



DOI: 10.22363/2312-8143-2024-25-4-413-426

УДК 623.746.-519

EDN: BSHWDV

Научная статья / Research article

## Автоматизированная посадка беспилотного летательного аппарата на подвижную платформу с использованием нейронных сетей

В.А. Суслов<sup>ID✉</sup>, С.В. Гагарский

Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова, Санкт-Петербург, Россия  
✉ vlarsu@mail.ru

### История статьи

Поступила в редакцию: 10 июня 2024 г.

Доработана: 16 августа 2024 г.

Принята к публикации: 10 сентября 2024 г.

### Заявление о конфликте интересов

Авторы заявляют об отсутствии конфликта интересов.

### Вклад авторов

Нераздельное соавторство.

**Аннотация.** Разработка беспилотных летательных аппаратов является одним из перспективных направлений для гражданской авиации, которое имеет широкий спектр применения. Нейронные сети могут быть обучены для принятия решений в реальном времени, адаптируясь к изменяющимся условиям на поле боя и обеспечивая оптимальное выполнение поставленных задач. Среди многих задач навигации и управления БПЛА сохраняет актуальность проблема автоматической посадки беспилотного летательного аппарата на подвижную посадочную площадку (корабли, транспортные средства, специализированные площадки). Особую актуальность носит автоматизированная посадка беспилотного летательного аппарата на подвижный носитель. В связи с этим авторами исследуется система автоматической посадки беспилотного летательного аппарата (БПЛА) на подвижную платформу с использованием нейросетевых технологий. Метод исследования основан на применении искусственных нейронных сетей для разработки адаптивной системы управления, способной принимать решения в реальном времени при выполнении посадочных маневров. В результате разработан алгоритм управления, обеспечивающий точную посадку БПЛА на движущиеся платформы различного типа (корабли, транспортные средства, специализированные площадки), что позволяет существенно расширить радиус действия беспилотных аппаратов и повысить эффективность их применения в различных условиях эксплуатации.

**Ключевые слова:** приемопередающее устройство, взлетно-посадочная установка, лидар, взлетно-посадочной платформы, нейросеть, подвижный носитель

### Для цитирования

Сулов В.А., Гагарский С.В. Автоматизированная посадка беспилотного летательного аппарата на подвижную платформу с использованием нейронных сетей // Вестник Российского университета дружбы народов. Серия: Инженерные исследования. 2024. Т. 25. № 4. С. 413–426. <http://doi.org/10.22363/2312-8143-2024-25-4-413-426>



## Automated Landing of an Unmanned Aerial Vehicle on a Mobile Platform Using Neural Networks

Vladislav A. Suslov  , Sergey V. Gagarsky

Baltic State Technical University “Voenmeh” named after D.F. Ustinov, *Saint Petersburg, Russia*

 vlarsu@mail.ru

### Article history

Received: June 10, 2024

Revised: August 16, 2024

Accepted: September 10, 2024

### Conflicts of interest

The authors declare that there is no conflict of interest.

### Authors' contribution

Undivided co-authorship.

**Abstract.** The development of unmanned aerial vehicles is one of the promising directions for civil aviation with a wide range of applications. Neural networks can be trained for real-time decision making, adapting to changing battlefield conditions and ensuring optimal task execution. Among many UAV navigation and control tasks, the challenge of automatic landing of an unmanned aerial vehicle on a mobile landing platform (ships, vehicles, specialized platforms) remains relevant. Automated landing of an unmanned aerial vehicle on a mobile carrier is particularly significant. The article examines the system of automatic landing of an unmanned aerial vehicle (UAV) on a mobile platform using neural network technologies. The research method is based on the application of artificial neural networks for developing an adaptive control system capable of real-time decision-making during landing maneuvers. As a result of the research, a control algorithm was developed that ensures precise landing of UAVs on various types of moving platforms (ships, vehicles, specialized platforms), which significantly expands the operational range of unmanned vehicles and increases their efficiency in various operating conditions.

**Keywords:** receiving and transmitting device, runway, lidar, runway, neural network, mobile carrier

### For citation

Suslov VA, Gagarsky SV. Automated landing of an unmanned aerial vehicle on a mobile platform using neural networks. *RUDN Journal of Engineering Research*. 2024;25(4):413–426. (In Russ.) <http://doi.org/10.22363/2312-8143-2024-25-4-413-426>

## Введение

Применение нейронных сетей для автоматизированной посадки беспилотного летательного аппарата (БПЛА) на подвижную платформу является актуальной задачей. Нейронные сети, обученные на данных о движении платформы и внешних условиях, позволяют адаптироваться к изменениям в реальном времени, обеспечивая надежность операций даже в нестабильных условиях. В таких системах управления нейронные сети обрабатывают данные с сенсоров и камер, включая показатели ветра, перемещения платформы и пространственные параметры, что способствует высокой точности посадки.

Разработка систем автоматической посадки БПЛА на подвижные платформы является ак-

туальным направлением исследований. Анализ научной литературы показывает значительный прогресс в этой ОБПЛАсти. Ю.П. Токарев [1] исследовал методы управления БПЛА в общем воздушном пространстве. А.В. Платунова и соавторы [2] рассмотрели особенности формирования адаптивных законов управления. М.Д. Никанорова, Е.В. Заболотская [3] представили численные расчеты аэродинамических характеристик БПЛА.

Применение нейронных сетей для управления БПЛА рассматривается в [4–5]. Особый интерес представляют исследования В.В. Щербинина [6] по алгоритмам корреляционно-экстремальной навигации и работы А.Г. Смольского [7] по нейросетевой обработке геопространственной информации.

Система управления БПЛА состоит из приемопередающих устройств и электронных блоков управления, обеспечивающих точное позиционирование БПЛА на подвижной платформе. Эти системы поддерживают обмен информацией с использованием оптических сенсоров, что позволяет динамично корректировать траекторию для успешной посадки даже при изменяющихся условиях окружающей среды.

Задача точной посадки БПЛА на подвижную платформу требует дальнейшего исследования, особенно в части применения нейронных сетей для обработки данных с оптических сенсоров в реальном времени [8; 9].

Цель исследования — разработка алгоритмов для автоматизированной посадки БПЛА на подвижные носители с использованием нейронных сетей. Данные технологии могут применяться для выполнения задач в удаленных и опасных зонах, снижая зависимость от человеческого вмешательства и повышая безопасность операций.

## 1. Описание системы автоматической посадки

### 1.1. Общая структура системы

Предположим, что у нас есть один БПЛА с симметрично расположенными винтами. Также имеется круговая плоская платформа, которая движется в горизонтальной плоскости в пределах обнаружения квадрокоптером. Радиус платформы незначительно превышает расстояние от центра масс до края любого из винтов. Целью является достижение состояния, когда квадрокоптер садится на движущуюся платформу. Решение задачи посадки БПЛА на подвижный носитель включает в себя создание программно-аппаратного комплекса, который содержит подсистему механизации данного процесса и алгоритм управления, реализованный в виде программной части, обеспечивающей работу подсистемы механизации (рис. 1).

Программно-аппаратный комплекс разработан для координации двух управляемых объектов: стола взлетно-посадочной установки

(ВПУ) и беспилотного летательного аппарата. В процессе посадки на движущуюся платформу стол ВПУ может перемещаться независимо от корпуса носителя. Управляемое движение стола обеспечивается электромеханическими приводами, которые стабилизируют платформу по всем шести степеням свободы. Это позволяет компенсировать движение носителя, обеспечивая точную позицию для посадки [10; 11].



Рис. 1. Взаимное расположение БПЛА и ВПУ

Источник: выполнено В.А. Сусловым, С.В. Гагарским

Figure 1. Relative positioning of the Unmanned Aerial Vehicle and the Takeoff and Landing Gear

Source: done by V.A. Suslov, S.V. Gagarsky

Стол взлетно-посадочной установки связан с носителем через подсистему механизации и может двигаться в трехмерной системе координат, следуя кинематической траектории носителя. Движение платформы, как правило, характеризуется случайной составляющей, обусловленной типом носителя: движущееся судно подвержено качке, а автотранспортное средство — неровностям дороги. Подсистема механизации не только сглаживает линейные и угловые колебания, но также синхронизирует движение стола с траекторией БПЛА.

В процессе посадки БПЛА подвергается воздействию внешних сил, таких как порывы ветра и турбулентные потоки, создаваемые движением вокруг стола взлетно-посадочной установки. Эти факторы добавляют случайные коле-

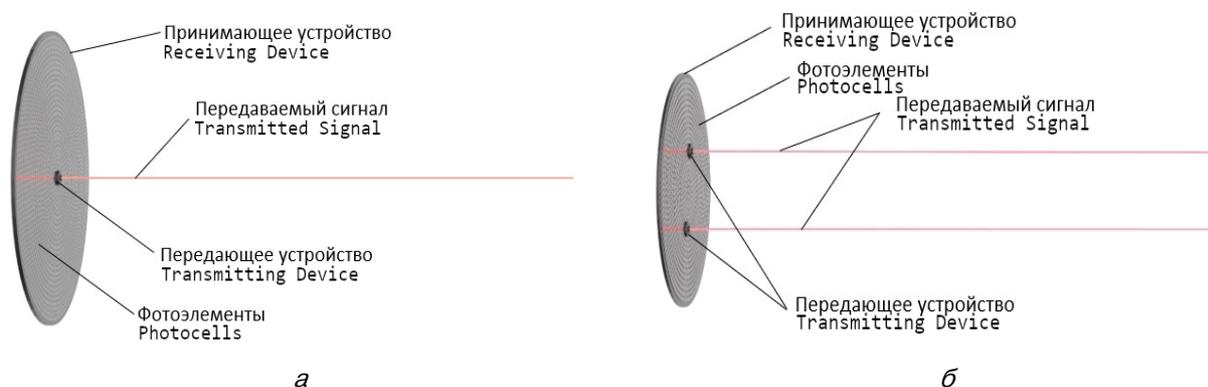
бания в траекторию БПЛА во время захода на посадку, что требует постоянной корректировки его положения и ориентации.

### 1.2. Приемопередающие устройства

Для точного определения взаимного положения БПЛА и ВПУ на столе и на днище БПЛА устанавливается приемопередающее устройство (ППУ), включающее лидар и матрицу фоторезисторов. Лидар располагается в центре, а фо-

торезисторы образуют концентрические круги вокруг него, фиксируя лазерные лучи, направленные с противоположного устройства [12].

Ось лидара ориентирована перпендикулярно к матрице фоторезисторов. Когда БПЛА входит в зону посадки, лидары и фоторезисторы обоих устройств активируются, регистрируя рассогласование осей лидара, что позволяет вычислить относительное пространственное положение объектов (см. рис. 2).



**Рис. 2.** Состав приемопередающего устройства:  
*а* — взлетно-посадочная установка; *б* — беспилотный летательный аппарат  
 Источник: выполнено В.А. Сусловым, С.В. Гагарским

**Figure 2.** Composition of the transceiver unit:  
*a* — Takeoff and Landing Gear; *b* — Unmanned Aerial Vehicle  
 Source: done by V.A. Suslov, S.V. Gagarsky

Излучатель каждого ППУ передает сигнал на приемник соседнего ППУ. Передаваемый сигнал содержит координаты луча на ЭБУ с точностью дискретизации фоторезисторов, расположенных на панели (рис. 2). Оба сенсорных модуля регистрируют и передают координаты принятого луча на ЭБУ, где происходит расчет относительного положения двух объектов — ВПУ и БПЛА. Дополнительно в центре диска сенсорного модуля ВПУ установлен дальномер, который определяет длину луча от поверхности ВПУ до точки следа на поверхности БПЛА [13; 14].

Для расчетов необходимо учитывать следующие параметры, которые регистрируются и передаются в ЭБУ:

1) две координаты луча от ВПУ на поверхности модуля БПЛА;

2) четыре координаты лучей от БПЛА на поверхности ВПУ (по две координаты на каждый луч);

3) три длины лучей ВПУ и БПЛА.

На основе этих девяти параметров настроенная нейронная сеть определяет шесть координат взаимного положения одного объекта относительно другого.

### 1.3. Схема системы управления

Специальная сенсорная система (см. рис. 3.) включает несколько устройств.

1. Приемопередающее устройство ВПУ.

2. Приемопередающее устройство БПЛА.

Приемопередающие устройства ВПУ и БПЛА являются ключевыми компонентами системы управления, предназначенными для об-

мена данными в процессе посадки БПЛА на ВПУ. Эти устройства передают и принимают сигналы, содержащие информацию о координатах и ориентации ВПУ относительно БПЛА, что необходимо для точной и безопасной посадки. Приемопередающее устройство ВПУ непрерывно обновляет данные о взаимном положении ВПУ и БПЛА в реальном времени, позволяя системе управления точно рассчитывать коррекции для успешной посадки. Обмен информацией между устройствами осуществляется через оптические сенсорные модули обоих устройств.

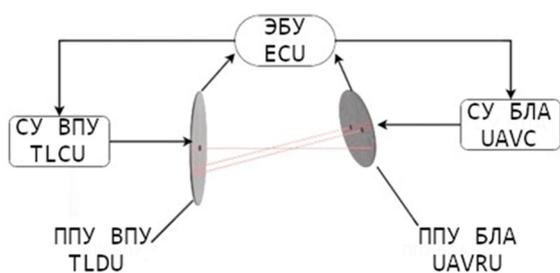


Рис. 3. Состав сенсорной системы

Источник: выполнено В.А. Сусловым, С.В. Гагарским

Figure 3. Composition of the sensor system  
Source: done by V.A. Suslov, S.V. Gagarsky

3. *Электронный блок управления (ЭБУ) взаимного положения.* ЭБУ взаимного положения играет центральную роль в системе управления, координируя действия между ВПУ и БПЛА. Он обрабатывает данные, поступающие от сенсорных модулей и приемопередающих устройств, и управляет движениями обоих объектов для успешного выполнения стыковки. ЭБУ анализирует и интерпретирует данные о шести координатах взаимного положения (три линейных смещения и три угловых поворота на углы Эйлера), которые определяются нейронной сетью на основе информации с оптических сенсоров. Он также рассчитывает необходимые коррекции для точной посадки БПЛА на ВПУ, передает команды системам управления, поддерживая стабильность и безопасность операции.

4. *Система управления (СУ) ВПУ.*

5. *Система управления (СУ) БПЛА.*

Системы управления БПЛА и ВПУ отвечают за координацию и управление процессом

посадки БПЛА на ВПУ. СУ БПЛА управляет действиями летательного аппарата, используя данные от сенсоров, включая приемопередающее устройство, для выполнения точных маневров посадки. СУ ВПУ управляет движениями платформы, поддерживая стабильность и координацию для успешного завершения операции. Обе системы управления тесно взаимодействуют, обмениваясь информацией и координируя действия для безопасной и точной посадки БПЛА на ВПУ.

## 2. Алгоритм расчета геометрии взаимного положения взлетно-посадочной платформы и беспилотного летательного аппарата

Алгоритм расчета взаимного положения ВПУ и БПЛА направлен на точное позиционирование в пространстве для успешной стыковки во время посадки. В процессе используется взаимодействие координатных систем, одна из которых жестко связана с ВПУ, а другая — с БПЛА. Алгоритм реализует расчет геометрии взаимного положения ВПУ и БПЛА с использованием данных от сенсорных модулей [8].

### 2.1. Структура алгоритма

1. *Система сенсоров.* Используются два ППУ — одно установлено на ВПУ, другое на БПЛА. Они обмениваются информацией с помощью оптических сенсоров, передавая данные о текущем положении и принимая данные с противоположного устройства.

2. *Нейронная сеть для обработки данных.* Поступающие данные обрабатываются заранее обученной нейронной сетью, которая преобразует их в шесть координат, включающих три линейных смещения центра тяжести и три угловых поворота (углы Эйлера). Эти координаты поступают в систему управления для корректировки позиции БПЛА.

3. *Компоненты ППУ.* Каждое ППУ оснащено:

– *передающим устройством* (оптический излучатель) — на БПЛА устанавливаются два

устройства, разнесенных на заданное расстояние, а на ВПУ — одно устройство, расположенное в центре;

– *принимаящим устройством* (сенсорная панель с фотодиодами/фоторезисторами) — для регистрации светового сигнала от противоположного ППУ.

4. *Обратная связь и корректировка*. Алгоритм поддерживает обратную связь, что позволяет корректировать взаимное положение для точной стыковки.

## 2.2. Описание алгоритма

Суть данного алгоритма — в создании обратной связи между двумя управляемыми объектами, что позволяет им динамично реагировать на изменения относительного положения и ориентации в пространстве для точной стыковки. Успех данной процедуры зависит от точности сенсорных систем и эффективности алгоритма управления, который должен быстро и точно рассчитывать необходимые корректирующие маневры для достижения совмещения систем координат для безопасной стыковки БПЛА с ВПУ.

*Сбор данных*. Сенсорные датчики на ВПУ и БПЛА регистрируют положение лазерного луча, который светит на них с противоположного объекта (например, излучатели на БПЛА направляют лучи на датчики ВПУ, и наоборот).

1. *Анализ регистрации*. По показаниям сенсорных датчиков определяется, на каких координатах ( $X$ ,  $Y$ ) каждый из лучей попадает на соответствующую сенсорную панель. Это дает информацию об отклонении относительно идеального положения центров лидаров.

2. *Алгоритм расчета геометрии*. На основе этих данных алгоритм рассчитывает нужное положение и ориентацию ВПУ и БПЛА, чтобы совместить их системы координат так, чтобы оси лидаров обоих объектов совпадали.

3. *Корректирующие маневры*. ВПУ и БПЛА независимо друг от друга выполняют маневры для корректировки своего положения таким образом, чтобы координаты положения следа их

луча на смежном теле совпадали с идеальными координатами на собственном ППУ

4. *Обратная связь и повторение*. После каждого действия данные снова собираются и анализируются для уточнения последующих корректирующих действий. Этот процесс повторяется до тех пор, пока системы координат не совпадут настолько, чтобы обеспечить успешную посадку БПЛА на подвижную платформу в условиях боевых действий.

## 3. Нейросетевая система управления

Для обучения нейронной сети, использующейся для ориентации положения беспилотного летательного аппарата (БПЛА) в пространстве, была разработана программа на языке Python с использованием библиотеки Keras. В этой программе используются данные, полученные из имитационной модели, созданной в SolidWorks, включая углы поворота, смещения, координаты излучателей на БПЛА и контрольные замеры.

С помощью этой программы производятся расчеты следов первого и второго излучателей БПЛА на поверхности верхней платформы взлетно-посадочной установки (ВПУ), а также строятся матрицы координат следа излучателя БПЛА на ВПУ.

Использование нейронных сетей для определения ориентации БПЛА в пространстве является более эффективным и целесообразным подходом по сравнению с вычислениями на основе алгоритмов. Это позволяет решать задачи автоматизации полетов беспилотных аппаратов, так как требуется обработка большого объема информации, выполнение множества вычислений, обеспечение автоматической посадки БПЛА и высокой точности посадки.

Для успешного обучения нейронной сети необходимо сгенерировать большое количество вариантов взаимного положения ВПУ и БПЛА в пределах ограничений размеров сенсорной зоны ВПУ и БПЛА по всем девяти координатам пространственного положения. Это обеспечит нейронной сети достаточное разнообразие данных для эффективного обучения и точного определения положения БПЛА в пространстве.

Перед началом работы с нейронными сетями необходимо загрузить данные для обучения модели. В контексте работы над алгоритмом расчета геометрии взаимного положения взлетно-посадочной платформы и беспилотного летательного аппарата данные обычно импортируются из таблицы.

Шаги подготовки данных включают несколько этапов.

1. *Импорт данных из файла*: данные из файла загружаются в переменные X и Y для последующего использования в обучении модели.

2. *Нормализация данных*: каждая характеристика масштабируется и преобразуется таким образом, чтобы находиться в заданном диапазоне обучающей выборки, обычно между нулем и единицей.

Предварительная обработка данных может включать следующие шаги:

1) *заполнение пропущенных значений*: обеспечивает полноту исходных данных;

2) *нормализация данных*: приведение данных к общему масштабу для снижения влияния различных единиц измерения;

3) *удаление выбросов*: исключение некорректных или необычных значений, которые могут исказить результаты модели;

4) *преобразование данных*: приведение данных к необходимому формату для обучения модели, например, преобразование категориальных переменных в числовой формат или использование кодирования с одним активным состоянием для обработки категориальных признаков.

Масштабирование данных играет важную роль в обучении модели машинного обучения, поскольку обеспечивает равномерный вклад всех признаков в процесс обучения, что в конечном итоге может улучшить производительность модели.

Для оценки производительности модели машинного обучения данные обычно разделяют на обучающую и тестовую выборки. Обучающая выборка используется для обучения модели, в то время как тестовая выборка используется для проверки производительности модели на новых, ранее не виденных данных.

После разделения данных на обучающую и тестовую выборки также создается валидационная выборка. Этот подход позволяет оценить производительность модели на независимых от обучающей выборки данных и помогает выявить ее способность обобщать на новые данные.

### **3.1. Пример разбивки набора данных на тренировочные, тестовые и валидационные (алгоритм расчета геометрии взаимного положения взлетно-посадочной платформы и беспилотного летательного аппарата)**

Для создания устойчивой модели нейронной сети данные разделяются на три выборки: тренировочную (70 %), тестовую (15 %) и валидационную (15 %). В Python для этой задачи используется функция `train_test_split` из библиотеки `scikit-learn`:

```
X_train, X_val, X_test, Y_train, Y_val, Y_test = train_test_split(
    X_scale, Y_scale, test_size=0.3, random_state=4)
X_val, X_test, Y_val, Y_test = train_test_split(
    X_val_and_test, Y_val_and_test, test_size=0.5, random_state=4)
```

После разделения данных выполняется нормализация и проверка на репрезентативность. Это необходимо для обеспечения стабильности и качества модели при обработке новых данных.

### **3.2. Пример построения модели нейронной сети (алгоритм расчета геометрии взаимного положения взлетно-посадочной платформы и беспилотного летательного аппарата)**

При работе над алгоритмом используются данные библиотеки класса `Sequential`, который группирует линейный набор слоев в переменную `model`. В данном примере используется следующая архитектура нейронной сети (рис. 4).

1. *Входной слой*: Плотный (Dense) слой с 9 нейронами, соответствующими входным параметрам.

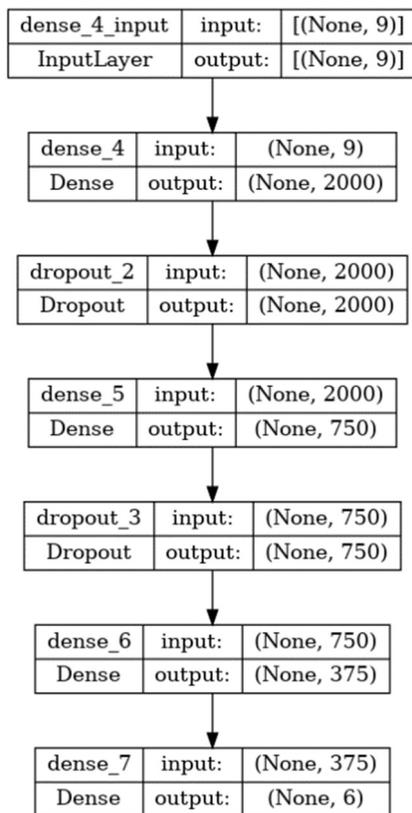
2. *Скрытые слои*:

– первый скрытый слой с 2000 нейронами и функцией активации ReLU;

– второй и третий скрытые слои по 1500 нейронов каждый с ReLU;

– два слоя Dropout для предотвращения переобучения.

3. *Выходной слой*: с 6 нейронами и функцией активации Hard Sigmoid для предсказания 6 параметров.



**Рис. 4.** Схема нейронной сети

Источник: выполнено В.А. Суловым, С.В. Гагарским

**Figure 4.** Neural network diagram

Source: compiled by V.A. Suslov, S.V. Gagarsky

Эта архитектура нейронной сети может быть эффективной для задачи расчета геометрии взаимного положения взлетно-посадочной платформы и беспилотного летательного аппарата.

### 3.3. Компиляция модели (сборка и настройка модели для обучения)

После определения архитектуры модели требуется компилировать ее с выбранными параметрами. Компиляция модели включает выбор оптимизатора, функции потерь и метрик для оценки производительности.

```
model.compile(optimizer='adam',
              loss='msle', metrics=['accuracy'])
```

На этом этапе следует выполнить следующие действия:

- выбрать оптимизатор;
- выбрать функцию потерь (loss function);
- выбрать метрику (metrics);
- установить точность (accuracy).

После компиляции модель будет готова для обучения. В процессе обучения модели в библиотеке Keras следует выполнить следующие шаги:

- передать подготовленные данные в модель;
- выбрать количество эпох (*epochs*) и размер пакета (*batch size*);
- выполнить процесс обучения с помощью метода *fit()*.

Перед началом обучения модели важно передать подготовленные данные в метод *fit()*, используя обучающую выборку *x\_train* и *y\_train*.

### 3.4. Пример обучения модели нейронной сети (алгоритм расчета геометрии взаимного положения взлетно-посадочной платформы и беспилотного летательного аппарата)

Обучение модели происходит в течение фиксированного количества эпох (итераций на наборе данных), где эпоха представляет собой один полный проход по всей обучающей выборке. Для определения подходящего числа эпох рекомендуется отслеживать метрики производительности модели на валидационной выборке. Размер пакета определяет количество образцов, передаваемых модели за одну итерацию, обычно выбирается так, чтобы обеспечить хорошую скорость обучения и уложиться в доступную память.

В данном случае нейронная сеть обучалась в течение 100 эпох, используя тестовые входные и выходные данные, которые были предварительно получены при разбиении полного набора данных. Объем данных (количество строк), передаваемых модели между вычислениями функции потерь, выбран равным 32.

Процесс обучения включает передачу подготовленных данных в метод *fit()* с указанием

числа эпох и размера пакета. Метод `fit()` обучает модель на протяжении фиксированного числа эпох, обновляя веса с использованием выбранного оптимизатора и минимизируя функцию потерь.

При обучении модели рекомендуется сохранять историю метрик производительности для последующего анализа и визуализации. По завершении обучения модель будет готова для использования в предсказаниях или оценке производительности на новых данных.

Используются переменные `'X_train'` и `'Y_train'` для хранения тренировочных данных, полученных при разбиении исходного набора данных.

Параметры метода `fit()`:

- `batch_size`: количество выборок для обновления градиента;
- `epochs`: количество эпох для обучения модели;
- `validation_data`: данные, используемые для оценки потерь и метрик модели в конце каждой эпохи, но не для обучения модели.

```
hist = model.fit(X_train, Y_train,
                 batch_size=32, epochs=100,
                 validation_data=(X_val, Y_val)).
```

Метод `evaluate()` используется для оценки производительности модели в тестовом режиме. Этот метод возвращает значение потерь (*loss*) и значения выбранных метрик для модели на тестовом наборе данных.

Пример использования метода `evaluate()`:

```
accuracy = model.evaluate(X_test, Y_test)
print('Loss: %.6f' % (accuracy[0]))
print('Accuracy: %.2f%%' % (accuracy[1]*100)),
```

где, *model* — ваша обученная модель, *X\_test* — тестовые входные данные, *Y\_test* — соответствующие этим данным ожидаемые выходные данные.

После выполнения метода `evaluate()` получим значение потерь модели на тестовом наборе данных, а также значения выбранных вами метрик, которые позволяют оценить производительность модели на новых, ранее не виденных данных.

Когда модель обучена, мы можем сохранить ее для будущего использования или передачи другим. В процессе разработки модели с использованием Keras может возникать необходимость в отладке и улучшении модели для достижения лучших результатов.

Анализ и визуализация результатов помогут вам лучше понять, как ваша модель работает и что может быть улучшено. В Keras вы можете использовать историю обучения, полученную в процессе обучения, для анализа метрик производительности, таких как точность и функция потерь.

### **3.5. Пример регуляции модели нейронной сети (алгоритм расчета геометрии взаимного положения взлетно-посадочной платформы и беспилотного летательного аппарата)**

Перед использованием нейронной сети необходимо провести оценку достоверности результатов вычислений сети на тестовом наборе входных векторов. Для этого следует использовать тестовый набор, компоненты которого отличаются от компонентов набора, использованного для обучения.

Важно обеспечить разнообразие и репрезентативность тестового набора данных, чтобы получить объективную оценку производительности модели. При выборе тестового набора следует учитывать, что он должен содержать данные, которые модель не видела в процессе обучения, чтобы оценить ее способность к обобщению на новые, ранее неизвестные данные.

После проведения тестирования на тестовом наборе данных можно оценить работу модели и принять решение о ее дальнейшем использовании или необходимости дополнительной настройки.

Регуляризация модели нейронной сети может быть полезным методом для предотвращения переобучения и улучшения обобщающей способности модели. Различные методы регуляризации, такие как *L1* и *L2 регуляризация*, *Dropout* и *Batch Normalization*, могут помочь улучшить производительность модели на новых данных и сделать ее более устойчивой к шуму во входных данных.

Дополнительно для оценки производительности модели на тестовом наборе данных можно использовать метрики, такие как точность (*accuracy*), полнота (*recall*), точность (*precision*) и *F1-мера*. Эти метрики помогут оценить качество работы модели и ее способность правильно классифицировать данные.

Важно помнить, что регуляризация и тестирование модели на разнообразных данных являются важными шагами в разработке нейронных сетей, чтобы обеспечить их эффективную работу и способность к обобщению на новые ситуации.

### 3.6. Проверка модели при помощи тестового набора данных

После выполнения метода `evaluate()` модель возвращает значение потерь и значения выбранных метрик для модели в тестовом режиме. В следующем примере кода:

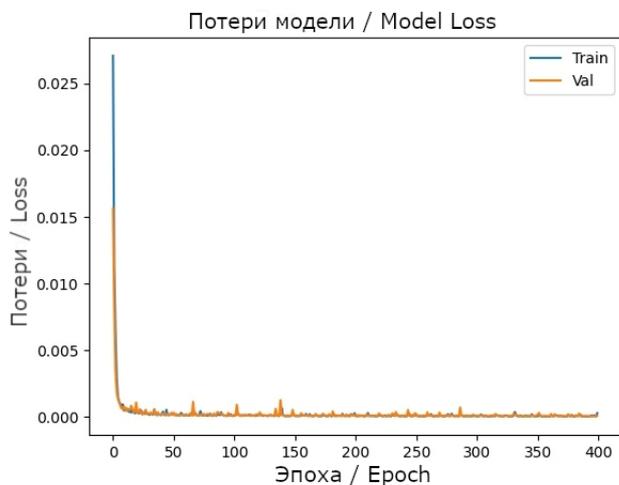
```
accuracy = model.evaluate(X_test, Y_test)
print('Loss: %.6f' % (accuracy[0]))
print('Accuracy: %.2f%%' % (accuracy[1]*100))
```

Эти числовые значения позволяют оценить, насколько хорошо модель справляется с клас-

сификацией данных. Точность (accuracy) указывает на процент правильно классифицированных образцов, а функция потерь (loss function) измеряет разницу между предсказанными значениями и фактическими значениями.

Интерпретация точности и функции потерь помогает понять, насколько хорошо модель обучена и насколько точно она делает предсказания. Точность близка к 100 % и указывает на высокую способность модели к правильной классификации данных, а низкое значение функции потерь говорит о том, что модель хорошо соответствует данным.

Эти результаты могут быть использованы для сравнения различных моделей или для оценки улучшений в процессе обучения. Дополнительно визуализация метрик производительности модели на графиках может помочь более наглядно представить ее эффективность и изменения во время обучения. На этом этапе нейросеть возвращает значение потерь и значения показателей для модели в тестовом режиме (рис. 5). При данных настройках нейросеть достигает точности в 98,17 %, а функция потерь выдает результат  $7.0177e-06$  (рис. 6).

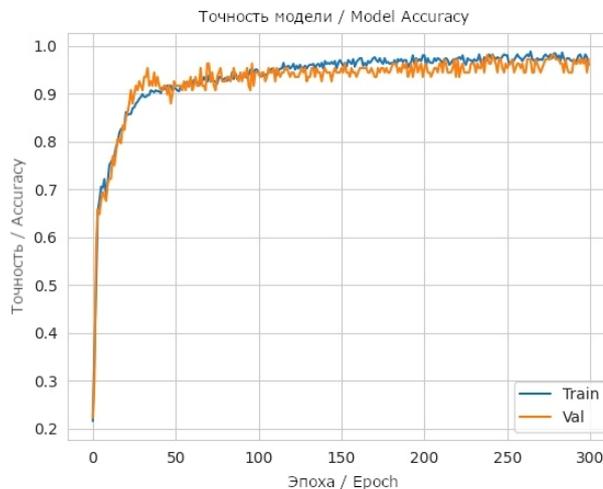


**Рис. 5.** Потери модели

Источник: выполнено В.А. Сусловым, С.В. Гагарским

**Figure 5.** Model losses

Source: done by V.A. Suslov, S.V. Gagarsky



**Рис. 6.** Точность модели

Источник: выполнено В.А. Сусловым, С.В. Гагарским

**Figure 6.** Model accuracy

Source: done by V.A. Suslov, S.V. Gagarsky

Для проверки модели используются тестовые входные данные. Полученные предсказанные данные можно сравнить с тестовыми выходными данными, которые являются эталонными. Также можно подавать как одиночные строки с 9 входными данными, так и несколько строк подряд, эмулируя подачу данных на нейронную сеть от управляющего пункта управления ППУ.

При сравнении компонентов эталонных векторов с соответствующими компонентами выходных векторов нейронной сети видно, что почти все точки лежат на прямой, что свидетельствует о правильной работе сети на тестовом наборе данных (рис. 7).

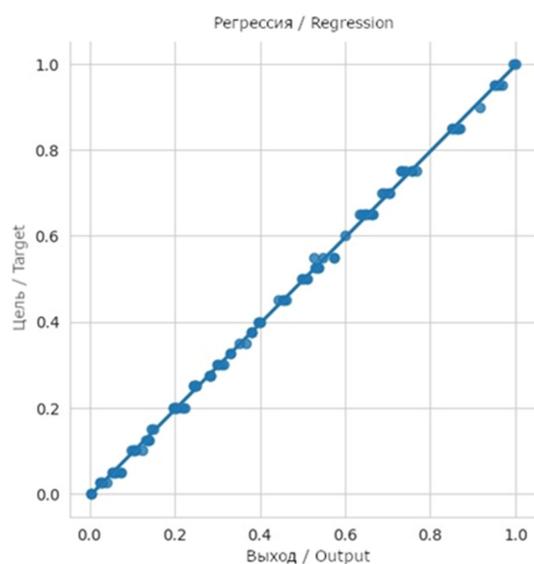


Рис. 7. Регрессия

Источник: выполнено В.А. Сусловым, С.В. Гагарским

Figure 7. Regression

Source: done by V.A. Suslov, S.V. Gagarsky

Рассмотрено несколько распространенных методов сохранения и загрузки модели, таких как регуляризация, сокращение размерности и аугментация данных. Сохранение и загрузка модели являются важными процессами при работе с глубоким обучением. Они позволяют сохранить результаты обучения и использовать модель в других приложениях или продолжить обучение с сохраненной точки.

После того как модель обучена и улучшена в Keras, ее можно сохранить для последующего использования без необходимости повторного обучения. Загруженная модель будет готова к использованию для предсказаний или оценки производительности на новых данных.

Для визуализации процесса обучения и оценки модели представлены графики потерь и точности модели на тренировочном и валидационном наборах данных. Также представлен график регрессии, отображающий соотношение между предсказанными и истинными значениями. Кроме того, приведен пример использования функции `plot_model` из **TensorFlow Keras** для визуализации архитектуры модели.

Все эти шаги помогают не только оценить производительность модели, но и лучше понять ее работу, а также обеспечить возможность последующего использования и дальнейшего улучшения.

#### 4. Результаты исследования

В результате разработки и тестирования нейросетевой системы управления достигнута точность работы системы 98.17 %, что подтверждает высокую эффективность выбранной архитектуры. Получено минимальное значение функции потерь ( $7.0177e-06$ ), свидетельствующее о стабильности обучения.

Реализована оптимальная архитектура нейронной сети с входным слоем (9 нейронов) для обработки данных с сенсоров — тремя скрытыми слоями (2000, 1500, 1500 нейронов) с функцией активации ReLU, двумя слоями Dropout для предотвращения переобучения, выходным слоем (6 нейронов) с функцией активации Hard Sigmoid.

Создан эффективный алгоритм определения взаимного положения БПЛА и платформы, основанный на использовании приемопередающих устройств с лидарами и матрицами фоторезисторов и обработке данных 9 параметров положения для определения 6 координат пространственного положения.

Разработана программно-аппаратная система, включающая механизацию процесса по-

садки, алгоритмы управления на базе нейронных сетей, систему сенсоров для точного позиционирования.

## Заключение

Задача точного и плавного опускания БПЛА на подвижную посадочную платформу в условиях боевых действий представляет значительные сложности, даже при наличии современных технологий и систем управления. Это обуславливает необходимость разработки и внедрения передовых технологий, таких как нейронные сети, для повышения эффективности посадочного процесса в условиях реального времени и изменяющихся внешних факторов.

Нейронные сети играют важную роль в автоматизации посадки БПЛА на подвижные платформы, позволяя системе обучаться на широком наборе данных и принимать решения в реальном времени. Адаптируясь к переменам в окружающей среде, нейронные сети учитывают такие факторы, как движение платформы, погодные условия, препятствия и другие параметры, что обеспечивает стабильную и безопасную посадку.

Применение нейронных сетей не только улучшает точность посадки, но и повышает безопасность операции, позволяет быстро реагировать на изменение обстоятельств и улучшает координацию военных действий. Таким образом, нейросетевые технологии в автоматизированной посадке БПЛА способствуют значительному увеличению эффективности и успешности операций, особенно в боевых условиях, где время играет критическую роль. Проведенное исследование подтвердило эффективность применения нейронных сетей для решения задачи автоматической посадки БПЛА на подвижную платформу. Разработанная система демонстрирует высокую точность и надежность, что особенно важно при эксплуатации в сложных условиях.

В работе рассмотрены подходы к автоматическому управлению БПЛА с использованием нейросетевых регуляторов, проанализированы сценарии захода на посадку в условиях верти-

кальных ветровых возмущений и представлены архитектуры нейросетевых регуляторов для управления в продольном канале. Также приведены результаты моделирования и обучения сети, показавшие высокую точность в выполнении двух управленческих законов, обеспечивающих стабильность полета и точность посадки. Дальнейшее совершенствование нейронных сетей и их применение в подобных задачах могут значительно улучшить эффективность и безопасность авиационных операций. Практическая значимость результатов заключается в возможности их применения для повышения эффективности и безопасности операций с использованием БПЛА, особенно в условиях, требующих высокой точности позиционирования.

## Список литературы

1. *Токарев Ю.П.* Методы управления беспилотными летательными аппаратами в общем воздушном пространстве: дис. ... канд. техн. наук: 05.22.13. Москва, 2022. 168 с.
2. *Платунова А.В., Клишин А.Н., Илюхин С.Н.* Особенности формирования адаптивных законов управления высокоточными летательными аппаратами // *Инженерный вестник*. 2016. № 10. С. 5. EDN: XСMPBV
3. *Никанорова М.Д., Заболотская Е.В.* Численный расчет аэродинамических характеристик беспилотного летательного аппарата // *Политехнический молодежный журнал МГТУ им. Н.Э. Баумана*. 2019. № 4 (33). С. 1–14. <https://doi.org/10.18698/2541-8009-2019-4-464>
4. *Кореванов С., Казин В.В.* Искусственные нейронные сети в задачах навигации беспилотных летательных аппаратов // *Научный вестник МГТУ ГА*. 2014. № 201. С. 46–49. EDN: RYFUSP
5. *Долгов Е.Н.* Искусственный интеллект для управления летательными аппаратами // *Молодой ученый*. 2021. № 16 (358). С. 81–86. EDN: DDХGPE
6. *Щербинин В.В., Васильева Ю.С., Чижевская О.М., Шевцова Е.В.* Методы функционирования и алгоритмы корреляционно-экстремальной системы навигации летательных аппаратов на основе цветового зрения // *Гироскопия и навигация*. 2013. Т. 4, № 1. С. 39–49. <https://doi.org/10.1134/S2075108713010082>
7. *Смольский А.Г., Коваленко С.Н., Мухута М.В.* Нейросетевой алгоритм обработки геопространственной информации с беспилотного летательного аппарата // *Геоинформационные системы военного назначения: теория и практика применения: материалы IX Респ. науч.-практ. конф., Минск, 12 мая 2023 г. / Белорус. гос. ун-т; редкол.* С. 12–16. EDN: ZVCQOK

8. Макаров И.М., Лохин В.М., Манько С.В. Искусственный интеллект и интеллектуальные системы управления. Москва : Наука, 2006. 333 с.

9. Гафаров Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения : учебное пособие. Казань : КФУ, 2018. 320 с.

10. Сизов А.В., Ипполитов С.В., Савченко А.Ю., Малышев В.А. Способ автономной коррекции инерциальной навигационной системы БПЛА на основе геоинформационных технологий // Моделирование, оптимизация и информационные технологии. 2019. Т. 7 (1). С. 183–195. <https://doi.org/10.26102/2310-6018/2019.24.1.030>

11. Иванова И.А., Никонов В.В., Царева А.А. Способы организации управления беспилотными летательными аппаратами // Актуальные проблемы гуманитарных и естественных наук. 2014. № 11–1. С. 56–63. EDN: TCIQMH

12. Коротин П.С., Алексеев Е.Г. Опыт применения Keras как Front end для tensorflow // Сборник трудов XXIII научно-практической конференции МРСУ. Саранск, 2019. С. 301–306. EDN: KVNIOF

13. Ткачев Н.М., Федяев О.И. Параметрическое описание моделей глубоких нейронных сетей в библиотеке Keras // Программная инженерия: методы и технологии разработки информационно-вычислительных систем. Донецк, 2018. С. 112–118. EDN: ZVPZTL

14. Гагарский С.В., Суслов В.А. Система посадки вертолета на палубу. Санкт-Петербург : Автономные системы, 2022. 95 с.

15. Федосеева Н.А., Загвоздкин М.В. Перспективные оБПЛАсти применения беспилотных летательных аппаратов // Научный журнал. 2017. Т. 22 (9). С. 26–29. EDN: ZSUMLX

16. Попов А.Н. Методы планирования траектории движения беспилотного летательного аппарата // Известия СамНЦ РАН. 2017. Т. 19 (1–2). С. 364–370. EDN: ZTPOIN

17. Маммадов А.З. Модель инерциальной навигации для беспилотных летательных аппаратов // Universum: технические науки. 2021. № 5(86). С. 5–9. <https://doi.org/10.32743/UniTech.2021.86.5.11683>

18. Библиотека Keras. URL: <https://keras.io/> (дата обращения: 17.05.2024)

19. Автономная навигационная система. URL: <https://info.wikireading.ru/84077> (дата обращения: 27.04.2024)

20. Сенсоры беспилотников. URL: <https://www.energovector.com/energoznanie-kak-robotyorientiruyutsya-v-prostranstve.html> (дата обращения: 27.04.2024)

## References

1. Tokarev YuP. *Methods of Controlling Unmanned Aerial Vehicles in Common Airspace Using Flight Information with Automatic Dependent Surveillance*: Ph.D.

thesis in Technical Sciences: 05.22.13. Moscow; 2022. (In Russ.)

2. Platonova AV, Klishin AN, Ilyukhin SN. Features of Forming Adaptive Control Laws for High-Precision Aircraft. *Engineering Bulletin*. 2016;(10):5. (In Russ.) EDN: XCOMPBB

3. Nikanorova MD, Zabolotskaya EV. Numerical calculation of aerodynamic characteristics unmanned aerial vehicle. *Politechnical student journal*. 2019;4(33):1–14. <https://doi.org/10.18698/2541-8009-2019-4-464>

4. Korevanov S, Kazin VV. Application of artificial neural networks in problems of general and comparative navigation methods of UAV. *Civil aviation high technologies*. 2014;(201):46–49. (In Russ.) EDN: RYFUSP

5. Dolgov EN. Artificial Intelligence for Aircraft Control. *Young Scientist*. 2021;16(358):81–86. (In Russ.) EDN: DDYGPE

6. Scherbinin VV, Vasil'eva YS, Chizhevskaya OM, Shevtsova EV. Functioning methods and algorithms of color vision-based correlation-extremal aircraft navigation system. *Gyroscopy and Navigation*. 2013;4(1):39–49. <https://doi.org/10.1134/S2075108713010082>

7. Smolsky AG, Kovalenko SN, Mikhuta MV. Neural network algorithm for processing geospatial information from an unmanned aerial vehicle. In: *Geoinformation systems for military purposes: theory and practice of application*. Minsk; 2023. p. 12–16. (In Russ.) EDN: ZVCQOK

8. Makarov IM, Lokhin VM, Manko SV. *Artificial Intelligence and Intelligent Control Systems*. Moscow: Nauka Publ.; 2006. (In Russ.)

9. Gafarov FM, Galimyanov AF. *Artificial Neural Networks and Applications: Textbook*. Kazan: Kazan Federal University; 2018. (In Russ.)

10. Sizov AV, Ippolitov SV, Savchenko AYu, Malyshchev VA. Method of autonomous correction of inertial navigation system of unmanned aircraft on the basis of modern geoinformation technologies. *Modeling, Optimization and Information Technologies*. 2019;7(1):183–195. (In Russ.) <https://doi.org/10.26102/2310-6018/2019.24.1.030>

11. Ivanova IA, Nikonov VV, Tsareva AA. Methods of Organizing Control of Unmanned Aerial Vehicles. *Actual problems of the humanities and natural sciences*. 2014;(11–1):56–63. (In Russ.) EDN: TCIQMH

12. Korotin PS, Alekseev EG. The experience of using Keras as a Front end for tensorflow. In: *XXIII scientific and practical conference of National Research Mordovian State University*. Saransk; 2019. p. 301–306. EDN: KVNIOF

13. Tkachev N, Fedyaev O. Parametric description of deep neural network models in the Keras library. In: *Software Engineering: Methods and Technologies*. Donetsk; 2018. p. 112–118. EDN: ZVPZTL

14. Gagarsky SV, Suslov VA. *Helicopter Deck Landing System*. Saint Petersburg: Autonomous Systems LLC; 2022. (In Russ.)

15. Fedoseeva NA, Zagvozdkin MV. Promising Areas of Application of Unmanned Aerial Vehicles. *Scientific Journal*. 2017;22(9):26–29. (In Russ.) EDN: ZSUMLX

16. Popov AN. Planning methods of movement trajectory of unmanned aerial vehicle. *Izvestia of Samara Scientific Center RAS*. 2017;19(1–2):364–370. (In Russ.) EDN: ZTPOIN

17. Mammadov AZ. Model inertial navigation for unmanned aerial vehicles. *Universum: Technical Sciences*.

2021;5(86):5–9. (In Russ.) <https://doi.org/10.32743/UniTech.2021.86.5.11683>

18. *Keras Library*. Available from: <https://keras.io/> (accessed: 17.05.2024).

19. *Autonomous Navigation System*. Available from: <https://info.wikireading.ru/84077> (accessed: 27.04.2024).

20. *UAV Sensors*. Available from: <https://www.energovector.com/energoznanie-kak-roboty-orientiruyutsya-v-prostranstve.html> (accessed: 27.04.2024).

#### Сведения об авторах:

**Сулов Владислав Артемович**, аспирант кафедры стартовых и технических комплексов ракет и космических аппаратов, Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова, Санкт-Петербург, Россия; eLIBRARY SPIN-код: 2143-8843, ORCID: 0009-0007-1028-5993; e-mail: vlarsu@mail.ru

**Гагарский Сергей Васильевич**, кандидат технических наук, доцент кафедры стартовых и технических комплексов ракет и космических аппаратов, Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова, Санкт-Петербург, Россия; e-mail: svgagarski@mail.ru

#### About the authors

**Vladislav A. Suslov**, Post-graduate student of the Department of Launch and Technical Complexes of Rockets and Spacecraft, Baltic State Technical University “Voenmeh” named after D.F. Ustinov, Saint Petersburg, Russia; eLIBRARY SPIN-code: 2143-8843, ORCID: 0009-0007-1028-5993; e-mail: vlarsu@mail.ru

**Sergey V. Gagarsky**, Candidate of Technical Sciences, Associate Professor of the Department of Launch and Technical Complexes of Rockets and Spacecraft, Baltic State Technical University “Voenmeh” named after D.F. Ustinov, Saint Petersburg, Russia; e-mail: svgagarski@mail.ru