



## Основы формальной теории систем реального времени

А. А. Зеленский<sup>а</sup>, канд. техн. наук, доцент, [orcid.org/0000-0002-3464-538X](https://orcid.org/0000-0002-3464-538X)

А. А. Грибков<sup>а</sup>, доктор техн. наук, ведущий научный сотрудник, [orcid.org/0000-0002-9734-105X](https://orcid.org/0000-0002-9734-105X), [andarmo@yandex.ru](mailto:andarmo@yandex.ru)

<sup>а</sup>НПК «Технологический центр», Шокина пл., 1, стр. 7, Москва, 124498, РФ

**Введение:** важность и выраженная специфика вычислительных систем реального времени обуславливает востребованность построения абстрагированной от внешнего мира строго формализованной системы, включающей в себя необходимый и достаточный комплекс определяющих параметров. **Цель:** создать формальную теорию вычислительных систем, работающих в условиях ограничения времени. **Результаты:** предложена новая формальная теория систем реального времени, позволяющая формализовать представление систем управления реального времени, что, по мнению авторов, расширит возможности их анализа и синтеза. Объектом предлагаемой формальной теории является цикл системы реального времени, определяемый исходя из заданной композиции функций, обеспечивающей выполнение системой своего функционального назначения, и представляемый в виде конфигурации, формируемой из акторов, распределенных по группам, потокам исполнения и порядку исполнения в каждом потоке. Формальная теория систем реального времени включает в себя набор стандартных и специальных символов, формул. Посредством этих символов и формул определяются возможные для заданной композиции акторов варианты конфигураций, удовлетворяющие установленным требованиям, а для каждой из этих конфигураций – достижимые длительности цикла, потребные ресурсы и количество портов ввода и вывода для связи акторов. Определено место и роль формальной теории систем реального времени в определении осуществимости вычислительной системы, констатирована возможность двух вариантов конфигурирования: статического и динамического. Последнее необходимо в случае переопределения конфигурации цикла в процессе его отработки.

**Ключевые слова** — формальная теория, реальное время, система управления, композиция, конфигурация, цикл, ресурсы, порты, осуществимость.

**Для цитирования:** Зеленский А. А., Грибков А. А. Основы формальной теории систем реального времени. *Информационно-управляющие системы*, 2025, № 5, с. 2–10. doi:10.31799/1684-8853-2025-5-2-10, EDN: JNQQQK

**For citation:** Zelenskii A. A., Gribkov A. A. Fundamentals of the formal theory of real-time systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2025, no. 5, pp. 2–10 (In Russian). doi:10.31799/1684-8853-2025-5-2-10, EDN: JNQQQK

### Введение

Основным трендом развития вычислительных систем в продолжении всего времени с появления ЭВМ является повышение их производительности. За последние 30 лет производительность (вычислительная мощность) наиболее мощной компьютерной системы в мире выросла в 12,4 млн раз (со 140,4 GFLOPS в 1994 г. до 1,742 EFLOPS в 2024 г., <https://top500.org/resources/top-systems/>). Такой рост производительности позволяет успешно решать задачи, для которых в свое время были созданы ЭВМ, — научные расчеты с большим объемом вычислений без жестких ограничений по времени [1].

Важной составляющей социально-экономического развития в последние десятилетия стало широкое внедрение информационных технологий во все отрасли экономики и сферы общественного потребления. Характер требований к вычислительным системам при этом изменяется. Актуальность сложных научных расчетов, требующих высокой производительности, по-прежнему высокая, и она не будет снижаться [2]. Но также растет значимость быстродействия вычислительных систем, определяющего их способность выполнять ограниченный

комплекс вычислительных операций за малый интервал времени [3, 4]. Необходимость быстродействия вычислительных систем возникает в системах управления, в том числе с наиболее жестким ограничением длительности указанного интервала времени (цикла управления) — вычислительных системах жесткого реального времени ([https://www.eremex.ru/upload/iblock/eb6/rtos\\_dev\\_book.pdf](https://www.eremex.ru/upload/iblock/eb6/rtos_dev_book.pdf)), для которых нарушения временных ограничений недопустимы (фатальны с точки зрения достижения цели управления) [5].

Косвенным показателем быстродействия вычислительной системы является время ее отклика. По данному показателю за последние 30 лет прогресса не наблюдается: время отклика даже для сравнительно несложных вычислительных систем с архитектурой фон Неймана [6] (например, ПК) не опускается ниже 30–50 мс [7], что для современных систем управления реального времени неприемлемо большой интервал.

В настоящее время человеческая цивилизация находится в процессе перехода к цивилизации когнитивных технологий [8, 9], технологической основой которой являются когнитивные системы, служащие для интеллектуального управления машинами, в том числе управления

в реальном времени. Можно констатировать, что вычислительные системы реального времени являются важнейшей составляющей современных технологий, а в будущем их значимость продолжит повышаться.

Важность и выраженная специфика вычислительных систем реального времени обуславливает востребованность формализации таких систем, построения абстрагированной от внешнего мира строго формализованной системы, включающей в себя необходимый и достаточный комплекс определяющих параметров.

Имеющиеся в настоящее время работы в области вычислительных систем реального времени пока не обеспечивают необходимой степени формализации их представления, в частности не позволяют напрямую вводить в описание системы исходное ограничение по временной и пространственной вычислительной сложности (длительности цикла вычислений и доступной памяти). Указанные ограничения в рамках существующих подходов вводятся опосредованно, на стадии анализа конечного результата функционирования вычислительной системы реального времени, удовлетворяющей или не удовлетворяющей требованиям, накладываемым на систему скоростью изменений (физических или иных параметров) объекта управления [10–12]. Существенная часть современных исследований посвящена проблематике программного обеспечения операционных вычислительных систем реального времени (real-time operating system, RTOS) [13, 14]. Успешная программная реализация — необходимое условие эффективности вычислительных систем реального времени, однако ей должно предшествовать структурное определение реализуемого системой вычислительного алгоритма, не определяемое в рамках оптимизации программного обеспечения. Базой для решения задачи успешной реализации вычислительного алгоритма реального времени является формализация его представления в рамках формальной теории.

Определенность формальной теории требует удовлетворения следующим четырем основным условиям [15, 16].

Во-первых, необходимо определить ограниченное множество символов, из которых по заданным правилам формируются выражения. Эти символы могут быть равноценными (как буквы в алфавите) или соответствовать принципиально различным понятиям (например, элемент и свойство элемента).

Во-вторых, требуется сформировать определяющую систему совокупность выражений — множество формул. Множество формул должно обеспечивать взаимосвязанное и внутренне непротиворечивое представление системы. При

этом данное множество не является ограниченным уже определенными формулами, но может дополняться, оставаясь конечным.

В-третьих, следует задать аксиомы формальной системы (если в этом есть необходимость).

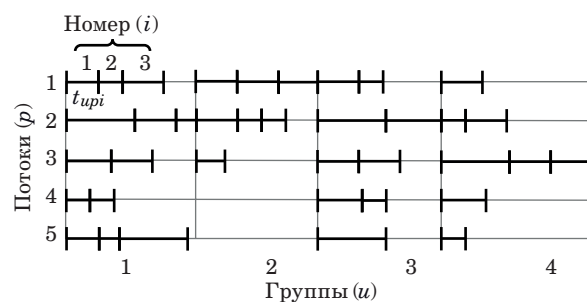
В-четвертых, необходимо определить правила вывода, определяющие отношения между формулами.

## Цикл системы реального времени

Любая формальная теория служит цели описания какого-либо объекта или класса объектов — реальных или абстрактных. Объектом, описанию которого служит формальная теория систем реального времени, является цикл системы управления реального времени, представляющий собой реализацию композиции элементов (функций), необходимой и достаточной для выполнения системой своей функции управления. Напомним, что системы реального времени — это почти исключительно системы управления. Именно для систем управления возникает необходимость замыкания в цикл обратной связи совокупности элементов-функций, контролирующих состояние объекта управления.

Цикл системы управления реального времени (рисунок) представляет собой замкнутый цикл, ограниченный интервалом времени, за который выполняется совокупность необходимых операций управления — элементов цикла, которые можно отождествить с функциями. Элементы, используемые в процессе управления, взаимосвязаны, что обуславливает необходимость формирования между ними каналов связи и устанавливает ограничения на порядок их выполнения. Кроме того, каждый из элементов требует ресурсов (вычислительных или других), что накладывает ограничение на их одновременное задействование: используемые ресурсы не должны превышать имеющихся в наличии.

В результате цикл системы управления реального времени представляет собой компози-



- Диаграмма цикла системы управления
- Control system cycle diagram

цию элементов с некоторой конфигурацией [17, 18], заключающейся в распределении элементов по группам, потокам и последовательности выполнения в каждом из потоков. Разбиение на группы отражает необходимость выполнения элементов одних групп строго после других (после завершения выполнения последнего элемента из предыдущей группы), разбиение на потоки — возможность выполнения некоторых (не связанных между собой) элементов параллельно [19].

Свойства элементов в составе цикла управления, основными из которых являются автономность, обмен данными (через заданные каналы связи), интеграция для выполнения общих задач, определяют предпочтительность акторной модели [20, 21] представления композиции элементов.

## Символы и выражения

Для описания цикла системы управления реального времени как конфигурации заданной композиции элементов-акторов необходимо использовать различные символы, как стандартные, так и специальные, вводимые формальной теорией систем реального времени.

К числу используемых стандартных символов относятся:  $\sum$  — сумма;  $\cup$  — дизъюнкция;  $\forall$  — квантор всеобщности;  $\min(\dots)$ ,  $\max(\dots)$  — минимум и максимум;  $+$ ,  $-$ ,  $\times$  — операторы суммы, разности и умножения;  $\{\dots\}$  — оператор множества и др.

К специальным символам относятся:

$\tau$  — длительность цикла системы управления, которая может либо задаваться как начальное условие, следующее из требований по точности задания параметров объекта управления (например, для управления движением рабочего органа — исходя из допустимой погрешности задания траектории и скорости движения), либо определяться по итогам оптимизации;

$W$  — общие доступные ресурсы системы управления: вычислительные (в большинстве рассматриваемых случаев), энергетические и др.;

$S$  — количество портов ввода и вывода, связывающих элементы системы (акторы), которое может задаваться как условие (в виде максимально допустимого количества, устанавливаемого исходя из требований сложности технической реализации), либо определяться по итогам оптимизации;

$a$  — актор — элемент системы управления, отождествляемый с определенной функцией (набором функций), который может иметь два варианта индексов ( $a_j$  — для исходного множества композиции и  $a_{upi}$  — для определения позиции

в конфигурации: группа  $u$ , поток  $p$  и номер  $i$  в потоке) и определяется парой параметров  $\{t, w\}$  — длительностью выполнения актора  $t$  и потребными для него ресурсами  $w$ ;

$Q = Q(\{a_j\}, K_Q)$  — композиция акторов, определяемая как совокупность двух множеств: исходного множества акторов  $\{a_j\}$  и матрицы  $K_Q$  отношений акторов исходного множества, задающей попарные отношения акторов по последовательности их выполнения; множество значений, принимаемых элементом матрицы  $K_Q$  ( $j_1, j_2\} \in \{-1, 0, 1\}$ , где «-1» соответствует случаю, когда актор  $j_2$  выполняется до  $j_1$ ; «1» соответствует случаю, когда актор  $j_2$  выполняется после  $j_1$ ; «0» соответствует случаю, когда порядок выполнения акторов не имеет значения;

$R = R(\{a_{upi}\}, Z_R)$  — конфигурация цикла, определяемая как совокупность двух множеств: множества  $\{a_{upi}\}$  акторов в данной конфигурации, представляющего собой трехмерный (по группе, потоку и номеру в потоке) массив акторов, задаваемых в виде их индексов в исходном множестве композиции, и матрицы  $Z_R$  портовости акторов, определяющей для каждого актора в данной конфигурации цикла минимально необходимые каналы связи с другими акторами и представляющей собой трехмерный массив с элементами в виде двумерных массивов;

$R$  — операторы конфигурирования, означающие получение конфигурации цикла системы управления на основе ее композиции, имеющие четыре варианта:  $R(Q, \{\tau\}, W, S)$  — конфигурирование по критерию минимизации длительности цикла при установленных ограничениях по доступным ресурсам системы и допустимому числу портов;  $R(Q, \{W\}, \tau, S)$  — конфигурирование по критерию минимизации требуемых ресурсов системы при установленных ограничениях по длительности цикла и допустимому числу портов;  $R(Q, \{S\}, \tau, W)$  — конфигурирование по критерию минимизации необходимого числа портов при установленных требованиях к длительности цикла и ограничениях по доступным ресурсам системы;  $R(Q, \tau, W, S, 1)$  — конфигурирование при установленных требованиях к длительности цикла, ограничениях по доступным ресурсам системы и числу портов, обеспечивающее получение первой годной конфигурации.

Последний вариант конфигурирования может быть востребован в случае динамического конфигурирования, при котором конфигурирование цикла производится во время его выполнения. В результате полноценная оптимизация не представляется возможной, однако возможен поиск годного варианта, занимающий существенно меньше времени.

При необходимости возможно расширение функционала данного варианта конфигури-

рования за счет введения критерия сравнения нескольких годных вариантов (по  $\tau$ ,  $W$  или  $S$ ) и выбора из них. В этом случае оператор конфигурирования имеет вид  $\mathbf{R}(Q, \tau, W, S, n, f)$ , где  $n$  — число рассматриваемых годных конфигураций;  $f \in \{\tau, W, S\}$  — критерий выбора из годных.

### Аксиома и формулы

Приведенное выше выражение  $R = R(\{a_{upi}\}, \mathbf{Z}_R)$ , раскрывающее содержание конфигурации цикла, является (единственной) аксиомой, на которой базируется формальная теория систем реального времени. Согласно этой аксиоме, цикл системы управления определяется ее конфигурацией  $R$ , которая характеризуется распределением связанных между собой (через порты ввода и вывода данных) акторов  $a$ , обладающих длительностью выполнения и потребностью в ресурсах, по группам ( $u$ ), потокам ( $p$ ) и последовательности ( $i$ ) в потоке. Данная аксиома представляет собой выбор формы представления цикла системы реального времени. При этом доказательство правильности или единственности выбранного представления в рамках самой формальной теории не предполагается.

Центральной формулой формальной теории систем реального времени является формула для определения длительности цикла для заданной  $g$ -й конфигурации [22]

$$\tau(R_g) = \sum_{u=1}^{u_{\max}} \max \left( \bigcup_{p=1}^{p_{u\max}} \sum_{i=1}^{i_{up\max}} t_{upi} \right), \quad (1)$$

где  $u_{\max}$  — количество групп акторов;  $p_{u\max}$  — количество потоков в  $u$ -й группе;  $i_{up\max}$  — количество акторов в  $p$ -м потоке  $u$ -й группы.

Длительность цикла может определяться при конфигурировании цикла по критерию минимизации длительности цикла (для заданной композиции) либо задаваться как начальное условие в виде ограничения на максимальную длительность цикла. Это ограничение определяется требованиями к точности воспроизводства параметров объекта управления. Для систем управления движением максимальная длительность цикла управления, как показали исследования авторов, не должна превышать отношения допустимой погрешности позиционирования подвижного рабочего органа к скорости его движения.

Минимальная длительность цикла для заданной композиции

$$\tau(Q) = \min \left( \bigcup_{g=1}^{g_{\max}} \tau(R_g) \right), \quad (2)$$

где  $g_{\max}$  — количество возможных конфигураций для заданной композиции.

Задание ограничения на максимальную длительность цикла управления выглядит следующим образом:

$$\tau(Q) \leq \tau_{\max}. \quad (3)$$

Наряду с длительностью цикла управления конфигурация также характеризуется потребными ресурсами и необходимым количеством портов ввода и вывода. Максимальная потребность в ресурсах за цикл для заданной конфигурации

$$W(R_g) = \max(W_u(R_g, t), u \leq u_{\max}, t \leq \tau(R_g)), \quad (4)$$

где  $W_u(R_g, t) = \sum_{p=1}^{p_{u\max}} w_{upi}$  — потребные ресурсы для выполнения акторов  $u$ -й группы;  $w_{upi}$  — потребные ресурсы для актора  $a_{upi}$ , а индекс  $i$  определяется из системы уравнений

$$\begin{cases} t_0 + \sum_{q=1}^{i-1} t_{upq} \leq t < t_0 + \sum_{q=1}^i t_{upq} \\ t_0 = \sum_{v=1}^u \max \left( \bigcup_{p=1}^{p_{v\max}} \sum_{i=1}^{i_{vp\max}} t_{vpi} \right) \end{cases}$$

Минимальная потребность в ресурсах за цикл для заданной композиции

$$W(Q) = \min \left( \bigcup_{g=1}^{g_{\max}} W(R_g) \right). \quad (5)$$

Необходимое число портов одного актора для заданной конфигурации [23]

$$\begin{aligned} s_{upi} = & \sum_{v=1}^{u-1} \sum_{\eta=1}^{p_{u\max}} \min \left( - \bigcup_{\lambda=1}^{i_{u\eta\max}} \mathbf{K}_Q(H(u\eta i), H(v\eta \lambda)), 0 \right) + \\ & + \sum_{\eta=1}^{p_{u\max}} \left( \max \left( \bigcup_{\lambda=1}^{i_{u\eta\max}} \mathbf{K}_Q(H(u\eta i), H(u\eta \lambda)), 0 \right) - \right. \\ & \left. - \min \left( \bigcup_{\lambda=1}^{i_{u\eta\max}} \mathbf{K}_Q(H(u\eta i), H(u\eta \lambda)), 0 \right) \right) + \\ & + \sum_{v=1}^{u+1} \sum_{\eta=1}^{p_{u\max}} \max \left( \bigcup_{\lambda=1}^{i_{u\eta\max}} \mathbf{K}_Q(H(u\eta i), H(v\eta \lambda)), 0 \right), \quad (6) \end{aligned}$$

где  $H(\dots)$  — функция преобразования позиции актора в конфигурации в номер этого актора в исходном множестве композиции.

Необходимое число портов всех акторов для заданной конфигурации

$$S(R_g) = \sum_{u=1}^{u_{\max}} \sum_{p=1}^{p_{u\max}} \sum_{i=1}^{i_{up\max}} s_{upi}. \quad (7)$$



Минимальное число портов всех акторов для заданной композиции

$$S(Q) = \min \left( \bigcup_{g=1}^{g_{\max}} S(R_g) \right). \quad (8)$$

Наряду с формулами, определяющими отношения между основными понятиями формальной теории, представленными в виде символов, в множество формул также могут быть включены рассмотренные ранее операторы конфигурирования, представляющие собой функции преобразования композиции во множество конфигураций или в выбранную конфигурацию, удовлетворяющую заданным требованиям.

Операторы конфигурирования не могут быть выполнены в виде вычислений по заданным формулам. Конфигурирование осуществляется посредством специальных алгоритмов, в рамках которых построение каждой конфигурации выполняется в виде последовательных операций. При этом альтернативные конфигурации могут выстраиваться параллельно.

К числу основных алгоритмов относятся: алгоритм конфигурирования по критерию длительности ( $\tau$ ), алгоритм конфигурирования по критерию ресурсов ( $W$ ), алгоритм конфигурирования по критерию портовости акторов ( $S$ ) и алгоритм поиска лучшей по критерию  $f$  из  $n$  годных конфигураций:

$$\begin{aligned} R_Q^\tau &= R(Q, \{\tau\}, W_{\max}, S_{\max}); \\ R_Q^W &= R(Q, \tau_{\max}, \{W\}, S_{\max}); \\ R_Q^S &= R(Q, \tau_{\max}, W_{\max}, \{S\}); \\ R_Q^{nf} &= R(Q, \tau_{\max}, W_{\max}, S_{\max}, n, f), \end{aligned} \quad (9)$$

где  $R_Q^\tau, R_Q^W, R_Q^S$  — наилучшая конфигурация композиции при оптимизации по критериям  $\tau, W, S$ ;  $R_Q^{nf}$  — лучшая по критерию  $f$  из найденных  $n$  годных конфигураций композиции, удовлетворявших всем ограничениям.

Возможные варианты выполнения указанных алгоритмов были предложены в работах [22, 23]. Как показали исследования, запись алгоритмов осуществляется с использованием исключительно стандартных символов.

Правила вывода формул формальной теории систем реального времени соответствуют логике третьего порядка, оперирующей символами переменных (например, индексами акторов) и служебными символами (скобками и т. д.), логическими операторами (конъюнкцией и др.) и кванторами применительно не только к переменным, но также ко множествам и множествам множеств [24]. В формальной теории систем реального времени множества образуют акторы (в составе композиции и в составе конфигура-

ции), коэффициенты их отношений, матрицы портовости, варианты конфигурации для заданной композиции и др. В рамках логики как первого, так и высших порядков правилами вывода являются *modus ponens*  $\left( \frac{A, A \rightarrow B}{B} \right)$  и правило обобщения  $\left( \frac{A(x)}{\forall x A(x)} \right)$ . Соответственно, эти же правила определяют вывод формул в формальной теории систем реального времени.

Конкретными правилами, определенными в рамках формальной теории систем реального времени, являются уже приведенные нами выше правило определения композиции акторов  $Q = Q(\{a_j\}, K_Q)$ , правило определения конфигурации цикла  $R = R(\{a_{up_i}\}, Z_R)$  (также являющееся аксиомой), а также правила определения наилучших конфигураций (9).

## Вычислимость и реализуемость композиции

Одним из результатов проведенных ранее исследований [23] является выдвижение представления о характеристической функции, бинарно описывающей осуществимость вычислительной системы. Характеристическая функция представляет собой произведение трех вспомогательных функций: вычислимости, реализуемости и доверия. Функция доверия относится к уровню технической реализации вычислительной системы, поэтому оставим ее за пределами анализа в данной статье и обратим внимание на две оставшиеся функции.

Рассмотренная в данной статье формальная теория систем реального времени может использоваться в качестве инструмента фактической верификации вычислительной системы на предмет ее реализуемости. Однако определение реализуемости вычислительной системы реального времени — задача, решаемая после определения вычислимости композиции, обеспечивающей выполнение вычислительной системой своего функционального назначения.

Необходимо констатировать, что поскольку множество переменных (связанных с акторами), определяющих вычислительную систему, перечислимо, то композиция всегда вычислима. Это, однако, не означает, что она вычислима за приемлемый (для выполнения вычислительной системой своего функционального назначения) интервал времени. Данная композиция может не иметь годной конфигурации, удовлетворяющей требованиям выполнения вычислительной системой своего функционального назначения.

Конфигурирование цикла системы реального времени может осуществляться без ограничений

на длительность выполнения конфигурирования или с установленными ограничениями на длительность конфигурирования. Первому варианту соответствует статическое конфигурирование, второму — динамическое. Необходимость в динамическом конфигурировании возникает в случае корректирования конфигурации во время выполнения цикла системы реального времени. Обновление конфигурации при этом происходит в продолжение одного или жестко лимитированного числа циклов.

Наличие для заданной композиции годной конфигурации означает, что статическое конфигурирование возможно. Соответственно, обеспечивается реализуемость композиции с фиксированной (заранее определенной) конфигурацией.

Реализуемость композиции в случае динамического конфигурирования — задача, не всегда имеющая решение. Это обусловлено чрезвычайно высокой сложностью оптимизационных алгоритмов конфигурирования, предполагающих сравнительный анализ всех возможных вариантов конфигурации по заданным критериям оптимизации (длительность цикла, потребные ресурсы, необходимое количество портов для взаимодействия акторов). Как показали исследования [23], минимальная сложность оптимизационной задачи, достижимая при использовании лучших алгоритмов сортировки [25, с. 206–207], соответствует нотации  $O(n! \cdot 2^{n-1} \cdot \log n)$ , где  $n$  — количество акторов на этапе с наибольшим количеством акторов.

Некоторое снижение сложности задачи конфигурирования может быть достигнуто в случае отказа от оптимизации и использования вместо нее поиска годного варианта (первого годного или лучшего из нескольких годных, отбираемого по заданному критерию). Однако даже в этом случае успешность конфигурирования не гарантирована. Неизбежным может оказаться увеличение длительности цикла (если это возможно), наращивание ресурсов системы и ее портовости либо совершенствование используемых акторов в направлении повышения их быстродействия и снижения ресурсоемкости. При этом решается задача минимизации коррекции задания [11].

## Выводы

Резюмируем проведенное в статье исследование.

1. Одним из актуальных направлений развития информационных технологий в последние десятилетия является создание систем управления реального времени. Это обуславливает востребованность формализации таких систем, в том числе в виде формальной теории, удовлетворяющей существующим требованиям к таким теориям.

2. Объектом описания формальной теории систем реального времени является цикл системы реального времени, представляющий собой реализацию композиции элементов (функций), необходимой и достаточной для выполнения системой своей функции.

3. Для описания цикла системы управления в рамках формальной теории используются как стандартные символы, так и специальные, вводимые формальной теорией. К числу параметров, для которых используются специальные символы, относятся длительность цикла, общие доступные ресурсы системы, количество портов ввода и вывода, акторы с различными индексами, композиция и конфигурация акторов, а также операторы конфигурирования.

4. Формулы формальной теории систем реального времени служат определению для заданной композиции акторов конфигураций, удовлетворяющих определенным требованиям по длительности цикла, потребным ресурсам и количеству портов ввода и вывода. Кроме того, формулы формальной теории позволяют определять достижимые значения длительности цикла, потребных ресурсов и количества портов для любой заданной конфигурации и всех возможных конфигураций заданной композиции.

5. Правила вывода формул предлагаемой формальной теории соответствуют логике третьего порядка. Это обусловлено тем, что в рамках теории необходимо оперировать символами, операторами и кванторами не только применительно к переменным, но и к множествам и множествам множеств.

6. Формальная теория систем реального времени является одним из инструментов определения осуществимости вычислительной системы. Совместный анализ связанных с формальной теорией функции реализуемости и функции вычислимости композиции выявляет двойственность вычислимости, всегда достижимой для систем реального времени в случае статического конфигурирования, но не всегда возможной при динамическом конфигурировании, когда конфигурация цикла управления переопределяется в процессе выполнения цикла. Это обуславливает специальный алгоритм динамического конфигурирования, а также может требовать корректирования исходных ограничений по длительности цикла, используемым ресурсам, портам, быстродействию и ресурсоемкости акторов-функций, формирующих композицию.

## Финансовая поддержка

Исследование выполнено при поддержке Российского научного фонда по гранту № 24-19-00692, <http://rscf.ru/project/24-19-00692/>

## Литература

1. Парфенов А. В., Чудинов С. М. Тенденции развития технологии вычислительной техники. *Научные ведомости Белгородского государственного университета. Сер. Экономика. Информатика*, 2016, № 16 (237), вып. 39, с. 98–106. EDN: WXTRPX. <http://dspace.bsuedu.ru/handle/123456789/59980> (дата обращения: 01.05.2025).
2. Якобовский М. В., Корнилина М. А. Развитие суперкомпьютерных технологий в ИММ РАН и ИПМ им. М.В. Келдыша РАН. *Computational Mathematics and Information Technologies*, 2024, № 8(1), с. 12–28. doi:10.23947/2587-8999-2024-8-1-12-28
3. Ильясов Б. Г., Макарова Е. А., Закиева Е. Ш., Габдуллина Э. Р. Методологические основы моделирования и интеллектуального управления промышленным комплексом как сложным динамическим многоагентным объектом. *Современные наукоемкие технологии*, 2020, № 11-2, с. 288–293. doi:10.17513/snt.38376, EDN: NYNXMO
4. Knyazeva H. Strategies of dynamic complexity management. *Foresight and STI Governance*, 2020, vol. 14, no. 4, pp. 34–45. doi:10.17323/2500-2597.2020.4.34.45
5. Burns A. Scheduling hard real-time systems: A review. *Software Engineering Journal*, 1991, vol. 6, iss. 3, pp. 116–128. doi:10.1049/sej.1991.0015
6. Беркс А., Голдстейн Г., Нейман Дж. Предварительное рассмотрение логической конструкции электронного вычислительного устройства. *Кибернетический сборник. Вып. 9: сб. переводов: пер. с англ./ред. А. А. Ляпунов, О. Б. Лупанов. М., Мир, 1964, с. 7–67.*
7. Attig C., Rauh N., Franke T., Krems J. F. *System latency guidelines then and now – is zero latency really considered necessary?* Engineering Psychology and Cognitive Ergonomics/ Ed. D. Harris. Part II, LNAI 10276. Cham, Switzerland, Springer International Publishing AG, 2017, pp. 2–14. doi:10.1007/978-3-319-58475-1\_1
8. Малащенко А. В., Нисневич Ю. А., Рябов А. В. Становление постиндустриальной общепланетарной цивилизации. *Вопросы теоретической экономики*, 2020, № 2, с. 86–98. doi:10.24411/2587-7666-2020-10205, EDN: CKZNQG
9. Грибков А. А. Человек в цивилизации когнитивных технологий. *Философия и культура*, 2024, № 1, с. 22–33. doi:10.7256/2454-0757.2024.1.69678, EDN: KAPIMN. [https://nbpublish.com/library\\_read\\_article.php?id=69678](https://nbpublish.com/library_read_article.php?id=69678) (дата обращения: 01.05.2025).
10. Литвин В. Г., Чернышова Е. В., Литвин Ю. В. Оценка показателей работы вычислительной системы реального времени, содержащей узлы с параллельным выполнением процессов. *Информационные технологии и вычислительные системы*, 2014, № 2, с. 16–25. EDN: SGRVHP
11. Furugyan M. G. Designing a real-time computing system with specified characteristics. *Systems Analysis and Operations Research*, 2025, no. 64, pp. 63–71. doi:10.1134/S1064230725700054
12. Buttazzo G. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Fourth Ed. Springer, 2024. 509 p.
13. Takado K., Yokoyama T., Yoo M. A real-time operating system for physical and logical time-triggered distributed computing. *Proceedings of the 2024 6th Asia Pacific Information Technology Conference (APIT '24)*, 2024, pp. 78–85. doi:10.1145/3651623.3651635
14. Khan S. Real-Time Operating System (RTOS) with different application: A systematic mapping. *European Journal of Engineering and Technology Research*, 2021, vol. 6, no. 1, pp. 100–103. doi:10.24018/ejers.2021.6.1.2322
15. Lange S., Grieser G., Jantke K. P. Advanced elementary formal systems. *Theoretical Computer Science*, 2003, no. 298, pp. 51–70. doi:10.1016/S0304-3975(02)00418-8
16. Соколенко М. Формальная система в полном объеме. *Ученые записки Таврического национального университета им. В. И. Вернадского. Сер. Философия. Культурология. Политология. Социология*, 2013, т. 26 (65), № 4, с. 384–388.
17. Свиридов А. С. Конфигурирование информационных систем с точки зрения систем управления. *Известия ЮФУ. Технические науки*, 2014, № 6 (155), с. 168–172. EDN: SFLDLL
18. Зеленский А. А., Грибков А. А. Конфигурирование память-ориентированной системы управления движением. *Программные системы и вычислительные методы*, 2024, № 3, с. 12–25. doi:10.7256/2454-0714.2024.3.71073, EDN: TTQBB. [https://nbpublish.com/library\\_read\\_article.php?id=71073](https://nbpublish.com/library_read_article.php?id=71073) (дата обращения: 01.05.2025).
19. Ежова Н. А., Соколинский Л. Б. Обзор моделей параллельных вычислений. *Вестник ЮУрГУ. Сер. Вычислительная математика и информатика*, 2019, т. 8, № 3, с. 58–91. doi:10.14529/cmse190304, EDN: NGIIVA
20. Rinaldi L., Torquati M., Mencagli G., Danelutto M., Menga T. Accelerating actor-based applications with parallel patterns. *Proceedings of the 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, 2019, Pavia, Italy, 13–15 Feb. 2019. doi:10.1109/EMPDP.2019.8671602
21. Shah V., Vaz Salles M. A. Reactors: A case for predictable, virtualized actor database systems. *SIGMOD '18: Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 259–274. doi:10.1145/3183713.3183752
22. Зеленский А. А., Грибков А. А. Значение фактора портовости для конфигурирования цикла акторной системы управления. *Моделирование, оптимизация и информационные технологии*, 2025, № 13(1). doi:10.26102/2310-6018/2025.48.1.037, EDN: YEZNVD. <https://moitvvt.ru/ru/journal/pdf?id=1839>



23. Зеленский А. А., Грибков А. А. Характеристическая функция акторной вычислительной системы. *Моделирование, оптимизация и информационные технологии*, 2024, № 12(4). doi:10.26102/2310-6018/2024.47.4.015, EDN: FRGGNU. <https://moitvivr.ru/ru/journal/pdf?id=1722> (дата обращения: 01.05.2025).

24. Hetzl S. *Higher-Order Logic*. Vienna University of Technology, January 25, 2019. <https://dmg.tuwien.ac.at/hetzl/teaching/hol.20190125.pdf> (дата обращения: 01.05.2025).

25. Кнут Д. Э. Искусство программирования. Т. 3. *Сортировка и поиск*. М., Вильямс, 2018. 832 с.

UDC 004.032.3

doi:10.31799/1684-8853-2025-5-2-10

EDN: JNQQQK

### Fundamentals of the formal theory of real-time systems

A. A. Zelenskii<sup>a</sup>, PhD, Tech., Associate Professor, orcid.org/0000-0002-3464-538X

A. A. Gribkov<sup>a</sup>, Dr. Sc., Tech., Leading Researcher, orcid.org/0000-0002-9734-105X, andarmo@yandex.ru

<sup>a</sup>Scientific-Manufacturing Complex «Technological Centre», 1, bld. 7, Shokina Sq., 124498, Zelenograd, Moscow, Russian Federation

**Introduction:** The importance and expressed specificity of real-time computing systems determines the demand for the construction of a strictly formalized system abstracted from the outside world, including a necessary and sufficient set of determining parameters. **Purpose:** To create a formal theory of computing systems operating under time constraints. **Results:** We propose a new formal theory of real-time systems, which allows formalizing the representation of real-time control systems. We expect it will expand the possibilities of the analysis and synthesis of these systems. The object of the proposed formal theory is the cycle of a real-time system. The cycle is defined on the basis of a given composition of functions which ensure the fulfillment of the system's functional purpose, and is represented as a configuration formed of actors which are distributed by groups, execution threads and the order of execution in each thread. The formal theory of real-time systems includes a set of standard and special symbols and formulas. These symbols and formulas are used to determine the configuration options that meet the specified requirements for a given actor composition; in addition, for each of these configurations, the achievable cycle times, required resources, and the number of input and output ports for actor communication are determined. In conclusion, we define the place and role of the formal theory of real-time systems in a feasibility study and state the possibility of two configuration options: static and dynamic. The latter may be necessary to redetermine a cycle configuration during its execution.

**Keywords** – formal theory, real time, control system, composition, configuration, cycle, resources, ports, feasibility.

**For citation:** Zelenskii A. A., Gribkov A. A. Fundamentals of the formal theory of real-time systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2025, no. 5, pp. 2–10 (In Russian). doi:10.31799/1684-8853-2025-5-2-10, EDN: JNQQQK

### Financial support

The research was supported by the Russian Science Foundation under grant No. 24-19-00692, <http://rscf.ru/project/24-19-00692/>

### References

1. Parfenov A. V., Chudinov S. M. Tendencies of development of computer technology. *Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta. Ser. Ekonomika. Informatika*, 2016, no. 16 (237), iss. 39, pp. 98–106. EDN: WXTRPX. Available at: <http://dspace.bsuedu.ru/handle/123456789/59980> (accessed 1 May 2025) (In Russian).
2. Yakobovskiy M. V., Kornilina M. A. Development of super-computer technologies at the Institute of Mathematical Modelling and Keldysh Institute of Applied Mathematics of Russian Academy of Sciences. *Computational Mathematics and Information Technologies*, 2024, vol. 8, no. 1, pp. 12–28. <https://doi.org/10.23947/2587-8999-2024-8-1-12-28> (In Russian).
3. Ilyasov B. G., Makarova E. A., Zakieva E. Sh., Gabdullina E. G. Methodological foundations of modeling and intelligent management of an industrial complex as a complex dynamic multi-agent object. *Modern High Technologies*, 2020, no. 11-2, pp. 288–293 (In Russian). <https://doi.org/10.17513/snt.38376>, EDN: NYNXMO
4. Knyazeva H. Strategies of dynamic complexity management. *Foresight and STI Governance*, 2020, vol. 14, no. 4, pp. 34–45. <https://doi.org/10.17323/2500-2597.2020.4.34.45>
5. Burns A. Scheduling hard real-time systems: A review. *Software Engineering Journal*, 1991, vol. 6, iss. 3, pp. 116–128. <https://doi.org/10.1049/sej.1991.0015>
6. Burks A. W., Goldstein H. H., von Neumann J. Preliminary discussion of the logical design of an electronic computing instrument. In: von Neumann J. *Collected Works*. Vol. V. Pergamon Press, Oxford – Lnd. – N. Y. – Paris, 1963, pp. 34–79.
7. Attig C., Rauh N., Franke T., Krems J. F. *System latency guidelines then and now – is zero latency really considered necessary?* In: *Engineering Psychology and Cognitive Ergonomics*. Ed. D. Harris. Part II, LNAI 10276. Cham, Switzerland, Springer International Publishing AG, 2017, pp. 2–14. [https://doi.org/10.1007/978-3-319-58475-1\\_1](https://doi.org/10.1007/978-3-319-58475-1_1)
8. Malashenko A. V., Nisnevich Yu. A., Ryabov A. V. Formation of post-industrial all-planetary civilization. *Issues of Economic Theory*, 2020, no. 2, pp. 86–98 (In Russian). <https://doi.org/10.24411/2587-7666-2020-10205>, EDN: CKZNRG
9. Gribkov A. A. Human beings in the civilization of cognitive technologies. *Philosophy and Culture*, 2024, no. 1, pp. 22–33. <https://doi.org/10.7256/2454-0757.2024.1.69678>, EDN: KAPIMN. Available at: [https://nbpublish.com/library\\_read\\_article.php?id=69678](https://nbpublish.com/library_read_article.php?id=69678). (accessed 1 May 2025) (In Russian).
10. Litvin V. G., Chernyshova E. V., Litvin Yu. V. Assessment index of the realtime computing system with parallel performance of work. *Journal of Information Technologies and Computing Systems*, 2014, no. 2, pp. 16–25 (In Russian). EDN: SGRVHP
11. Furugyan M. G. Designing a real-time computing system with specified characteristics. *Systems Analysis and Operations Research*, 2025, no. 64, pp. 63–71. doi:10.1134/S1064230725700054
12. Buttazzo G. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Fourth Ed. Springer, 2024. 509 p.
13. Takado K., Yokoyama T., Yoo M. A real-time operating system for physical and logical time-triggered distributed computing. *Proceedings of the 2024 6th Asia Pacific Information*



- Technology Conference (APIT '24), 2024, pp. 78–85. doi:10.1145/3651623.3651635
14. Khan S. Real-Time Operating System (RTOS) with different application: A systematic mapping. *European Journal of Engineering and Technology Research*, 2021, vol. 6, no. 1, pp. 100–103. doi:10.24018/ejers.2021.6.1.2322
15. Lange S., Grieser G., Jantke K. P. Advanced elementary formal systems. *Theoretical Computer Science*, 2003, no. 298, pp. 51–70. doi:10.1016/S0304-3975(02)00418-8
16. Sokolenko M. Full-content formal system. *Scientific Notes of Taurida National V. I. Vernadsky University. Series: Philosophy. Culturology. Political sciences. Sociology*, 2013, vol. 26 (65), no. 4, pp. 384–388 (In Russian).
17. Sviridov A. S. Configuration information systems in terms of management systems. *Izvestiya SFedU. Engineering Sciences*, 2014, no. 6 (155), pp. 168–172 (In Russian). EDN: SFLDLL
18. Zelenskii A. A., Gribkov A. A. Configuration of memory-oriented motion control system. *Software Systems and Computational Methods*, 2024, no. 3, pp. 12–25. <https://doi.org/10.7256/2454-0714.2024.3.71073>, EDN: TTQBB. Available at: [https://nbpublish.com/library\\_read\\_article.php?id=71073](https://nbpublish.com/library_read_article.php?id=71073) (accessed 1 May 2025) (In Russian).
19. Ezhova N. A., Sokolinsky L. B. Survey of Parallel Computation Models. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*, 2019, vol. 8, no. 3, pp. 58–91 (In Russian). <https://doi.org/10.14529/cmse190304>, EDN: NGIIVA
20. Rinaldi L., Torquati M., Mencagli G., Danelutto M., Menga T. Accelerating actor-based applications with parallel patterns. *Proceedings of the 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, 2019, Pavia, Italy. doi:10.1109/EMPDP.2019.8671602
21. Shah V., Vaz Salles M. A. Reactors: A case for predictable, virtualized actor database systems. *SIGMOD '18: Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 259–274. <https://doi.org/10.1145/3183713.3183752>
22. Zelenskii A. A., Gribkov A. A. An importance of the portability factor for configuring the cycle of real-time actuator control system. *Modeling, Optimization and Information Technology*, 2025, no. 13(1). <https://doi.org/10.26102/2310-6018/2025.48.1.037>, EDN: YEZNVD. Available at: <https://moitvvt.ru/ru/journal/pdf?id=1839> (accessed 1 May 2025) (In Russian).
23. Zelenskii A. A., Gribkov A. A. Characteristic function of an actor computing system. *Modeling, Optimization and Information Technology*, 2024, no. 12(4). <https://doi.org/10.26102/2310-6018/2024.47.4.015>, EDN: FRGGNU. Available at: <https://moitvvt.ru/ru/journal/pdf?id=1722> (accessed 1 May 2025) (In Russian).
24. Hetzl S. *Higher-Order Logic*. Vienna University of Technology. January 25, 2019. Available at: <https://dmgtuwien.ac.at/hetzl/teaching/hol.20190125.pdf> (accessed 1 May 2025).
25. Knuth D. E. *Iskusstvo programmirovaniya. T. 3. Sortirovka i poisk* [The Art of Programming. Vol. 3. Sorting and searching]. Moscow, Vil'yams Publ., 2018. 832 p. (In Russian).