ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2020, № 3, с. 50–63

КИБЕРБЕЗОПАСНОСТЬ В СИСТЕМАХ УПРАВЛЕНИЯ

УДК 519.711.3

УПРАВЛЕНИЕ ДИНАМИЧЕСКОЙ ИНФРАСТРУКТУРОЙ СЛОЖНЫХ СИСТЕМ В УСЛОВИЯХ ЦЕЛЕНАПРАВЛЕННЫХ КИБЕРАТАК[1]

© 2020 г. Д. П. Зегжда[a], Д. С. Лаврова[a,]*, Е. Ю. Павленко[a]

[a] СПбПУ Петра Великого, Санкт-Петербург, Россия

**e-mail: lavrova@ibks.spbstu.ru*

Поступила в редакцию 05.06.2019 г.
После доработки 20.08.2019 г.
Принята к публикации 30.09.2019 г.

Рассматривается проблема управления динамической инфраструктурой сложных систем в условиях целенаправленных кибератак. Предложен подход к управлению, состоящий в нахождении за минимальное время сценария саморегуляции, устраняющего деструктивное воздействие за счет нахождения альтернативного маршрута реализации целевой функции. Предложено характеризовать процесс саморегуляции, оценивая устойчивость системы к кибератакам. Предложены подходы к оценке устойчивости для систем с централизованным и децентрализованным управлением.

DOI: 10.31857/S0002338820020134

Введение. Современные технологии привели к появлению сложных, крупномасштабных систем, компоненты которых имеют доступ к сети Интернет и способны обмениваться данными друг с другом посредством использования сетевых технологий. Как правило, такие системы имеют динамическую инфраструктуру (таких, как беспилотная авиация, робототехнические системы), и управление ими приходится осуществлять в условиях целенаправленного информационного деструктивного воздействия (кибератаки). Кибератаки способны удаленным образом нарушить процесс управления или вмешаться в него для навязывания своих алгоритмов управления, примеры чего в современном мире встречаются весьма часто [1, 2].

Практика показывает нарастающий объем кибератак на сложные системы с динамической инфраструктурой (ССДИ). Динамическая инфраструктура сложных систем определяется переменным составом среды системы (возможностью активации или отключения каких-либо узлов системы) и высокой вариативностью маршрутов передачи данных между входными и выходными узлами системы. Сетевая архитектура таких систем может быть не только клиент-серверной, но и одноранговой, в таком случае процесс передачи данных между узлами может быть вариативным. Значительная часть атак направлена на промышленные ССДИ: в период с 2012 по 2016 г. количество инцидентов безопасности в промышленных системах выросло более чем в 2 раза и почти достигло отметки в 300 инцидентов безопасности за год, при этом более половины из них относятся к критическим отраслям жизнедеятельности, таким, как энергетика, оборонная промышленность, связь, здравоохранение и транспорт [3, 4].

Среди получивших за последние годы широкую огласку инцидентов нарушения безопасности ССДИ наиболее критичными являлись атаки на систему диспетчеризации службы 911 в г. Балтимор, США (март 2018 г.), на нефтехимический завод в Саудовской Аравии (август 2017 г.), а также массовая атака, нацеленная на значимые объекты критической информационной инфраструктуры (КИИ) РФ (6 апреля 2018 г.).

С каждым годом число кибератак на объекты промышленной инфраструктуры продолжает расти: по данным Лаборатории Касперского, доля атакованных компьютеров промышленных систем (автоматизированных систем управления промышленными процессами) в первом полугодии 2018 г. в мире выросла на 3.5% и составила 41.2% [5]. За год этот показатель увеличился на 4.6% в соответствии с рис. 1.

[1] Работа выполнена в рамках гранта Президента РФ для государственной поддержки ведущих научных школ Российской Федерации НШ-2992.2018.9 (соглашение 075-15-2019-1066).

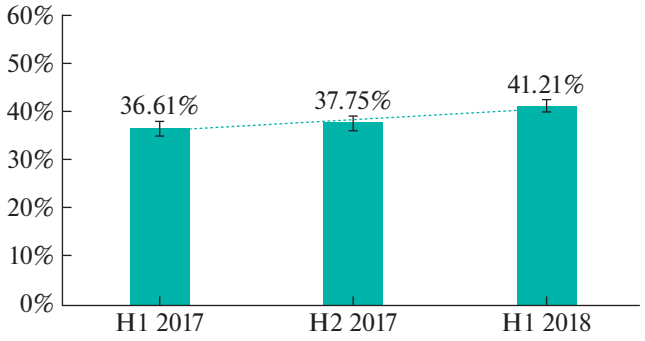

Рис. 1. Рост доли атакованных компьютеров промышленных систем в первом полугодии 2018 г. по данным АО "Лаборатория Касперского"

Без решения этой проблемы управление современными сложными системами, как правило, имеющими критическое назначение, в условиях целенаправленных кибератак становится невозможным [6]. Сложившееся положение ставит перед специалистами по управлению новую задачу, состоящую в сохранении работоспособности сложных систем в условиях целенаправленного и интеллектуального деструктивного управляющего воздействия, которое и представляет собой кибератака. Эта задача отличается от задач, которые решала до настоящего времени теория управления, в связи с тем, что возмущающий фактор носит не природный, а искусственный и, более того, интеллектуальный характер, поскольку в пределе представляет собой такое же информационное воздействие, сгенерированное противоборствующей системой управления, причем, в общем случае, более сложной и интеллектуальной.

Кибератака представляет собой деструктивное информационное воздействие на цифровое управление исполнительными механизмами [2, 7], цель которого – получение возможности управления работой системы и протекающими в ней физическими процессами, а не нарушение конфиденциальности, целостности или доступности информации. Выход необратимых физических процессов из-под контроля может привести к катастрофическим последствиям.

Главной задачей управления в условиях кибератак является поддержание ССДИ в работоспособном состоянии. Управление может заключаться во внесении изменений в параметры протекающего процесса или в параметры динамической инфраструктуры системы, в которой эти процессы выполняются. Для управления системой в условиях целенаправленных кибератак необходимо вносить изменения именно в параметры инфраструктуры, поскольку ее компоненты (зачастую доступные из сети Интернет) и являются объектом кибератак.

ССДИ представляют собой гибридную систему с изменяющейся структурой, содержащей переменное число компонентов, связанных друг с другом и охваченных общим информационным контуром. Математическое описание имеет высокую сложность, в связи с чем построение алгоритмов управления по параметрам для таких систем затруднено. Для решения задачи управления такими системами в условиях кибератак авторы предлагают подход к управлению их инфраструктурой по состоянию, исходя из избыточности компонентов таких систем, дублирования коммуникационных связей и функциональной взаимозаменяемости компонентов, позволяющих в случаях кибератак осуществить реконфигурацию системы для сохранения возможности реализации рабочего процесса.

1. Постановка задачи. Предлагается подход к концепции управления ССДИ как к регуляции инфраструктуры объекта на основе интеллектуального выбора алгоритма реконфигурации схемы взаимодействия компонентов для устранения последствий кибератаки.

1.1. Описание предлагаемого подхода. Подход распространяется на распространенный класс сложных систем с динамической инфраструктурой, например, на системы Интернета вещей, многоагентные системы, автоматизированные системы управления технологическими процессами, интеллектуальные роботизированные комплексы и т.д. [8], и включает:

представление рабочего процесса в виде маршрута на множестве связей между компонентами систем, маршрут вариативен за счет избыточности и взаимозаменяемости компонентов;

систематизацию кибервоздействий и их атрибуцию по точке входа и виду изменения маршрута;

разработку алгоритма управления, осуществляющего саморегуляцию систем, позволяющего восстановить или построить нарушенные связи для сохранения системы в допустимом пространстве состояний;

оценку устойчивости ССДИ к кибератакам путем определения временных затрат на нейтрализацию кибератаки для сокращения переходных режимов.

Учитывая, что кибератака является интеллектуальным воздействием, распределенным во времени с непредсказуемой точкой входа и алгоритмом изменения рабочего маршрута, управление состоит в нахождении подходящего сценария саморегуляции, обеспечивающего построение альтернативного маршрута за минимальное время, исходя из механизма атаки.

Это обстоятельство отличает предлагаемый подход от задач обеспечения надежности и отказоустойчивости, поскольку эти процессы предсказуемы, нецеленаправленны и носят накопительный характер.

Предлагаемый в работе подход может распространяться как на сложные системы централизованной архитектуры, так и децентрализованной, в которой единая система управления отсутствует за счет распределения управляющих функций между всеми компонентами.

1.2. Связанные работы. Значительное число исследований посвящено управлению самонастраивающимися динамическими системами сложной архитектуры. Прямое применение уравнения управления системой по состоянию для современных киберфизических систем приводит к трудностям формального описания ввиду изменения гибридной природы объекта и разнообразия режимов работы и каналов взаимодействия с окружающей средой. По этим причинам методология автоматического адаптивного управления такими системами развивалась по пути робастных, стохастических, интерактивных, игровых методов с использованием нечетких логических выводов или эвристик. Подобные методы показывают низкую чувствительность к целенаправленным внешним возмущениям. В направлении развития адаптивных методов по вектору выхода системы можно отметить методы адаптации высшего порядка [9, 10], требующие значительной вычислительной мощности, что приводит к увеличению времени реагирования, что совершенно неприемлемо для случая информационных воздействий. Направления исследований в области адаптивного управления сосредоточены либо в области создания комбинированных методов, либо в декомпозиции задачи адаптации по различным уровням представления, что затруднено для киберфизических систем ввиду разной физической природы протекающих в системе процессов и внешних возмущений.

Проблема построения систем, способных к саморегуляции, также широко освещена в различных научных источниках. В [11] авторами описывается разработанная ими система PROtEUS, основная цель которой – интегрировать данные процессов, реализуемых физическими компонентами (сенсорами, актуаторами, контроллерами) киберфизических систем с данными информационных и человекоориентированных процессов. Авторы демонстрируют применимость предложенного ими подхода и системы PROtEUS на примере системы “Умный дом”, представляющую собой простейшую киберфизическую систему, работа которой связана непосредственно с событиями, инициируемыми человеком – пользователем системы.

Авторы [12] отмечают, что важной проблемой при создании саморегулирующихся киберфизических систем является высокий уровень неопределенности времени выполнения процессов. Для реализации предложенного подхода авторы описывают три механизма саморегуляции: механизм совместной сенсорики, механизм изоляции неисправного компонента системы и механизм усовершенствования режимов переключения между компонентами. Проведенные экспериментальные исследования продемонстрировали высокую способность системы к поддержанию корректного функционирования и к восстановлению системы после ошибок, возникших во время работы.

Существует также ряд исследований, посвященных саморегуляции динамических адаптивных систем (DAS). В частности, авторы [13] утверждают, что DAS-система должна быть способна справиться с непредсказуемыми событиями, возникающими в процессе ее работы. Однако в этом случае существует проблема неопределенности, которая не может быть решена человеком. Синтезируя понятия неопределенности из других дисциплин, авторы [13] пересматривают понятие с точки зрения DAS и предлагает таксономию потенциальных источников неопределенности. Предложен шаблон для описания различных типов неопределенности, содержащий такие поля, как источник неопределенности, ситуации, приводящие к ее возникновению, воздействия на систему и стратегии разрешения неопределенности.

В [14] представлена разработанная авторами эталонная модель под названием FOrmal Reference Model for Self-adaptation (FORMS), ориентированная на создание единого метода описания, сравнения и оценки альтернативных архитектурных решений. Модель включает в себя несколько формально описанных примитивов моделирования, соответствующих ключевым особенностям саморегулирующихся программных систем, а также набор отношений между этими примитивами. Описываются результаты применения FORMS в нескольких программных системах. Благодаря использованию модели, для всех систем удалось с высокой точностью выделить их ключевые особенности функционирования. Также авторы отмечают, что FORMS может быть применена при создании и формализации архитектурных шаблонов саморегуляции. Таким образом, перечисленные выше работы применяют саморегуляцию для поддержания функциональности без учета возможных воздействий.

В [15] рассматривается расширяемая архитектура, обеспечивающая поддержание устойчивого функционирования электронных систем при возникновении аварийных ситуаций или иных, стрессовых для системы, состояний. В основе предложенной авторами архитектуры лежит искусственная иммунная система. Использование биоинспирированных подходов, в частности иммунных систем, является достаточно распространенной практикой при построении саморегулирующихся систем [16, 17]. Теме построения саморегулирующихся электронных систем также посвящено большое количество источников, в частности практическое руководство по проектированию саморегулирующихся систем [18], авторы которого представляют обзор реконфигурируемых устройств, особенное внимание уделяя отказоустойчивым приложениям и концепции системной интеграции.

Существенные результаты содержатся в [19], описывающей алгоритм децентрализованного управления группой мехатронных устройств, однако рассмотренный подход не учитывает кибератаки на систему. Следует отметить, что приведенные выше подходы к созданию саморегулирующихся систем и к управлению сложными системами направлены, прежде всего, на обеспечение отказоустойчивости работы системы и на восстановление ее функциональности в случае сбоя или иной аварийной ситуации. Рассмотренные подходы не учитывают кибератаки, отличительной особенностью которых является их непредсказуемость (в отличие от аварийных ситуаций), неформализуемость и интеллектуальность. Все эти подходы не ориентированы на компенсацию деструктивных воздействий, реализуемых над системой, следовательно, они не будут эффективны для решения задачи управления ССДИ в условиях целенаправленных кибератак.

Важным отличием предлагаемого в данной статье подхода от существующих является его универсальность и ориентированность на любые типы систем, так как он направлен на управление инфраструктурой, а не конкретными процессами, а также способность обеспечить управление динамической инфраструктурой в условиях целенаправленных деструктивных кибератак, реализуемых злоумышленниками, которые обладают знаниями о системе и принципах ее функционирования и управления.

1.3. Модель управления. Рассматриваемые системы имеют сложную структуру и включают в свой состав большое число разнородных компонентов, реализующих различные физические и технологические процессы. Системы могут существовать в многоальтернативном режиме, однако реализуемые ими процессы имеют вход и выход, а их компоненты взаимодействуют, осуществляя саморегуляцию ее инфраструктуры, в которой протекают процессы, в случае внесения изменений в ее работу [20–23].

Рассмотрим абстрактную ССДИ. Для описания такой системы предлагается использовать относительно простую математическую модель на основе теории ориентированных графов, операции над которой позволяют с требуемой полнотой отслеживать последствия кибератак (без идентификации их механизма) и эффективность мер противодействия.

Модель функционирования системы может быть представлена в виде ориентированного графа $G = \langle V, \mathrm{E}, \mathrm{R} \rangle$, где $V = \{v_1, v_2, \ldots, v_n\}$ – множество его вершин (структурные компоненты), $E = \{e_1, e_2, \ldots, e_m\}$ – множество его ребер (межкомпонентные связи), $R = R_{ij}$ – множество маршрутов графа G, элементы которого представляют собой совокупность различных путей из вершины v_i в вершину v_j.

Каждая вершина v_i графа описывается кортежем $\langle \beta, type, \mu, \vartheta \rangle$, где β – идентификатор вершины, *type* – тип компонента, μ – множество параметров вершины, $\vartheta = \{\rho_1^m, \rho_2^m, \ldots\}$ – множество

функций, поддерживаемых узлом, где индекс m обозначает режим выполнения функции (использует ли данный узел функциональность ρ_i в текущем процессе или нет).

Каждая связь характеризуется набором $\langle \delta, \omega, e_i \rangle$, где δ – идентификатор узла, $\omega = \{\upsilon_1, \upsilon_2, \ldots, \upsilon_k\}$ – множество параметров, характеризующих соединение. Любой технологический процесс, протекающий в системе, есть набор рабочих путей, $R = \{R_{ij}\}$, $R_{ij} = r_{ij}^{(1)}, r_{ij}^{(2)}, \ldots, r_{ij}^{(k)} = \langle v_i, \ldots, v_j \rangle$, $k = \overline{1, |R_{ij}|}$. Рабочие пути представлены множеством $R_p \subseteq R$ на множестве маршрутов графа G, что в функциональном смысле есть набор функций $\Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_m\}$, ассоциированных с узлами системы и выполняемых над входными данными и внешними системами. В соответствии с введенными выше обозначениями технологический процесс может быть описан с помощью отображения $F_p : \Phi \to \Omega$, где $\Omega = \{\rho_1^m | \mathrm{m} = 1, v_i \in R_p\}$ – множество функций вершин системы, задействованных в процессе.

Система подвергается кибератакам, направленным на вывод из строя отдельных узлов, на нарушение (возникновения) дуг, количество которых характеризует интенсивность кибератак Z, что может быть выражено оператором в виде $Z \to G(V, E, R) \to G'(V', E', R')$.

Для оценки динамических свойств системы предлагается разбить компоненты системы на динамические звенья с известной передаточной функцией W_i. В соответствии с теорией автоматического управления динамические звенья представляют собой некоторую часть системы управления либо всей системы, описываемую уравнением определенного вида. Для описания целевой функции, которую реализует система, выполняется разбиение графа G, моделирующего систему, на кластеры, которые способны взаимодействовать друг с другом.

Каждое динамическое звено системы имеет передаточную функцию $W_i(s)$, которая по определению есть отношение изображения выходного воздействия $Y(s)$ к изображению входного $X(s)$ при нулевых начальных условиях: $W_i(s) = Y(x)/X(s)$. Вид и способ вычисления передаточной функции для совокупности динамических звеньев и всей системы в целом будет зависеть от типа соединений между динамическими звеньями. Изменение состояния системы может быть оценено на основе вычисления передаточной функции W_i.

Рассмотренная модель обладает полнотой, поскольку графовая структура позволяет отразить все типы кибератак на такие системы, связанных с внесением изменений в число узлов системы, с нарушением связей между узлами, а также с изменениями параметров функционирования узлов и связей между ними. За счет данной модели также возможно учитывать критичность узла, в отношении которого реализуются кибератаки.

1.4. Формализация задачи управления инфраструктурой в условиях кибератак. Введем следующие обозначения: Q – некоторый показатель качества процесса, реализуемого системой, B – текущая структура системы, а вектор $z(t)$ описывает возмущение, связанное с кибератакой.

В таком случае задачу управления ССДИ в условиях кибервоздействий на основе саморегуляции можно формализовать следующим образом: для любого внешнего возмущения Z, изменившего структуру систем от B до B', нужно найти такой вектор управления U (в виде строки), который обеспечит компенсацию последствий кибервоздействия за счет изменения структуры системы до состояния B'', такого, что качество параметров системы изменится несущественно: $|Q^{B''} - Q^B| < \varepsilon$:

$$U = U_0(x) - w^{\mathrm{T}}(x,t)C, \tag{1.1}$$

$$\frac{dC}{dt} = \alpha w(x,t)\frac{\partial L(x)}{\partial x}Bu(x), \tag{1.2}$$

где C – вектор компенсирующего воздействия (в виде строки), возникающего в результате саморегуляции системы, $L(x)$ – выбранная мера оценки корректности функционирования системы, $w(x, t)$ – некоторая известная матричная функция, α – численный параметр, u – матрица управления системой.

Таким образом, общая задача обеспечения саморегуляции сложных систем может быть представлена как поиск сюръективного отображения $\psi : \Gamma \to D$, (где Γ – множество состояний, D – область корректного функционирования), осуществляющего перевод текущего состояния системы $x(t) \in \Gamma$, в котором система находится в момент времени t с учетом деструктивного воздей-

ствия $z(t)$, которое было оказано на систему, в область корректного функционирования D: $\forall t$ найти $\psi(\varphi(t)) : x''(t+1) = x(t) + \psi(\varphi(t))$, $x'' \in D$, при $G \in G_{\text{доп}}$, $Q \in Q_{\text{доп}}$.

Качество реализации технологического процесса формализуется следующим путем:

$$Q = Q(\Phi = \{\varphi_1, \varphi_2, ..., \varphi_m\}, \vartheta = \{\rho_1^m, \rho_2^m, ...\}). \tag{1.3}$$

В случае, если протекающие процессы периодичны, Q может иметь смысл самоподобия. Некоторые ССДИ, в частности – производственные, имеют целевую функцию – набор периодических физических процессов, значения параметров которых инвариантны во времени (принадлежат определенному диапазону значений). Свойство инвариантности во времени в данном случае предлагается рассматривать как свойство самоподобия, и показатель качества Q может представлять собой метрику оценки самоподобия (например, значение коэффициента Херста).

2. Предлагаемый подход к управлению. В общем виде подход к управлению включает представление рабочего процесса ССДИ в виде маршрута на графе, моделирующем систему, локализацию деструктивных информационных воздействий и их атрибуцию в соответствии с изменением графа для поиска сценария саморегуляции системы в соответствии с типом распознанного деструктивного кибервоздействия, и применение его к системе.

2.1. Принципы подхода и модель деструктивных информационных воздействий. Рассмотрим принципы предлагаемого подхода:

Принцип инвариантности по отношению к типу возмущающих воздействий. Компенсаторный механизм включается не для блокирования конкретных типов атак, а для восстановления состояния системы в допустимой зоне. В терминах графовой модели любое возмущающее воздействие представляет собой унарную операцию над графом G (или совокупность унарных операций). Следовательно, если любое воздействие приводит к изменению графа G, оно может быть обнаружено, и для него может быть выработано корректирующее воздействие, также описанное в терминах теории графов.

Принцип функциональной полноты системы саморегуляции, состоящий в существовании хотя бы одного алгоритма решения задачи саморегуляции с помощью установленного набора операций на графе. Возмущающее воздействие в рамках графовой модели однозначно определяется набором унарных операций над графом и соответствующим изменением показателя Q.

Принцип минимизации операций заключается в выборе такой альтернативы, построенной с помощью оператора M, которая будет включать в себя меньшее число операций, чем другие. Минимизация числа действий над графом системы для перехода в новый режим:

$$\forall Z \exists M(G, F, Q) \Rightarrow M(m_1 \dots m_n) : \{x_i m_i x_{i+1}\} \quad \text{для} \quad i = \overline{1,k}, \quad k \to \min. \tag{2.1}$$

Угрозы безопасности, реализуемые путем деструктивных информационных воздействий на систему, могут быть представлены в рамках графовой модели как преобразования графа G, моделирующего систему (табл. 1). Дискретная структура системы позволяет формализовать полный набор деструктивных информационных воздействий (оператор Z) в виде формальных изменений графа G.

2.2. Модель алгоритма управления. Для любого возмущающего воздействия Z (воздействия из данного множества отличаются типом, реализующими их механизмами и воздействиями на граф G) существует алгоритм Λ коррекции текущего состояния графа G, такой, что на этом состоянии может быть построен маршрут, характеризующий выполнение функций системы F_G, моделируемой этим графом, с сохранением значения показателя качества Q в определенной (заданной) зоне:

$$\forall z_i \in Z \; \exists \Lambda, \quad \Lambda : (z_i : G \to G_z) \to G', \tag{2.2}$$

$$G' : F_G = F_{G'}, \quad Q' \in [Q_{\min}; Q_{\max}]. \tag{2.3}$$

Алгоритм должен быть выполнен за конечное число шагов. Введем оператор h, $h : (\Lambda : (z_i : G \to G_z) \to G') \to \{h_1, h_2, ..., h_n\}$. Таким образом, число шагов в алгоритме должно быть ограничено некоторым n, $n \in Z$, где Z – множество целых чисел. Алгоритм должен выполняться за минимальное время, и это время должно быть меньше времени прогнозирования возмущающего воздействия.

$$\xi : (\Lambda : (z_i : G \to G_z) \to G') \to \Delta t_\Lambda, \quad \Delta t_\Lambda < \Delta t_\theta. \tag{2.4}$$

Таблица 1. Пример реализуемых деструктивных информационных воздействий Z, отражаемых унарными графовыми операциями

Унарная операция, выражаемая оператором Z	Воздействие на граф в результате применения операции	Пример реализации киберугрозы, выражающийся как изменение в графе	Интерпретация последствия атак
1. Удаление вершины v_i из графа (орграфа) G, получение нового графа G'	Удаляется вершина и все инцидентные ей ребра (дуги): $G = \langle V, E\rangle$, $G' = \langle V', E'\rangle$, $V' = V\backslash\{v_i\}$, $E' \subseteq E$	Атака отказа в обслуживании, отключение устройства или частичная неработоспособность	Сокращение производительности
2. Удаление ребра (дуги) e_{ij} из графа (орграфа) G, получение нового графа (орграфа) G'	Удаляется только ребро (дуга) e_{ij}, вершины v_i, v_j остаются: $G = \langle V, E\rangle$, $G' = \langle V', E'\rangle$, $V' = V$, $E' = E\backslash\{e_{ij}\}$	Изменение правил работы или настроек системы, запрещающие общение устройства с каким-либо другим	Выход из строя узла или нарушение его режима
3. Сильное удаление ребра e из гиперграфа G	Удаляется ребро и все инцидентные ему вершины: $G = \langle V, E, R\rangle$, $G' = \langle V', E', R'\rangle$, $V' == V\backslash\{v_i \mid R(v_i, e) \equiv 1\}$ $E' = E\backslash\{e\}$	Изменение правил работы или настроек системы, запрещающее общение группы устройств друг с другом с последующим выводом данной группы устройств из строя	Изменение режима работы или последовательности связи узлов
4. Стягивание – удаление ребра (дуги) и отождествление его концевых вершин, получение нового графа (орграфа) G'	Из графа (орграфа) G удаляется ребро e_{ij}, а вершины v_i, v_j заменяются такой новой вершиной v_k, что все дуги в графе (орграфе) G, инцидентные v_i и v_j, становятся инцидентными новой вершине v_k: $G' = \langle V', E'\rangle$, $V' = (V\backslash\{v_i, v_j\}) \cup \{v_k\}$, $E' = E\backslash\{e_{ij}\}$	Сложная атака, сочетающая в себе атаку “воронки” и запрет соединения между узлами	Исключение технологического этапа

Здесь оператор ξ позволяет вычислить время, необходимое для работы алгоритма коррекции, Δt_θ – время, за которое осуществляется прогнозирование значения показателя качества Q.

2.2.1. Алгоритм управления инфраструктурой. Для автоматизации корректирующего воздействия нет необходимости в восстановлении структуры графа G в том виде, в каком она была до начала реализации злоумышленником деструктивного воздействия и до изменения прогнозируемого значения показателя качества Q.

В связи с этим для коррекции необходимо порождать множество маршрутов, эквивалентных маршруту, характеризующему выполнение функции F на графе G. Поскольку текущий маршрут требует внесения изменений, будем считать, что переходим к новому графу G', с другим множеством рабочих маршрутов R', включающем в себя как те маршруты, которых не коснулось деструктивное влияние злоумышленника, так и новые, порожденные маршруты, позволяющие нейтрализовать такое воздействие, предупредив его или устранив.

Таблица 2. Некоторые сценарии саморегуляции в терминах графовой модели

Унарная операция	Методы противодействия	Ответное воздействие на граф
1. Удаление вершины v_i из графа (орграфа) G, получение нового графа G'	1. Активация резервной вершины v_i', обладающей не меньшим функционалом. Если вершина участвовала в маршруте R, новый маршрут R' повторяет старый, но идет через вершину v_i' вместо вершины v_i. 2. Поиск альтернативного маршрута целевой функции (поиск уже работающей вершины с аналогичным функционалом)	$G = \langle V, E\rangle$, $G' = \langle V', E'\rangle$, $G'' = \langle V'', E''\rangle$, $V' = V\setminus\{v_i\}, E' \subseteq E$, 1) $V'' = V'\cup\{v_i'\}$, $E' \subseteq E'' \subseteq E$, $\varphi(v_i') \subseteq \varphi(v_i)$, $\theta(v_i) = \theta(v_i')$; $R = \{v_j,\ldots,v_i,\ldots,v_k\}$, $R' = \{v_j,\ldots,v_i',\ldots,v_k\}$; 2) $R' = \pi(F)$
2. Удаление ребра (дуги) e_{ij} из графа (орграфа) G, получение нового графа (орграфа) G'	1. Создание нового ребра (дуги) e_{ij}' между вершинами v_i, v_j. Если маршрут проходил через удаленное ребро, то маршрут использует новое ребро вместо старого. 2. Поиск альтернативного маршрута с возможностью использовать вершины v_i, v_j. 3. Поиск альтернативного маршрута с запретом на использование вершин v_i, v_j	$G = \langle V, E\rangle$, $G' = \langle V', E'\rangle$, $G'' = \langle V'', E''\rangle$, $V'' = V' = V$, $E' = E\setminus\{e_{ij}\}$; 1) $E'' = E\cup\{e_{ij}'\}$, $R = \{v_l,\ldots,e_{ij},\ldots,v_k\}$, $R' = \{v_l,\ldots,e_{ij}',\ldots,v_k\}$; 2) $R' = \pi(F)$, $v_i, v_j \notin R'$; 3) $R' = \pi(F)$
3. Сильное удаление ребра e из гиперграфа G	1. Восстановление удаленных вершин, создание нового ребра e', которое объединяло бы все вершины, которые объединяло ребро e. 2. Восстановление только тех из удаленных вершин, которые участвовали в реализации целевой функции; создание нового ребра e', которое обеспечило бы те связи между восстановленными вершинами, которые требуются для реализации целевой функции. 3. Поиск альтернативного маршрута	$G = \langle V, E,\rangle$, $G' = \langle V', E', R'\rangle$, $G'' = \langle V'', E'', R''\rangle$, $E' = E\setminus\{e\}$; 1) $V'' = V$, $E'' = E'\cup\{e'\}$, $\omega(e') = \omega(e)$, $R = R'$; 2) $V'' \subseteq V'$, $E'' = E'\cup\{e'\}$, $\omega(e') \subseteq \omega(e)$, $R = R'$; 3) $E'' = E'$, $V'' = V'$, $R' = \pi(F)$
4. Стягивание – удаление ребра (дуги) и отождествление его концевых вершин, получение нового графа (орграфа) G'	1. Удаление ребер, инцидентных новой вершине, восстановление стянутого ребра (дуги), запрет на использование новой вершины для реализации целевой функции. 2. Удаление ребер, инцидентных новой вершине, восстановление ребер, необходимых для реализации целевой функции, запрет на использование новой вершины для реализации целевой функции. 3. Удаление ребер, инцидентных новой вершине, поиск альтернативного маршрута, не включающего новый узел	$G = \langle V, E\rangle$, $G' = \langle V', E'\rangle$, $G'' = \langle V', E''\rangle$, $V' = (V\setminus\{v_i, v_j\})\cup\{v_k\}$, $E' = E\setminus\{e_{ij}\}$; 1) $E'' = (E'\setminus\{e_{ik}, e_{kj}\})\cup\{e_{ij}\}$, $S' = S$; 2) $(E'\setminus\{e_{ik}, e_{kj}\}) \subseteq E'' \subseteq E$, $R' = R$; 3) $E'' = E'\setminus\{e_{ik}, e_{kj}\}$, $R' = \pi(F)$, $v_k \notin R'$

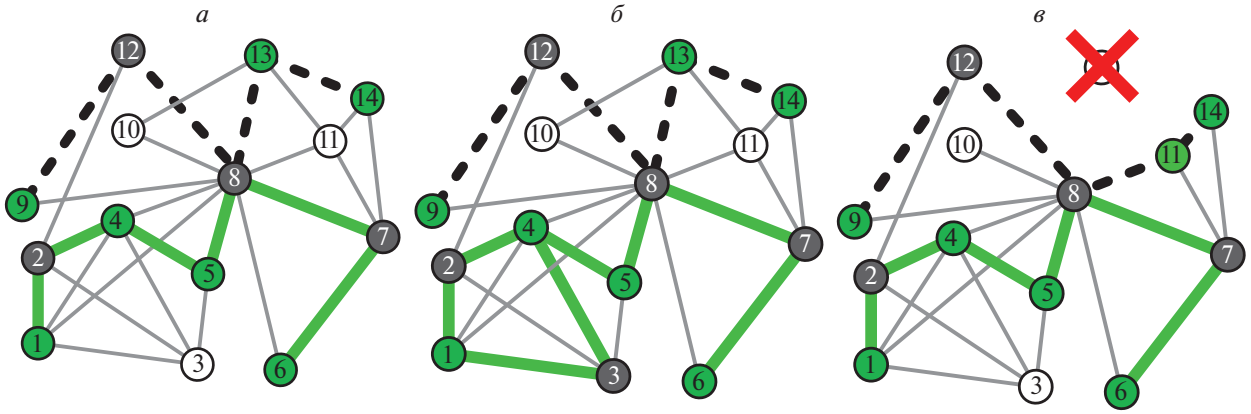

Рис. 2. Примеры сценариев саморегуляции

Введем понятие эквивалентности маршрутов, считая, что деструктивным воздействиям подвергся подграф $g \in G$, в частности узлы k и l, соединенные одной или несколькими дугами. Эквивалентной считается такая пара маршрутов R_{ij} и R'_{ij}, что:

1) вершины k и l, принадлежащие R_{ij}, не входят в маршрут R'_{ij}: $k,l \notin V'$;

2) существует непустое множество вершин v', которым инцидентны ребра, ранее инцидентные вершинам k и l;

3) на обоих маршрутах R_{ij} и R'_{ij} реализуется функция F, $G \to R = \{R_{ij}\}$, $F_{G'} = F_G$, $F_{G'} : G' \to R' = \{R'_{ij}\}$;

4) оба маршрута R_{ij} и R'_{ij} сохраняют значение Q в заданных пределах, $Q' \in \varepsilon = [Q_{\min}, Q_{\max}]$.

Для автоматической коррекции графа G необходимо получить множество эквивалентных маршрутов и затем, используя критерий оптимальности маршрутов, выбрать один из них для применения к текущему состоянию системы.

При генерации множества эквивалентных маршрутов осуществляются:

1) структурные изменения: замена всего маршрута или его части. Такие изменения реализуются с помощью описанных ранее унарных операций над графом;

2) функциональные изменения:

изменение функций отдельных узлов – пусть узел выполняет другую функцию, а не ту, которую выполнял раньше;

спецификация функций, выполняемых узлами (корректировка параметров выполнения этих функций).

2.2.2. Сценарии саморегуляции. В общем виде для ССДИ любого типа общая схема алгоритма саморегуляции реализует поиск сценария саморегуляции в зависимости от типа деструктивного воздействия, а также ограничений, накладываемых на время выполнения сценария, и на возможность коммуникации узлов друг с другом.

Для систем различного типа могут быть использованы различные сценарии саморегуляции. В общем виде все сценарии реализуют противодействия деструктивным информационным воздействиям в рамках графовой модели. Описание некоторых сценариев саморегуляции представлено в табл. 2, примеры сценариев представлены на рис. 2.

Часть рис. 2, *а* отражает начальное состояние системы: в ней протекают два процесса: № 1 (вершины 1-2-4-5-8-7-6) и № 2 (вершины 9-12-8-13-14). Узлы 3, 10 и 11 неактивны и зарезервированы. Часть рис. 2, *б* иллюстрирует активацию узла 3 и использование его для выполнения процесса с целью снижения нагрузки на узел 2. Часть рис. 2, *в* отражает исключение скомпрометированного узла 13, активацию узла 11 и построение через него маршрута от узла 8 к узлу 14.

2.3. Оценка устойчивости ССДИ к кибератакам. Характеристикой процесса саморегуляции может являться устойчивость системы – способность системы к корректному функционированию в условиях деструктивных информационных воздействий. Необходимо отметить, что для ССДИ построение функции Ляпунова (или аналогичного показателя устойчиво-

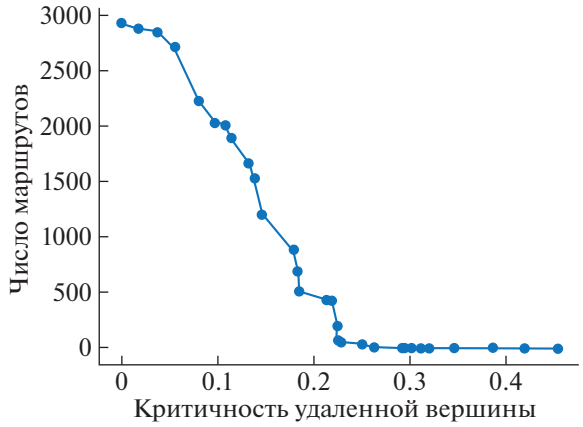

Рис. 3. Зависимость устойчивости от критичности удаленных вершин по экспериментальным данным

сти) затруднено сложностью вычисления показателей чувствительности (вычисление производных от влияющих информационных воздействий) с учетом изменения структурного состава системы в процессе реструктуризации. Поэтому предлагается использовать определение устойчивости, аналогичное формулировке, предложенной Дж. Николисом [24].

Пусть ϖ – граница состояния полной функциональности системы, ε – заданное положительное число, тогда система устойчива, если траектория изменения ее состояния никогда не приближается к ϖ ближе, чем на ε.

2.3.1. ССДИ с централизованным управлением. Для систем различного типа оценка устойчивости может быть получена различными способами. Например, для систем с централизованным управлением устойчивость может быть представлена как избыточность системы – число допустимых маршрутов реализации целевой функции, обеспечивающих саморегуляцию системы. Математически задача поиска маршрутов, обеспечивающих саморегуляцию системы и восстановление целевой функции, сводится к задаче поиска путей фиксированной длины на ориентированном графе G', являющемся подграфом G, который моделирует всю систему. Использование подграфа базируется на том факте, что нет необходимости в перестроении всей целевой функции, а только некоторой ее части. Введем матрицу смежности графа G', обозначим ее $A_{[V\times V]} = \alpha_{i,i}$, где i пробегает значения от 1 до k: $i = \overline{1,k}$. Поскольку нам требуется найти все пути длины $k-1$ (между k вершинами), возведем матрицу смежности $A_{[V\times V]}$ в степень k, тогда значение $\gamma_{1,k}$ новой матрицы, находящееся на пересечении строки под номером 1 и столбца под номером k, и будет отражать искомое число маршрутов между кластерами, обеспечивающих саморегуляцию. Назовем это мощностью пространства саморегуляции: $\gamma_{1,k} \in A^k_{[V\times V]}$.

Для демонстрации возможности применяемого подхода была смоделирована атака на ССДИ с централизованным управлением, представляющую собой систему управления процессом многоступенчатой очистки воды. Моделировалось удаление вершины, обладающей определенной степенью критичности. В качестве критичности вершины предлагается использовать отношение числа рабочих маршрутов, проходящих через вершину, к общему числу рабочих маршрутов. Зависимость устойчивости (числа возможных рабочих маршрутов) от критичности удаленной вершины представлена на рис. 3.

2.3.2. ССДИ с децентрализованным управлением. Для таких систем, имеющих децентрализованную инфраструктуру, предлагается оценивать динамические свойства системы на основе вычисления времени регулирования τ с использованием передаточной функции $W_i(s)$ для динамических звеньев системы. Время регулирования представляет собой минимальное время, по истечении которого, начиная с момента начала воздействия на систему, выходное значение отклоняется от установившегося на величину Δ, не превышающую некоторую постоянную (обычно принимают за 5%). В предположении о линейности системы приблизительно вре-

мя регулирования может быть найдено по переходной характеристике $h(t)$, которая вычисляется по формуле Хевисайда (если уравнение $X(s) = 0$ не имеет кратных корней):

$$h(t) = \frac{Y(0)}{X(0)} + \sum_{i=1}^{n} \frac{Y(s_i)}{s_i X'(s_i)} e^{s_i t}, \tag{2.5}$$

где $X'(s_i)$ – первая производная от $X(s)$ при $s = s_i$, s_i – корни характеристического уравнения $X(S) = 0$.

Тогда время регулирования системы τ может быть найдено из неравенства

$$\frac{Y(0)}{X(0)} + \sum_{i=1}^{n} \frac{Y(s_i)}{s_i X'(s_i)} e^{s_i \tau} - \psi \le 0.005, \tag{2.6}$$

где ψ – установившееся значение выходной величины. Преобразуем и получим оценку времени регулирования:

$$\tau \le \left[\ln\left(0.005 + \psi - \frac{Y(0)}{X(0)} \right) - \ln \sum_{i=1}^{n} \frac{Y(s_i)}{s_i X'(s_i)} e^{s_i \tau} \right] \Bigg/ \sum_{i=1}^{n} s_i . \tag{2.7}$$

Очевидно, что интересующее нас максимально допустимое время регулирования – верхняя временная граница – будет получено в случае равенства (2.7). Таким образом, получена верхняя оценка времени саморегуляции.

2.3.3. Условия выполнения саморегуляции за заданное время. Приведем условия выполнения саморегуляции за заданное время. Для оценки времени саморегуляции введем: Γ – множество возможных состояний системы, D – множество корректных состояний системы, $x(t)$ – состояние системы в момент времени t, $\varphi(t)$ – воздействия от управляющих узлов в момент времени t, $z(t)$ – деструктивное воздействие на систему в момент времени t, $f(x(t))$ – нейтрализующее воздействие на деструктивное воздействие. Рассматривается $z(t) : x(t + 1) = x(t) + \varphi(t) + z(t) \in \Gamma \backslash D$, так как другие случаи не имеют смысла.

Пусть $\nexists f : \Gamma \to D$, тогда $z(t) : x(t+1) = x(t) + \varphi(t) + z(t) \in D$, $x(t+1) = x(t) + \varphi(t) + z(t) \in D$ – противоречие.

Три случая, описывающие все возможные отношения между t_f и t_z:

1) $t_f > t_z$, $t_f = t_z + \Delta t : x(t+1) - f(x(t) + \varphi(t)) + z(t) = x(t) + f(\varphi(t_z)) + z(t_z) = z(\Delta t) = x(t) + z(\Delta t)) \in \Gamma \backslash D$;

2) $t_f = t_z : x(t+1) = f(x(t) + \varphi(t)) + z(t) = x(t) + f(\varphi(t_f)) + z(t_z) + z(\Delta t) = x(t) \in \Gamma$;

3) $t_f < t_z$, $t_z = t_f + \Delta t : x(t+1) = f(x(t) + \varphi(t)) + z(t) = x(t) + f(\varphi(t_z)) + z(t_z) + f(\varphi(\Delta t)) = x(t) + f(\varphi(\Delta t) \in D$.

Таким образом, $t_f < t_z$.

Время саморегуляции $T_c = T_D + T_V + T_E$, T_D – время поиска маршрута на графе, T_E – время передачи данных до узлов, которые необходимо переконфигурировать, T_V – время применения изменений на узлах.

3. Применение предлагаемого подхода и оценка регулируемости. Для проведения экспериментальных исследований по оценке устойчивости ССДИ разработанная система моделирования использовала данные об архитектуре, компонентах и принципах работы автоматической системы распределения водоснабжения в “умном городе”, а также данные о смоделированных кибератаках на систему [25]. Для оценки числа маршрутов в зависимости от времени их выполнения была построена накопительная функция, значение которой является числом маршрутов, время выполнения которых меньше значения χ (рис. 4).

Число альтернативных рабочих маршрутов может быть интерпретировано как мера приспособляемости при управлении [26, 27].

При моделировании атаки, заключающейся в последовательном удалении вершин графа, была определена траектория движения системы при моделировании атак (рис. 5).

Линии A и A' ограничивают минимальную производительность (A) и максимально допустимое время реконфигурации (A'). Минимально допустимое число маршрутов задано линией B.

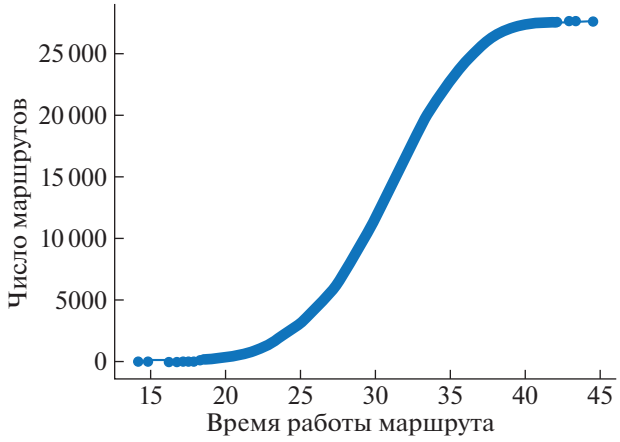

Рис. 4. Накопительная функция для времени выполнения маршрутов

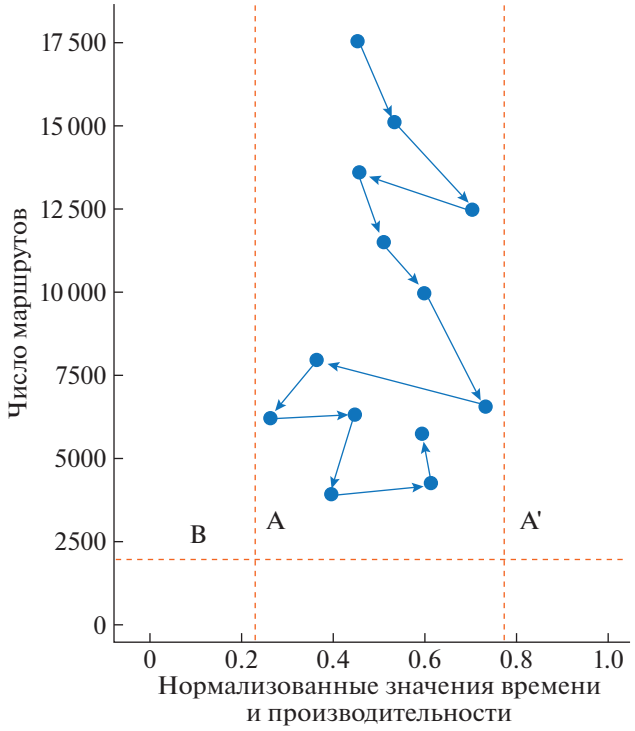

Рис. 5. Траектория движения системы при моделировании атак

Заключение. Предложен подход к управлению ССДИ, к которым относятся, в том числе, современные технические системы критического назначения, такие, как умный город, системы автоматизации распределения ресурсов или планирование передачи энергии. Предложена концепция управления такими системами в условиях кибератак на основе саморегуляции инфраструктуры, что позволяет обеспечить устойчивость функционирования системы.

Ключевой особенностью предложенного подхода является его универсальность и применимость к сложным системам любого типа и архитектуры, в частности, были рассмотрены примеры для ССДИ с централизованным и децентрализованным управлением. Управление производится над компонентами динамической инфраструктуры сложных систем, обеспечивая их саморегуляцию для поддержания необходимых условий корректного протекания физических и технологических процессов, реализуемых системой.

В основе подхода лежит графовая модель системы, позволяющая описать ее динамическую инфраструктуру, а также учитывать критичность узлов инфраструктуры, на которые оказывается деструктивное воздействие. Разработанная модель обладает полнотой с точки зрения типов кибератак, которые могут быть реализованы на инфраструктуру системы.

Отличием предложенного подхода от рассмотренных подходов к построению саморегулирующихся систем является его способность учитывать не только ситуации, связанные с нарушением отказоустойчивости системы, но и целенаправленные информационные воздействия, реализуемые злоумышленником, который обладает знаниями о реализуемых в системе процессах управления.

СПИСОК ЛИТЕРАТУРЫ

1. *Positive Research 2018*. Сборник исследований по практической безопасности. https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/Positive-Research-2018-rus.pdf (дата обращения 10.04.2019).
2. *Зегжда Д.П., Васильев Ю.С., Полтавцева М.А., Кефели И.Ф., Боровков А.И.* Кибербезопасность прогрессивных производственных технологий в эпоху цифровой трансформации // Вопросы кибербезопасности. 2018. № 2. С. 2–15.
3. *Зегжда П.Д., Васильев Ю.С., Зегжда Д.П.* Обеспечение безопасности автоматизированных систем управления технологическими процессами на объектах гидроэнергетики // Изв. РАН. Энергетика. 2016. № 3. 22 с.
4. *Bakhshi Z., Balador A., Mustafa J.* Industrial IoT Security Threats and Concerns by Considering Cisco and Microsoft IoT Reference Models // Wireless Communications and Networking Conf. Workshops (WCNCW). Barcelona. 2018. С. 173–178.
5. Ландшафт угроз для систем промышленной автоматизации. Первое полугодие 2018 [Электронный ресурс]. URL: https://ics-cert.kaspersky.ru/media/H1_2018_ICS_REPORT_RUS.pdf (дата обращения 28.05.2019).
6. *Петренко С.А.* Концепция поддержания работоспособности киберсистем в условиях информационно-технических воздействий // Тр. ИСА РАН. 2009. Т. 41. С. 175–193.
7. *Зегжда Д.П., Павленко Е.Ю.* Гомеостатическая стратегия безопасности киберфизических систем // Проблемы информационной безопасности. Компьютерные системы. 2017. № 3. С. 9–22.
8. *Zegzhda D.P., Poltavtseva M.A., Lavrova D.S.* Systematization and Security Assessment of Cyber-physical Systems // Automatic Control and Computer Sciences. 2017. Т. 51. № . 8. С. 835–843.
9. *Фрадков А.Л.* Адаптивное управление в сложных системах. М.: Наука, 1990.
10. *Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е.* Интеллектное управление динамическими системами. М.: Физматлит, 2000. 352 с.
11. *Seiger R., Huber S., Schlegel T.* Proteus: An Integrated System for Process Execution in Cyber-physical Systems // Business-Process and Information Systems Modeling. Stockholm: Springer, 2015. P. 265–280.
12. *Gerostathopoulos I., Skoda D., Plasil F., Bures T., Knauss A.* Architectural Homeostasis in Self-adaptive Software-intensive Cyber-physical Systems // European Conf. on Software Architecture. Copenhagen: Springer, 2016. P. 113–128.
13. *Ramirez A.J., Jensen A.C., Cheng B.H.* A Taxonomy of Uncertainty for Dynamically Adaptive Systems // Proceedings of the 7th Intern. Sympos. on Software Engineering for Adaptive and Self-Managing Systems. Zurich: IEEE Press, 2012. P. 99–108.
14. *Weyns D., Malek S., Andersson J.* Forms: a Formal Reference Model for Self-adaptation // Proceedings of the 7th International Conference on Autonomic Computing. Washington: ACM, 2010. P. 205–214.
15. *Tyrrell A., Timmis J., Greensted A., Owens N.* Evolvable Hardware, a Fundamental Technology for Homeostasis // Proceedings of the IEEE Workshop on Evolvable and Adaptive Hardware. Honolulu. WEAH, 2007. С. 40–45.
16. *Cohen I.* Tending Adam's Garden: Evolving the Cognitive Immune Self. N.Y.: Elsevier Acad. Press, 2000.
17. *Andrews P., Timmis J.* Inspiration for the Next Generation of Artificial Immune Systems. // Proceedings of the 4th Intern. Conf. on Artificial Immune Systems (ICARIS 2005). Alberta: Springer-Verlag, 2005. P. 126–138.
18. *Greenwood G., Tyrrell A.* Introduction to Evolvable Hardware: A Practical Guide for Designing Self-Adaptive Systems. N.Y.: Wiley-IEEE Press, 2006.
19. *Каляев И.А., Мельник Э.В.* Децентрализованные системы компьютерного управления. Ростов н/Д.: ЮНЦ РАН, 2011.
20. *Филимонов А.Б., Филимонов Н.Б., Тихонов В.Ю.* Задача прохождения лабиринта интеллектуальными агентами // Мехатроника, автоматизация, управление. 2016. № 11. Т. 17. С. 750–761.
21. *Филимонов А.Б., Филимонов Н.Б.* Автомат ограничений управляемых динамических процессов // Изв. ЮФУ. Техн. науки. 2013. № 3 (140). С. 155–161.

22. *Васильев С.Н.* К управляемости нелинейных систем при фазовых ограничениях и постоянно действующих возмущениях // Изв. АН СССР. Техн. кибернетика. 1993. № 1. С. 77–82.
23. *Федосов Е.А.*, *Инсаров В.В.*, *Селивохин О.С.* Системы управления конечным положением в условиях противодействия среды. М.: Наука, 1989.
24. *Николис Дж.* Динамика иерархических систем: Эволюционное представление // М.: Мир, 1989. 488 с.
25. *Goh J., Adepu S., Junejo K. N., Mathur A.* A Dataset to Support Research in the Design of Secure Water Treatment Systems // Intern. Conf. on Critical Information Infrastructures Security. Paris: Springer, 2016. С. 88–99.
26. *Tsypkin Ya.Z.* Learning in Robust Control Systems // Intelligent Control Systems: Theory and Applications. N.Y.: IEEE Press, 1996.
27. *Страшнов Е.В.*, *Торгашев М.А.* Моделирование динамики электроприводов виртуальных роботов в имитационно-тренажерных комплексах // Мехатроника, автоматизация, управление. 2016. № 11. Т. 17. С. 762–768.