
3. *Галушкин А.И.* Нейрокомпьютеры. – Кн.3. – М.: ИПРЖР, 2000.
4. *Романчук В.А., Ручкин В.Н.* Разработка программных средств анализа нейропроцессорных систем // Вестник РГРТУ. – 2010. – Вып. 32, №2. – С.61–67.
5. *Романчук В.А., Ручкин В.Н., Фулин В.А.* Разработка модели сложной нейропроцессорной системы // Цифровая обработка сигналов. – 2012. – № 4. – С.70–74.
6. *Ручкин В.Н., Романчук В.А., Фулин В.А.* Когнитология и искусственный интеллект. – Рязань: Узорочье, 2012.
7. *Романчук В.А., Ручкин В.Н.* Алгоритмы анализа вычислительных структур на базе нейропроцессоров // Вестник РГРТУ. – 2012. – Вып.40, №2. – С.60–66.
8. *Романчук В.А., Ручкин В.Н.* Оценка результатов моделирования вычислительных систем на базе нейропроцессоров // Известия Тульского гос. ун-та: "Технические науки". – 2013. – Вып. 9-2. – С.194-203.

Статья представлена к публикации членом редколлегии О.С. Амосовым.

E-mail:

Романчук Виталий Александрович – v.a.romanchuk@yandex.ru.