

Данный текст является русскоязычной версией опубликованной на английском языке статьи и представлен в авторской редакции только на данном сайте!

UDC 519.87

Doi: 10.31772/2587-6066-2020-21-1-21-27

For citation: Raskina A. V., Videnin S. A., Chzhan E. A., Yusupova R. R. Peculiarities of design software architecture of adaptive information processing, modeling and control systems. *Siberian Journal of Science and Technology*. 2020, Vol. 21, No. 1, P. 21–27. Doi: 10.31772/2587-6066-2020-21-1-21-27

Для цитирования: Раскина А. В., Виденин С. А., Чжан Е. А., Юсупова Р. Р. Особенности организации программной архитектуры адаптивных систем обработки информации, моделирования и управления // Сибирский журнал науки и технологий. 2020. Т. 21, № 1. С. 21–27. Doi: 10.31772/2587-6066-2020-21-1-21-27

PECULIARITIES OF DESIGN SOFTWARE ARCHITECTURE OF ADAPTIVE INFORMATION PROCESSING, MODELING AND CONTROL SYSTEMS

A. V. Raskina, S. A. Videnin, E. A. Chzhan, R. R. Yusupova

Siberian Federal University
79, Svobodnyy Av., Krasnoyarsk, 660041, Russian Federation
E-mail: raskina.1012@gmail.com

The article proposes an approach to developing the architecture of a service-oriented information processing system, modeling and process control. The system, which is being developed, is a tool for identifying, predicting and controlling discrete-continuous processes. Its mathematical apparatus is based on nonparametric algorithms of identification and control. The software architecture includes the following main modules: the module for processing data, modeling and forecasting output process variables and the process control module. The first module includes data preprocessing algorithms: normalization, centering and analysis of outliers and omissions. The modeling module is an algorithm for research and recovery dependencies between process variables, process identification using nonparametric estimation of the regression function from observations. The last module is an implementation of nonparametric dual control algorithms. Control devices built on the basis of these algorithms perform functions of both object control and its study.

The article discusses the application of architectural solutions based on two proven approaches in the field of software development: the composite approach and the service-oriented approach.. The main principles of composite architecture as a set of software systems with many characteristics that perform a specific task and service-oriented architecture as a modular approach to software development are described. The advantages of the applied composite service-oriented architecture over other variants of software architecture for control systems are shown, in particular, monolithic software architecture is compared with composite service-oriented architecture. This means that a researcher can use a single operation, which is a logically isolated, repeated task related to the production process of the enterprise. At the same time, it is necessary to ensure positive results when integrating with existing software products of enterprises which greatly complicates and requires the development of new components, as well as support for the "inherited" parts of the system.

Keywords: service-oriented architecture, software development, design of process control systems.

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ПРОГРАММНОЙ АРХИТЕКТУРЫ АДАПТИВНЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ, МОДЕЛИРОВАНИЯ И УПРАВЛЕНИЯ

А. В. Раскина, С. А. Виденин, Е. А. Чжан, Р. Р. Юсупова

Сибирский федеральный университет
Российская Федерация, 660041, г. Красноярск, просп. Свободный, 79
E-mail: raskina.1012@gmail.com

В статье предложен подход к разработке архитектуры сервис-ориентированной системы обработки информации, моделирования и управления технологическими процессами. Разрабатываемая система представляет собой инструментарий для идентификации, прогнозирования и управления дискретно-непрерывными процессами, математический аппарат которой основан на непараметрических алгоритмах идентификации и управления. Программная архитектура состоит из нескольких основных модулей: модуль обработки данных, моделирования и прогноза выходных переменных процесса, модуль управления технологическим процессом. Первый модуль включает в себя алгоритмы предобработки данных: нормализация, центрирование и анализ выбросов и пропусков. Модуль моделирования представлен алгоритмическим функционалом для исследования и восстановления зависимостей между переменными процесса, идентификации процесса с использованием непараметрической оценки функции регрессии по наблюдениям. Последний модуль – это реализация непараметрических алгоритмов дуального управления. Управляющие устройства, построенные на основании данных алгоритмов, выполняют функции как управления объектом, так и его изучения.

В статье обсуждаются вопросы применения архитектурных решений, основанных на двух зарекомендовавших себя в области разработки программного обеспечения подходах – композитном и сервис-ориентированном. Описываются основные принципы композитной архитектуры как набора программных систем с множеством характеристик, которые выполняют определенную задачу, и сервис-ориентированной архитектуры как модульного подхода к разработке программного обеспечения. Показаны преимущества примененной композитной сервис-ориентированной архитектуры перед другими вариантами архитектур программного обеспечения для систем управления, в частности, в работе сравнивается монолитная программная архитектура с композитной сервис-ориентированной архитектурой. Выбранное архитектурное решение предоставляет возможность выстроить систему из набора независимых модулей, каждый из которых реализует отдельную операцию, которая является логически обособленной, повторяющейся задачей, являющейся составной частью производственного процесса предприятия. Использование описанного в работе подхода позволило достичь положительных результатов при интеграции с существующими программными продуктами предприятий, значительно сократить сложность и стоимость разработки новых компонентов, а также поддержки «унаследованных» частей системы.

Ключевые слова: сервис-ориентированная архитектура, разработка программного обеспечения, проектирование систем управления технологическими процессами.

Введение. На сегодняшний день для решения комплекса задач, связанных с управлением и моделированием сложных технологических процессов в различных отраслях промышленности, в частности аэрокосмической отрасли, используются программные и алгоритмические средства, позволяющие обеспечить функции идентификации и управления процессом. Одним из примеров подобного рода средств

являются адаптивные системы управления технологическими процессами, функционирование которых осуществляется с использованием непараметрической теории [1–5]. Использование непараметрических алгоритмов идентификации и управления позволяет осуществлять моделирование, прогноз и ведение процесса в рамках технологического регламента в условиях недостатка априорной информации, когда параметрическая структура модели исследуемого процесса остается неизвестной из априорной информации [6–9]. Эта ситуация довольно часто встречается при разработке компьютерных систем управления, так как в большинстве случаев исследователю приходится работать с малоизученными процессами, для которых не представляется возможным обоснованно подобрать параметрическую структуру модели. При этом к разрабатываемой системе управления предъявляется ряд требований, к которым относятся: синхронизации данных с различных точек ведения контроля; возможность работы с разнотипными данными (система самостоятельно производит преобразование данных в единый формат); возможность как автономной работы системы управления, так и коррекции управляющих действий системы экспертом-технологом; обеспечение безопасности путем осуществления шифрования и резервного копирования данных. Таким образом, основной задачей данного исследования является разработка и проектирование архитектуры программно-алгоритмического обеспечения, позволяющей оптимальным образом обеспечить функционирование системы.

Функциональная схема работы системы. Система обработки информации, моделирования и управления представляет собой набор подсистем, включающий в себя модули, представленные на рис. 1.

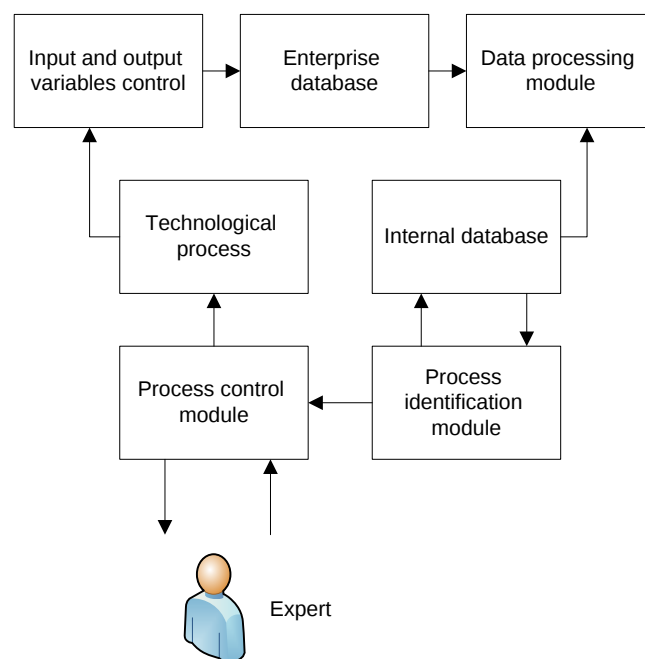


Рис. 1. Функциональная схема работы разрабатываемой адаптивной системы обработки информации, моделирования и управления

Fig. 1. Functional scheme of the developed adaptive information system for data processing, modeling and control

Контроль переменных технологического процесса осуществляется через интервал времени Δt . Вход и выход процесса представлен измерениями, формирующими выборку вида $\{u_i, x_i\}, i = \overline{1, s}$, где s – объем выборки; u_i, x_i – измерения входа и выхода объекта в момент времени t_i . Эта выборка содержится в блоке «База данных предприятия». Модуль

обработки данных осуществляет импорт данных различного формата, производит преобразование данных в единый формат, с которым работает система. Также в этом модуле реализованы алгоритмы предобработки данных: нормализация, центрирование, анализ выбросов и пропусков в выборке наблюдений. Результатом работы модуля является внутренняя база данных системы, которая содержит конвертированные данные в едином формате.

Модуль идентификации технологического процесса реализует непараметрические алгоритмы идентификации [10].

В работе рассматриваются классы технологических процессов управления, допускающие описание уравнениями вида:

$$x_t = F(x_{t-1}, \dots, x_{t-k}, u_t, \xi_t). \quad (1)$$

Здесь F – неизвестный функционал; k – порядок разностного уравнения, который ограничен $k \leq k_{\max}$; u_t – входная переменная объекта; x_t – выходная переменная объекта; индекс t – дискретное время; $\xi(t)$ – векторная случайная помеха.

Введем следующие обозначения:

$$z_t = (z_1, \dots, z_{k+1}) = (x_{t-1}, \dots, x_{t-k}, u_t), \quad (2)$$

тогда

$$x_t = F(z_t). \quad (3)$$

С учетом переобозначений (2) модель рассматриваемого технологического процесса может быть сведена к модели динамической системы в дискретном времени, когда на вход последней поступают не только переменные u_t , но и $x_{t-1}, x_{t-2}, \dots, x_{t-k}$ и т. д.

В рассматриваемых условиях в качестве непараметрической модели объекта может быть принята следующая непараметрическая оценка функции регрессии по данным наблюдений $\{x_i, u_i, i = \overline{1, s}\}$ [11]:

$$x_s^t = \frac{\sum_{i=1}^s x_i \cdot \Phi\left(\frac{u_s - u_i}{c_s^u}\right) \prod_{j=1}^k \Phi\left(\frac{x_{s-j} - x_{i-j}}{c_s^{x[j]}}\right)}{\sum_{i=1}^s \Phi\left(\frac{u_s - u_i}{c_s^u}\right) \prod_{j=1}^k \Phi\left(\frac{x_{s-j} - x_{i-j}}{c_s^{x[j]}}\right)}, \quad (4)$$

где $\Phi(\cdot)$ – колоколообразная функция; $c_s^u, c_s^{x[j]}$ – коэффициенты размытости ядра, которые при наличии обучающей выборки находятся из задачи минимизации показателя соответствия выхода объекта и выхода модели, основанного на методе скользящего экзамена, когда в непараметрической модели (4) по индексу i исключается q -е наблюдение переменной, предъявляемой для экзамена:

$$R(c_s^u, c_s^{x[1]}, \dots, c_s^{x[k]}) = \sum_{q=1}^s \left(x_s(u_q, x_{q-1}, \dots, x_{q-k}) - x_q \right)^2 = \min_{c_s^u, c_s^{x[1]}, \dots, c_s^{x[k]}} \sum_{q \neq i}^s \left(x_s(u_q, x_{q-1}, \dots, x_{q-k}) - x_q \right)^2, \quad (5)$$

где индекс i фигурирует в непараметрической модели (4). В качестве алгоритма оптимизации реализован неградиентный метод многомерной оптимизации – метод Нелдера – Мида, так как данный метод является эффективным при низкой скорости вычисления минимизируемой функции. Для выбора начальных вершин деформируемого многогранника была задана область возможных значений коэффициентов размытости ядерной функции $c_s \in [0.01, 4]$, из которой произвольным образом выбрана $n+k+1$ точка, где n – количество входных переменных; k – порядок разностного уравнения, которые образуют симплекс $n+k$ -мерного пространства.

Модуль управления технологическим процессом реализует схему управления, представленную на рис. 2.

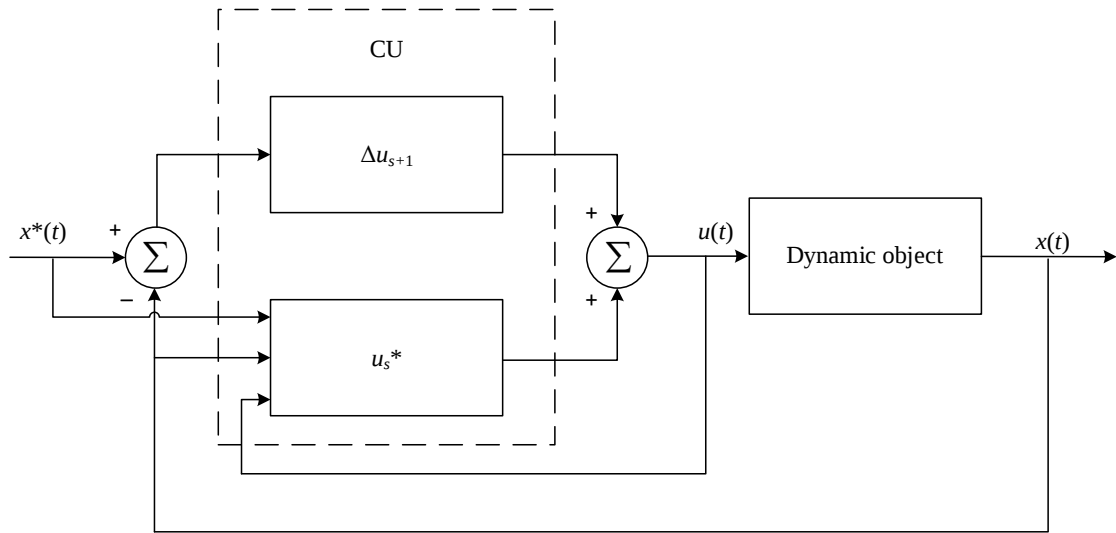


Рис. 2. Схема дуального управления динамическим объектом

Fig. 2. Scheme of dual control of a dynamic object

Непараметрический алгоритм дуального управления имеет вид [12]:

$$u_{s+1} = u_s^* + \Delta u_{s+1}, \quad (6)$$

где u_s^* – составляющая, накапливающая информацию об объекте исследования, а $\Delta u_{s+1} = \varepsilon(x_{s+1}^* - x_s)$ – «изучающие» поисковые шаги. Дуализм алгоритма (6) заключается в следующем. На первых тактах управления основную роль при формировании управляющих воздействий играет слагаемое Δu_{s+1} из формулы (6). Но уже по мере накопления информации об объекте возрастает роль слагаемого u_s^* .

В данном случае в качестве слагаемого u_s^* из уравнения (6) можно принять выражение:

$$u_s^* = \frac{\sum_{i=1}^s u_i \cdot \Phi\left(\frac{x_{s+1}^* - x_i}{c_s}\right) \prod_{j=1}^k \Phi\left(\frac{x_{s-j} - x_{i-j}}{c_s}\right)}{\sum_{i=1}^s \Phi\left(\frac{x_{s+1}^* - x_i}{c_s}\right) \prod_{j=1}^k \Phi\left(\frac{x_{s-j} - x_{i-j}}{c_s}\right)}. \quad (7)$$

Алгоритм управления технологическим процессом строится следующим образом. На основании правила выделения существенных переменных определяется порядок разностного уравнения модели динамического процесса k , который в дальнейшем используется при вычислении управляющих воздействий в (7), где присутствуют только те переменные, которые были отобраны алгоритмом. Качество управления оценивается по двум характеристикам:

1. Время регулирования t_p – время с начала управления до момента, когда выходная величина отличается от задания не более некоторой заданной величины α . Обычно принимают: $\alpha = 0,05_{уст}$.

2. Относительная ошибка управления W_p , равная суммарному отклонению фактического выхода процесса от задающего воздействия в течение всего времени регулирования по отношению к задающему воздействию, выраженная в относительных величинах, в %.

$$W_p = \frac{\frac{1}{s} \sum_{i=1}^s |x_i - x_i^*|}{x^*}.$$

Вычисленные значения управляющих воздействий u_{s+1} доступны к корректировке экспертом-технологом. Таким образом, разрабатываемая система представляет собой инструментарий, включающий в себя следующие основные модули: модуль обработки данных, моделирования и прогноза выходных переменных процесса, модуль управления технологическим процессом.

Программная архитектура системы. При проектировании программной архитектуры нами учитывался факт того, что каждый из модулей может быть легко заменен на новую его реализацию в случае изменения технологического процесса, а также дополнен новыми при расширении функционала системы.

На рис. 3 представлен пример одной и той же системы, построенной на двух различных архитектурах. Классический вариант системы представляет собой единое монолитное приложение, внутри которого можно выделить ряд тесно связанных подсистем, выстроенных вокруг конкретных больших бизнес-процессов. Хотя архитектура такой системы и представляет собой трехслойную архитектуру, ее высокая связность значительно осложняет сопровождение и внедрение нового функционала в такую систему. Кроме того, реализация нового бизнес-процесса приводит к разработке новой подсистемы,



большая часть исходного кода дублирует код уже имеющихся подсистем.

Рис. 3. Сравнение архитектур

Fig. 3. Architecture Comparison

Выделение базовых элементов из системы, а также разбиение бизнес-процессов на атомарные операции и их перенос в набор сервисов, не зависящих от реализации остальных компонентов системы, позволяет построить более гибкую систему на основе композитной архитектуры. При этом доступ к данным осуществляется через единый интерфейс, а доступ к сервисам может легко получить любая подсистема. К тому же такое разделение системы позволяет с меньшими трудозатратами выполнить разделение системы на несколько звеньев: сервер данных, сервер сервисов, клиентские приложения.

Проектируемое нами архитектурное решение основано на двух зарекомендовавших себя в области разработки программного обеспечения архитектурных подходах: композитный и сервис-ориентированный. Композитная архитектура – набор программных

систем с множеством характеристик, которые выполняют определенную задачу, а также разрабатываются в установленном порядке и на основе общего набора базовых средств. При этом операции интеграции и тестирования вытесняют операции проектирования и кодирования. Сервис-ориентированная архитектура – модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных взаимозаменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.

Главное отличие и преимущество композитной архитектуры перед SOA в том, что композитная архитектура обеспечивает гибкость на всех слоях (уровнях) приложения в то время, как SOA предоставляет гибкость только на одном слое – слое приложения (бизнес-логики) [13], так как композитная архитектура по своей сути является дальнейшим развитием сервис-ориентированной архитектуры и, соответственно, обладает всеми преимуществами сервис-ориентированного подхода в разработке программного обеспечения. Композитные приложения, как правило, являются дальнейшим развитием уже разработанной группы отдельных приложений, из которых выделяются базовые наборы компонентов и контейнеров для их дальнейшей интеграции друг с другом. В настоящее время идея построения сервис-ориентированных приложений на основе композитной архитектуры находит широкое распространение в различных областях разработки программного обеспечения и является предметом изучения во многих научно-исследовательских центрах. Одной из самых крупных подобных организаций является Институт инженерии программного обеспечения (англ., Software Engineering Institute, SEI) [14]. На данный момент институт является признанным международным лидером в изучении вопроса построения композитных сервис-ориентированных приложений. Композитная сервис-ориентированная архитектура – это подход к проектированию программных систем для организации уже существующих программных продуктов таким образом, чтобы разрозненные наборы сложных распределённых систем и приложений можно было превратить в сеть интегрированных простых и гибких ресурсов.

Основываясь на методологических рекомендациях SEI, мы спроектировали и применили гибридный архитектурный подход и таким образом получили композитную сервис-ориентированную программную систему, которая представляет собой совокупность сервисов, сервисных компонентов и компонентов доступа к данным, способных проектироваться и разворачиваться в одном приложении [15]. Взаимодействие между сервисами и различными компонентами осуществляется путем обмена сообщениями.

В примененной нами композитной сервис-ориентированной архитектуре важную роль играют сервисные компоненты, которые являются строительными блоками модулей нашей программной системы. Каждый компонент размещается в определенном сервисном контейнере. Сообщения передаются в сервисные контейнеры, а затем внутри них перенаправляются в соответствующие сервисные компоненты. Кроме того, сервисные контейнеры могут взаимодействовать и между собой. Например, предназначенное для какого-либо процесса сообщение сначала посылается в контейнер этого процесса, а затем сервисный контейнер обрабатывает полученную информацию и перенаправляет сообщения на соответствующие сервисные компоненты внутри себя. Сервисные компоненты в разработанной гибридной архитектуре решают следующие задачи:

- описывают и реализуют операции предметной области;
- задают правила операций;
- передают сообщения между компонентами системы.

Сами сервисы обеспечивают взаимодействие между композитным приложением и его потребителями. Связь с сервисом осуществляется по заданным протоколам (например, SOAP/HTTP). Компоненты доступа к данным служат для извлечения и модификации данных на основе сообщений, переданных от сервисных компонентов. Связи описывают способы обмена информацией между сервисами и сервисными компонентами, между

различными сервисными компонентами или сервисными компонентами и компонентами доступа к данным.

Применяемая композитная сервис-ориентированная архитектура предоставляет возможность выстроить систему из набора независимых модулей, каждый из которых реализует отдельную операцию, которая является логически обособленной, повторяющейся задачей, являющейся составной частью производственного процесса предприятия. Более того, сервисы могут быть реализованы независимо от языков программирования и других технических особенностей реализации, что дает возможность использовать различные технологии. Также сервисы могут быть написаны в независимости от других служб системы, необходимо только знание интерфейса используемых сервисов, то есть службы будут слабосвязанными. Использование описанного в работе архитектурного подхода также позволило достичь положительных результатов при интеграции с существующими программными продуктами предприятий, значительно сократить сложность и стоимость разработки новых компонентов, а также поддержки «унаследованных» частей системы.

Заключение. Работа посвящена комплексной разработке системы, предназначенной для предварительной обработки данных, моделирования и управления многомерными дискретно-непрерывными процессам. Ядро системы составляют непараметрические алгоритмы. Представлены два базовых алгоритма для моделирования процесса неизвестной структуры и управления в условиях неполноты данных об объекте. Разработанная архитектура системы носит гибкий характер – замена ключевых модулей не влечет за собой необратимых последствий при функционировании системы в целом.

References

1. Levine W. S. et al. Control system advanced methods. – Boca Raton, FL : CRC press, 2011, 50 p.
2. Hovakimyan N., Cao C. Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation. Society for Industrial and Applied Mathematics, 2010.
3. Jianling Q., Guang M. Design of glass furnace control system based on model-free adaptive controller. *Second International Conference on Computer Modeling and Simulation*. IEEE, 2010, Vol. 4, P. 130–133.
4. Eren-Oruklu M. et al. Adaptive system identification for estimating future glucose concentrations and hypoglycemia alarms. *Automatica*. 2012, Vol. 48, No. 8, P. 1892–1897.
5. Wu L., Qiu X., Guo Y. A simplified adaptive feedback active noise control system. *Applied Acoustics*. 2014, Vol. 81, P. 40–46.
6. Medvedev A. V. *Osnovy teorii adaptivnyh system* [Basic theory of adaptive systems]. Krasnoyarsk, SibGAU Publ., 2015, 525 p.
7. Medvedev A. V. [The theory of nonparametric systems. Simulation]. *Vestnik SibGAU*. 2010, Vol. 4 (30), P. 4–9 (In Russ.).
8. Raskina A. V. [Determination of the structure of a linear dynamic object in nonparametric identification problems]. *Vestnik SibGAU*. 2016, Vol. 4, P. 891–896 (In Russ.).
9. Bannikova A. V. Medvedev A. V. [On objects with memory management in a non-parametric uncertainty]. *Vestnik SibGAU*. 2014, Vol. 5(57), P. 26–37 (In Russ.).
10. Raskina A. V. *Neparametricheskie algoritmy identifikatsii i dual'nogo upravleniya dinamicheskimi ob"ektami*. Dokt. Diss. [Nonparametric algorithms of identification and dual control of dynamic objects. Doct. Diss.]. Krasnoyarsk, 2018, 122 p.
11. Nadaraja Je. A. *Neparametricheskie ocenki plotnosti verojatnosti i krivoj regressii* [Nonparametric estimation of probability density and the regression curve]. Tbilisi, 1983, 194 p.
12. Fel'dbaum A. A. *Osnovy teorii optimal'nyh avtomaticheskikh system* [Fundamentals of the theory of optimal automatic systems]. Moscow, Fizmatgiz Publ., 1963, 552 p.

13. SOA in the Real World. Available at: <http://www.microsoft.com/en-us/download/details.aspx?id=16187> (accessed 20.12.2019).
14. Northrop L. Software Product Lines Essentials. Pittsburg: SEI Carnegie Mellon University, 2008, 85 p.
15. Videnin S. A., Kostyuk A. V., Vasil'ev E. V. [Service-oriented architecture in modern information systems]. *Sovremennye nauchnye issledovaniya i innovacii*. 2016, Vol. 7. P. 121–123 (In Russ.).

Библиографические ссылки

1. Control system advanced methods / Levine W. S. et al. Boca Raton, FL : CRC press, 2011. 50 p.
2. Hovakimyan N., Cao C. Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation. Society for Industrial and Applied Mathematics, 2010.
3. Jianling Q., Guang M. Design of glass furnace control system based on model-free adaptive controller // Second International Conference on Computer Modeling and Simulation. IEEE. 2010. Vol. 4. P. 130–133.
4. Adaptive system identification for estimating future glucose concentrations and hypoglycemia alarms / Eren-Oruklu M. et al. // Automatica. 2012. Vol. 48, no. 8. P. 1892–1897.
5. Wu L., Qiu X., Guo Y. A simplified adaptive feedback active noise control system // Applied Acoustics. 2014. Vol. 81. P. 40–46.
6. Медведев А. В. Основы теории адаптивных систем / СибГАУ. Красноярск, 2015. 525 с.
7. Медведев А. В. Теория непараметрических систем моделирование // Вестник СибГАУ. 2010. № 4 (30). С. 4–9.
8. Раскина А. В. Определение структуры линейного динамического объекта в задачах непараметрической идентификации // Вестник СибГАУ. 2016. № 4. С. 891–896.
9. Банникова А. В. Медведев А. В. Об управлении объектами с памятью в условиях непараметрической неопределенности // Вестник СибГАУ. 2014. № 5(57). С. 26–37.
10. Раскина А. В. Непараметрические алгоритмы идентификации и дуального управления динамическими объектами : дисс. ... канд. техн. наук. Красноярск, 2018. 122 с.
11. Надарая Э. А. Непараметрические оценки плотности вероятности и кривой регрессии. Тбилиси : Изд. Тбил. у-та, 1983. 194 с.
12. Фельдбаум А. А. Основы теории оптимальных автоматических систем. М. : Физматгиз, 1963. 552 с.
13. SOA in the Real World [Электронный ресурс]. URL: <http://www.microsoft.com/en-us/download/details.aspx?id=16187> (дата обращения: 20.12.2019).
14. Northrop L. Software Product Lines Essentials. L. Northrop. Pittsburg: SEI Carnegie Mellon University. 2008. 85 p.
15. Виденин С. А., Костюк А. В., Васильев Э. В. Сервис-ориентированная архитектура в современных информационных системах // Современные научные исследования и инновации. 2016. № 7. С. 121–123.

© Raskina A. V., Videnin S. A., Chzhan E. A., Yusupova R. R., 2020

Raskina Anastasia Vladimirovna – Cand. Sc., associate professor of Department of Information Systems, Siberian Federal University, School of Space and Information Technology. E-mail: raskina.1012@yandex.ru.

Videnin Sergey Aleksandrovich – Cand. Sc., associate professor of Department of Information Systems, Siberian Federal University, School of Space and Information Technology. E-mail: videninserg@mail.ru.

Chzhan Ekaterina Anatolyevna – Cand. Sc., associate professor of Department of Intelligent Control Systems, Siberian Federal University, School of Space and Information Technology. E-mail: ekach@list.ru.

Yusupova Ramilya Ramilevna – student, Siberian Federal University, School of Space and Information Technology. E-mail: sr.eagleowl@gmail.com.

Раскина Анастасия Владимировна – кандидат технических наук, доцент кафедры информационных систем, Сибирский федеральный университет, Институт космических и информационных технологий. E-mail: raskina.1012@yandex.ru.

Виденин Сергей Александрович – кандидат педагогических наук, доцент кафедры информационных систем, Сибирский федеральный университет, Институт космических и информационных технологий. E-mail: videninserg@mail.ru.

Чжан Екатерина Анатольевна – кандидат технических наук, доцент кафедры интеллектуальных систем управления, Сибирский федеральный университет, Институт космических и информационных технологий. E-mail: ekach@list.ru.

Юсупова Рамиля Рамилевна – студент, Сибирский федеральный университет, Институт космических и информационных технологий. E-mail: sr.eagleowl@gmail.com.