

Использование искусственных иммунных систем для управления летательными аппаратами

И. Ф. Астахова^{1*}, Ю. В. Хицкова²

¹ ФГБОУ ВО «Воронежский государственный университет», г. Воронеж, Российская Федерация
Адрес: 394018, Российская Федерация, г. Воронеж, Университетская площадь, д. 1

² ФГБОУ ВО «Воронежский государственный технический университет», г. Воронеж, Российская Федерация

Адрес: 394006, Российская Федерация, г. Воронеж, ул. 20-летия Октября, д. 84

* astachova@list.ru

Аннотация

Работа посвящена разработке компьютерной системы для управления летательными аппаратами на основе искусственных иммунных систем. Теория была предложена в книге Дасгупты. В статье будет предложена реализация к применению управления летательным аппаратом. Реализация алгоритма автономного режима траекторий движения летательного аппарата выполнена на языке программирования Python. Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью. Выбор данной технологии в первую очередь основывается на её распространенности в сфере разработки машинного обучения, что, в свою очередь, гарантирует наличие большого количества сторонних библиотек, а также длительную поддержку и возможность расширения проекта в будущем. При программной реализации использовались библиотеки NumPy и SciPy. NumPy – пакет для научных вычислений, основными преимуществами которого являются удобство в использовании, оптимизированность, большой набор встроенных математических функций, направленность на работу с многомерными массивами, а также постоянная техническая поддержка со стороны разработчика.

SciPy – библиотека с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов. При реализации проекта использовалась в целях получения решения уравнения Риккати. Также стоит отметить, что при разработке данной информационной системы были составлены некоторые схемы и диаграммы, проектирование которых осуществлялось в таких приложениях, как ERwin Process Modeler и StarUML. ERwin Process Modeler – это инструмент для моделирования, анализа, документирования и оптимизации бизнес-процессов, разработанный компанией Logic Works. Ранее назывался Vpwin.

Ключевые слова: автономный летающий аппарат, траектория, Python-язык, NumPy, SciPy, ERwin Process Modeler, StarUML

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Астахова И. Ф., Хицкова Ю. В. Использование искусственных иммунных систем для управления летательными аппаратами // Современные информационные технологии и ИТ-образование. 2026. Т. 22, № 1. С. 147-154. <https://doi.org/10.25559/SITITO.022.202601.147-154>

© Астахова И. Ф., Хицкова Ю. В., 2026



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Using Artificial Immune Systems for Aircraft Control

I. F. Astachova^{a*}, Ju. V. Khitskova^b

^a Voronezh State University, Voronezh, Russian Federation

Address: 1 Universitetskaya pl., Voronezh 394018, Russian Federation

^b Voronezh State Technical University, Voronezh, Russian Federation

Address: 84, 20-letiya Oktyabrya St., Voronezh 394006, Russian Federation

* astachova@list.ru

Abstract

The work is devoted to the development of a computer system for controlling aircraft based on artificial immune systems. The algorithm and theory were proposed in the book by Dasgupta. The article will propose an implementation for the application of aircraft control. The implementation of the algorithm of the autonomous mode of the aircraft trajectories is implemented in the Python programming language. Python is a high-level general-purpose programming language with dynamic strict typing and automatic memory management. The implementation of the algorithm of the autonomous mode of the aircraft trajectories is implemented in the Python programming language. The choice of this technology is primarily based on its prevalence in the field of machine learning development, which in turn guarantees the availability of a large number of third-party libraries, as well as long-term support and the possibility of expanding the project in the future. The Numpy and Scipy libraries were used in the software implementation. Numpy is a package for scientific computing, the main advantages of which are ease of use, optimization, a large set of built-in mathematical functions, focus on working with multidimensional arrays, as well as ongoing technical support from the developer. Scipy is an open source library designed to perform scientific and engineering calculations. During the project implementation, it was used to obtain a solution to the Riccati equation. It is also worth noting that during the development of this information system, some diagrams and charts were compiled, the design of which was carried out in applications such as ERwin Process Modeler and StarUML. ERwin Process Modeler is a tool for modeling, analyzing, documenting and optimizing business processes, developed by Logic Works. Formerly known as Bpwin.

Keywords: autonomous flying vehicle, trajectory, Python language, Numpy, Scipy, ERwin Process Modeler, StarUML

Conflict of interests: The authors declare no conflict of interest.

For citation: Astachova I.F., Khitskova Ju.V. Using Artificial Immune Systems for Aircraft Control. *Modern Information Technologies and IT-Education*. 2026;22(1):147-154. <https://doi.org/10.25559/SITITO.022.202601.147-154>



Введение

Существуют следующие интеллектуальные методы, которые применяются при разработке информационных систем: выделим биоинспирированные методы для управления летательными аппаратами:

1. Роевый интеллект (*Swarm Intelligence*) – это самое развитое направление, связанное с описанием поведения стай птиц, косяков рыб, колоний насекомых, оно связано с децентрализованным управлением, самоорганизацией, с коллективным поведением на основе простых правил (соблюдение дистанции, сохранение скорости, достижение цели). Основные алгоритмы: оптимизация муравьиной колонии (ACO) и метод пчелиной колонии (ABC). Эти методы используются для планирования маршрутов и распределения задач в рое [1].

2. Нейроморфные вычисления и импульсные нейронные сети (SNN): моделирование структуры и принципов работы биологического мозга для обработки сенсорной информации и управления. Обработка информации происходит в виде импульсов, асинхронности, способности к обучению в полете [2].

3. Эволюционные алгоритмы и обучение с подкреплением (как форма «искусственной эволюции»). Происходит имитация естественного отбора для оптимизации параметров контроллера или архитектуры нейросети. Основные принципы: «Особь» (алгоритмы управления) соревнуются, лучшие «скрещиваются» и «мутируют», порождая следующее поколение [3]. В это направление можно отнести искусственные иммунные системы и решение задач с их помощью.

4. Глубокое обучение с подкреплением (Deep RL) для обучения сложным маневрам [4].

Эта статья посвящена использованию искусственных иммунных систем в исследовании полетов летательных аппаратов. Появление искусственных иммунных систем в России связано с опубликованием книги под редакцией Д. Дасгупты [5], которая первая в мировой литературе целиком посвящена новой области информатики и математического моделирования – искусственным иммунным системам. Основные используемые семейства алгоритмов из класса биоинспирированных систем имеют много общего. Выделяя иммунные системы, можно сказать, что их существенное преимущество перед генетическими алгоритмами и искусственными нейронными сетями – это способность к быстрому обучению и наличие памяти. Другими словами, иммунные системы выбрали лучшее из существующих биоинспирированных алгоритмов и продолжают активно изучаться в целях применения их к различным областям знаний.

В связи с тем, что различные системы имеют разную способность к самосовершенствованию, считается удобным определить интеллектуальное управление на основе классификации его уровней и указать соответствующие им иммунные ответы [5]:

Уровень 0. Осуществляется робастное управление с обратной связью. Этому уровню соответствует первый уровень защиты – врожденный иммунитет, приводящий в состояние готовности следующий уровень защиты.

Уровень 1. Реализуется робастное управление с обратной связью и адаптивными управляющими параметрами. На этом

этапе Т-клетки активируют В-лимфоциты (адаптация) для нейтрализации антигена.

Уровень 2. Уровень оптимального управления: робастное адаптивное управление с обратной связью, которое минимизирует или максимизирует функцию качества. Макрофаги передают антиген Т-лимфоцитам с активацией Т-клеточного ответа (адаптация).

Уровень 3. Создается система планового управления, добавляется способность планирования заранее неопределенных ситуаций, имитирование и моделирование неопределенности. На этом шаге определённый антиген адаптируется вместе с конечным продуктом – антителом и видоизменяет регулятор. Следующие уровни интеллектуального управления только предстоит определить. Стоит отметить, что установление данных уровней интеллектуального управления с определёнными характеристиками определяется количественно и измеряется разными способами. Важным является и свойство расширяемости такой классификации, оставляющее в определении место для дальнейшего совершенствования с учетом неизбежных последующих нововведений.

Целью данной работы является рассмотрение специфики возможностей искусственных иммунных систем в задаче автономного управления летательными аппаратами, частичная программная реализация возможных этапов такой системы и исследование особенностей поставленной задачи с точки зрения разработки искусственных иммунных систем.

Будем пользоваться теорией, предложенной в книге Дасгупты [5]. В статье будет предложена реализация применительно к управлению летательным аппаратом, а именно, уточнение его подлинной траектории.

Реализация алгоритма автономного режима траекторий движения летательного аппарата выполнена на языке программирования Python¹ [6-16].

Python – высокоуровневый язык программирования общего назначения с динамической сторогой типизацией и автоматическим управлением памятью. Выбор данной технологии в первую очередь основывается на её распространенности в сфере разработки машинного обучения, что, в свою очередь, гарантирует наличие большого количества сторонних библиотек, а также длительную поддержку и возможность расширения проекта в будущем.

При программной реализации использовались библиотеки NumPy и SciPy.

NumPy – пакет для научных вычислений, основными преимуществами которого являются удобство в использовании, оптимизированность, большой набор встроенных математических функций, направленность на работу с многомерными массивами, а также постоянная техническая поддержка со стороны разработчика.

SciPy – библиотека с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов. При реализации проекта использовалась в целях получения решения уравнения Риккати.

Также стоит отметить, что при разработке данной информационной системы были составлены некоторые схемы и диаграммы, проектирование которых осуществлялось в таких прило-

¹ Selective theories of antibody formation [Electronic resource] // Immunology, 2025. URL: <http://imuno.net/59.php> (дата обращения: 14.12.2025).



жениях, как Erwin Process Modeler и StarUML.

Erwin Process Modeler – это инструмент для моделирования, анализа, документирования и оптимизации бизнес-процессов, разработанный компанией Logic Works. Ранее назывался Bpwin.

StarUML – программный инструмент визуального моделирования с открытым кодом, который поддерживает стандартизированный язык графического описания UML (*Unified Modeling Language*) для моделирования систем и программного обеспечения.

1 Алгоритм автономного режима

В работе Дасгупты Д. [5] приводится определение «иммунного адаптивного критика» (ИАК), который используется для определения реальной траектории летательного аппарата, при условии, что известны точки траектории, которые должны быть по плану полета. Оно возникло на стыке понятий о нейронных сетях, динамическом программировании и обучении с подкреплением. Как адаптивные, так и неадаптивные критики были предложены в качестве механизмов, способных моделировать свои будущие эксплуатационные параметры. Они представили формулы, необходимые для расчета траектории летательного аппарата, воспользуемся этими формулами для искусственной иммунной системы² [15-23].

Автономный режим. Этапы автономного режима работы включают построение набора вычислительных модулей и статистических данных. Согласно Дасгупте [5] рассмотрим этапы: Этап 1. Определяются четыре матрицы A_p (состояния траектории) с помощью метода моделирования динамики системы первого порядка. Матрица B пространства состояний выбрана постоянной (единичная матрица), также заданы 5 матриц Q , а матрица R считается постоянной (единичная). Каждая из четырех матриц A_p соответствует одному из вариантов маневров ЛА (летательного аппарата), выполняемых с различными временными константами. Таким образом, матрицы A_p – это диагональные матрицы размерности 3×3 , определяемые выбором временных констант τ_V , τ_H , τ_Ψ . Матрицы Q при этом должны были выбираться как $0.001R$, $0.01R$, $0.1R$, R и $10R$. Далее матрицы A_p , B , Q и R используются для вычисления 20 различных матриц S , при этом отдельные элементы матриц S служат модулями управления.

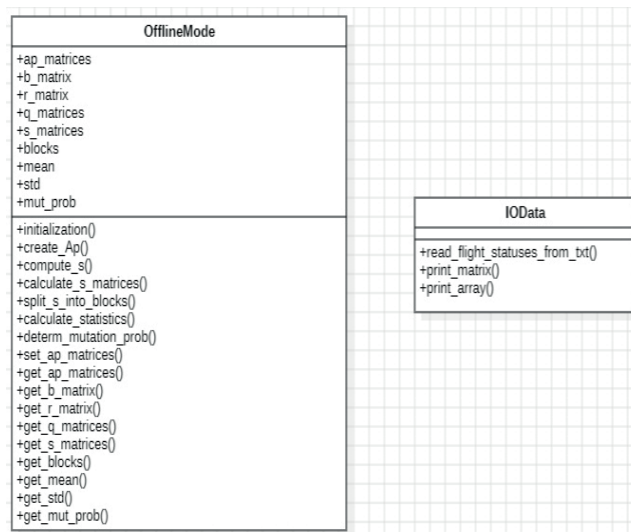
Этап 2. Матрицы S необходимо разбить на модули, и так как они имеют размерность 3×3 , да к тому же ещё симметричны, то всего получается 6 блоков (S_{11} , S_{12} , S_{13} , S_{22} , S_{32} , S_{33}) от каждой из 20 матриц, что в сумме даёт 120 априорных блоков. Этап 3. На этом шаге вычисляются статистики первого и второго порядка для каждого из элементов матриц S . Другими словами, для величин S_{11} , S_{12} , S_{13} , S_{22} , S_{32} , S_{33} вычисляются среднее значение μ и стандартное отклонение σ .

Этап 4. В заключение модули упорядочиваются по величине стандартного отклонения и определяются вероятности мутаций таким образом, чтобы модуль с наибольшим стандартным отклонением имел наибольшую вероятность мутации и так

далее³ [20-25]. Эти данные поступают в вычисления в оперативном режиме. В следующий раз при обращении к автономному режиму все повторяется.

2 Реализация автономного алгоритма

При реализации проекта были разработаны классы `IOData` и `OfflineMode`, представление которых можно увидеть на рис. 1. Данная диаграмма классов показывает поля и методы разработанных структур.



Р и с. 1. Диаграмма классов автономного режима генератора

F i g. 1. Class diagram of the generator's standalone mode

Источник: здесь и далее в статье все рисунки составлены авторами.

Source: Hereinafter in this article all figures were drawn up by the authors.

`IOData` – это класс, основной задачей которого является ввод/вывод информации. Метод `read_flight_statuses_from_txt()` считывает данные о состоянии траектории полёта из указанного файла формата txt и возвращает их в виде вложенных списков. Функции `print_matrix()` и `print_array()` служат для вывода матриц (двумерных массивов) и массивов на экран (в консоль) с предшествующим переданным сообщением.

`OfflineMode` – класс, предоставляющий набор встроенных полей и инструментов для осуществления алгоритма автономного режима. Все атрибуты структуры несут информативные названия и представляют из себя входные, промежуточные или итоговые результаты вычислений. Тип сохраняемых данных определяется структурой соответствующей информации. Класс `OfflineMode` включает в себя следующие методы: `initializer()` – некоторый метод, который принимает массив состояний траекторий полёта и инициализирует объект с полями `ap_matrices`, `b_matrix`, `r_matrix` и `q_matrices`.

² Ушаков С. А. Разработка и исследование алгоритмов решения задач распознавания на основе искусственных иммунных систем : дис. ... канд. тех. наук. Воронеж: ВГУ, 2016. 138 с. EDN: TRZGIU

³ Makoviy K., Khitskova Yu., Metelkin Ya. Server hardware resources optimization for virtual desktop implementation // Proceedings of the IV International Conference on Information Technology and Nanotechnology (ITNT-2017). Samara : Predpriyatie "Novaya tekhnika", 2017. P. 1394-1397. EDN: YQLRKB



create_ap() и compute_s() – функции, которые поштучно **вычисляют** матрицы Ap и S соответственно.

calculate_s_matrices() – метод, с помощью которого создается нужный набор матриц S.

split_s_into_blocks() – функция, чьей задачей является разбиение всех матриц S на модули.

calculate_statistics() – метод, использующийся для вычисления среднего значения и стандартного отклонения модулей матриц S (поле blocks).

determ_mutation_prob() – функция, упорядочивающая модули blocks по величине стандартного отклонения и вычисляющая вероятности мутаций (поле mut_prob).

set_ap_matrices() – задает новый набор входных данных.

get_...() – группа методов, направленных на получение данных из соответствующих полей (название методов строится по шаблону get_название_поля).

Таким образом, разработанные классы поддерживают широкий спектр методов и позволяют не только производить поэтапные вычисления, но и отслеживать получаемые данные на

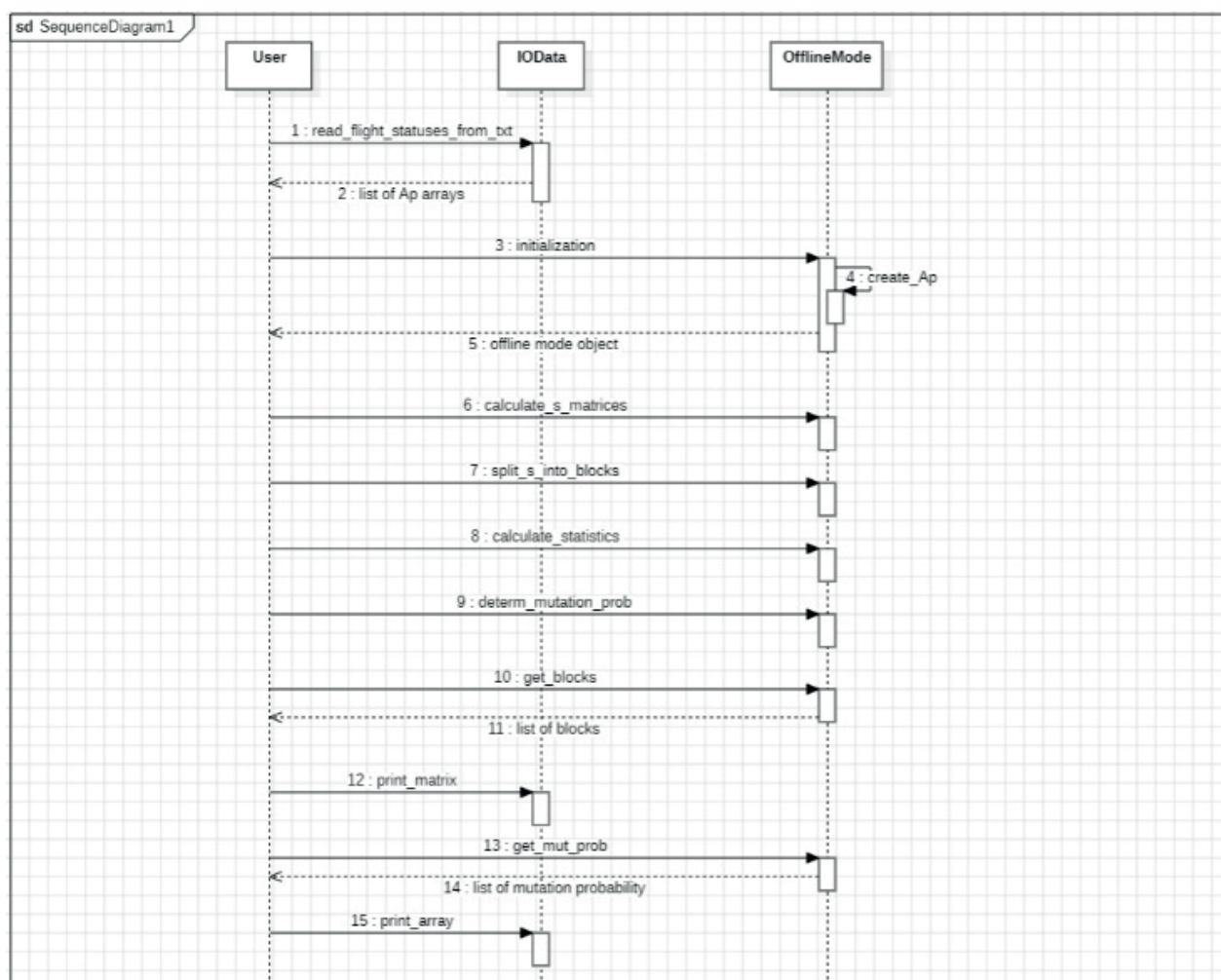
каждой из стадий. В свою очередь, это даёт возможность более детально отслеживать ход решения алгоритма и осуществлять контроль информации на каждом из действий.

Общую последовательность действий от считывания данных из файла до получения и вывода результата в консоль можно увидеть на рис. 2.

Представленная диаграмма демонстрирует минимальный набор поочерёдных действий со считыванием выходных данных из файла, посредством проведения которого алгоритм будет считаться выполненным.

При этом, как было сказано ранее, разработанные классы предоставляют возможность отслеживать полученные данные на каждом из этапов. Понятное изложение данной вариативности можно рассмотреть на диаграмме активности, изображенной на рис. 3.

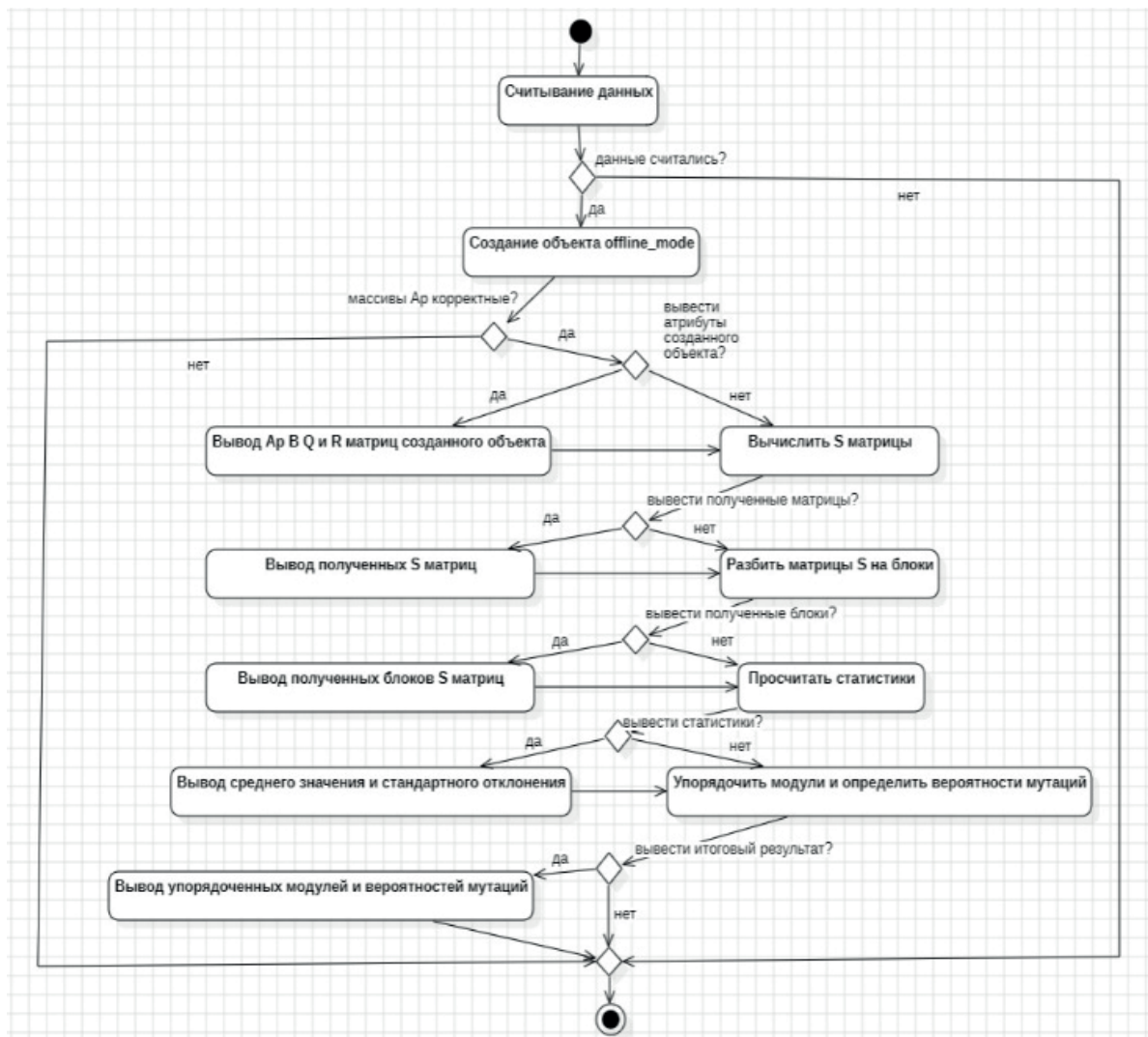
Анализ предложенных схем позволяет дополнить и сформировать полное представление о разработанной информационной системе.



Р и с. 2. Диаграмма последовательности автономного режима генератора

F i g. 2. Sequence diagram of the generator's standalone mode





Р и с. 3. Диаграмма активности разработанного автономного режима генератора
F i g. 3. Activity diagram of the developed generator standalone mode

Таким образом, получая на вход массив состояния полёта ЛА, алгоритм программы вычисляет и впоследствии разбивает матрицы S на блоки. Вычисленные по этим блокам статистики используются для их же упорядочивания и просчитывания соответствующих вероятностей мутаций. Итогом выполненных действий являются упорядоченные модули матриц S и соответствующие им вероятности мутаций, которые и считаются результатом разрабатываемой информационной системы. Впоследствии полученные данные используются в качестве входных параметров для Оперативного режима.

Обсуждение и заключение

Управление в автономном режиме превращает летательные аппараты в полноценные роботы, способные самостоятельно решать практические задачи. Цель исследований смещается с

пилотирования на постановку задачи и контроль результатов. Ключевые проблемы безопасности, надежности и регулирования остаются областями активных исследований и разработок, которые определяют скорость интеграции автономных летательных аппаратов в нашу повседневную жизнь. Сила искусственных иммунных систем для управления летательными аппаратами — не в замену классической теории управления, а в дополнение ее на тех участках, где требуется распознавание сложных паттернов и адаптация к непредвиденным ситуациям. Основное направление исследований, вытекающее из анализа современной литературы — это создание гибридных интеллектуальных систем, сочетающих иммунные алгоритмы с нейросетями, нечеткой логикой и эволюционными вычислениями для создания нового поколения автономных, надежных и «умных» летательных аппаратов [5], [20-25].



References

- [1] Skobtsov Y. Artificial Immune Systems – Models and Applications. In: Kravets A.G., Bolshakov A.A., Shcherbakov M. (eds) Cyber-Physical Systems: Intelligent Models and Algorithms. *Studies in Systems, Decision and Control*. Vol. 417. Cham: Springer; 2022. p. 29-42. https://doi.org/10.1007/978-3-030-95116-0_3
- [2] Floreano D., Mattiussi C. Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies. MIT Press; 2008. 544 p.
- [3] Kaufmann E., et al. Champion-Level Drone Racing using Deep Reinforcement Learning. *Nature*. 2023;620:982-987. <https://doi.org/10.1038/s41586-023-06419-4>
- [4] Kaufmann E., Bauersfeld L., Loquercio A., Müller M., Koltun V., Scaramuzza D. Champion-level drone racing using deep reinforcement learning. *Nature*. 2023;620:982-987. <https://doi.org/10.1038/s41586-023-06419-4>
- [5] Dasgupta D. Artificial Immune Systems and Their Applications. Springer Publishing Company, 2014. 306 p.
- [6] VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data. USA, O'Reilly Media; 2016. 548 p.
- [7] Idris I. NumPy Cookbook. Great Britain: Packt Publishing; 2012. 226 p.
- [8] Oliphant T.E. Guide to NumPy. 2nd Edition. USA, GreateSpace Independent Publishing Platform; 2015. 364 p.
- [9] McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter. 3rd Edition. USA: O'Reilly Media; 2017. 579 p.
- [10] Grus J. Data Science from Scratch: First Principle with Python. USA: O'Reilly Media; 2019. 406 p.
- [11] Muller A.C., Guido S. Introduction to Machine Learning with Python: A Guide for Data Scientists. USA: O'Reilly Media; 2016. 400 p.
- [12] Brownley C.W. Foundations for Analytics with Python Non-Programmer to Hacker. USA: O'Reilly Media; 2016. 300 p.
- [13] Burnet F.M. A Modification of Jerne's theory of antibody production using the concept of clonal selection. *CA: A Cancer Journal for Clinicians*. 1976;26(2):119-121. <https://doi.org/10.3322/canjclin.26.2.119>
- [14] McLaughlin R.G., Perhinschi M.G. Partitioned artificial immune system for detection and identification of autonomous flight vehicle abnormal conditions. *International Journal of Intelligent Unmanned Systems*. 2020;9(4):237-255. <https://doi.org/10.1108/IJIUS-11-2019-0064>
- [15] Neri F., Cotta C., Moscato P. Handbook of Memetic Algorithms. Heidelberg: Springer Berlin; 2012. 370 p. <https://doi.org/10.1007/978-3-642-23247-3>
- [16] Sudholt D. Memetic algorithms with variable-depth search to overcome local optima. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO '08). New York, NY, USA: Association for Computing Machinery; 2008. p. 787-794. <https://doi.org/10.1145/1389095.1389251>
- [17] Smith J.E. Self-adaptative and coevolving memetic algorithms. *Studies in Computational Intelligence*. 2012;379:167188. https://doi.org/10.1007/978-3-642-23247-3_11
- [18] Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. Cambridge: MIT Press; 1998. 342 p.
- [19] Buzdalova A., Kononov V., Buzdalov M. Selecting evolutionary operators using reinforcement learning: Initial explorations. In: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO Comp '14). New York, NY, USA: Association for Computing Machinery; 2014. p. 1033-1036. <https://doi.org/10.1145/2598394.2605681>
- [20] Perhinschi M.G., et al. Immunity-based flight envelope prediction at post-failure conditions. *Aerospace Science and Technology*. 2015;46:264-272. <https://doi.org/10.1016/j.ast.2015.07.014>
- [21] Astachova I.F., Kiseleva E.I. The algorithm is the use of artificial immune systems to optimize the target component of the information educational system. *Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies*. 2017;(2):61-65. (In Russ., abstract in Eng.) EDN: ZDWOMN
- [22] Astachova I.F., Ushakov S.A., Shashkin A.I., Belyaeva N.V. The application of artificial immune system for parallel process of calculation and their comparison with existing methods. *Journal of Physics: Conference Series*. 2019;1202:012003. <https://doi.org/10.1088/1742-6596/1202/1/012003>
- [23] Makoviy K.A., Hitskova Yu.V., Geras S.V. The use of hybrid method in the field of information technology. *Informacionnye tekhnologii v stroitel'nyh, social'nyh i ekonomicheskikh sistemah* = Information technologies in construction, social and economic systems. 2016;(1):120-124. (In Russ., abstract in Eng.) EDN: WYEHUH
- [24] Astachova I.F., Makoviy K.A., Khitskova Y.V. Possibilities for predicting the state of usability web resources. *Journal of Physics: Conference Series*. 2021;1902(1):012029. <https://doi.org/10.1088/1742-6596/1742-6596/1902/1/012029>
- [25] Astachova I., Kiseleva E. The Application of the Artificial Immune System for Design, Development and Using of the Hybrid System in Education. In: Sukhomlin V., Zubareva E. (eds.) Modern Information Technology and IT Education. SITITO 2017. *Communications in Computer and Information Science*. Vol. 1204. Cham: Springer; 2021. p. 67-75. https://doi.org/10.1007/978-3-030-78273-3_7

Поступила 14.12.2025; одобрена после рецензирования 11.02.2026; принята к публикации 18.03.2026.

Submitted 14.12.2025; approved after reviewing 11.02.2026; accepted for publication 18.03.2026.



Об авторах:

Астахова Ирина Федоровна, профессор кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики, ФГБОУ ВО «Воронежский государственный университет» (394018, Российская Федерация, г. Воронеж, Университетская площадь, д. 1), доктор технических наук, профессор, **ORCID: <https://orcid.org/0000-0002-2627-8508>**, astachova@list.ru

Хицкова Юлия Владимировна, доцент кафедры систем управления и информационных технологий в строительстве факультета информационных технологий и компьютерной безопасности, ФГБОУ ВО «Воронежский государственный технический университет» (394006, Российская Федерация, г. Воронеж, ул. 20-летия Октября, д. 84), кандидат экономических наук, доцент, **ORCID: <https://orcid.org/0000-0001-6322-8633>**, julia.hickova@gmail.com

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Irina F. Astachova, Professor of the Chair of Computer Hardware, Faculty of Applied Mathematics, Informatics and Mechanics, Voronezh State University (1 Universitetskaya pl., Voronezh 394018, Russian Federation), Dr. Sci. (Eng.), Professor, **ORCID: <https://orcid.org/0000-0002-2627-8508>**, astachova@list.ru

Yulia V. Khitskova, Associate Professor of the Chair of Control Systems and Information Technologies in Construction, Faculty of Information Technology and Computer Safety, Voronezh State Technical University (84, 20-letiya Oktyabrya St., Voronezh 394006, Russian Federation), Cand. Sci. (Econ.), Associate Professor, **ORCID: <https://orcid.org/0000-0001-6322-8633>**, julia.hickova@gmail.com

All authors have read and approved the final manuscript.

