

## ЧЕБЫШЕВСКИЙ СБОРНИК

Том 25. Выпуск 2.

УДК 004.046

DOI 10.22405/2226-8383-2024-25-2-127-138

**Модель контроллера с установленной программой поллинга для систем с векторным управлением**<sup>1</sup>

Е. В. Ларкин, В. С. Сальников, С. А. Скобельцын

**Ларкин Евгений Васильевич** — доктор технических наук, профессор, Тульский государственный университет (г. Тула).

*e-mail: elarkin@mail.ru*

**Сальников Владимир Сергеевич** — доктор технических наук, профессор, Тульский государственный университет (г. Тула).

*e-mail: vsalnikov.prof@yandex.ru*

**Скобельцын Сергей Алексеевич** — доктор физико-математических наук, Тульский государственный университет (г. Тула).

*e-mail: skbl@rambler.ru*

**Аннотация**

Разработан подход к конструированию программ цифровых контроллеров, выполняющих функции управления опросом сенсоров и исполнительных устройств, а также расчета векторных управляющих воздействий на объект. Исследованы три типа векторных алгоритмов расчета управляющих воздействий: решение систем разностных уравнений, вычисление цифровой свертки и ПИД регулятор. Показано, что контроллер, как реальный физический прибор, при реализации на нем программы поллинга является времязадающим элементом, и обеспечивает не только заданный период опроса периферийных устройств, но и задержки по времени между транзакциями. Произведена оценка временных интервалов по полумарковской модели алгоритма поллинга с гамильтоновым циклом управления транзакциями. На модели замкнутой линейной векторной системы управления показано влияние задержек по времени на такие характеристики системы, как перерегулирование и время выхода на установившийся режим. Разработана методика синтеза алгоритма поллинга по матричной модели системы управления, учитывающей реальные характеристики контроллера как прибора, включаемого в контур векторной обратной связи.

*Ключевые слова:* векторное управление, алгоритм поллинга, транзакция, задержка по времени, полумарковский процесс.

*Библиография:* 19 названий.

**Для цитирования:**

Е. В. Ларкин, В. С. Сальников, С. А. Скобельцын. Модель контроллера с установленной программой поллинга для систем с векторным управлением // Чебышевский сборник, 2024, т. 25, вып. 2, с. 127–138.

<sup>1</sup>Работа выполнена при поддержке гранта РФФ 24-29-20256

## CHEBYSHEVSKII SBORNIK

Vol. 25. No. 2.

UDC 004.046

DOI 10.22405/2226-8383-2024-25-2-127-138

**The controller with installed polling program model for vector control systems**

E. V. Larkun, V. S. Salnikov, S. A. Skobel'tsyn

**Larkun Evgeny Vasilievich** — doctor of technical sciences, professor, Tula State University (Tula).

*e-mail: elarkin@mail.ru*

**Salnikov Vladimir Sergeyeovich** — doctor of technical sciences, professor, Tula State University (Tula).

*e-mail: vsalnikov.prof@yandex.ru*

**Skobel'tsyn Sergey Alekseevich** — doctor of physical and mathematical sciences, Tula State University (Tula).

*e-mail: skbl@rambler.ru*

**Abstract**

An approach to digital controller's programs design, that perform the functions both sensors and actuators polling control, and control actions calculation, id worked out. Three types of control action computation vector algorithms: the solving systems of difference equations, calculation the digital convolution and PID controller are investigated: It is shown that the controller, as a real physical device, when implementing a polling program on it, is a time-setting element, and provides not only a specified peripheral devices polling period, but also time delays between transactions. An estimation of time intervals using the semi-Markov model of the polling algorithm with a Hamiltonian transaction management cycle is made. The model of a closed linear vector control system shows the influence of time delays on such system characteristics as overshoot and time to reach steady state. A method has been developed for polling algorithm synthesis using a control system matrix model that takes into account the real characteristics of the controller as the device included in the vector feedback loop.

*Keywords:* vector control, polling algorithm, transaction, time delay, semi-Markov process.

*Bibliography:* 19 titles.

**For citation:**

E. V. Larkun, V. S. Salnikov and S. A. Skobel'tsyn, 2024. "The controller with installed polling program model for vector control systems", *Chebyshevskii sbornik*, vol. 25, no. 2, pp. 127–138.

**1. Введение**

Основным направлением развития современной промышленности является широкое внедрение цифровых систем управления технологическим оборудованием на всех этапах производства [1], что предполагает использование в качестве аппаратной платформы контроллеров с последовательной интерпретацией команд программы управления [2]. Наиболее просто реализуемым и часто встречающимся способом организации доступа к сенсорам, измеряющим контролируемые параметры объекта управления, и исполнительным устройствам, осуществляющим воздействие на объект, является программный опрос периферийного оборудования, поллинг [3, 4]. Вследствие того, что программа поллинга интерпретируется контроллером в

реальном физическом времени, подобная реализация позволяет, с одной стороны, достаточно просто задавать требуемые периоды опроса сенсоров и приводов. С другой стороны, при реализации законов управления поллинг естественным образом приводит к возникновению задержек по времени между транзакциями, которые изменяют динамику работы цифровой системы управления в целом, по сравнению с аналоговой системой с теми же законами управления [5, 6]. В то же время, ухудшение динамики системы управления при переходе на цифровые платформы не должно приводить к снижению качества выпускаемой продукции, в связи с чем возникает научная задача учета реальных физических характеристик контроллеров с установленной программой поллинга при анализе и синтезе систем векторного управления. Методики оценки временных интервалов и учета оценок при синтезе систем цифрового управления распространены в инженерной практике недостаточно, что подтверждает актуальность исследований в данной области.

## 2. Модель контроллера с программой поллинга

В типовой системе цифрового векторного управления, приведенной на рис. 1, объект управления (ОУ) и цифровой контроллер (К) разделены интерфейсом И. В контроллере текущее время определяется дискретной переменной  $n$ , вне цифрового контроллера события развиваются в физическом времени  $t$ .

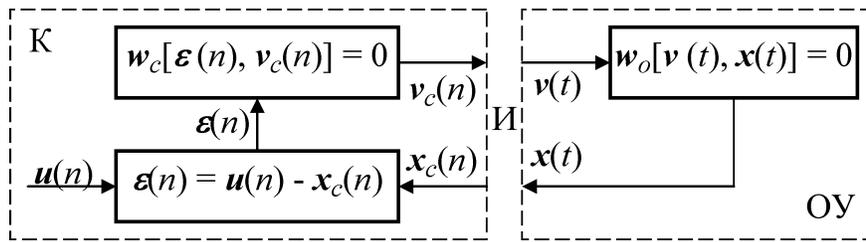


Рис. 1: Структурная схема цифровой векторной системы управления

Система функционирует следующим образом. Желаемое состояние ОУ задается цифровым вектором  $\mathbf{u}(n) = [u_1(n), \dots, u_k(n), \dots, u_K(n)]$ , где  $k$  – номер регулируемого параметра;  $K$  – количество регулируемых параметров. Текущее состояние ОУ,  $\mathbf{x}(t) = [x_1(t), \dots, x_k(t), \dots, x_K(t)]$ , измеряется с помощью сенсоров подсистемы и через интерфейс И в виде потока данных  $\mathbf{x}_c(n) = [x_{c,1}(n), \dots, x_{c,k}(n), \dots, x_{c,K}(n)]$ , где  $x_{c,k}(n)$  – код, формируемый на интерфейсе, по результатам оценки  $k$ -го параметра состояния  $x_k(t)$  в дискретный момент времени  $n$ , вводится в контроллер. В контроллере из векторов  $\mathbf{x}_c(n)$  и  $\mathbf{u}(n)$  формируется вектор кода ошибки

$$\boldsymbol{\varepsilon}(n) = \mathbf{u}(n) - \mathbf{x}_c(n), \quad (1)$$

где  $\boldsymbol{\varepsilon}(n) = [\varepsilon_1(n), \dots, \varepsilon_k(n), \dots, \varepsilon_K(n)]$  –  $K$ -элементный вектор разности между желаемым и текущим состояниями системы.

На следующем этапе в соответствии с выбранным законом управления

$$\mathbf{w}_c[\boldsymbol{\varepsilon}(n), \mathbf{v}_c(n)] = 0, \quad (2)$$

формируется поток векторов управляющих кодов  $\mathbf{v}_c(n) = [v_{c,1}(n), \dots, v_{c,k}(n), \dots, v_{c,K}(n)]$ , подаваемых после преобразования в аналоговые сигналы  $\mathbf{v}(t) = [v_1(t), \dots, v_k(t), \dots, v_K(t)]$  на исполнительные устройства ОУ.

Обработка данных (2) может быть реализована по одному из следующих алгоритмов: решение системы конечно-разностных уравнений [7], вычисление свертки [8]; реализация ПИД

регулятора [9, 10]. Каждый из перечисленных алгоритмов обеспечивает свою точность реализации закона управления и временную вычислительную сложность.

Алгоритмы, основанные на решении *системы конечно-разностных уравнений*, в общем случае имеют вид

$$\begin{aligned} w_{c,D,l} [n, \Delta^{0(v,1)} v_{c,1}(n), \Delta^{1(v,1)} v_{c,1}(n), \dots, \Delta^{m(v,1)} v_{c,1}(n), \dots, \Delta^{M(v,1)} v_{c,1}(n), \dots, \\ \Delta^{0(v,k)} v_{c,k}(n), \Delta^{1(v,k)} v_{c,k}(n), \dots, \Delta^{m(v,k)} v_{c,k}(n), \dots, \Delta^{M(v,k)} v_{c,k}(n), \dots, \\ \Delta^{0(v,K)} v_{c,K}(n), \Delta^{1(v,K)} v_{c,K}(n), \dots, \Delta^{m(v,K)} v_{c,K}(n), \dots, \Delta^{M(v,K)} v_{c,K}(n), \dots, \\ \Delta^{0(\varepsilon,l)} \varepsilon_l(n), \Delta^{1(\varepsilon,l)} \varepsilon_l(n), \dots, \Delta^{m(\varepsilon,l)} \varepsilon_l(n), \dots, \Delta^{M(\varepsilon,l)} \varepsilon_l(n)] = 0; \end{aligned} \quad (3)$$

$$1 \leq l \leq K,$$

где  $w_{c,D,l} [\dots]$  – скалярные функции;  $\Delta^{m(v,k)} v_{c,k}(n)$ ,  $\Delta^{m(\varepsilon,l)} \varepsilon_l(n)$  – конечные разности;  $m(\varepsilon, l)$ ,  $m(v, k)$  – индекс-функции, обозначающие порядок конечной разности, в которых символы, перечисленные в скобках, указывают на переменную, к которой относятся, и на индекс переменной в системе;  $M(\varepsilon, l)$  и  $M(v, k)$  – максимальные значения порядков;

$$\Delta^\kappa \rho(n) = \sum_{\iota=0}^{\kappa} (-1)^\iota \cdot C_\kappa^\iota \cdot \rho(n + \kappa - \iota), \quad C_\kappa^\iota = \frac{\kappa!}{\iota! (\kappa - \iota)!}.$$

Решение системы (3) без учета начальных условий может быть представлено в следующем виде:  $v_{c,k}(n) = w_{v,k} [\varepsilon_1(n), \dots, \varepsilon_l(n), \dots, \varepsilon_K(n)]$ . Вычислительная сложность алгоритма (3) определяется видом функций  $w_{c,D,l} [\dots]$  и значениями порядков  $M(\varepsilon, l)$  и  $M(v, k)$ .

Алгоритм свертки в общем случае имеет вид

$$\begin{aligned} v_{c,k}(n) = w_{\tilde{n},C,k} \{ \varepsilon_1[n], \dots, \varepsilon_1[n - m(v, k, 1)], \dots, \varepsilon_1[n - M(v, k, 1)], \tilde{v}_{k,1}[0], \dots, \tilde{v}_{k,1}[m(v, k, 1)], \\ \dots, \tilde{v}_{k,1}[M(v, k, 1)], \dots, \varepsilon_l[n], \dots, \varepsilon_l[n - m(v, k, l)], \dots, \varepsilon_l[n - M(v, k, l)], \tilde{v}_{k,l}[0], \dots, \\ \tilde{v}_{k,l}[m(v, k, l)], \dots, \tilde{v}_{k,l}[M(v, k, l)], \varepsilon_K[n], \dots, \varepsilon_K[n - m(v, k, K)], \dots, \varepsilon_K[n - M(v, k, K)], \\ \tilde{v}_{k,K}[0], \dots, \tilde{v}_{k,K}[m(v, k, K)], \dots, \tilde{v}_{k,K}[M(v, k, K)] \}, \quad 1 \leq k \leq K, \end{aligned} \quad (4)$$

где  $w_{c,C,k} \{\dots\}$ ,  $1 \leq k \leq K$  – скалярные функции, в общем случае нелинейные, от значений элементов вектора  $\varepsilon(n)$ , как текущих, так и полученных ранее;  $[n - m(v, k, l)]$  – ранее прошедшие дискретные моменты времени;  $\tilde{v}_{k,l}[m(v, k, l)]$ ,  $0 \leq m(v, k, l) \leq M(v, k, l)$  – дискретные значения импульсного отклика фильтра сигнала  $\varepsilon_l$  при формировании сигнала  $v_{c,k}$ ;  $M(v, k, l) + 1$ ,  $1 \leq k, l \leq K$  – величины апертур соответствующих фильтров.

Точность воспроизведения закона управления и вычислительная сложность процедуры расчета управляющего воздействия  $v_{c,k}$  при обработке сигнала ошибки  $\varepsilon_l$  по каждому регулируемому параметру  $x_k$  напрямую определяются величиной апертуры  $M(v, k, l) + 1$ . При необходимости достижения высокой точности вычислительная сложность алгоритма свертки может быть весьма значительной.

Одной из разновидностей законов управления, реализуемых в алгоритме свертки, является алгоритм  $w_{c,PID}$  ПИД регулятора, имеющий вид

$$v_{c,PID,k}(n) = \sum_{l=1}^K \left\{ \tilde{w}_{k,l,P} \varepsilon_l(n) + \tilde{w}_{k,l,D} [\varepsilon_l(n) - \varepsilon_l(n-1)] + \tilde{w}_{k,l,In} \sum_{m(In,k,l)=0}^n \varepsilon_l[m(In, k, l)] \right\}, \quad (5)$$

где  $\tilde{w}_{k,l,P}$ ,  $\tilde{w}_{k,l,D}$ ,  $\tilde{w}_{k,l,In}$  – коэффициенты при пропорциональной ( $P$ ), дифференциальной ( $D$ ) и интегральной составляющей ( $In$ ), соответственно;  $[\varepsilon_l(n) - \varepsilon_l(n-1)]$  – конечная разность;

$\sum_{m(In,k,l)=0}^n \varepsilon_l[m(In, k, l)]$  – конечная сумма;  $1 \leq k, l \leq K$ .

Следует отметить, что конечная сумма значений  $\varepsilon_l [m(In, k, l)]$  накапливается в процессе пошаговой реализации алгоритма (5) на предыдущих этапах и не оказывает существенного влияния на его вычислительную сложность.

Элементы вектора  $\mathbf{x}_c(n)$  вводятся, а вектора  $\mathbf{u}_c(n)$  выводятся из контроллера в результате работы программы поллинга (polling). В общем случае алгоритм поллинга имеет структуру, показанную на рис. 2 [11] и включает неисполнимые операторы  $a_{0(a)}$  «Начало» и  $a_{J(a)+1}$  «Конец», а также исполнимые операторы  $a_{1(a)}, \dots, a_{j(a)}, \dots, a_{J(a)}$ , реализующие функции транзакций и обработки данных. Циклическим алгоритм делает обратная связь  $[a_{J(a)+1}, a_{0(a)}]$ , показанная на рис. 2 штриховой стрелкой.

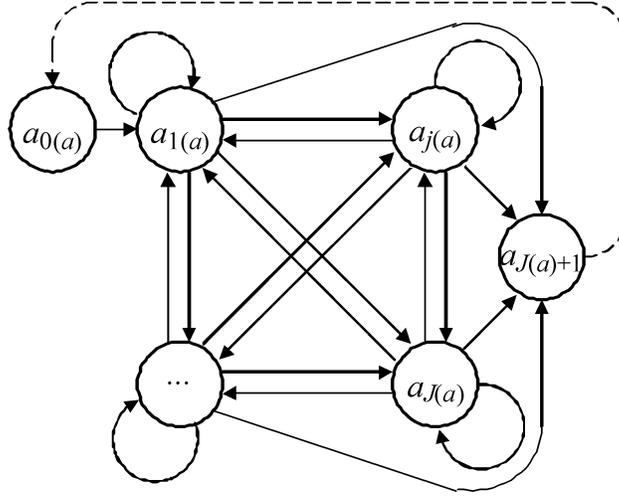


Рис. 2: Структура программы поллинга

Исполнимые операторы алгоритма осуществляют как опрос периферийных устройств, так и программную обработку данных. Принято в структуре, подобной приведенной на рис. 2, представлять только операторы транзакций, а обработку данных включать в операторы транзакций [11, 12]. Вследствие того, что техническими средствами контроллера алгоритм интерпретируется в реальном физическом времени  $t$  последовательно, оператор за оператором, между транзакциями естественным образом образуются временные интервалы, наличие которых должно учитываться при включении контроллера в контур управления.

Величины временных интервалов могут быть оценены с применением полумарковской модели интерпретации алгоритма [13, 14, 15]

$$\mu = \{A, \mathbf{h}(t)\}, \quad (6)$$

где  $A = \{a_{0(a)}, a_{1(a)}, \dots, a_{j(a)}, \dots, a_{J(a)}, a_{J(a)+1}\}$  – множество состояний полумарковского процесса, совпадающее с множеством операторов алгоритма;  $\mathbf{h}(t) = [h_{j(a),n(a)}(t)] = \mathbf{p} \otimes \mathbf{f}(t) - [J(a) + 2] \times [J(a) + 2]$  полумарковская матрица;  $\mathbf{p} = [p_{j(a),n(a)}] - [J(a) + 2] \times [J(a) + 2]$  стохастическая матрица;  $\mathbf{f}(t) = [f_{j(a),n(a)}(t)] - [J(a) + 2] \times [J(a) + 2]$  матрица плотностей распределения времени пребывания процесса в состояниях множества  $A$  до переключения в сопряженные состояния.

Правильно сконструированный алгоритм обладает следующими свойствами:

– для каждого оператора алгоритма (состояния полумарковского процесса)  $a_{j(a)}$ , существует хотя бы один путь  $a_{0(a)} \rightarrow a_{j(a)}$  и хотя бы один путь  $a_{j(a)} \rightarrow a_{J(a)+1}$ , т.е. алгоритм поллинга является циклическим без заикливания, а полумарковский процесс при замкнутой обратной связи является возвратным.

– на вероятности переключения из операторов (состояний)  $a_{j(a)}$  накладываются ограничения

$$\sum_{n(a)=0(a)}^{J(a)} p_{j(a),n(a)} = \begin{cases} 1, & j(a) \neq J(a) + 1 \text{ или } j(a) = J(a) + 1 \text{ (обратная связь замкнута);} \\ 0, & j(a) = J(a) + 1 \text{ (обратная связь разомкнута);} \end{cases}$$

– на время интерпретации операторов алгоритма (пребывания в состояниях полумарковского процесса)  $a_{j(a)}$ , накладываются ограничения:

$$0 < T_{j(a),n(a)\min} \leq \arg f_{j(a),n(a)}(t) \leq T_{j(a),n(a)\max} < \infty, \quad \int_0^{\infty} f_{j(a),n(a)}(t) dt = 1.$$

Таким образом, в полумарковской матрице  $\mathbf{h}(t)$  модели (6) с разомкнутой обратной связью  $[J(a) + 1]$ -я строка имеет вид  $[0, \dots, 0, \dots, 0]$ , а в модели  $\mathbf{h}'(t)$  с замкнутой обратной связью она равна  $[f_{J(a)+1,0(a)}(t), \dots, 0, \dots, 0]$ , где  $f_{J(a)+1,0(a)}(t)$  – плотность распределения времени работы операционной системы контроллера по перезапуску программы.

Одной из важных характеристик программы поллинга является период следования транзакций [8]. Для самого общего случая он может быть оценен по модели (6) согласно следующей зависимости

$$f_c(t) = \mathbf{I}_{0(a)}^R L^{-1} \left[ \sum_{\xi=1}^{\infty} \{L[\mathbf{h}(t)]\}^{\xi} \right] \mathbf{I}_{J(a)+1}^C * f_{J(a)+1,0(a)}(t), \quad (7)$$

где  $f_c(t)$  – плотность распределения времени от текущего до следующего перезапуска программы поллинга;  $\mathbf{I}_{0(a)}^R$  –  $[J(a) + 2]$ -элементный вектор-строка, нулевой элемент которого равен единице, а остальные элементы равны нулю;  $\mathbf{I}_{J(a)+1}^C$  –  $[J(a) + 2]$ -элементный вектор-столбец, последний,  $[J(a) + 1]$ -й элемент которого равен единице, а остальные элементы равны нулю;  $L[\dots]$ ,  $L^{-1}[\dots]$  – прямое и обратное преобразование Лапласа, соответственно.

Математическое ожидание от  $f_c(t)$  дает собственно период опроса  $T_c$ , а дисперсия  $D_c$  – оценку шума поллинга

$$T_c = \int_0^{\infty} t f_c(t) dt; \quad D_c = \int_0^{\infty} (t - T_c)^2 f_c(t) dt.$$

В реальном алгоритме поллинга (полумарковской модели (6)) на структуру может быть наложено следующее ограничение: в течение одного цикла каждое устройство ОУ (сенсор или привод) опрашивается по одному разу, а следовательно, исполнимые операторы  $a_{1(a)}, \dots, a_{j(a)}, \dots, a_{J(a)}$  (последовательность смены состояний полумарковского процесса) образуют гамильтонов цикл на графе, показанном на рис. 2. Схема линейно-циклического алгоритма, представляющего один из возможных гамильтоновых циклов, приведена на рис. 3 а, где показано  $a_{k(H)} \in \{a_{1(a)}, \dots, a_{j(a)}, \dots, a_{J(a)}\}$ ,  $1(H) \leq k(H) \leq 2K(H)$ .

Кроме интервала опроса  $T_c$  алгоритм поллинга формирует также и временные интервалы между транзакциями (рис. 3 б). Транзакции ввода данных от сенсоров отмечены стрелками с круглым наконечником, транзакции вывода управляющих воздействий на приводы отмечены стрелкам с ромбическим наконечником. Без нарушения общности можно считать, что внутри гамильтонова цикла первой транзакцией является ввод элемента  $x_{c,1}$  вектора состояния  $\mathbf{x}_c$ , после чего в порядке, определенном алгоритмом, вводятся оставшиеся элементы указанного вектора и выводятся элементы вектора  $\mathbf{v}_c$ .

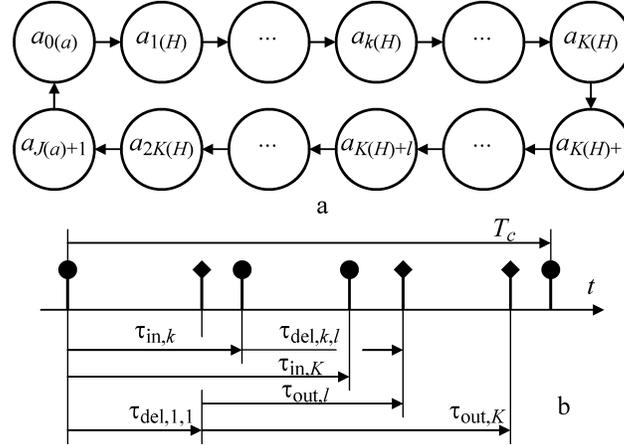


Рис. 3: Реализация гамильтонова пути на исполнимых операторах (а) и порядок следования транзакций (б)

Временные интервалы между началом ввода  $x_{c,1}$  и началами вводов  $x_{c,2}, \dots, x_{c,k}, \dots, x_{c,K}$  порождают перекося данных  $\tau_{in,1}, \dots, \tau_{in,k}, \dots, \tau_{in,K}$  при вводе, где  $\tau_{in,1} = 0$ . Временные интервалы  $\tau_{out,1}, \dots, \tau_{out,k}, \dots, \tau_{out,K}$  между началом вывода элемента  $v_{c,1}$  вектора  $\mathbf{v}_c$  и началами выводов элементов  $v_{c,2}, \dots, v_{c,l}, \dots, v_{c,K}$ , порождают перекося данных  $\tau_{out,1}, \dots, \tau_{out,l}, \dots, \tau_{out,K}$  при выводе  $\mathbf{v}_c$ . Временной интервал  $\tau_{del,k,l}$ , алгоритмически формируемый между вводом  $x_{c,k}$  и выводом  $v_{c,l}$ , дает часть времени чистого запаздывания формирования  $l$ -го элемента вектора управления относительно  $k$ -го элемента вектора состояния. Полное время чистого запаздывания определяется используемым в системе алгоритмом обработки данных ( $w_{c,D}$ ,  $w_{c,C}$  или  $w_{c,PID}$ )

$$\tau_{k,l} = \tau_{del,k,l} + N_w T_c, \quad (8)$$

где  $\tau_{del,k,l} = (\tau_{del,1,1} + \tau_{out,l}) - \tau_{in,k}$ ;  $N_w$  – количество вводов элементов вектора  $\mathbf{x}_c$ , которые необходимо совершить для расчета  $v_{c,l}$ .

В алгоритме (3)  $N_w = \max \{M(v, 1), \dots, M(v, k), \dots, M(v, K)\}$ . В алгоритме (4)  $N_w = \max \{M(v, k, 1), \dots, M(v, k, l), \dots, M(v, k, K)\}$ . В алгоритме (5)  $N_w = 1$ , поскольку в общем случае в алгоритме используется конечная разность порядка не выше первого, а конечная сумма накапливается в течение всех циклов работы алгоритма.

Текущее значение дискретного аргумента  $n$  может быть представлено в виде  $n = t/T_c$ . В этом случае при  $T_c \rightarrow 0$  конечные разности в (3), (5) преобразуются в производные. Конечная сумма в (5) и дискретная свертка в (4) преобразуются в интеграл и континуальную свертку, соответственно. Общее для (3), (4), (5) выражение (2), после линейризации, добавления к нему (1) и применения к системе преобразования Лапласа может быть представлено как

$$\mathbf{V}_c(s) = \mathbf{W}_{\tilde{n}}(s) \cdot [\mathbf{U}(s) - \mathbf{X}_{\tilde{n}}(s)], \quad (9)$$

где  $\mathbf{V}_c(s) = L[\mathbf{v}_c^\theta(t)]$ ,  $\mathbf{U}(s) = L[\mathbf{u}^\theta(t)]$ ,  $\mathbf{x}_c(t)$ ,  $\mathbf{v}_c(t)$ ,  $\mathbf{u}(t)$  – векторы, получаемые из  $\mathbf{x}_c(n)$ ;  $\mathbf{v}_c(n)$ ,  $\mathbf{u}(n)$  путем подстановки вместо дискретного аргумента  $n$  непрерывного времени  $t$ ;  $\theta$  – операция транспонирования;  $\mathbf{W}_c(s) = [W_{c,k,l}(s)]$  – матрица передаточных функций;  $W_{c,k,l}(s) = V_{c,k}(s)/\varepsilon_l(s)$  – передаточные функции, определяемые из реализованных алгоритмов (3), (4) или (5);  $\varepsilon(s) = L[\varepsilon(t)]$ ;  $\varepsilon(t)$  – вектор, получаемый из  $\varepsilon(n)$  путем подстановки вместо дискретного аргумента  $n$  непрерывного времени  $t$ ;  $s$  – переменная Лапласа (оператор дифференцирования).

В свою очередь, линейный ОУ может быть представлен как

$$\mathbf{X}(s) = \mathbf{W}_o(s) \cdot \mathbf{V}(s), \quad (10)$$

где  $\mathbf{V}(s) = L[v^\theta(t)]$ ;  $\mathbf{X}(s) = L[x^\theta(t)]$ ;  $\mathbf{W}_o(s) = [W_{o,k,l}(s)]$  – матрица передаточных функций, которая получается из модели ОУ  $\mathbf{w}_o[\mathbf{v}(t), \mathbf{x}(t)] = 0$  (рис. 1) с помощью операций линеаризации и выполнения преобразования Лапласа.

Из (9) и (10) следует, что без учета временного фактора формирование состояний ОУ описывается матричной системой

$$\mathbf{X}(s) = [\mathbf{E} + \mathbf{W}_c(s) \cdot \mathbf{W}_o(s)]^{-1} \cdot \mathbf{W}_c(s) \cdot \mathbf{W}_o(s) \cdot \mathbf{U}(\omega), \quad (11)$$

где  $\mathbf{E}$  – единичная диагональная  $K \times K$  матрица.

Задержки по времени при выводе элементов вектора  $\mathbf{V}(s)$  относительно момента ввода элементов вектора  $\mathbf{X}(s)$ , оцененные в соответствии с (8), согласно теореме о смещении в действительной области могут быть представлены в виде  $K \times K$  матрицы передаточных функций интерфейса  $\mathbf{I}$

$$\mathbf{W}_I(s) = \begin{bmatrix} \exp(-s\tau_{1,1}) & \dots & \exp(-s\tau_{1,l}) & \dots & \exp(-s\tau_{1,K}) \\ & & \dots & & \\ \exp(-s\tau_{k,1}) & \dots & \exp(-s\tau_{k,l}) & \dots & \exp(-s\tau_{k,K}) \\ & & \dots & & \\ \exp(-s\tau_{K,1}) & \dots & \exp(-s\tau_{K,l}) & \dots & \exp(-s\tau_{K,K}) \end{bmatrix}. \quad (12)$$

С учетом (12) выражение (11) принимает вид

$$\mathbf{X}(s) = [\mathbf{E} + \mathbf{W}_I(s) \cdot \mathbf{W}_c(s) \cdot \mathbf{W}_o(s)]^{-1} \cdot \mathbf{W}_I(s) \cdot \mathbf{W}_c(s) \cdot \mathbf{W}_o(s) \cdot \mathbf{U}(\omega). \quad (13)$$

Матрица передаточных функций интерфейса оказывает существенное влияние на динамику процесса программного управления. Изменение динамических характеристик замкнутой системы обусловлено изменением характеристического уравнения, которое в данном случае принимает вид

$$\det[\mathbf{E} + \mathbf{W}_I(s) \cdot \mathbf{W}_c(s) \cdot \mathbf{W}_o(s)] = 0. \quad (14)$$

Отличительной особенностью (14) является наличие в характеристическом уравнении комплексных экспонент, которые зависят от временных интервалов (8), создаваемых алгоритмом поллинга. Комплексные экспоненты изменяют местоположение корней на комплексной плоскости [16, 17], а следовательно, и характер переходных процессов [18, 19].

### 3. Методика конструирования программ поллинга

Из изложенного следует методика конструирования программ поллинга, реализующих системы векторного управления сложными многоконтурными объектами. Исходными данными для реализации методики является модель ОУ и технические требования к цифровому контроллеру, как к платформе для размещения управляющей программы.

Порядок моделирования системы управления с контроллером на основе программы поллинга представляется следующим:

1) Разработка математической модели ОУ  $\mathbf{w}_o[\mathbf{v}(t), \mathbf{x}(t)] = 0$ , включающей модель собственно объекта, а также модели приводов и сенсоров, обеспечивающих функционирование ОУ.

2) Выбор закона управления  $\mathbf{w}_c[\varepsilon(n), \mathbf{v}_c(n)] = 0$  для реализации в виде алгоритма и программы поллинга.

3) Определение платформы для реализации системы управления и проверка реализуемости выбранного закона на заданной платформе.

4) Моделирование процесса управления без учета задержек по времени, вносимых контроллером как реальным физическим прибором, реализующим алгоритм поллинга в реальном времени. При необходимости корректировка закона управления.

5) Оценка задержек по времени между транзакциями в алгоритме поллинга и моделирование системы управления с их учетом. При невыполнении требований к системе управления корректировка алгоритма поллинга, закона управления, выбранной цифровой платформы и/или обеспечивающей аппаратуры ОУ.

Рассмотрим применение методики на примере, в котором ОУ описывается матричным уравнением

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{s} \begin{bmatrix} \frac{20}{0.0002s^2 + 0.03s + 1} & \frac{5}{0.04s + 1} \\ \frac{5}{0.04s + 1} & \frac{20}{0.0002s^2 + 0.03s + 1} \end{bmatrix} \begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix}.$$

Вследствие того, что в ОУ присутствует астатизм первого порядка, из ПИД регулятора исключен канал интегрирования и закон управления имеет вид ПД регулятора:

$$\begin{pmatrix} 0.15 [\varepsilon_1(n) - \varepsilon_1(n-1)] + 1.5\varepsilon_1(n) \\ 0.15 [\varepsilon_2(n) - \varepsilon_2(n-1)] + 1.5\varepsilon_2(n) \end{pmatrix} = \begin{pmatrix} 0.01 [v_1(n) - v_1(n-1)] + v_1(n) \\ 0.01 [v_2(n) - v_2(n-1)] + v_2(n) \end{pmatrix},$$

$$\begin{bmatrix} \varepsilon_1(n) \\ \varepsilon_2(n) \end{bmatrix} = \begin{bmatrix} u_1(n) - x_1(n) \\ u_2(n) - x_2(n) \end{bmatrix}.$$

На вход системы подаются единичные ступенчатые функции Хевисайда

$$u_1(n) = u_2(n) = \begin{cases} 0, & \text{при } n \leq 0; \\ 1, & \text{при } n > 0. \end{cases}$$

Результат моделирования представлен на рис. 4, где на рис. 4 а показан переходный процесс в модели, не учитывающей запаздывание в контурах обратной связи; на рис. 4 б, с, d учтено запаздывание  $\tau$ , которое равно, соответственно, 0.0025, 0.005 и 0.0079 с.

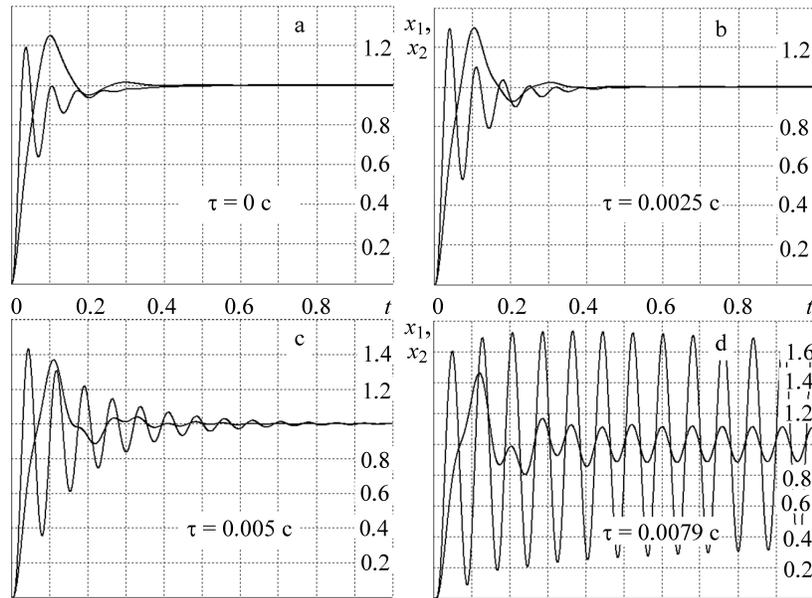


Рис. 4: Переходные процессы в системе без учета (а) и с учетом задержек (б, с, d) вносимых программой поллинга

Как следует из примера, с увеличением времени запаздывания в указанных диапазонах время выхода системы на установившийся режим равно, соответственно 0.4 с, 0.45 с, 0.9 с,  $> 1$  с (неустойчивое функционирование). Перерегулирование при реакции системы на единичную функцию Хевисайда равно: по каналу  $x_1$  – 25%, 30%, 35%, 45%; по каналу  $x_2$  – 20%, 30%, 40%, 60%.

## 4. Заключение

Основным результатом данной работы следует считать модель функционирования цифрового контроллера как физического прибора, осуществляющего расчет вектора управляющих воздействий на многоконтурный объект на основании информации, получаемой от сенсоров, в реальном физическом времени.

Подобная модель позволяет оценить задержки по времени, вносимые контроллером и учитывать эти задержки при синтезе алгоритма поллинга. Практическим результатом является методика конструирования программ поллинга для систем векторного управления.

Дальнейшие исследования в данной области могут быть направлены на установление зависимостей, прямую связывающих временную вычислительную сложность программы поллинга, структурную сложность объекта управления и требуемые параметры функционирования системы управления.

## СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Malin Löfving M., Säfsten K., Winroth M. Manufacturing strategy formulation, leadership style and organizational culture in small and medium-sized enterprises // IJMTM. 2016. Vol. 30. No. 5. P. 306–325.
2. Landau I. D., Zito G. Digital Control Systems, Design, Identification and Implementation. Springer, 2006. 484 p.
3. Larkin E. V., Ivutin A. N. Estimation of Latency in Embedded Real-Time Systems // 3-rd Mediterranean Conference on Embedded Computing (MECO-2014). 2014. June 15-19. Budva, Montenegro, 2014. P. 236–239.
4. Kilian C. T. Modern control technology: Components and systems. Thompson Delmar Learning. 2000. 608 p.
5. Babishin V., Taghipour S. Optimal maintenance policy for multicomponent systems with periodic and opportunistic inspections and preventive replacements // Applied Mathematical Modelling. 2016. V. 40. №. 23. P. 10480–10505.
6. Aström J., Wittenmark B. Computer Controlled Systems: Theory and Design. Tsinghua University Press. Prentice Hall, 2002. 557 p.
7. Meyer-Baese U. Digital signal processing. Springer-Verlag Berlin, Heidelberg, 2004. 523 p.
8. Yeh Y.-C., Chu Y., Chiou C. W. Improving the sampling resolution of periodic signals by using controlled sampling interval method // Computers & Electrical Engineering, 2014. Vol. 40. N. 4. P. 1064–1071.
9. Ang K. N., Chong G., Li Y. PID control system analysis, design and technology // IEEE Transactions of control systems technology. 2005. V. 13. N 4. P. 559–576.

10. O'Dwier A. PID compensation of time delay processes 1999-2002: a survey // Proceedings of the American control conference, USA, Denver, Colorado. 2003. P. 1494–1499.
11. Larkin E. V., Nguyen V. S., Privalov A. N. Simulation of digital control systems by nonlinear objects // Lecture Notes on Data Engineering and Communications Technologies. 2022. V. 124. P. 711–721.
12. Larkin E., Privalov A., Bogomolov A., Akimenko T. Digital Control of Continuous Production with Dry Friction at Actuators // Smart Innovation, Systems and Technologies, 2022. V. 232. P. 427–436.
13. Bielecki T. R., Jakubowski J., Niewęgłowski M. Conditional Markov chains: Properties, construction and structured dependence // Stochastic Processes and their Applications. V. 127, N. 4. 2017. P. 1125–1170.
14. Ching W. K., Huang X., Ng M. K., Siu T. K. Markov Chains: Models, Algorithms and Applications / International Series in Operations Research & Management Science. V. 189. Springer Science + Business Media NY, 2013. 241 p.
15. Janssen J., Manca R. Applied Semi-Markov processes. Springer US, 2006. 310 p.
16. Toshiharu Sugie. Simple explanation of Routh-Hurwitz criterion for undergraduate education // Systems, control and information. 2021. Vol. 65. N 7. P. 257–270.
17. Bodoson M. Explaining Routh-Hurwitz criterion: A tutorial presentation // IEEE Control systems magazine. 2020. Vol. 40, N 1. P. 45–51.
18. Wu M., He Y., She J. H., Liu G. P. Delay-dependent criteria for robust stability of time-varying delay systems // Automatica, 2004. Vol. 40, N. 8. P. 1435–1439.
19. Zhang X. M., Min W. U., Yong H. E. Delay dependent robust control for linear systems with multiple time-varying delays and uncertainties // Control & Decision, 2004. Vol. 19. N. 5. P. 496–500.

## REFERENCES

1. Malin Löfving, M., Säfsten, K. & Winroth, M. 2016. “Manufacturing strategy formulation, leadership style and organizational culture in small and medium-sized enterprises”, *IJMTEM*, no. 30 (5), pp. 306–325.
2. Landau, I. D. & Zito, G. 2006. *Digital Control Systems, Design, Identification and Implementation*, Springer, 484 pp.
3. Larkin, E. V. & Ivutin, A. N. 2014. “Estimation of Latency in Embedded Real-Time Systems”, *3-rd Mediterranean Conference on Embedded Computing (MECO-2014)*, June 15-19. Budva, Montenegro, pp. 236–239.
4. Kilian, C. T. 2000. *Modern control technology: Components and systems*, Thompson Delmar Learning, 608 pp.
5. Babishin, V. & Taghipour, S. 2016. “Optimal maintenance policy for multicomponent systems with periodic and opportunistic inspections and preventive replacements”, *Applied Mathematical Modelling*, no. 40 (23), pp. 10480–10505.

6. Aström, J. & Wittenmark, B. 2002. *Computer Controlled Systems: Theory and Design*, Tsinghua University Press. Prentice Hall, 557 pp.
7. Meyer-Baese, U. 2004. *Digital signal processing*, Springer-Verlag Berlin, Heidelberg, 523 pp.
8. Yeh, Y.-C., Chu, Y. & Chiou C. W. 2014. “Improving the sampling resolution of periodic signals by using controlled sampling interval method”, *Computers & Electrical Engineering*, no. 40 (4), pp. 1064–1071.
9. Ang, K. N., Chong, G. & Li, Y. 2005. “PID control system analysis, design and technology”, *IEEE Transactions of control systems technology*, no. 13 (4), pp. 559–576.
10. O’Dwier A. 2003. “PID compensation of time delay processes 1999-2002: a survey”, *Proceedings of the American control conference*, USA, Denver, Colorado, pp. 1494–1499.
11. Larkin E. V., Nguyen V. S. & Privalov A. N. 2022. “Simulation of digital control systems by nonlinear objects”, *Lecture Notes on Data Engineering and Communications Technologies*, vol. 124, pp. 711–721.
12. Larkin, E., Privalov, A., Bogomolov, A. & Akimenko, T. 2022. “Digital Control of Continuous Production with Dry Friction at Actuators”, *Smart Innovation, Systems and Technologies*, vol. 232, pp. 427–436.
13. Bielecki T. R., Jakubowski, J. & Niewęglowski, M. 2017. “Conditional Markov chains: Properties, construction and structured dependence”, *Stochastic Processes and their Applications*, no. 127 (4), pp. 1125–1170.
14. Ching, W. K., Huang, X., Ng, M. K. & Siu, T. K. 2013. “Markov Chains: Models, Algorithms and Applications”, *International Series in Operations Research & Management Science*, vol. 189. Springer Science + Business Media NY, 241 pp.
15. Janssen, J. & Manca, R. 2006. *Applied Semi-Markov processes*, Springer US, 310 pp.
16. Toshiharu, Sugie. 2021. “Simple explanation of Routh-Hurwitz criterion for undergraduate education”, *Systems, control and information*, no. 65 (7), pp. 257–270.
17. Bodoson, M. 2020. “Explaining Routh-Hurwitz criterion: A tutorial presentation”, *IEEE Control systems magazine*, no. 40 (1), pp. 45–51.
18. Wu, M., He, Y., She, J. H. & Liu, G. P. 2004. “Delay-dependent criteria for robust stability of time-varying delay systems”, *Automatica*, no. 40 (8), pp. 1435–1439.
19. Zhang, X. M., Min, W. U. & Yong, H. E. 2004. “Delay dependent robust control for linear systems with multiple time-varying delays and uncertainties”, *Control & Decision*, no. 19 (5), pp. 496–500.

Получено: 06.02.2024

Принято в печать: 28.06.2024