



АДАПТИВНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ БПЛА

Д. В. Анисимов

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Рассмотрены современные методы адаптивного управления беспилотными летательными аппаратами в условиях внешних возмущения и целенаправленных воздействий.

Выполнен сравнительный анализ традиционных подходов управления беспилотными летательными аппаратами, включая нелинейный синтез и гибридные методы. Особое внимание уделено интеграции алгоритмов машинного обучения в контур управления для решения задач навигации, планирования траектории и стабилизации полета. Представлена модель спуфинг-атаки, разработан адаптивный алгоритм защиты на основе статического обнаружения аномалий.

Создано программное обеспечение для моделирования трех типов траекторий движения беспилотного летательного аппарата под воздействием спуфинга с применением механизма машинного обучения.

Экспериментальная часть доказала состоятельность предложенного подхода при различных сценариях. Результаты показали перспективность гибридной архитектуры, сочетающей классические методы управления с адаптивными.

Ключевые слова: адаптивное управление, беспилотный летательный аппарат, машинное обучение, обнаружение аномалий, траектория управления, статистическая коррекция, спуфинг-атака.

Для цитирования:

Анисимов, Д. В. Адаптивные системы управления на основе машинного обучения для БПЛА / Д. В. Анисимов // Системный анализ и логистика. – 2025. – № 5(48). – с. 37-55. DOI: 10.31799/2077-5687-2025-5-37-55.

ADAPTIVE MACHINE LEARNING-BASED CONTROL SYSTEMS FOR UAVs

D. V. Anisimov

St. Petersburg State University of Aerospace Instrumentation

This paper examines modern methods for adaptive control of unmanned aerial vehicles (UAVs) under external disturbances and targeted influences.

A comparative analysis of traditional UAV control approaches, including nonlinear synthesis and hybrid methods, is performed. Particular attention is paid to machine learning algorithms in the control loop for solving navigation, trajectory planning, and flight management problems. A spoofing attack model is presented, and an adaptive defense algorithm based on static anomaly detection is developed.

Software for modeling three types of UAV trajectories under spoofing using a machine learning mechanism is created.

The experimental study demonstrated the feasibility of the proposed approach under various scenarios. Results from measuring the viability of a hybrid design combining classical and adaptive control methods are presented.

Keywords: adaptive control, unmanned aerial vehicle, machine learning, anomaly detection, control trajectory, statistical correction, spoofing attack.

For citation:

Anisimov, D. V. Adaptive machine learning-based control systems for UAVs / D. V. Anisimov // System analysis and logistics. – 2025. – № 5(48). – p. 37-55. DOI: 10.31799/2077-5687-2025-5-37-55.

Введение

Основная проблема при управлении беспилотными летательными аппаратами (БПЛА) – неопределенность модели и внешние возмущения – порывы ветра, намеренные воздействия на БПЛА (спуфинг) – которые могут существенно влиять на характеристики полета. Поэтому для решения проблемы разрабатываются различные адаптивные стратегии управления как одиночного БПЛА, так и роя.

Направленные на повышение автономности, устойчивости и эффективности в условиях неопределенности и внешних возмущений современные методы адаптивного управления беспилотными летательными аппаратами представляют собой сложный технический



комплекс. Такие методы критически важны для широкого спектра применений БПЛА: от наблюдения и доставки до разведки и операций по обеспечения безопасности. Традиционные системы управления полетом часто сталкиваются с серьезными изменениями в динамике, вызванные структурными повреждениями или отказами приводов. Адаптивные – предлагают решения, позволяющие корректировать при полете БПЛА стратегию управления в режиме реального времени, обеспечивая надежную работу и стабильность системы.

Современные адаптивные методы управления БПЛА

Российские научные работы выделяют несколько ключевых подходов и методов с их плюсами и минусами, что важно для выбора и применения в практике управления БПЛА (Таблица 1).

Таблица 1 – Методы адаптивного управления БПЛА

Метод	Описание	Преимущества	Недостатки
Нелинейный синтез [1]	Управление с учетом нелинейной динамики и возможностью адаптивной настройки параметров в реальном времени	Возможность адаптации к сложной динамике аппарата	Сложность реализации и вычислительных требований
Адаптивная самонастройка на основе нечеткой логики [1]	Использует правила нечеткой логики для настройки коэффициентов управления в реальном времени	Обеспечивает устойчивость, сохраняет качество при неизвестных параметрах, гибкость настройки	Требует разработки точных правил и функций Ляпунова, сложность настройки
Адаптивное управление по выходу [2]	Устойчив к ветровым возмущениям	Позволяет стабилизировать движение под воздействием возмущений и неопределенностей	Модельно-зависим, может требовать сложного синтеза алгоритмов
Гибридное адаптивное управление [3]	Комбинация классических и адаптивных алгоритмов управления для задач с высокой устойчивостью и адаптивностью	Эффективен при изменении параметров и внешних воздействиях, применим для манипуляторов на БПЛА	Высокие требования к вычислительным ресурсам, сложность настройки

Метод нелинейного синтеза ориентирован на управление с учетом полной нелинейной динамики БПЛА. Ключевое преимущество этого подхода – обеспечение гарантированной асимптотической устойчивости и возможность адаптации к сложной динамике аппарата, включая нестационарные системы, что позволяет достигать высокой точности и надежности управления в широком диапазоне режимов полета. Однако данный метод характеризуется повышенной вычислительной сложностью и трудоемкостью реализации, что затрудняет его применение в системах с ограниченными ресурсами.

Адаптивное управление на базе нечеткой логики предлагает иной подход, вместо точных математических моделей используется модель неопределенности через нечеткие множества и правила. Преимущество этого метода – гибкость в работе с неопределенностями и вариативностью параметров без необходимости в точной модели и сложных вычислениях. Такой подход хорошо справляется с внешними возмущениями и позволяет сохранить



устойчивость системы. Вместе с тем, качество управления сильно зависит от корректности и полноты нечетких правил, а их построение и обеспечение устойчивости через функции Ляпунова зачастую требуют значительной экспертной работы, что усложняет процесс проектирования.

Метод адаптивного управления по выходу выделяется способностью стабилизировать БПЛА при воздействии внешних факторов, даже при неполной измеримости всех состояний системы. Это дает преимущество в реальных условиях, где датчики могут быть ограничены либо подвержены ошибкам. Но при этом алгоритмы требуют модели состояния или ее приближения, а синтез управлений становится более сложным, что требует квалифицированного подхода к проектированию.

Гибридный метод адаптивного управления сочетает классические алгоритмы управления с адаптивными механизмами. Это обеспечивает универсальность и высокую адаптивность к изменяющимся условиям и параметрам объекта. Такие методы позволяют обойти отдельные ограничения чисто классических или сугубо адаптивных подходов, что делает их предпочтительным вариантом для современного комплексного управления БПЛА. Основной минус – высокие требования к ресурсам вычислительной и аппаратной части, а также большая сложность проектирования, тестирования и внедрения таких систем.

Обобщая, методы можно расположить по шкале применения и технических требований следующим образом: метод нелинейного синтеза лучше всего подходит для систем с высокой динамической точностью и ресурсами для сложных вычислений; нечеткая логика дает гибкость и адаптивность в условиях неопределенности, но требует детальной экспертной настройки; методы по выходу полезны при ограниченных возможностях измерения и неполных данных; гибридные методы – универсальное решение с компромиссом между адаптивностью и ресурсозатратами.

Основы машинного обучения в системах управления

Современные системы управления БПЛА сталкиваются с необходимостью функционирования в сложных, недетерминированных и динамически изменяющихся средах. Классические методы, основанные на точных математических моделях, зачастую не справляются с такой неопределенностью. Это обуславливает активное внедрение методов машинного обучения (МО), способных к обобщению и адаптации [4]. Систематизируем основные подходы к применению МО в контуре управления БПЛА.

Машинное обучение в задачах навигации и планирования траектории

Планирование миссии БПЛА можно разделить на два уровня: глобальный (маршрут) и локальный (обход препятствий).

На глобальном уровне МО используется для прогнозирования внешних условий что позволяет оптимизировать маршрут по энергии и времени. На локальном уровне наиболее перспективными являются гибридные подходы. Алгоритмы глубокого обучения, в частности сверточные нейронные сети (CNN), решают задачу семантической сегментации окружающего пространства, создавая карту препятствий. Эта карта используется классическими алгоритмами планирования, такими как RRTS (Rapidly-exploring Random Tree Star) или Model Predictive Control (MPC), для построения безопасной и выполнимой траектории [5]. Чистое сквозное обучение с подкреплением (RL) пока остается областью фундаментальных исследований из-за проблем с безопасностью и переносом из симуляции в реальный мир.

Низкоуровневое управление и стабилизация

Задача стабилизации и следования траектории требует высокой частоты и надежности управляющих сигналов. Прямая замена классических ПИД-регуляторов на глубокие нейронные сети несет в себе риски из-за недостаточной предсказуемости и сложности верификации. Поэтому более практичным для решения данных задач представляется использование МО для адаптации параметров существующих регуляторов. Нейросетевые системы могут в реальном времени подстраивать коэффициенты ПИД-регулятора в ответ на



изменения массы, инерции или аэродинамических характеристик аппарата [6]. Это повышает робастность системы управления без потери гарантий устойчивости, присущих классическим методам.

Восприятие окружающей среды с помощью компьютерного зрения

Данное направление является релевантным с точки зрения внедрения МО. Сверточные нейронные сети де-факто стали стандартом для решения задач обнаружения объектов, их классификации и семантической сегментации в видеопотоке с бортовой камеры [7]. Архитектуры типа U-Net и YOLO, оптимизированные для работы в реальном времени, позволяют БПЛА в полете идентифицировать статические и динамические препятствия, участки для безопасной посадки, объекты инфраструктуры. Интеграция семантической информации, полученной от CNN, в карты, построенные методами SLAM (Semantic SLAM), открывает путь к созданию автономных систем, способных "понимать" среду своего функционирования на концептуальном уровне [8].

Машинное обучение в контуре управления беспилотным летательным аппаратом: от эмпирических методов к системной интеграции.

Если подвести итог рассмотренному материалу и отойти от частных случаев, можно наблюдать этап становления новой технологической парадигмы. Машинное обучение, в особенности глубокое обучение, прошел путь от экзотического инструмента до обязательного компонента в архитектуре сложных автономных систем. Сформулируем несколько тезисов, основанных на анализе многолетней эволюции данной проблематики.

1. Попытки заменить всю детерминированную архитектуру управления одной нейронной сетью, сколь угодно глубокой, оказались невыигрышными с инженерной точки зрения. Сила МО – не в замене, а в дополнении: система обретает робастность, когда алгоритмы МО берут на себя те функции, где они имеют неоспоримое преимущество (работа с неформализуемой априори информацией, распознавание образов, прогнозирование в условиях неполных данных). В то время как классические методы управления, верифицируемые и предсказуемые, остаются основой для синтеза управляющих воздействий, гарантирующих устойчивость и безопасность.
2. Более привычно управлять объектом, имея его точную динамическую модель. Для БПЛА это система дифференциальных уравнений, описывающая его движения. Однако МО позволяет отказаться от этого канона, при этом новым объектом управления становится не сам аппарат, а траектория его поведения в семантическом пространстве состояний.
3. В идеальной модели мира мы специально стараемся ограждать систему детерминированными правилами, например, "не приближаться к препятствиям ближе N метров". В реальном же стохастическом мире такие жесткие ограничения либо избыточны, либо невыполнимы. МО позволяет перейти от бинарной логики "безопасно/опасно" к непрерывной оценке и минимизации меры риска (энтропии).

Математическая модель и алгоритмы адаптивного управления на основе машинного обучения при движении беспилотного летательного аппарата по различным траекториям полета

На основе проведенного обзора и сформулированных тезисов, а также референсных работы [9-11] создано программное обеспечение (ПО), моделирующее три вида движения БПЛА под действием спуфинг-атаки с применением МО. Выбранным языком программирования является Python по причине обеспечения высокой гибкости и контроля над каждым этапом алгоритма, также программирование на данном языке предоставляет широкий арсенал инструментов для визуализации и интерпретации результатов.

Общий вид главного окна программы представлен на Рис. 1 – имеется возможность:



- 1) выбирать один из трех видов движения БПЛА;
- 2) регулировать степень спуфинг-атаки, выражаемой в работе через интенсивность, с помощью интерактивного «ползунка»;
- 3) выбирать отображение траекторий (в работе и в ПО используется три вида: исходная – *зеленая*; атакованная, т.е. траектория БПЛА под действием спуфинга – *красная пунктирная*; защищенная, т.е. траектория БПЛА после срабатывания методов МО при детекции атаки – *синяя пунктирная*);
- 4) отслеживать время, с возможностью выбора коэффициента времени (скорость моделирования) для различных сценариев и объемов моделирования;
- 5) отслеживать работу программы с помощью встроенного окна логирования;
- 6) экспортировать данные моделирования в формате .csv для дальнейшего анализа;
- 7) переключаться между трем вкладками с отрисовкой графиков движения, тем самым отслеживая процесс моделирования.

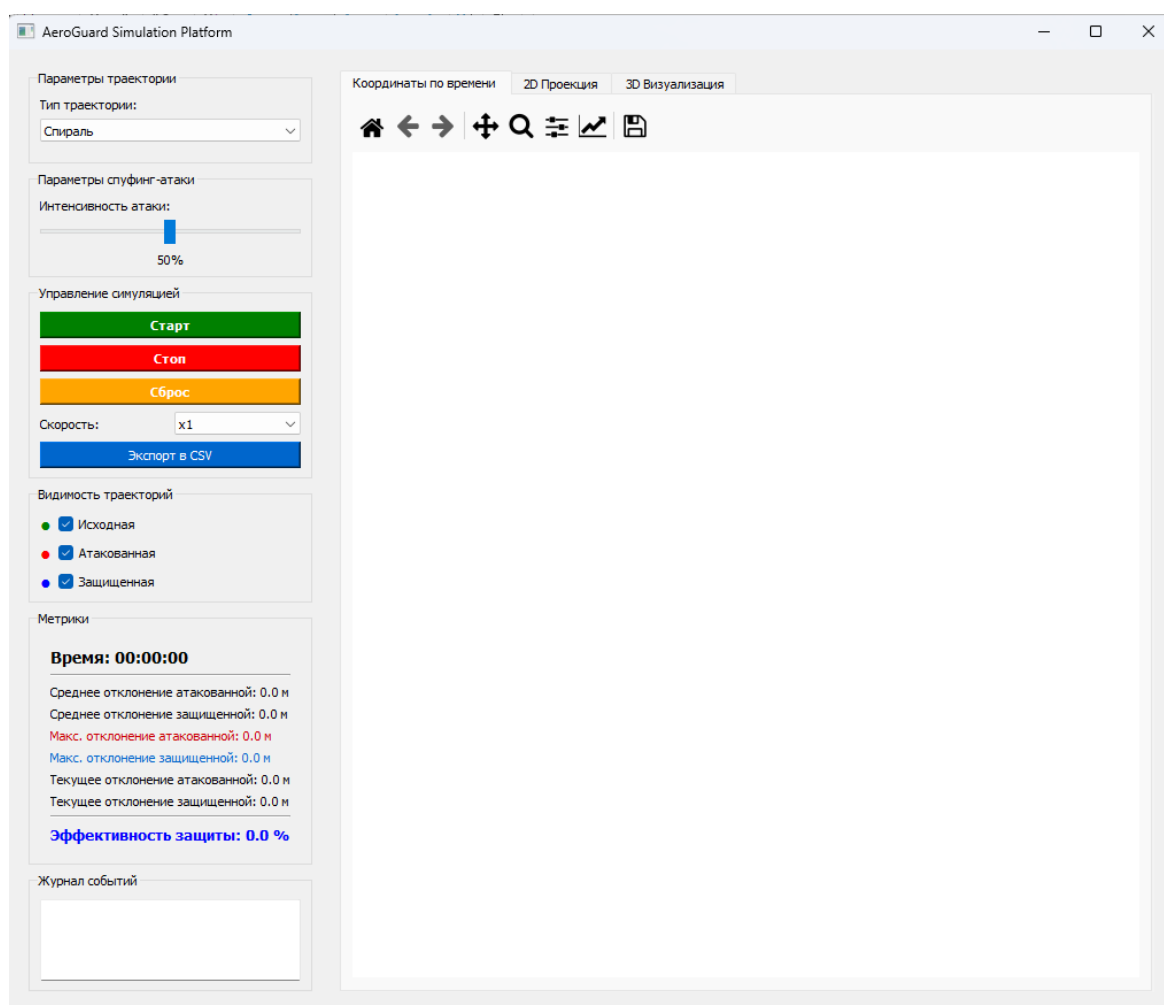


Рис. 1. Общий вид окна программы

Главный алгоритм работы (Рис. 2) приложения *AeroGuard Simulation Platform* представляет собой модульную архитектуру, где каждый компонент отвечает за определенную функциональность системы. При запуске приложения происходит последовательная инициализация всех модулей: сначала загружается графический интерфейс пользователя на базе PyQt5, затем создается движок симуляции, который координирует работу всех остальных модулей.



TrajectoryModule отвечает за генерацию исходных траекторий движения БПЛА согласно выбранному типу, *AttackModule* имитирует спуфинг-атаки, создавая искажения в координатах, а *DefenseModule* применяет алгоритм машинного обучения для обнаружения и смягчения атак. *VisualizationModule* накапливает данные всех трех траекторий для последующего отображения в графическом интерфейсе.

Главный цикл симуляции работает по принципу дискретного моделирования с заданным шагом времени. На каждом шаге происходит получение исходной позиции БПЛА, применение к ней атаки, затем коррекция с помощью алгоритма защиты, обновление графиков. Пользователь может управлять процессом симуляции через кнопки старт, стоп и сброс.

Алгоритм машинного (Рис. 3) в модуле защиты представляет собой адаптивную систему обнаружения аномалий, основанную на статистическом анализе поступающих данных. Механизм работает по принципу сравнения текущей атакованной позиции БПЛА с исходной траекторией, вычисляя евклидово расстояние между ними для количественной оценки отклонения.

При недостатке исторических данных (менее 5 точек) алгоритм возвращает позицию без коррекции, так как не может провести достоверный анализ. Как только накапливается достаточная выборка, начинается процесс статистического обучения. Система определяет присутствие атаки, если отклонение превышает базовый порог в 5 метров, и затем вычисляет коэффициент доверия к данным на основе степени отклонения.

Ключевой особенностью алгоритма является взвешенная коррекция позиции, которая реализует принцип адаптивной фильтрации. Весовой коэффициент коррекции зависит от уровня доверия: при высокой достоверности данных (близко к 0,90) система минимально корректирует позицию к исходной траектории, при низкой достоверности (близко к 0,15) максимально притягивает позицию к исходной траектории, компенсируя искажения атаки. Формула защищенной позиции представляет собой комбинацию исходной и атакованной позиций с динамически меняющимся весом в диапазоне от 75% до 90% в пользу исходной траектории.

Алгоритм генерации траекторий БПЛА обеспечивает создание трех типов маршрутов движения, каждый из которых предназначен для моделирования различных сценариев полета беспилотного аппарата. Прямолинейное движение реализуется простым линейным преобразованием времени в координаты с постоянными компонентами скорости по каждой оси (1). Траектория типа "спираль" создает расширяющуюся пространственную спираль с набором высоты, используя полярные координаты с растущим радиусом и тригонометрическими функциями для определения положения в горизонтальной плоскости, при этом вертикальная компонента растет линейно с течением времени (2). Траектория "зиг-заг" имитирует паттерн сканирования, характерный для разведывательных миссий, где аппарат движется вперед с постоянной скоростью, одновременно совершая резкие боковые маневры на основе знаковой функции синуса, что создает ступенчатые переходы в боковом направлении (3). Каждый тип траектории имеет свои параметрические настройки, такие как начальная позиция, скорости по осям, угловая скорость, рост радиуса и частота маневров, которые можно конфигурировать для различных сценариев моделирования.

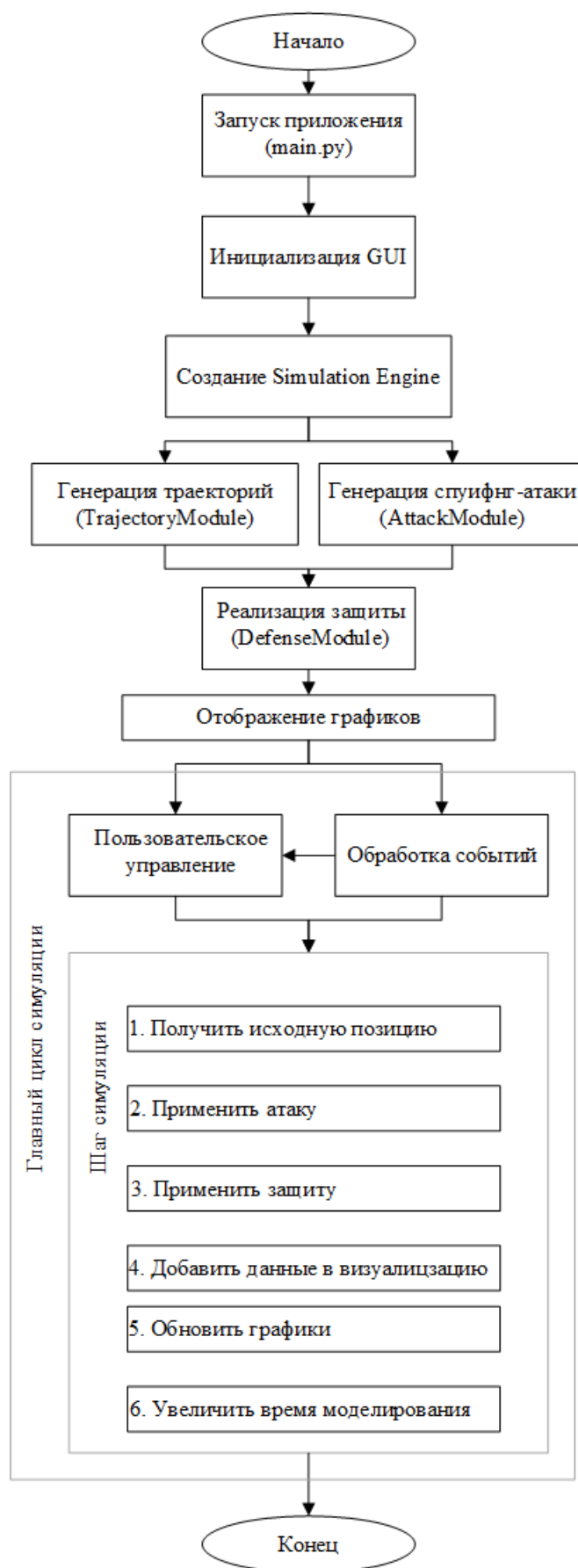


Рис. 2. Главный алгоритм работы приложения



Рис. 3. Алгоритм адаптивного обнаружения аномалий

$$r(t) = vt + p_0, \quad (1)$$

$$r(t) = \begin{bmatrix} (r_0 + kt) \cos \omega t \\ (r_0 + kt) \sin \omega t \\ v_z t \end{bmatrix}, \quad (2)$$



$$r(t) = \begin{bmatrix} v_x t \\ v_y \operatorname{sgn}(\sin \omega t) \\ v_z t \end{bmatrix}, \quad (3)$$

где $r(t) = [x(t) \ y(t) \ z(t)]^T$ – вектор положения БПЛА в момент времени t ;
 $v = [v_x \ v_y \ v_z]^T$ – вектор скорости БПЛА;
 $p_0 = [x_0 \ y_0 \ z_0]^T$ – вектор начального положения БПЛА;
 r_0 – начальный радиус спирали, м;
 k – коэффициент радиального роста, м/с;
 ω – угловая скорость, рад/с;
 $\operatorname{sgn}(x)$ – знаковая функция: $\operatorname{sgn} = \begin{cases} +1, & \text{если } x > 0 \\ -1, & \text{если } x \leq 0 \end{cases}$;
 $t \in [0, \tau]$ – временной отрезок;
 τ – продолжительность траектории (моделирования), с.

Алгоритм спуфинг-атаки реализует комплексный механизм генерации навигационных искажений, имитирующий реальные спутниковые спуфинг-атаки на БПЛА. Для обеспечения непредсказуемости и воспроизводимости одновременно, система использует детерминированную генерацию случайных чисел на основе времени симуляции.

Применение атаки заключается в наложении множественных источников на исходную позицию БПЛА. База – случайное смещение на основе нормального распределения величиной до 8 метров, умноженное на коэффициент интенсивности. Временные рывки создают резкие скачки позиции на расстояние до 12 метров, которые срабатывают через определенные интервалы времени для имитации периодических импульсных помех.

Синусоидальный компонент создает плавное уводящее воздействие с различными частотами и фазовыми сдвигами по каждой оси координат, что имитирует постепенное накопление ошибок. Экспоненциальный увод усиливается со временем, что моделирует эффект накопления систематических ошибок в навигационной системе. Постоянное смещение добавляет статическую ошибку, характерную для компрометированных навигационных данных.

Все вышеперечисленные компоненты объединяются с различными весовыми коэффициентами, создавая сложную пространственную деформацию исходной траектории. Итоговое искажение позиции применяется аддитивно к первоначальной позиции, в результате чего получается атакованная позиция БПЛА, которая воспроизводит воздействие спутникового спуфинга на реальном беспилотном аппарате.

Общая математическая модель спуфинг-атаки и каждой компоненты представлены выражениями (4)-(10).

Общая математическая модель:

$$r_a(t) = r_o(t) + \Delta(t), \quad (4)$$

Суммарное искажение:

$$\Delta(t) = \Delta_1(t) + \Delta_2(t) + 0,6\Delta_3(t) + 0,4\Delta_4(t) + 1,5\Delta_5(t), \quad (5)$$

Базовое смещение:

$$\Delta_1(t) = 8\alpha R, \quad (6)$$

Смещение импульсными скачками:

$$\Delta_2(t) = 12\alpha R \cdot \operatorname{sgn}\left(\frac{2\pi t}{T}\right), \quad (7)$$

Синусоидальное смещение:



$$\Delta_3(t) = 6\alpha \begin{bmatrix} \sin 2,5t \\ \sin 3,7t \\ \sin 2,9t \end{bmatrix}, \quad (8)$$

Нарастающее смещение:

$$\Delta_4(t) = 3\alpha R \cdot (1 + 5 \cdot 10^{-3}t)^\gamma, \quad (9)$$

Постоянное смещение:

$$\Delta_5(t) = 1,5\alpha Q, \quad (10)$$

где $r_a(t) = [x_a \ y_a \ z_a]^T$ – вектор положения БПЛА после проведения спуфинг-атаки;
 $r_o(t) = [x_o \ y_o \ z_o]^T$ – вектор исходного положения БПЛА;
 $\Delta(t) = [\Delta_x \ \Delta_y \ \Delta_z]^T$ – вектор суммарного смещения координат БПЛА;
 α – параметр интенсивности атаки;
 $R = [rand_1 \ rand_2 \ rand_3]^T$ – вектор трех независимых величин с нормальным распределением $N(0, 1)$;
 $sgn\left(\frac{2\pi t}{T}\right)$ – периодическая функция, имитирующая импульсные искажения спутникового сигнала при спуфинге;
 $(1 + 5 \cdot 10^{-3}t)^\gamma$ – множитель усиления искажения во времени t ;
 $Q = [Q_x \ Q_y \ Q_z]^T$ – вектор координат постоянного смещения (устанавливается при инициализации спуфинг-атаки).

Отметим, что в (5)-(10) весовые коэффициенты слагаемых и множителей подобраны экспериментальным путем таким образом, чтобы определять достаточный вклад каждой компоненты в искажение траектории движения БПЛА.

Проведение эксперимента

Для каждой траектории полета БПЛА проведем запуск разработанного приложения. Исходные параметры системы представлены в Таблице 2-5.

Таблица 2 – Исходные параметры траектории БПЛА

Прямолинейное движение
$[x_o \ y_o \ z_o] = [0 \ 0 \ 0]$ м; $[v_x \ v_y \ v_z] = [10 \ 5 \ 2]$ м/с
Движение по спирали
$[x_o \ y_o \ z_o] = [0 \ 0 \ 0]$ м; $r_0 = 10$ м; $k = 0,15$ м/с; $\omega = 0,3$ рад/с; $v_z = 1$ м/с
«Зиг-заг»
$[x_o \ y_o \ z_o] = [0 \ 0 \ 0]$ м; $v_x = 15$ м/с; $v_y = 20$ м/с; $v_z = 1,5$ м/с; $\omega = 0,1$ рад/с

Таблица 3 – Исходные параметры спуфинг-атаки на БПЛА

Интенсивность (для каждой из рассматриваемых траекторий)
75%
Компоненты искажения (начальные)
$\Delta_1 = 1$ м; $\Delta_2 = 12$ м; $\Delta_3 = 6$ м; $\Delta_4 = 3$; $\Delta_5 = 1,5$ м

Таблица 4 – Исходные данные для системы защиты от спуфинг-атаки на БПЛА

Размер истории: 5 точек траектории	Порог обнаружения атаки: 5 м
Коэф. адаптивного порога: 1,5	Весовой коэф. коррекции: 0,75...0,90
Коэф. доверия: max – 0,90 – при отклонении ≤ 2 м; min – 0,15 – при больших отклонениях	

Таблица 5 – Исходные параметры симуляции

Шаг времени симуляции: 0,5 с	Период обновления графиков: 200 мс
------------------------------	------------------------------------



Мах время симуляции: не ограничено	Общее кол-во точек: ограничено вычислительной способностью ПК
Скорость воспроизведения симуляции: x1, x2, x5, x0,5	

Возьмем ограничение по времени симуляции для каждой из траектории в 2 минуты. 2D- и 3D-графики, а также координатная зависимость по времени представлены для каждой траектории полета БПЛА на Рисунке 4-13. *Зеленая* – исходная траектория; *Красная пунктирная* – атакованная; *Зеленая пунктирная* – защищенная.

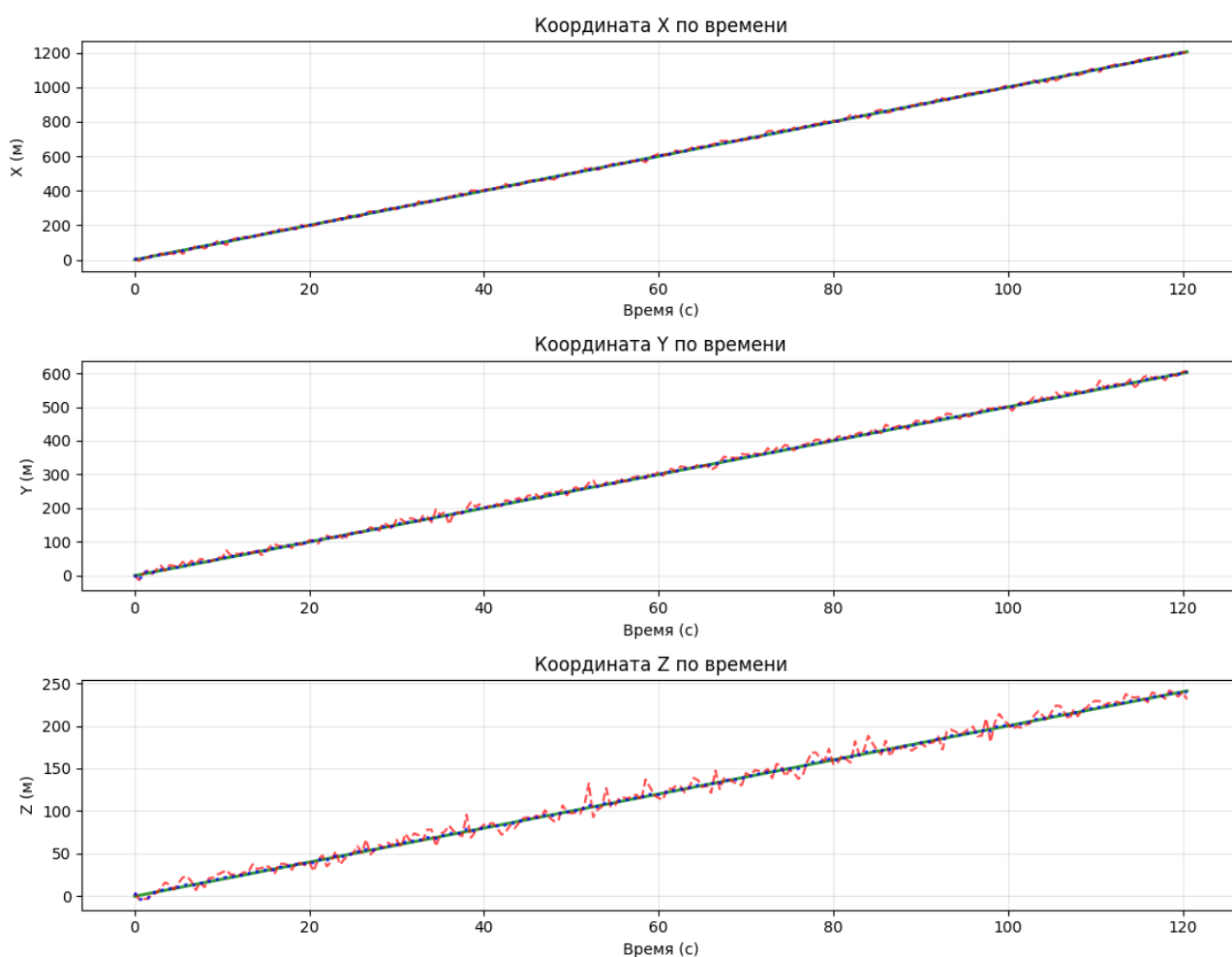


Рис. 4. Координатная зависимость по времени для прямолинейной траектории

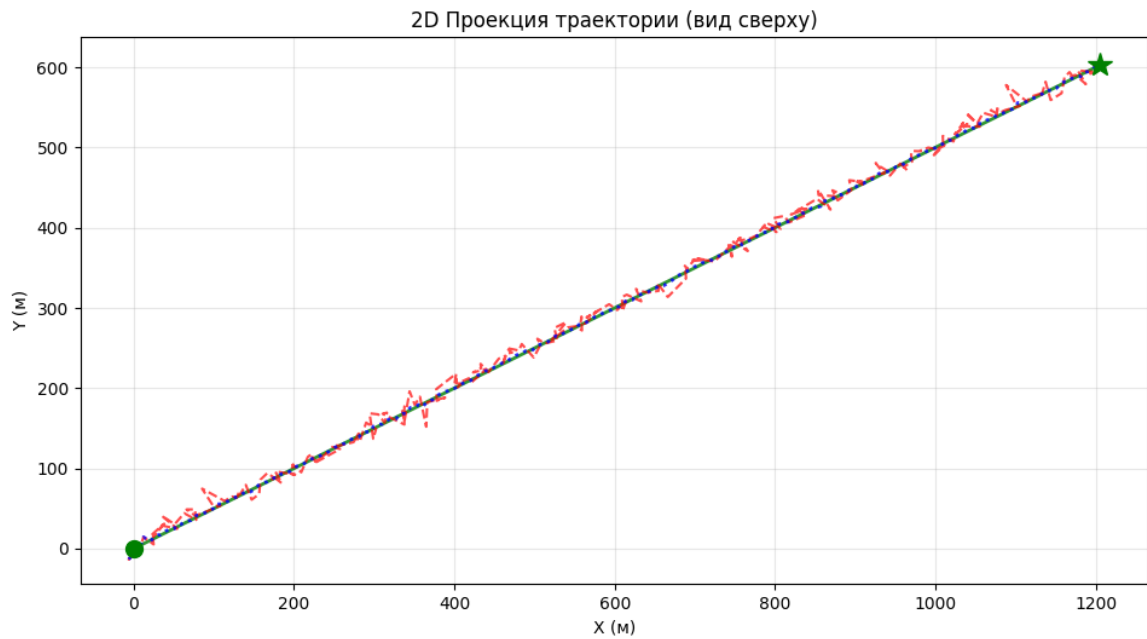


Рис. 5. Вид сверху на прямолинейную траекторию

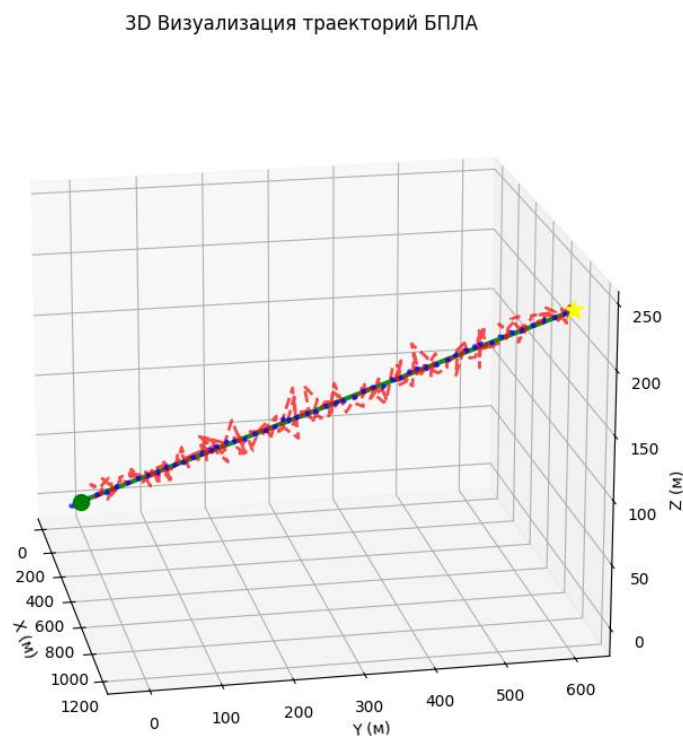


Рис. 6. 3D-график прямолинейного движения

Как видно из Рисунка 4, на большом интервале времени, каждая из траектории стремится друг к другу (что не так, исходя из Рис. 5-6), тогда более детально рассмотрим движение БПЛА в начальный момент времени (Рис. 7). На Рис. 7 отмечены 5 точек в начале движения БПЛА по вертикальной координате – что соответствует размеру истории, указанному в Таблице 4 и на Рис. 3.

Алгоритм защиты срабатывает после прохождения данного рубежа, однако же тех пор защищенная и атакованная траектории будут совпадать.

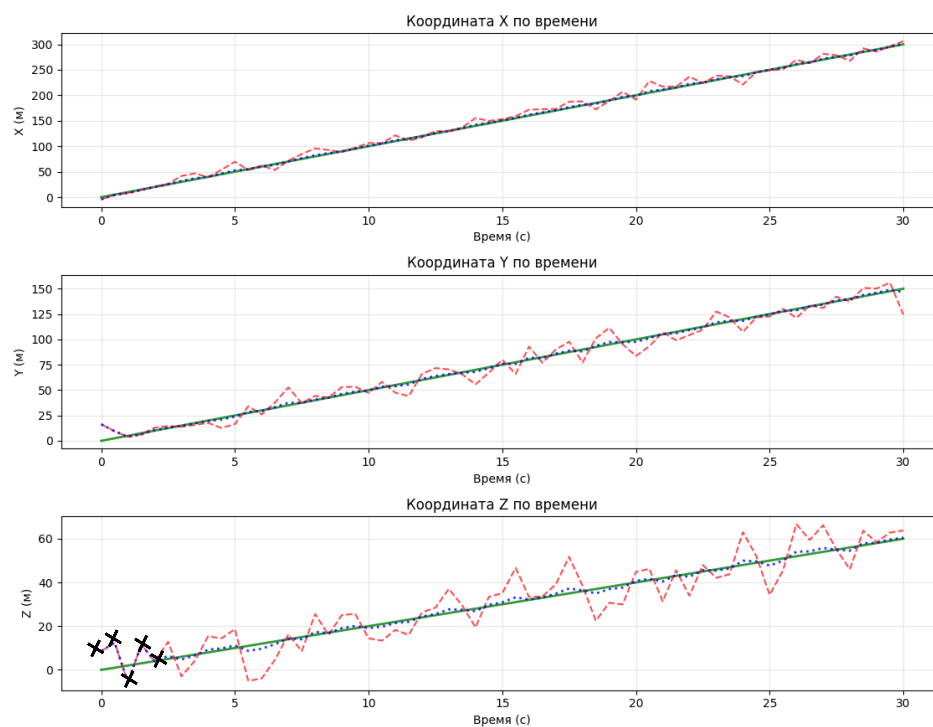


Рис. 7. Координатная зависимость по времени для прямолинейной траектории в начальный момент времени

Аналогично рассмотрим графики движения для оставшихся траекторий.

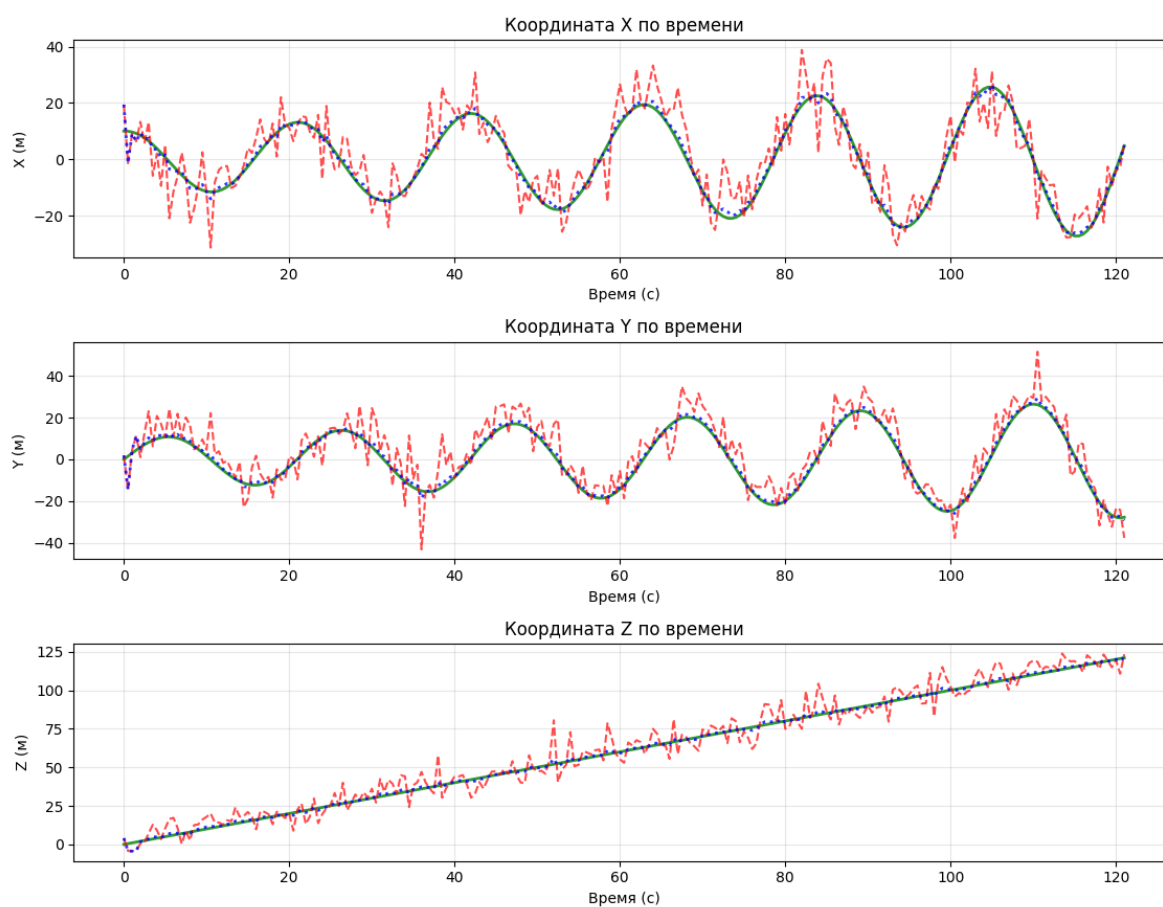


Рис. 8. Координатная зависимость по времени для траектории по спирали

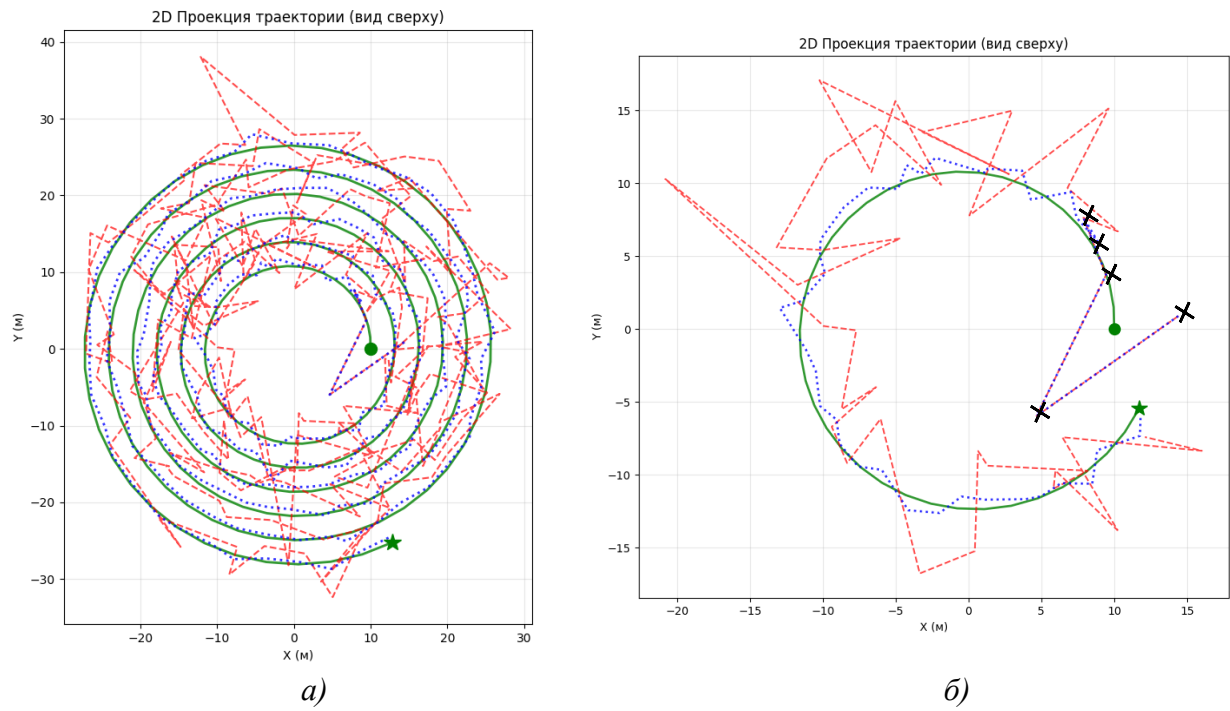


Рис. 9. Вид сверху на траекторию по спирали:
 а) за все время моделирования;
 б) в начальный момент времени

3D Визуализация траекторий БПЛА

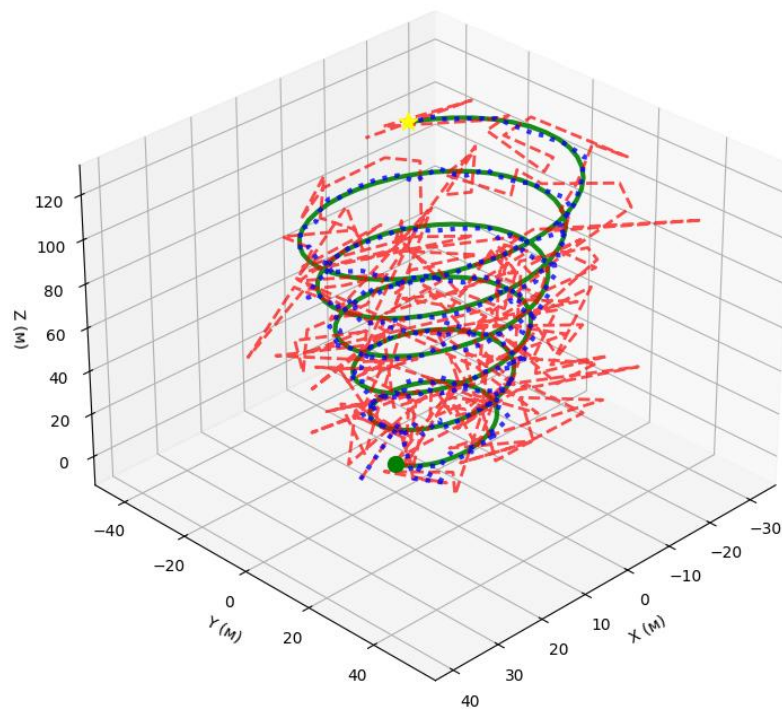


Рис. 10. 3D-график движения по спирали

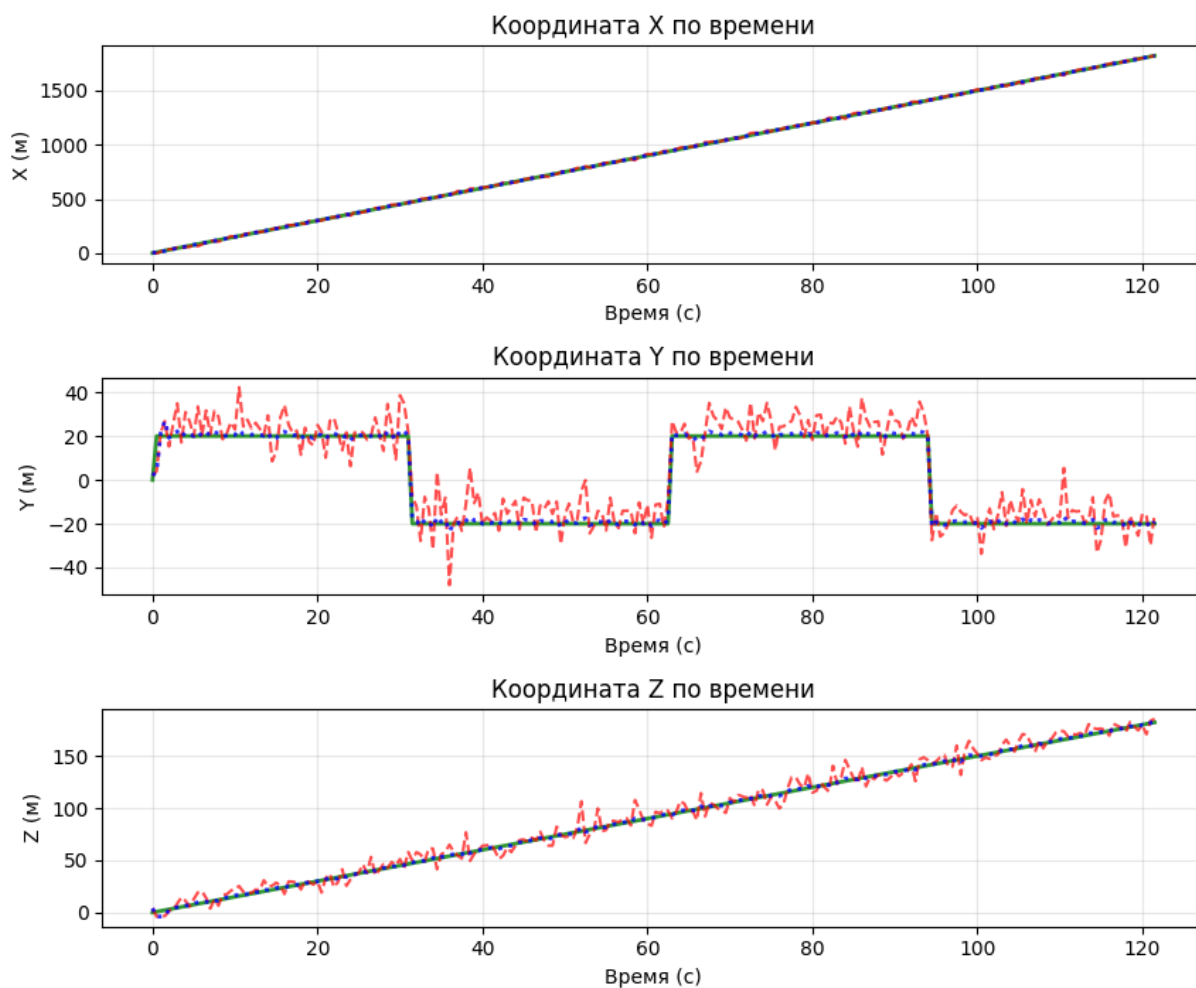


Рис. 11. Координатная зависимость по времени для траектории «зиг-заг»

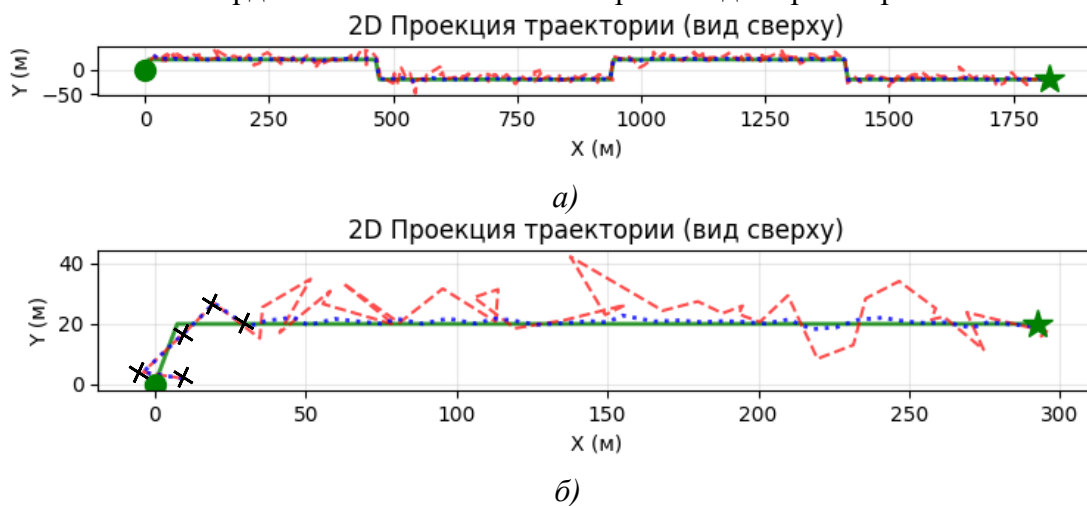


Рис. 12. Вид сверху на траекторию «зиг-заг»:

- а) за все время моделирования;
- б) в начальный момент времени



3D Визуализация траекторий БПЛА

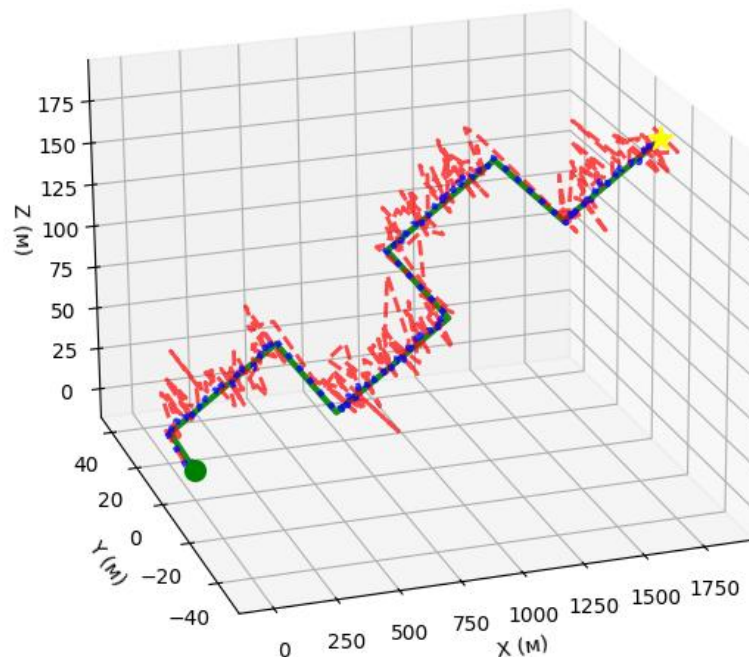


Рис. 13. 3D-график движения по «зиг-загу»

Анализ результатов

На основе экспортированных данных и их анализа для каждой траектории сформулированы тезисы проведенного эксперимента, которые в частности указывают на положительные и отрицательные стороны разработанного ПО, формируют дальнейший вектор проведения работ в рассматриваемой области.

На Рис. 14-16 представлены графики зависимости среднего отклонения атакованной/защищенной траектории и уровня доверия для каждой из 5 точек времени от всего времени моделирования. Как видно по графику для защищенной траектории – линия тренда нисходящая, что говорит о состоятельности алгоритма, разработанного ранее.

Прямолинейное движение/ Движение по спирали/ Движение по траектории «зиг-заг»:

Метрики траекторий:

- Среднее отклонение атакованной: 12,4/12,3/12,3 м;
- Среднее отклонение защищенной: 2,0 /2,1/2,1 м;
- Среднее снижение отклонения: 84/83/83 %;
- Максимальное отклонение атакованной: 31,9/31,8/31,9 м;
- Минимальное отклонение защищенной: 22,1/21,9/21,9 м.

Уровень доверия к данным траектории (на основе данных логирования):

- Средний уровень доверия: 0,42/0,41/0,41;
- Минимальный уровень доверия: 0,17/0,16/0,16 – пиковая атака;
- Максимальный уровень доверия: 0,90 – спокойный период функционирования.

Временные рамки моделирования для каждого уровня доверия:

- Общее время моделирования со средним уровнем доверия: 87,0/88,5/87,0 с;
- Общее время моделирования с низким уровнем доверия: 24,5/26,0/25,0 с;
- Общее время моделирования с высоким уровнем доверия: 7 с.

Метрики аномалий:

- Общее количество аномалий: 220/226/221;



- Среднее время между аномалиями: 0,54с;
- Период максимальной активности: с 12 до 72 с моделирования;
- Эффективность коррекции после обнаружения: 84/82/82 %.

Общая оценка:

Система защиты функционирует, показывает удовлетворительные результаты; испытывает трудности в определенных ситуациях, особенно при высокой частоте возникновения аномалий.

Средний уровень доверия ниже оптимального (0,70). Минимальный уровень доверия (0,16) очень низок – система находится на грани неспособности адекватно оценивать ситуацию при пиковых атаках.

Эффективность коррекции находится на отличном уровне – система нейтрализует подавляющее большинство искажений, вызванных атаками.

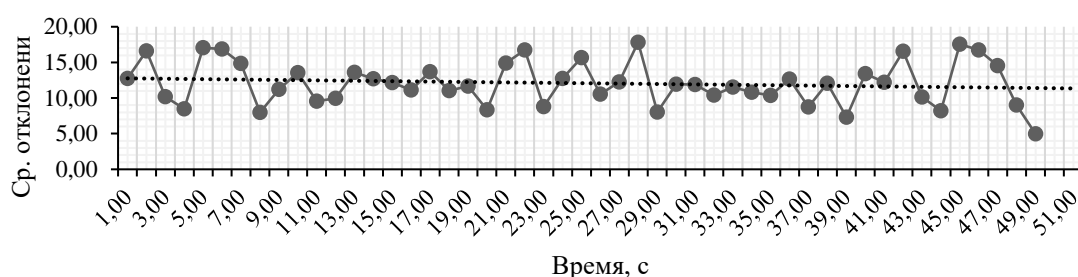


Рис. 14. График среднего отклонения атакующей траектории для набора из каждых 5 точек

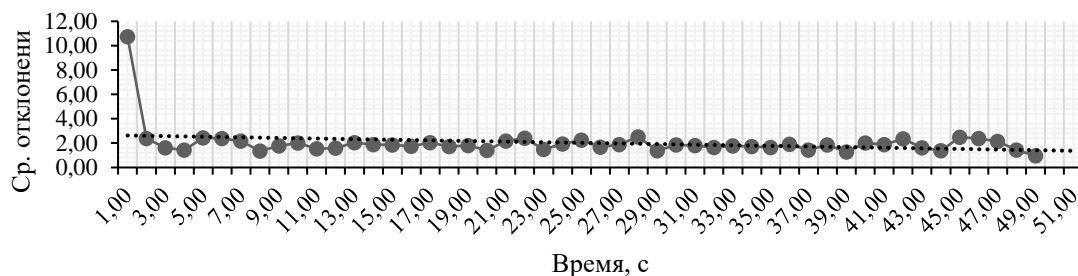


Рис. 15. График среднего отклонения защищенной траектории для набора из каждых 5 точек

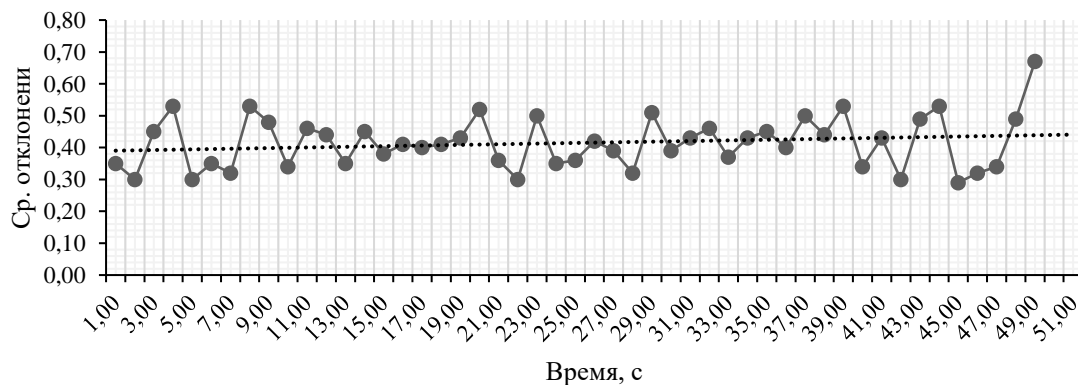


Рис. 16. График среднего уровня доверия для набора из каждых 5 точек



Заключение

Разработанная математическая модель спуфинг-атаки на навигационную систему БПЛА, учитывающая пять компонент искажения, вместе с предложенным адаптивным алгоритмом защиты на основе статистического обнаружения аномалий с динамической коррекцией позиции в зависимости от уровня доверия к данным показала свою состоятельность.

Проведенные численные эксперименты для трех типов траекторий движения БПЛА показали высокую эффективность разработанного алгоритма защиты со средним снижением отклонения траектории на 82-84% при интенсивности атаки 75%. Максимальное отклонение атакованной траектории достигало 31,9 метров, тогда как минимальное отклонение защищенной траектории составило 21,9 метров.

В ходе анализа метрик системы выявлено, что средний уровень доверия к данным составил 0,41-0,42, что указывает на необходимость дальнейшей оптимизации. При пиковых атаках уровень доверия снижался до критических значений 0,16-0,17, однако эффективность коррекции сохранялась на уровне 82-84%.

Таким образом сформулированы следующие рекомендации по совершенствованию системы: 1) оптимизировать алгоритм расчета доверия для повышения среднего уровня с текущих до оптимальных; 2) усилить защиту при пиковых атаках путем введения дополнительных механизмов детектирования критических ситуаций при снижении доверия.

СПИСОК ЛИТЕРАТУРЫ

1. *Муслимов Т. З.* Применение нечеткого адаптивного управления с эталонной моделью в задачах группового полета БПЛА // Беспилотные транспортные средства с элементами искусственного интеллекта: материалы VI Всероссийского научно-практического семинара. – М., 2021. – С. 135-137.
2. *Путов В. В.* Адаптивное управление продольным движением беспилотного летательного аппарата / В. А. Путов // Известия СПбГЭТУ ЛЭТИ. – 2017. – № 4. – С. 35-43.
3. *Веселов Г. Е., Лихтин С. С.* Проблема управления манипуляторами на БПЛА мультироторного типа // Системный синтез и прикладная синергетика: Сб. научных трудов XII Всероссийской научной конференции. – Ростов-на-Дону, 2024. – С. 336-341.
4. *Kaufmann E.* Deep Drone Racing: From Simulation to Reality with Domain Randomization / E. Kaufmann // IEEE Transactions on Robotics (T-RO). – 2020. – 16 p.
5. *Loquercio A.* Learning High-Speed Flight in the Wild / A. Loquercio // Science Robotics. – 2021. – 23 p.
6. *Nguyen V.-T.* Robust adaptive nonlinear PID controller using radial basis function neural network for ballbots with external force / Nguyen Van-Truong // Engineering Science and Technology, an International Journal. – 2025. – 14 p.
7. *Romera E.* "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation / E. Romera // IEEE Transactions on Intelligent Transportation Systems. – 2018. – P. 263-272.
8. *Rosinol A.* Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping / A. Rosinol // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). – 2020. – 8 p.
9. *Анисимов Д. В., Кнжяский А. Ю.* Обзор методов контроля достоверности данных в автоматизированных системах управления движением летательных аппаратов // Аэрокосмическое приборостроение и эксплуатационные технологии: Материалы Шестой Международной научной конференции. – Санкт-Петербург, 2025 года. – С. 5-10.
10. *Кугаевских А. В.* Классические методы машинного обучения / А. В. Кугаевских, Д. И. Муромцев, О. В. Кирсанова – СПб: Университет ИТМО, 2022. – 53 с.



11. *Басан Е. С.* Робототехника, автоматизация и системы управления / *Е. С. Басан* / Информатика и автоматизация. – 2021. – Том 20 № 6. – С. 1368-1394.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Анисимов Дмитрий Валерьевич

Студент-магистр

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Россия, 190000, Санкт-Петербург, ул. Большая Морская, д.67, лит. А

E-mail: anisimovdv03@gmail.com

INFORMATION ABOUT THE AUTHOR

Anisimov Dmitrii Valerievich

Graduate student

Saint-Petersburg State University of Aerospace Instrumentation

67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia

E-mail: anisimovdv03@gmail.com

Дата поступления: 19.11.2025

Дата принятия: 20.11.2025