



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 198 за 2018 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Сорокин А.В., [Широбоков М.Г.](#)

Коррекция и
прогнозирование
орбитального движения
космических аппаратов с
помощью искусственных
нейронных сетей

Рекомендуемая форма библиографической ссылки: Сорокин А.В., Широбоков М.Г. Коррекция и прогнозирование орбитального движения космических аппаратов с помощью искусственных нейронных сетей // Препринты ИПМ им. М.В.Келдыша. 2018. № 198. 28 с. doi:[10.20948/prepr-2018-198](https://doi.org/10.20948/prepr-2018-198)
URL: <http://library.keldysh.ru/preprint.asp?id=2018-198>

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

А.В. Сорокин, М.Г. Широбоков

Коррекция и прогнозирование орбитального
движения космических аппаратов с помощью
искусственных нейронных сетей

Москва — 2018

А.В. Сорокин, М.Г. Широбоков

Коррекция и прогнозирование орбитального движения космических аппаратов с помощью искусственных нейронных сетей

Рассматривается задача перелета между двумя точками пространства с использованием искусственных нейронных сетей. Предполагается, что бортовой компьютер космического аппарата оснащен искусственной нейронной сетью, отвечающей за прогноз движения, и сетью, отвечающей за расчет импульсов коррекции движения. Обе нейронные сети обучены в рамках невозмущенной задачи двух тел. Цель работы – исследование возможностей предлагаемых нейронных сетей для коррекции движения в возмущенной задаче двух тел.

Ключевые слова: искусственная нейронная сеть, орбитальное движение, прогнозирование движения, коррекция движения

A. V. Sorokin, M. G. Shirobokov

Correction and prediction of the spacecraft's orbital motion by artificial neural networks

The problem of transfer between two points of space exploiting the artificial neural networks is considered. It is assumed that the onboard computer of the spacecraft is equipped by the two neural networks: the predictive one and the controlling one. Both the neural networks are trained within the framework of the unperturbed two-body problem model. The purpose of the work is to study the capabilities of the proposed neural networks for the correction of motion in cases of perturbations.

Key words: artificial neural network, orbital motion, motion prediction, motion correction

Работа поддержана грантом Российского фонда фундаментальных исследований (РФФИ) 18-31-00403.

Введение

В последнее время активно повышается внимание к методам машинного обучения в различных областях науки. Методы машинного обучения включают в себя различные инструменты обработки и обобщения информации, такие как: машины опорных векторов, алгоритмы бустинга, нейронные сети, каждая из которых имеет свою специфику. В настоящей работе упор делается на применении искусственных нейронных сетей (ИНС) в задачах орбитальной динамики.

По сравнению с традиционными вычислительными методами, ИНС обладают следующими преимуществами: способность решать задачи, в которых неизвестны зависимости между входными и выходными данными, устойчивость к шумам во входных данных, адаптирование к изменениям окружающей среды, а также отказоустойчивость при аппаратной реализации, т.е. незначительное падение производительности при неблагоприятных условиях. Это значит, что с учетом высоких требований надежности, предъявляемых к космическим системам, нейронные сети могут быть подспорьем для бортовой реализации алгоритмов управления.

С одной стороны, интерес к нейронным сетям в области орбитальной динамики обусловлен повышенными требованиями к степени автономности космических аппаратов. При потере связи с наземной станцией или нештатной ситуации на борту аппарат должен самостоятельно брать на себя управление и выполнять поставленные перед ним задачи. Автономность особенно важна для аппаратов, движущихся на низких орбитах, так как для них плотность атмосферы является вторым по значимости фактором после гравитационного притяжения. С использованием методов обучения с учителем нейронные сети могут помочь уточнить модель атмосферы во время движения аппарата [1,2].

С другой стороны, поиск траекторий, оптимальных с точки зрения времени полета или затрат топлива, представляет собой сложный и ресурсозатратный процесс, так как поведение сходимости традиционных методов оптимизации зависит от «адекватных» начальных приближений. Искусственные нейронные сети помогают частично решать эти проблемы. Например, нейронные сети, оптимизация синаптических связей в которых выполняется эволюционными алгоритмами, методами обучения с подкреплением способны синтезировать регуляторы для полетов космических аппаратов с малой тягой к Луне [3,4,5]. Алгоритмы оптимизации, рассмотренные в этих работах, не требуют начального приближения. Другой пример использования искусственных нейронных сетей — управление формациями спутников в условиях неустойчивой динамики вокруг коллинеарных точек либрации [6].

В настоящей работе производится моделирование ситуации, когда космический аппарат управляет своим орбитальным движением с помощью ИНС. Перейдем к постановке задачи.

1. Постановка задачи

Рассмотрим перелет космического аппарата (КА) из начальной точки в конечную точку за заданное время в рамках возмущенной модели двух тел. Введем инерциальную систему координат $Oxyz$ с началом в притягивающем центре и запишем систему уравнений:

$$\begin{cases} \ddot{\vec{r}} + \mu \frac{\vec{r}}{r^3} = \vec{F} \\ \vec{r}(t_0) = \vec{R}_1 \\ \vec{r}(t_1) = \vec{R}_2 \end{cases}$$

где μ — гравитационный параметр, \vec{r} — радиус-вектор, соединяющий центр масс КА и притягивающий центр, \vec{F} — сумма возмущающих сил, действующих на КА, t_0 и t_1 — начальное и конечное время перелета, \vec{R}_1 и \vec{R}_2 — векторы начальной и конечной точек (см. рисунок 1; здесь $\vec{R}_1 = [1, 0, 0]$, $\vec{R}_2 = [0, 2, 0]$ в безразмерной системе единиц, в которой гравитационный параметр μ равен 1).

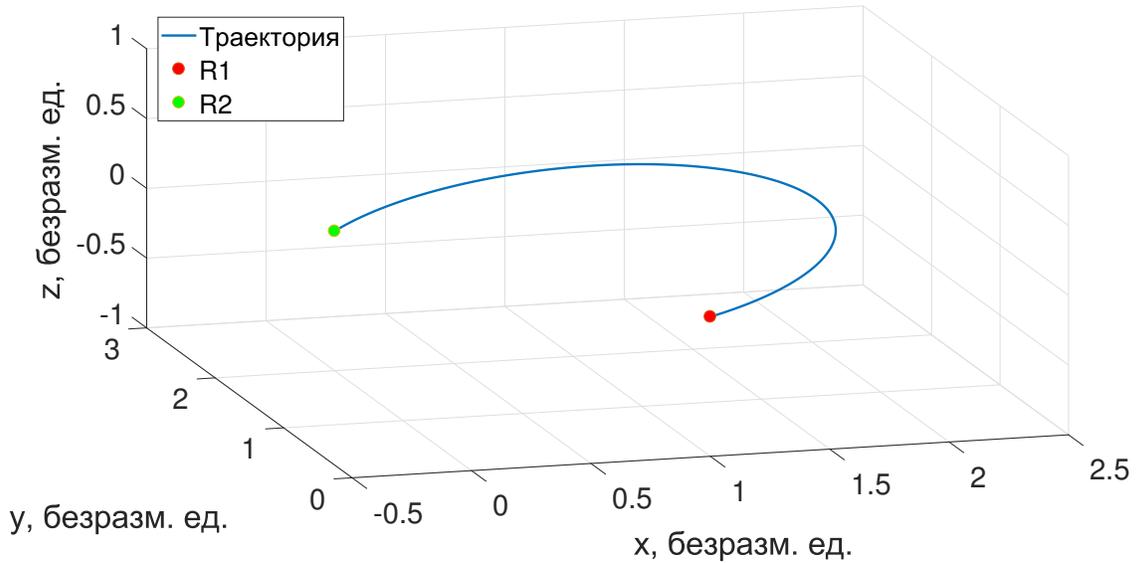


Рис. 1. Перелет из $\vec{R}_1 = [1, 0, 0]$ в $\vec{R}_2 = [0, 2, 0]$

Ставится задача поиска решения данной системы (двухточечная краевая задача). Эта задача эквивалентна поиску скоростей в точках \vec{R}_1 и \vec{R}_2 .

Решение задачи можно найти различными способами: например, использовать эффективные методы решения задачи Ламберта в невозмущенной задаче двух тел, методы простой или параллельной пристрелки.

В данной работе для поиска решения предлагается воспользоваться ИНС. А именно, вводится нейронная сеть, отображающая положения \vec{R}_1 , \vec{R}_2 и время перелета T в требуемые скорости \vec{V}_1 и \vec{V}_2 . Эта нейронная сеть называется управляющей. Обучение управляющей нейронной сети легче всего выполнить в рамках невозмущенной задачи двух тел, так как в этом случае для генерирования выборки можно воспользоваться известными эффективными алгоритмами решения задачи Ламберта. Решение, найденное управляющей нейронной сетью в рамках невозмущенной задачи двух тел, не приведет КА к цели, поэтому его движение следует скорректировать. Коррекцию осуществляет та же управляющая нейронная сеть. Чтобы определить момент времени, когда необходимо провести коррекцию, движение КА следует спрогнозировать. Для этого вводится вторая нейронная сеть – прогнозирующая. Она также обучается в рамках невозмущенной задачи двух тел. Таким образом, движение КА в возмущенной среде определяется работой двух нейронных сетей (управляющей и прогнозирующей), обученных в рамках невозмущенной задачи двух тел. В связи с этим возникает еще одна задача: тестирование предложенных нейронных сетей.

Итак, в данной работе ставятся следующие задачи:

1. Разработать прогнозирующую нейронную сеть, подготовить для нее обучающую выборку, подобрать алгоритм обучения и обучить ее.
2. Разработать корректирующую нейронную сеть, подготовить для нее обучающую выборку, подобрать алгоритм обучения и обучить ее.
3. Оценить точность решения, полученного ИНС, оценить количество коррекций, величины среднего и суммарного импульсов при различных величинах возмущающих ускорений, проверить влияние количества нейронов в сети на точность коррекций, проверить устойчивость решения к начальному условию.

Для выполнения цели работы сначала в главе 2 приводится общая информация об ИНС и существующих алгоритмах их обучения. Далее в главе 3 описываются методы построения обучающих выборок, нейронных сетей и алгоритмов обучения. В главе 4 проводится моделирование разработанных ИНС, приводятся результаты исследования. В заключительной главе перечисляются главные выводы работы.

2. Введение в нейронные сети

С математической точки зрения, нейронная сеть — это взвешенный ориентированный граф, в узлах которого находятся функции, называемые активационными. Процесс изменения весов в графе называется ее обучением. Формально обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации. Существует несколько способов обучения ИНС, таких как обучение с учителем, обучение без учителя и обучение с подкреплением. Каждый способ имеет свою типологию задач. Обучение с учителем позволяет решать задачу регрессии и классификации объектов. В этом случае для настройки параметров сеть использует массив данных, состоящий из входных сигналов и соответствующих им выходных сигналов. Метод обучения нейронных сетей без учителя позволяет решать задачу кластеризации, поиска ассоциаций и сокращения размерности. Алгоритмы обучения с подкреплением выдают стратегию, приписывающую состояниям окружающей среды действия, которые должна предпринять система в этих состояниях. В данной главе подробно рассматриваются методы обучения с учителем, соответствующие нейронные сети и их архитектуры.

Структура графа ИНС определяется следующим образом: выделяется множество вершин, называемых входным слоем нейронной сети. Далее в граф определенным образом добавляется еще одно множество вершин, называемое скрытым слоем, после чего определяются ребра, связывающие два слоя. Процедура добавления скрытых слоев итеративна, то есть таким образом можно построить N скрытых слоев. Последний слой нейронной сети называется выходным. После построения графа произвольным образом инициализируются веса в его ребрах. На рисунке 2 приведен пример многослойной нейронной сети.

В каждом узле графа размещена некоторая функция единственного аргумента. Данная функция называется активационной и обладает следующими особенностями: монотонность и дифференцируемость. Пример активационной функции — сигмоида:

$$\phi(x) = \frac{1}{1 + e^{-x}}, \quad \phi'(x) = \phi(x) \cdot (1 - \phi(x)).$$

Каждый слой сети независимо друг от друга обладает одной и той же активационной функцией. После того как была определена структура нейронной сети, через входной слой по сети распространяется сигнал. Под входным сигналом будем понимать некоторый вектор, длина которого совпадает с количеством нейронов входного слоя сети. Выходной сигнал — это, соответственно, вектор, длина которого равна количеству нейронов

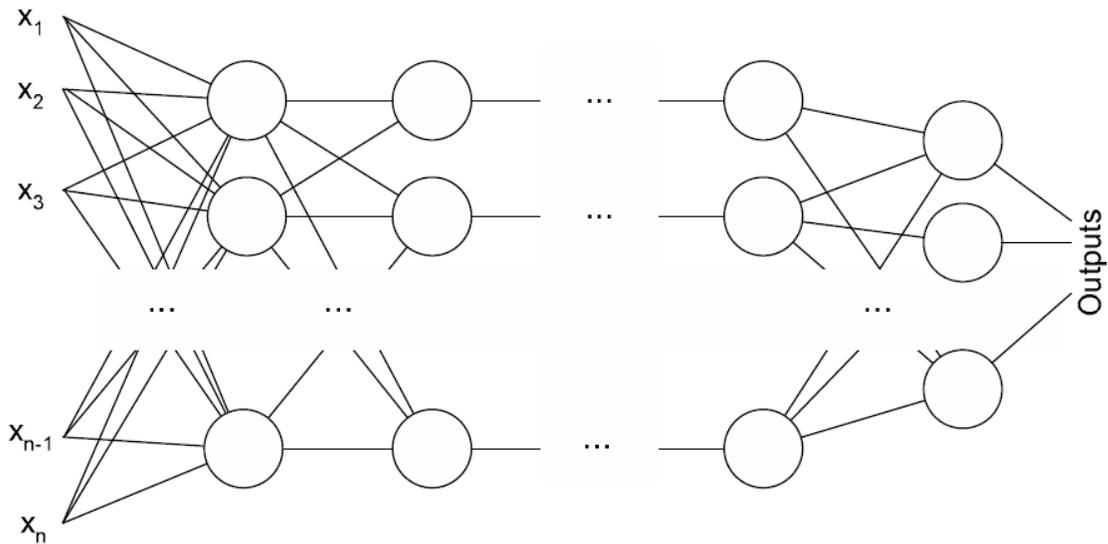


Рис. 2. Многослойная нейронная сеть

в выходном слое сети. В каждом узле графа вычисляется значение активационной функции от линейной комбинации сигнала. Поскольку веса в графе в начальный момент были определены произвольно, то значения в выходном слое также будут произвольными. Обучение с учителем предполагает наличие обучающей выборки типа входной-выходной сигнал. Под обучающей выборкой будем понимать конечный набор из входных и соответствующих им выходных сигналов. После того как входной сигнал распространился по сети, необходимо ввести некоторый функционал ошибки, для того чтобы оценить меру совпадения результата работы сети — предсказанного сигнала, и соответствующего выходного сигнала. Существует множество подобных функционалов, к примеру, средний модуль отклонения (Mean Absolute Error):

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \tilde{y}_i|,$$

где y_i — i -я компонента выходящего сигнала, \tilde{y}_i — i -я компонента предсказанного сигнала, а N — количество компонент выходного сигнала. Поскольку сигнал — это вектор, то под его компонентами будем понимать векторные компоненты. В итоге принцип работы ИНС заключается в том, чтобы минимизировать имеющийся функционал посредством изменения весов в графе. Большинство методов оптимизации являются градиентными, именно поэтому накладывалось условие дифференцируемости на активационные функции. Одним из основных методов минимизации функционалов ошибки в ИНС, который используется в

данной работе, является метод обратного распространения ошибки. Рассмотрим принцип работы этого алгоритма на примере простейшего функционала ошибки, приведенного ранее — $MAE = \frac{1}{N} \sum_{i=1}^N |d_i - t_i|$, где d_i — i -я компонента выходящего сигнала, t_i — i -я компонента предсказанного сигнала, а N — количество компонент выходного сигнала. Введем необходимые обозначения. Пусть пример обучения представлен парой (\vec{x}, \vec{d}) , где \vec{x} — входной вектор, а \vec{d} — желаемый отклик. Пусть в сети L слоев, тогда для j -го нейрона l -го слоя линейная комбинация выглядит следующим образом: $v_j^l = \sum_{i=0}^M w_{ji}^l \cdot y_i^{l-1}$, где y_i^{l-1} — выходной сигнал нейрона i , расположенного в слое $l - 1$, w_{ji}^l — вес связи нейрона j слоя l с нейроном i слоя $l - 1$. В случае сигмоидальной функции активации выходной сигнал нейрона j слоя l имеет следующий вид: $y_j^l = \phi_j(v_j^l)$. Если $l = 0$, т.е. нейрон j находится во входном слое, то $y_j^0 = x_j$, где x_j — j -я компонента входного вектора \vec{x} . Если же нейрон j находится в выходном слое, то $y_j^L = t_j$. Таким образом, j -я компонента функционала ошибки равна $e_j = \frac{1}{N} |d_j - t_j|$. Для минимизации ошибки будем двигаться в направлении антиградиента. Значения локальных градиентов δ_j^l в узлах графа задаются следующей формулой [7]:

$$\delta_j^l = \begin{cases} e_j^L \cdot \phi_j'(v_j^L) & \text{для нейрона } j \text{ выходного слоя } L \\ \phi_j'(v_j^l) \cdot \sum_k \delta_k^{l+1} \cdot \delta_{kj}^{l+1} & \text{для нейрона } j \text{ скрытого слоя } l \end{cases}$$

Следовательно, изменение весов в l -м слое будет происходить по следующей формуле: $w_{ji}^l \leftarrow w_{ji}^l + \lambda \cdot \delta_j^l \cdot y_i^{l-1}$. Процесс итеративный и завершается в том случае, когда функционал ошибки достигает определенного значения.

Качество обучения нейронных сетей напрямую зависит от выбора функционала ошибки, метода оптимизации и архитектуры сети, однако наряду с проблемой выбора этих характеристик сети существует также проблема переобучения. Суть явления заключается в том, что ИНС в процессе обучения начинает оптимизировать параметры таким образом, что точность аппроксимации обучающей выборки будет значительно выше, чем точность аппроксимации отличающейся выборки из того же распределения. Существует множество способов борьбы с этой проблемой. Один из самых эффективных методов называется dropout. Принцип действия метода заключается в следующем: вначале определяется количество слоев и нейронов в ИНС, после чего строится полный ориентированный граф. После все ребра в графе пронумеровываются, и с заданной вероятностью p исключается каждое. В результате получается некоторый произвольным образом связанный граф. Далее строится ансамбль таких графов, и каждый обучается на одной и той

же обучающей выборке. Итоговый результат получается усреднением весов для всех графов. Эффективность данного подхода показана экспериментально [8].

3. Решение задачи

Рассмотрим задачу перелета КА из начальной точки \vec{R}_1 в конечную точку \vec{R}_2 за заданное время, но в рамках невозмущенной модели. Необходимо найти траекторию перелета, удовлетворяющую заданным условиям:

$$\begin{cases} \ddot{\vec{r}} = -\mu \frac{\vec{r}}{r^3} \\ \vec{r}(t_0) = \vec{R}_1 \\ \vec{r}(t_1) = \vec{R}_2 \end{cases}$$

Данная задача называется задачей Ламберта. Ее численное решение, реализованное в среде MATLAB, — функция *lambert.m* [9] — принимает в качестве аргументов радиус-векторы заданных точек и время, выделенное на перелет, а возвращает вектор-скорости в заданных точках. Скорость в начальной точке рассчитывается таким образом, чтобы попасть в конечную точку.

Задачу из постановки можно решить, используя функцию *lambert.m*. Для этого введем некоторый промежуток времени Δt , не зависящий от длительности перелета. При интегрировании возмущенной траектории спустя каждый промежуток времени Δt будем использовать значение текущего фазового вектора в качестве начального условия для интегрирования невозмущенной траектории методом Рунге-Кутты 4-го порядка с постоянным шагом. В случае, когда евклидова норма разности радиус-вектора конечной точки и радиус-вектора конечной точки, полученного путем интегрирования, будет превышать заранее заданную величину ε , будем применять импульсную коррекцию при помощи *lambert.m* таким образом, чтобы попасть в конечную точку.

Теперь перейдем к решению поставленной задачи при помощи нейронных сетей. Осуществим переход следующим образом: вместо интегрирования невозмущенной траектории методом Рунге-Кутты построим прогнозирующую нейронную сеть, которая также по значению текущего фазового вектора будет рассчитывать значение конечного радиус-вектора. Теперь, в случае, когда евклидова норма разности радиус-вектора конечной точки и радиус-вектора конечной точки, полученного прогнозирующей сетью, будет превышать величину ε , будем применять импульсную коррекцию, но уже при помощи корректирующей нейронной сети. Приведем пример псевдокода:

```

if  $\|\vec{R}_2 - NET_{progn}(\vec{y}_i)\| > \varepsilon$  then
    apply correction with  $NET_{corr}(\vec{R}_i)$ ;
else
    do not apply correction;
end if

```

где \vec{R}_2 – конечный радиус вектор, $\vec{y}_i = [R_x^i, R_y^i, R_z^i, V_x^i, V_y^i, V_z^i]$ – текущий фазовый вектор, $NET_{progn}(\vec{y}_i)$ – радиус-вектор конечной точки, полученный прогнозирующей сетью, \vec{R}_i – текущий радиус-вектор, $NET_{corr}(\vec{R}_i)$ – вектор скорости, полученный корректирующей сетью.

Перейдем к составлению обучающих выборок. Поскольку для задачи прогнозирования численное решение можно построить для любой точки фазового пространства, то в случае нейронной сети необходимо определить ограниченную область фазового пространства, в которой будет составлена выборка. В качестве этой области будем брать некоторую трубку траекторий. Определим два шара, первый – радиуса ε_1 с центром в точке, задаваемой начальным радиус-вектором, второй – радиуса ε_2 с центром в точке, задаваемой вектором начальной скорости. Составим в этих шарах конечный набор начальных условий следующим образом: компоненты радиус-вектора будем выбирать из равномерного распределения с нулевым математическим ожиданием на шаре радиуса ε_1 . Компоненты вектора скорости выберем из того же распределения, но на шаре радиуса ε_2 . Теперь для каждого начального условия проинтегрируем с постоянным шагом соответствующую траекторию в рамках невозмущенной модели. Таким образом, для нейронной сети, прогнозирующей движение, обучающая выборка будет состоять из радиус-векторов, принадлежащих траекториям, и соответствующих им конечных радиус-векторов: $S = \{S_i\}_{i=1}^M$, где $S_i = \{[\vec{R}_n^i, \vec{R}_{end}^i]\}_{n=1}^N$, M – количество траекторий, а N – количество точек на траектории. На рисунке 3 приведен набор траекторий для обучающей выборки прогнозирования.

В задаче коррекции численное решение базируется на функции *lambert.m*, таким образом, необходимо построить нейросетевой аналог отображения, задаваемый этой функцией, в ограниченной области пространства. Для этого построим трубку фиксированного радиуса δ и равномерно распределим в ней конечное количество позиций КА. Для каждой позиции посчитаем скорость, необходимую для того, чтобы попасть в конечную точку. Таким образом, для нейронной сети, корректирующей движение, обучающая выборка будет состоять из радиус-векторов, принадлежащих трубке, соответствующих им вектор-скоростей и времени полета: $S = \{[\vec{R}_i, t_i, \vec{V}_i^{lambt}]\}_{i=1}^N$, где N – количество заданных позиций. Включать радиус-вектор конечной точки в выборку нет

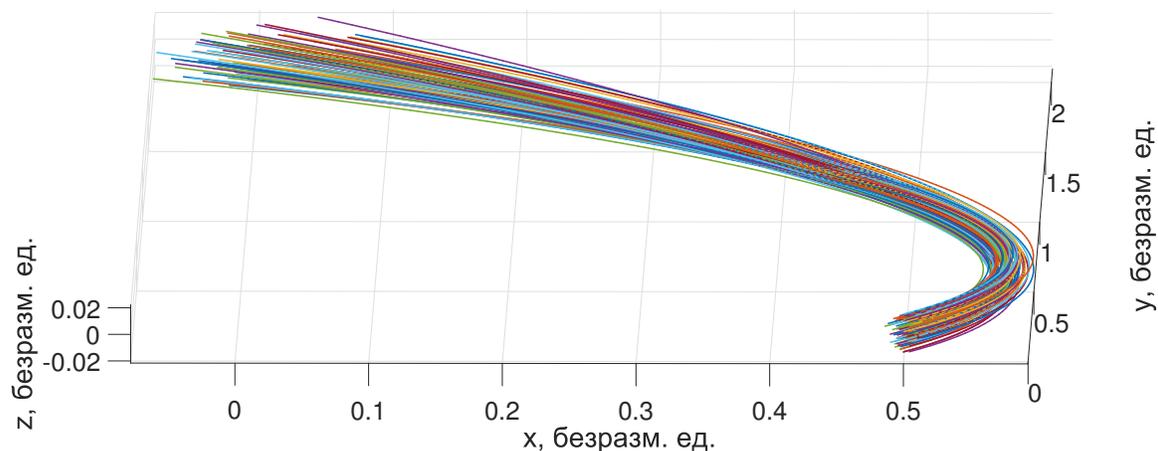


Рис. 3. Визуализация траекторий для обучающей выборки прогнозирования

смысла, поскольку информация о нем хранится в скорости, рассчитанной функцией *lambert.m*. На рисунке 4 приведена визуализация обучающей выборки для задачи коррекции.

После того как обучающие выборки составлены, необходимо построить нейронные сети, способные их аппроксимировать. Обе нейронные сети должны решать задачу многомерной нелинейной регрессии, другими словами, для каждой выборки необходимо построить аппроксимирующее отображение. Таким образом, выбор архитектуры сети и функционалов ошибки значительно сужается. В 1957 году А.Н. Колмогоровым и В.И. Арнольдом была доказана теорема о представлении непрерывных функций нескольких переменных суперпозицией непрерывных функций одной переменной, которая в 1989 году была переформулирована и доказана Д. Цыбенко для нейронных сетей: любая функция нескольких переменных может быть представлена трехслойной ИНС с прямыми полными связями между входным и скрытым слоями с N нейронами входного слоя, не менее чем $2N + 1$ нейронами скрытого слоя с ограниченными функциями активации и нейронами выходного слоя с неизвестными функциями активации. Следовательно, архитектура сети будет задана следующим образом: полный граф с 3 слоями: входным, скрытым и выходным. Функционал ошибки выберем из физических соображений. Поскольку обе нейронные сети должны минимизировать евклидово расстояние, то в качестве функционала выберем

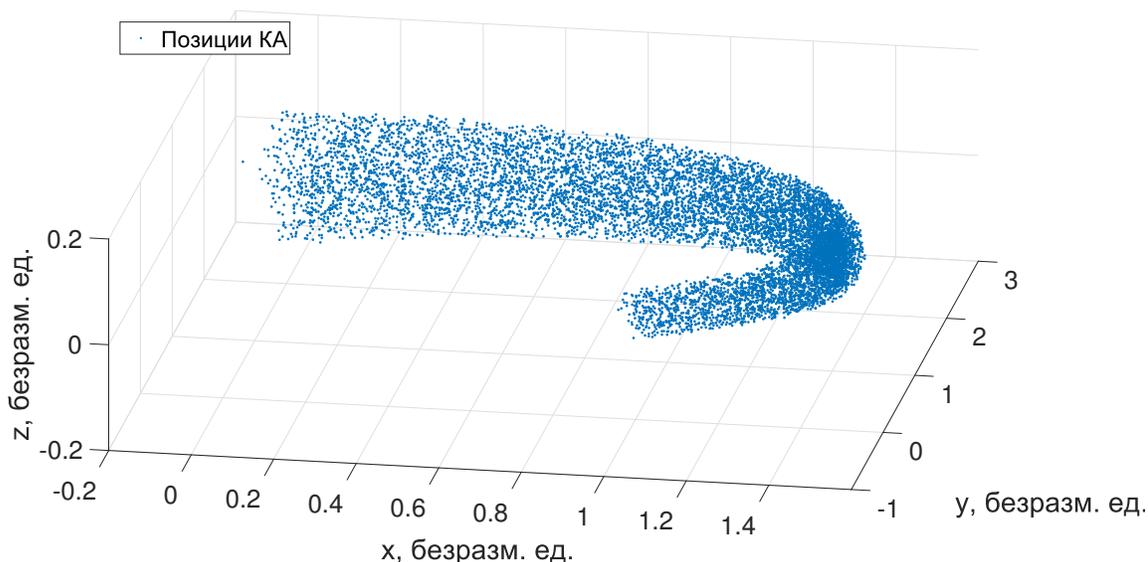


Рис. 4. Визуализация обучающей выборки для коррекции

среднеквадратичное отклонение (Mean Squared Error):

$$MSE = \frac{1}{N} \sum_{n=1}^N (y_n - \tilde{y}_n)^2,$$

где N — количество примеров в обучающей выборке, y_n — выходной обучающий пример, а \tilde{y}_n — желаемый отклик сети. Теперь определимся с активационными функциями. Для скрытого слоя в качестве нелинейной активационной функции выберем плотность нормального распределения. Она является дифференцируемой и монотонной, поэтому удовлетворяет определению активационной функции. Полученная нейронная сеть называется радиально-базисной нейронной сетью. Функционировать данная сеть будет в соответствии с данной формулой

$$f(\vec{x}|\vec{\theta}) = \omega_0 + \sum_{n=1}^M \omega_n \exp\left(-\frac{\|\vec{x} - \vec{\mu}_n\|^2}{2\sigma_n^2}\right),$$

где вектор параметров $\vec{\theta} = [\vec{\omega}, \vec{\mu}, \vec{\sigma}]$, а $\vec{\mu} = [\vec{\mu}_1, \dots, \vec{\mu}_M]$. Здесь M — это количество нейронов в скрытом слое. Инициализация параметров будет происходить следующим образом: коэффициенты линейной комбинации ω_n задаются из равномерного распределения с нулевым математическим ожиданием, а дисперсия полагается единичной. Вектора математического ожидания определим

из физического смысла следующим образом: разобьем аппроксимируемое множество — обучающую выборку — на кластеры. Под кластером подразумевается плотное скопление объектов выборки. Количество кластеров определяется количеством нейронов в скрытом слое сети. Для каждого кластера определим его центр масс с помощью метода *к-средних*. Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров. Таким образом, вектора математического ожидания будут выбраны как центры масс своих кластеров. Физический смысл заключается в том, что каждый нейрон будет аппроксимировать свой кластер плотностью нормального распределения с заданными параметрами. В итоге гиперповерхность, задаваемая выборкой, будет аппроксимирована суперпозицией плотностей нормального распределения. Подобная инициализация параметров является хорошим начальным приближением для градиентного метода, оптимизирующего эти параметры. В качестве градиентного метода будет выбран алгоритм Левенберга–Марквардта [10]. Подробно рассмотрим процесс оптимизации функции потерь. Распишем более подробно вектор параметров сети: $\vec{\theta} = [\vec{\omega}, \vec{\mu}, \vec{\sigma}]$. Его длина равна $M \cdot d + 2M + 1$, где d — размерность выборки, поскольку $\vec{\mu} = [\vec{\mu}_1, \dots, \vec{\mu}_M]$ — это вектор составленный из векторов математических ожиданий всех кластеров, $\vec{\omega} = [\omega_0, \omega_1, \dots, \omega_M]$, а $\vec{\sigma} = [\sigma_1, \sigma_2, \dots, \sigma_M]$. Запишем линейное приближение аппроксимирующей функции: $f(\vec{x}|\vec{\theta} + \Delta\vec{\theta}) - f(\vec{x}|\vec{\theta}) = J\Delta\vec{\theta}$, где J — якобиан функции $f(\vec{x}|\vec{\theta})$ в точке $\vec{\theta}$ размерности $(N \times M \cdot d + 2M + 1)$, N — объем обучающей выборки.

$$J = \begin{pmatrix} \frac{\partial f(\vec{x}_1, \vec{\theta})}{\partial \theta_1} & \dots & \frac{\partial f(\vec{x}_1, \vec{\theta})}{\partial \theta_{M \cdot d + 2M + 1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(\vec{x}_N, \vec{\theta})}{\partial \theta_1} & \dots & \frac{\partial f(\vec{x}_N, \vec{\theta})}{\partial \theta_{M \cdot d + 2M + 1}} \end{pmatrix}.$$

Для нахождения приращения $\Delta\vec{\theta}$ приравняем нулю вектор частных производных функционала ошибки по параметрам, для этого представим его следующим образом: $MSE(\vec{\theta}) = \|\vec{y} - f(\vec{\theta} + \Delta\vec{\theta})\|^2$, где $\vec{y} = [y_1, \dots, y_N]^T$, а $f(\vec{\theta} + \Delta\vec{\theta}) = [f(x_1|\vec{\theta} + \Delta\vec{\theta}), \dots, f(x_N|\vec{\theta} + \Delta\vec{\theta})]^T$. Преобразуем функционал: $\|\vec{y} - f(\vec{\theta} + \Delta\vec{\theta})\|^2 = (\vec{y} - f(\vec{\theta} + \Delta\vec{\theta}))^T (\vec{y} - f(\vec{\theta} + \Delta\vec{\theta})) = \vec{y}^T \vec{y} - 2\vec{y}^T f(\vec{\theta} + \Delta\vec{\theta}) + f^T(\vec{\theta} + \Delta\vec{\theta}) f(\vec{\theta} + \Delta\vec{\theta})$. Дифференцируя полученное выражение по параметрам, получим

$$\frac{\partial MSE(\vec{\theta})}{\partial \vec{\theta}} = (J^T J)\Delta\vec{\theta} - J^T(\vec{y} - f(\vec{\theta})) = 0.$$

Таким образом, $\Delta\vec{\theta} = (J^T J)^{-1} J^T(\vec{y} - f(\vec{\theta}))$. Матрица $J^T J$ может оказаться

вырожденной, поэтому вводят параметр регуляризации $\lambda \geq 0$

$$\Delta \vec{\theta} = (J^T J + \lambda E)^{-1} J^T (\vec{y} - f(\vec{\theta})),$$

где E — единичная матрица подходящей размерности. Полученное приращение параметра $\vec{\theta}$ есть приращение для всех весов в нейронной сети.

4. Моделирование

Перед тем как приступить к моделированию решения, сделаем несколько замечаний, касающихся обучающей выборки для задачи коррекции. Во-первых, траектория, используемая для составления выборки, не является непрерывной, а состоит из дискретного набора точек. Нужно учитывать данный факт при инициализации позиций КА, иначе могут получиться незаполненные области в трубке. Поскольку рассматриваются эллиптические траектории, а интегрирование происходит с постоянным шагом, то точки проинтегрированной траектории будут находиться на различных расстояниях друг от друга. При составлении выборки невозмущенная траектория помещалась в трубку постоянного радиуса δ , после чего в ней равномерно распределялись позиции КА, но в реальных условиях распределение позиций КА происходит в сферах радиуса δ с центрами в точках, образующих траекторию, поэтому в случае отстояния каких-то точек траектории на расстояние, большее 2δ , область между ними останется незаполненной.

Следующее замечание затрагивает типы траекторий, поскольку не все траектории являются одинаковыми с точки зрения аппроксимации нейронными сетями. Есть два таких типа — плоская траектория, у которой не происходит изменения значений по одной из координатных осей, и обычная траектория, где все компоненты меняются со временем. Принципиальное различие заключается в том, что в случае плоской траектории нейронная сеть будет аппроксимировать нулевой координатный столбец, однако сделать этого не удастся, поскольку аппроксимация будет напрямую зависеть от силы и вида возмущения. Решение данной проблемы кроется в несколько измененном подходе к заполнению трубки, другими словами, распределение координат позиций КА должно быть индивидуальным для каждой оси. Например, в случае плоской траектории, лежащей в плоскости z , значения компоненты z позиций КА должны быть равномерно распределены на отрезке $[-10^{-4}, 10^{-4}]$ безразмерных единиц, иначе слабые возмущения не будут аппроксимироваться. Учитывая данные замечания, удастся получить аппроксимацию для любых типов траекторий.

Будем называть совокупность двух обученных нейронных сетей моделью. После того как были составлены выборки и обучены нейронные сети,

осталось валидировать полученные результаты. Для этого необходимо проверить корректность работы модели для различных траекторий при различных возмущениях, оценить количество коррекций в зависимости от силы возмущения, оценить величины норм суммарного и среднего импульсов, затраченных на коррекцию, а также проверить устойчивость полученного решения к начальному приближению.

Для каждой траектории, задаваемой своими начальными условиями, необходимо обучать свою модель. Начнем с тестирования работы набора моделей для различных траекторий и возмущений. Для этого произвольным образом составим набор из пяти начальных и конечных точек и обучим, соответственно, пять моделей. Количество нейронов и минимальный порог функционала ошибки определим для всех моделей одинаковыми. Положим $M = 64$ нейронов, а $MSE_{threshold} = 10^{-5}$. В качестве возмущений будем брать возмущения, не меняющиеся во времени, и переменные возмущения одного и того же порядка малости. Оценка работы модели будет заключаться в проверке точности совпадения концов невозмущенной траектории и траектории, полученной нейронной сетью, а также совпадении двух траекторий. Точность прогнозирования положим равной 100 метрам. Приведем таблицу 1, описывающую зависимость норм разностей конечных точек траекторий от постоянных возмущений. Все модели из таблицы тестировались на возмущениях одинаковых порядков, но константы при возмущениях для каждой модели полагались разными. Константы выбирались из равномерного распределения с нулевым математическим ожиданием с помощью фиксирования потока (seed) в генераторе случайных чисел.

		Постоянное возмущение								
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$	10^{-5}
Модель	1	0.01	4,01	1.67	1.55	1.52	1.49	1.48	1.47	1.46
	2	0.008	7.65	2.27	2.02	2.02	2.05	2.10	2.11	2.12
	3	0.006	9.02	1.53	1.30	1.18	0.87	0.74	0.64	0.57
	4	0.01	11.06	3.80	3.61	2.96	1.93	1.56	1.46	1.86
	5	0.004	6.09	2.95	1.82	1.39	1.26	0.91	0.89	0.81

Таблица 1. Нормы разностей концов траекторий

Первая модель аппроксимировала задачу попадания из $R_1 = [1, 0, 0]$ в $R_2 = [0, 2, 0]$, вторая — из $R_1 = [1, 0, 1]$ в $R_2 = [0, 2, 0]$, третья — из $R_1 = [0, 1, 0]$ в $R_2 = [2, 0, 1]$, четвертая — из $R_1 = [0.77, 0.02, 0.63]$ в $R_2 = [1.49, 0.99, 0.44]$ и пятая — из $R_1 = [0.84, 0.17, 0.05]$ в $R_2 = [0.72, 0.55, 1.06]$. Данный набор векторов,

задающий свои траектории, покрывает все их типы — плоские и обычные. По значениям из таблицы 1 построим рисунок 5, на котором изображена зависимость точности попадания КА в заданную цель от силы возмущения для всех моделей.

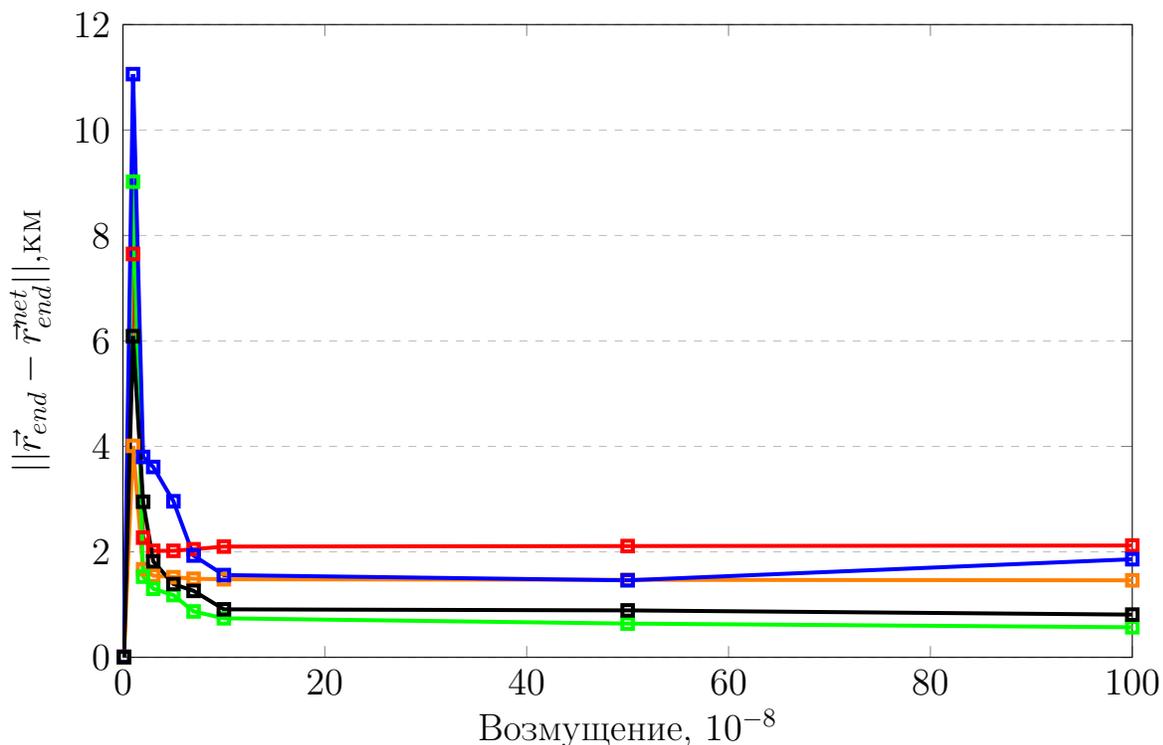


Рис. 5. Зависимость точности попадания КА в цель от силы возмущения

Данный пик можно объяснить тем, что при возмущениях порядка 10^{-7} количество коррекций не превышает пяти, это видно в таблице 2, следовательно, можно сделать вывод, что нескольких коррекций не хватает для того, чтобы добиться такой же точности, как и в случае многих коррекций. Возникает вопрос: что нужно сделать, чтобы нескольких коррекций хватало для того, чтобы обеспечить такую же точность попадания в цель, как и в случае многих коррекций? Можно сделать предположение, что при увеличении числа нейронов в нейронной сети точность попадания для нескольких коррекций увеличится. Проверим данное предположение после анализа моделей на различных возмущениях.

Построим таблицу 2, описывающую количество коррекций для приведенных моделей и возмущений.

		Постоянное возмущение								
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$	10^{-5}
Модель	1	0	4	26	37	44	51	57	72	81
	2	0	1	6	17	30	37	41	67	75
	3	0	1	8	20	34	38	43	69	76
	4	0	1	20	32	40	46	53	74	80
	5	0	5	16	25	31	38	43	70	76

Таблица 2. Количество коррекций

По таблице 2 построим рисунок 6, на котором отложим зависимости количества коррекций от силы возмущения.

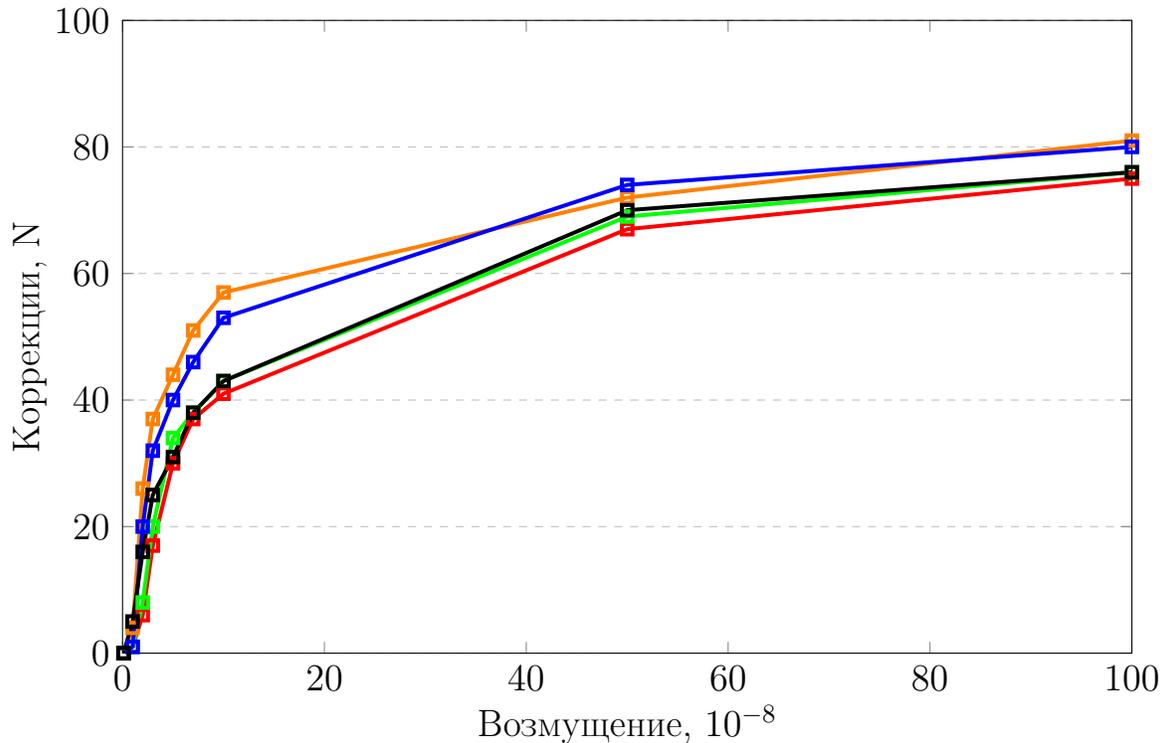


Рис. 6. Зависимость количества коррекций от силы возмущения

Количество коррекций растет при увеличении силы возмущения. Теперь проверим работу тех же моделей, но уже для переменных возмущений. Для этого построим таблицу 3, описывающую зависимость норм разностей концов траекторий от силы возмущения. В этот раз константы при возмущениях также будем выбирать из равномерного распределения с нулевым математическим ожиданием, но без фиксирования потока (seed) в генераторе случайных чисел.

		Переменное возмущение								
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$	10^{-5}
Модель	1	0.008	3.42	2.10	1.68	1.55	1.52	1.50	1.46	1.46
	2	0.009	11.41	2.91	2.01	2.02	2.04	2.09	2.11	2.12
	3	0.009	8.45	1.36	0.61	1.19	1.30	1.28	0.57	0.91
	4	0.009	5.30	3.23	3.26	2.84	2.42	2.03	2.00	2.00
	5	0.005	5.79	2.90	1.14	0.81	0.91	1.24	1.02	0.82

Таблица 3. Нормы разностей концов траекторий

Аналогичным образом изобразим на рисунке 7 результат работы моделей в зависимости от силы возмущения. Модели в таблице представлены в том же порядке, что и в таблице для постоянных возмущений.

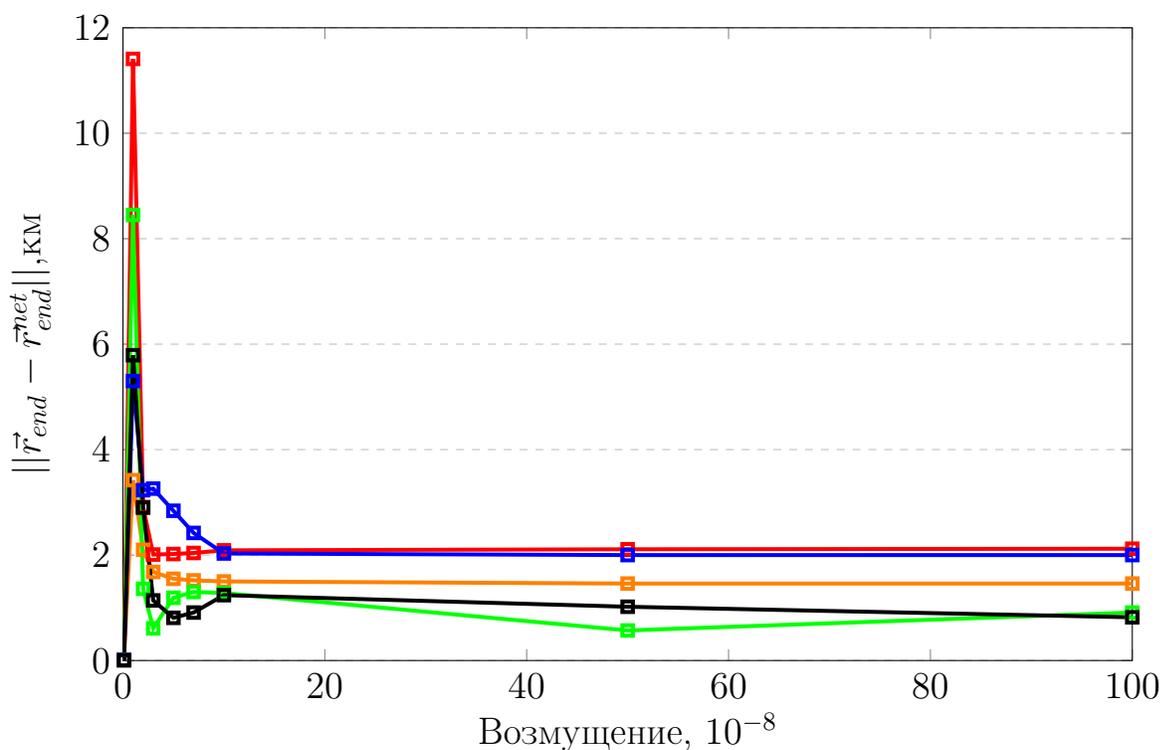


Рис. 7. Зависимость точности попадания КА в цель от силы возмущения

Сравнение результатов работы моделей при переменных и постоянных возмущениях показывает незначительное изменение соответствующих значений точностей, а значит, модели способны с некоторой точностью парировать любой из рассмотренных типов возмущения. Теперь построим таблицу 4, описывающую количество коррекций для переменных возмущений.

		Переменное возмущение								
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$	10^{-5}
Модель	1	0	5	13	25	37	41	48	72	78
	2	0	2	16	28	38	42	49	72	78
	3	0	2	17	28	38	42	50	72	78
	4	0	2	15	26	37	41	49	72	78
	5	0	2	17	28	38	42	49	72	78

Таблица 4. Количество коррекций

В таблице примечательно то, что для всех моделей количество коррекций для одних и тех порядков возмущений стало примерно одинаковым. Приведем рисунок 8, на котором изобразим зависимость количества коррекций от силы возмущения.

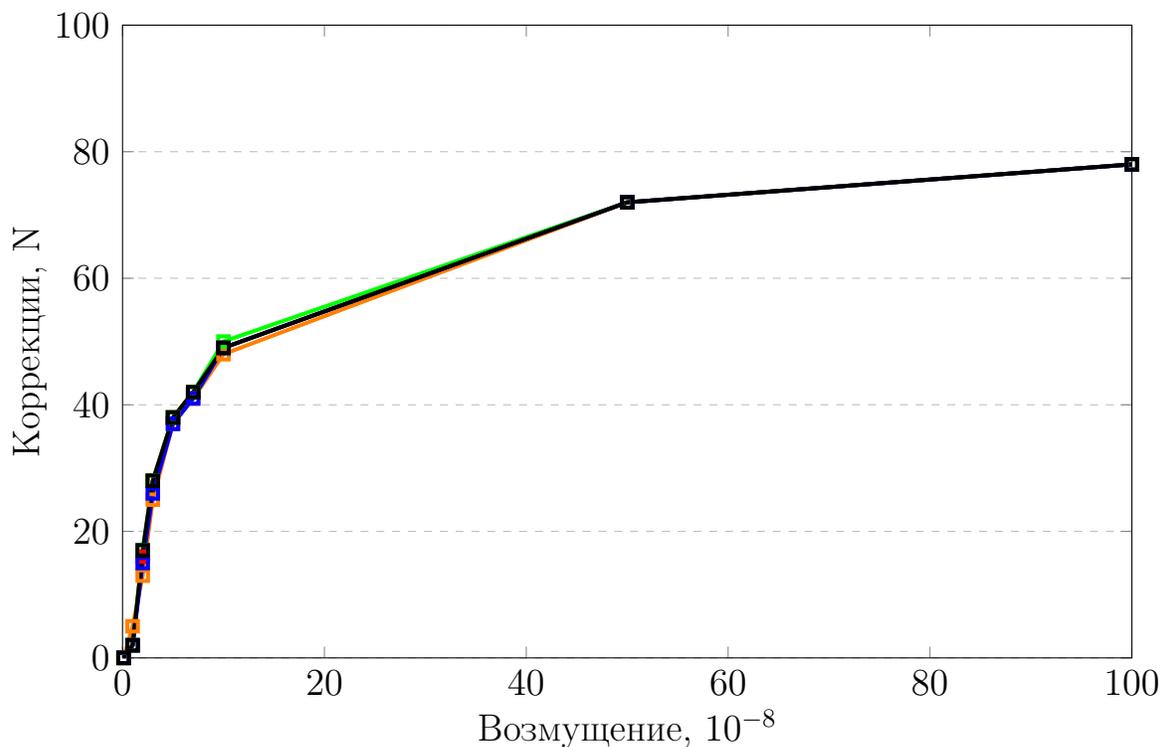


Рис. 8. Зависимость количества коррекций от силы возмущения

Действительно, кривые коррекций для всех моделей совпадают. Это объясняется тем, что случайные возмущения не имеют постоянного направления, поэтому возмущенная траектория будет колебаться относительно невозмущенной, таким образом, количество раз, когда прогнозирующая сеть предскажет выход конечной точки траектории из допустимого шара, будет примерно одинаковым.

Теперь оценим суммарные $\Delta v = \sum_{i=1}^N \|\vec{v}_i\|$ и средние $\Delta v_{avg} = \frac{1}{N} \sum_{i=1}^N \|\vec{v}_i\|$ затраты характеристической скорости, используемые для коррекции. Предыдущий эксперимент показал схожие результаты для постоянных и переменных возмущений, поэтому для оценки выберем любые из перечисленных возмущений, к примеру, переменные. Составим таблицу 5, описывающую зависимость суммарных затрат характеристической скорости для всех моделей, и таблицу 6, описывающую зависимость средних затрат характеристической скорости для всех моделей. По данным из таблицы 5 и 6 изобразим соответствующие рисунки 9 и 10. Приведенные скорости даны в размерных величинах м/с.

		Возмущение							
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$
Модель	1	0	102.82	58.09	86.38	101.48	102.03	120.98	179.27
	2	0	174.16	195.79	203.94	220.15	231.62	252.31	291.24
	3	0	128.02	105.95	107.47	135.44	166.09	183.72	229.08
	4	0	221.08	188.54	184.36	191.74	193.17	206.13	235.42
	5	0	239.05	189.47	208.15	238.68	257.19	275.67	322.45

Таблица 5. Суммарные затраты характеристической скорости

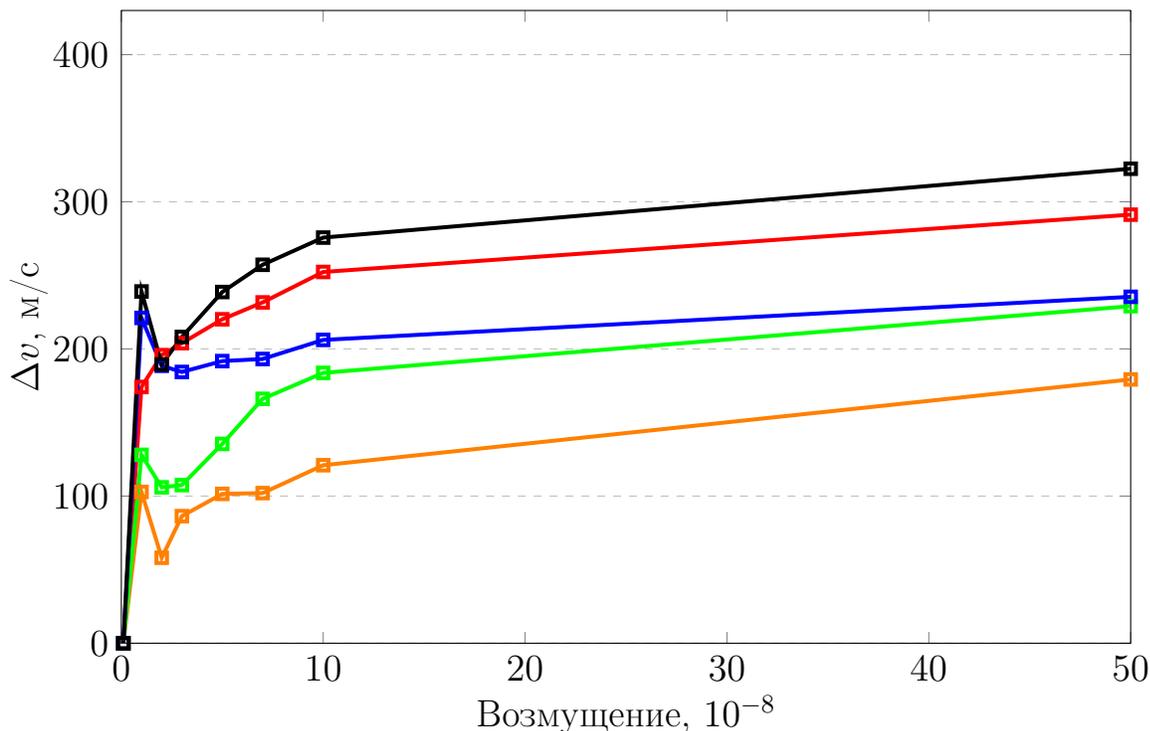


Рис. 9. Зависимость суммарных затрат хар. скорости от силы возмущения

		Возмущение							
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$
Модель	1	0	20.56	4.46	3.45	2.74	2.48	2.52	2.48
	2	0	21.77	12.23	7.28	5.79	5.51	5.14	5.03
	3	0	16.00	6.23	3.83	3.56	3.95	3.74	5.95
	4	0	31.58	12.56	6.82	5.18	4.71	4.20	4.65
	5	0	23.90	11.14	7.43	6.28	6.12	5.62	5.86

Таблица 6. Средние затраты характеристической скорости

С ростом возмущающей силы растет и затраченный импульс. Данная зависимость схожа с зависимостью количества коррекций от силы возмущения. Это естественный результат, поскольку суммарные затраты характеристической скорости будут пропорциональны количеству импульсных коррекций.

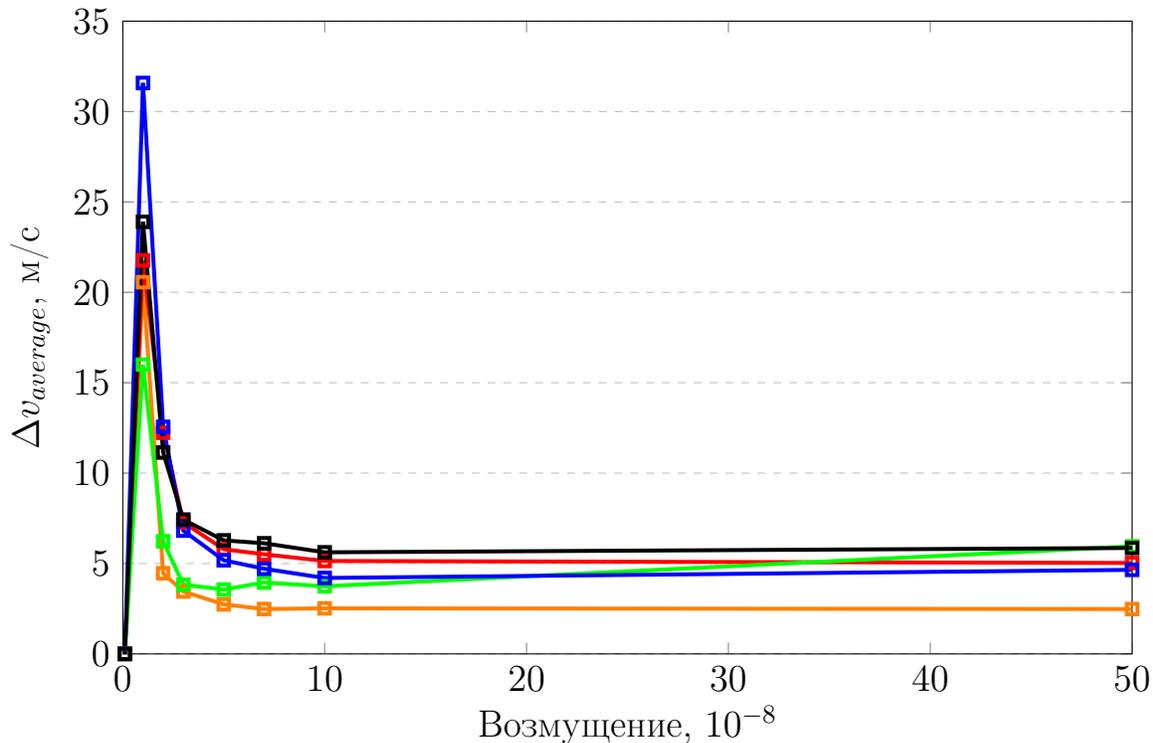


Рис. 10. Зависимость средних затрат хар. скорости от силы возмущения

В области малых возмущений порядка 10^{-7} безразмерных единиц для суммарных и средних затрат характеристической скорости регистрируются скачки, однако их природа принципиально разная. Скачок для средних затрат объясняется тем, что для возмущений этого порядка нейронная сеть делает всего несколько, но больших по величине импульсов. Однако особенность скачка сум-

марных затрат может объясняться предположением, сделанным ранее о количестве нейронов в сети. При большом количестве коррекций точность отдельной коррекции не определяет итоговый результат, но в случае малого их числа ситуация обратная. Таким образом, необходимо проверить влияние количества нейронов на точность отдельных коррекций. Для этого построим пять моделей для одной и той же обучающей выборки, задаваемой, к примеру, векторами $R_1 = [1, 0, 1]$ и $R_2 = [0, 2, 0]$. Число нейронов для моделей будет следующим: $M = 16, 32, 64, 128, 256$. Если скачок зависимости точности попадания КА в цель, а также скачок зависимости суммарных затрат характеристической скорости от возмущений будет сглаживаться при увеличении количества нейронов, то предположение окажется верным. Составим таблицу 7 для зависимости $\|\vec{r}_{end} - \vec{r}_{end}^{net}\|$ от силы возмущения и M . Изобразим рисунок 11 соответствующей зависимости.

		Возмущение								
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$	10^{-5}
Нейроны	16	0	5.63	2.17	2.77	3.07	3.18	3.25	3.25	3.28
	32	0	6.14	2.85	2.77	2.91	2.99	3.06	3.22	3.31
	64	0	5.70	1.77	1.71	1.71	1.72	1.73	1.77	1.78
	128	0	3.04	2.45	2.38	2.35	2.35	2.34	2.35	2.35
	256	0	2.08	2.04	1.96	1.94	1.94	1.93	1.93	1.93

Таблица 7. Нормы разностей концов траекторий

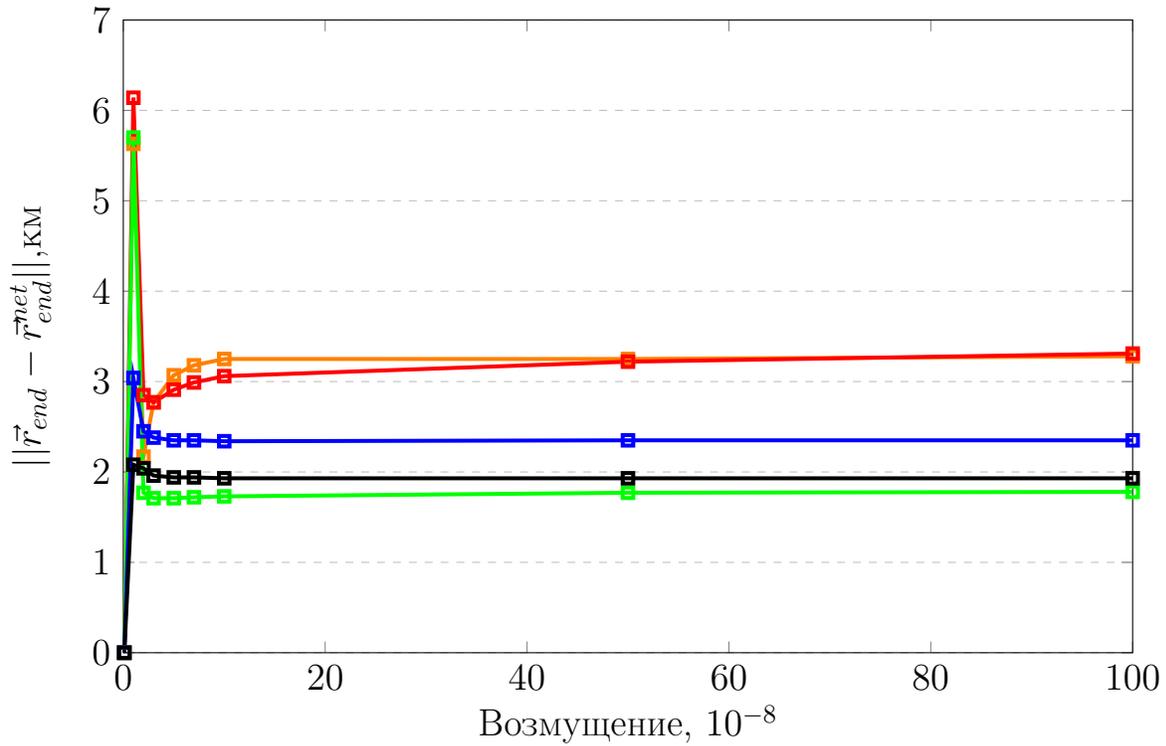


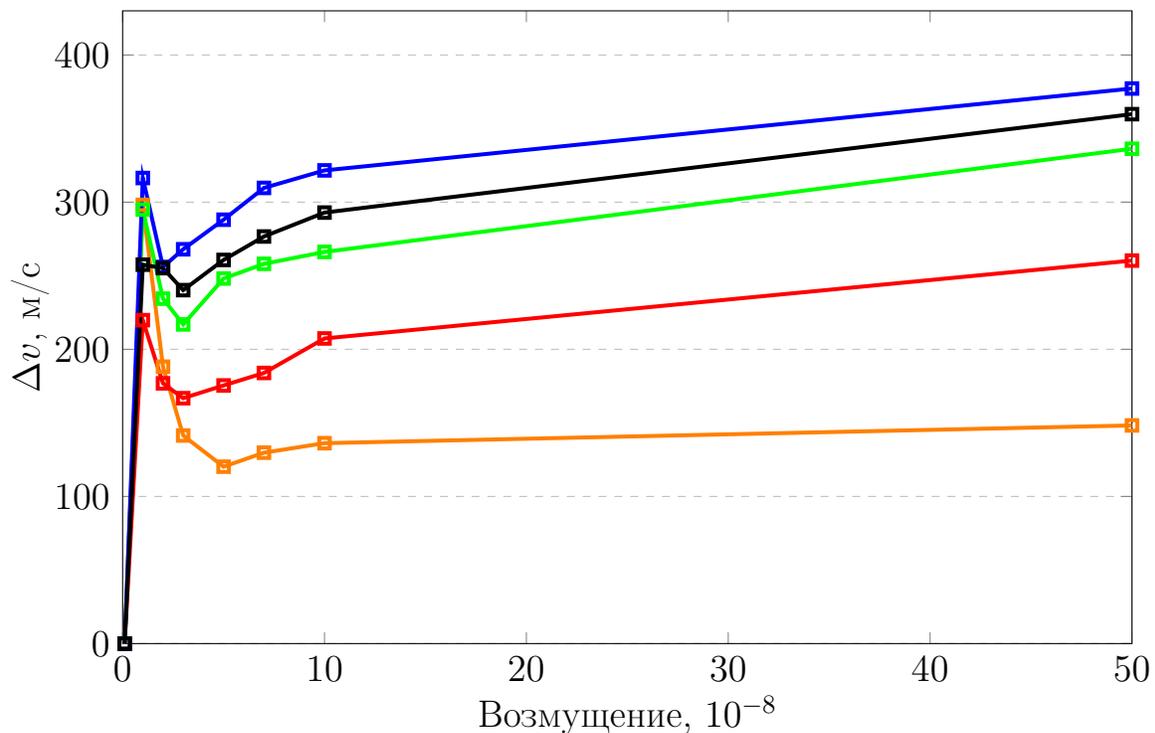
Рис. 11. Зависимость точности попадания КА в цель от силы возмущения и M

— $M = 16$; — $M = 32$; — $M = 64$; — $M = 128$; — $M = 256$;

По рисунку 11 можно понять две следующие вещи: во-первых, предположение было верным, поскольку величина скачка уменьшается при увеличении количества нейронов. Черная линия, отвечающая $M = 256$ нейронам, практически его не имеет. Во-вторых, точность попадания КА в цель не зависит от количества нейронов. Составим таблицу 8, описывающую зависимость суммарных затрат характеристической скорости от силы возмущения и количества нейронов в модели. По ней построим соответствующую зависимость и изобразим ее на рисунке 12.

		Возмущение							
		10^{-8}	10^{-7}	$2 \cdot 10^{-7}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$7 \cdot 10^{-7}$	10^{-6}	$5 \cdot 10^{-6}$
Нейроны	16	0	298.14	188.17	141.41	120.21	129.74	136.20	148.25
	32	0	219.79	176.87	166.78	175.43	183.92	207.37	260.28
	64	0	294.96	234.42	216.90	248.11	258.07	266.16	336.29
	128	0	316.45	255.95	267.98	288.06	309.67	321.58	377.22
	256	0	257.50	255.16	240.21	260.72	276.65	292.91	359.79

Таблица 8. Суммарные затраты характеристической скорости

Рис. 12. Зависимость суммарных затрат хар. скорости от силы возмущения и M

—■— $M = 16$; —■— $M = 32$; —■— $M = 64$; —■— $M = 128$; —■— $M = 256$;

В случае суммарных затрат ситуация аналогичная: кривые выравниваются при увеличении количества нейронов. Таким образом, экспериментально был показан факт влияния количества нейронов на точность отдельных коррекций.

Осталось проверить устойчивость решения: способность нейронных сетей аппроксимировать траектории, начальные точки которых не совпадают с точкой из постановки. Для этого инициализируем набор начальных условий для радиус-векторов в шаре радиуса ε_1 с центром в точке, задаваемой

начальным радиус-вектором. Начальные значения для векторов скоростей инициализируем в шаре радиуса ε_2 с центром в точке, задаваемой начальным вектором скорости. За ε_1 возьмем значение, в 10 раз превышающее значение, используемое при составлении обучающей выборки. За ε_2 тоже возьмем значение, используемое при составлении обучающей выборки, но увеличивать его не будем. В качестве проверочного функционала выберем опять норму разности конечных радиус-векторов. Проверять устойчивость будем на примере траектории с началом в $R_1 = [1, 0, 0]$ и концом в $R_2 = [0, 2, 0]$. Такой выбор обусловлен показательностью результата аппроксимации траектории нейронной сетью, поскольку данная траектория является плоской. Приведем таблицу 9, состоящую из норм разностей начальных и конечных векторов. Для каждого примера будем произвольно выбирать возмущение из диапазона значений $10^{-5} - 10^{-7}$ безразмерных единиц. В таблице 9 приведено десять значений норм разностей начальных и конечных векторов, но на практике было проделано порядка сотни экспериментов. Во всех случаях КА попадал в 10 километровую окрестность конечной точки. Учитывая этот результат, можно утверждать, что решение является устойчивым. На рисунке 13 изображена совокупность из невозмущенной траектории, аппроксимации нейронной сетью и возмущенной траектории.

$\ \vec{R}_1 - \vec{R}'_1\ $	60.03	77.59	80.72	23.03	58.12	51.02	16.53	95.35	87.34	97.07
$\ \vec{R}_2 - \vec{R}'_2\ $	2.20	1.87	2.55	1.66	2.52	6.72	1.68	3.82	3.18	5.74

Таблица 9. Зависимость норм разностей конечных векторов от начальных

Моделирование проводилось на персональном компьютере со следующими характеристиками: процессор Intel(R) Core I5-2520M с тактовой частотой 2.5 GHz, объем оперативной памяти 4 Gb. Вычисления проводились в среде MATLAB, версия R2017a.

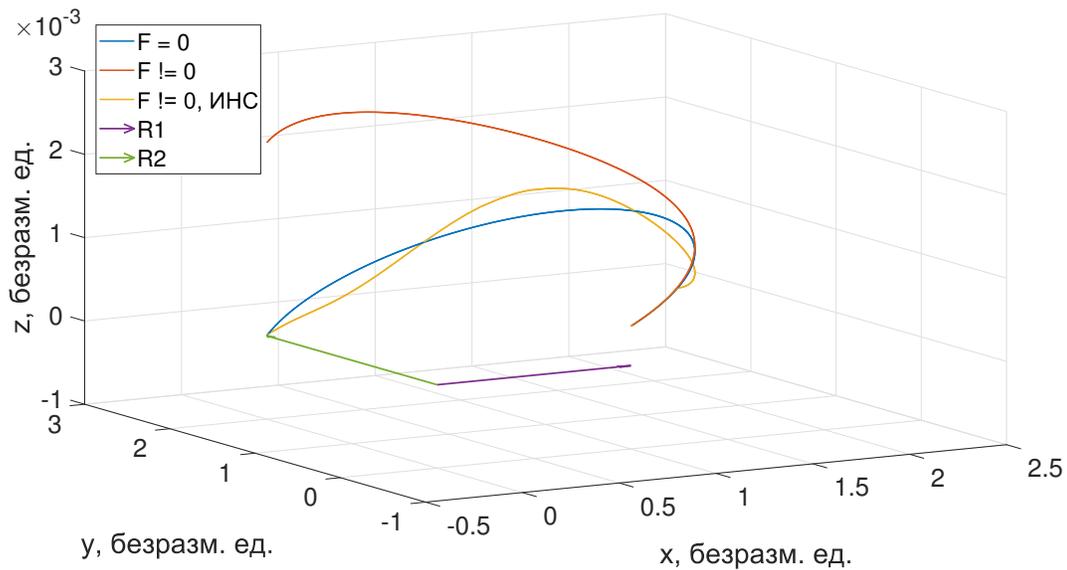


Рис. 13. Проверка устойчивости на примере плоской траектории

Заключение

Для задачи перелета между двумя точками разработаны две нейронные сети, отвечающие за коррекцию и прогноз орбитального движения аппарата в рамках возмущенной задачи двух тел. Исследования показали, что увеличение возмущающей силы приводит к возрастанию количества необходимых коррекций. В случаях, когда возмущающая сила достигает порядка 10^{-6} безразмерных единиц, рассматриваемая схема оказывается неэффективной с точки зрения количества коррекций, поэтому в таких случаях рекомендуются традиционные способы коррекции траектории. Тем не менее сети оказались способными решать задачу в случаях, когда начальное условие КА находится в некоторой окрестности исходной точки. Точность производимых коррекций увеличивается при увеличении нейронов в нейронных сетях. Затраты на коррекцию возрастают с ростом возмущающей силы и находятся в пределах от 50 до 400 м/с.

Список литературы

- [1] *Pérez D., Bevilacqua R.* Neural Network based calibration of atmospheric density models // *Acta Astronautica* 2015, V.110 (Supplement C), P. 58-76.
- [2] *Pérez C., David A.* Adaptive Lyapunov control and artificial neural networks for spacecraft relative maneuvering using atmospheric differential drag // Ph.D. Thesis. Rensselaer Polytechnic Institute. 2013.
- [3] *Dachwald B.* Low-thrust trajectory optimization and interplanetary mission analysis using evolutionary neurocontrol // Ph.D. Thesis. Bundeswehr University Munich, Munich, Germany. 2004.
- [4] *Dachwald B.* Optimization of very-low-thrust trajectories using evolutionary neurocontrol // *Acta Astronautica*. 2005. V. 57(2), P.175-185.
- [5] *Ohndorf A.* Optimization of low-thrust earth-moon transfers using evolutionary neurocontrol // *IEEE Congress on Evolutionary Computation*. 2009.
- [6] *Gurfil P.* Adaptive neural control of deep-space formation flying // *Journal of Guidance, Control, and Dynamics*. 2002, V. 26(3), P. 491–501.
- [7] *Haykin S.* *Neural Networks: A Comprehensive Foundation* (2nd Edition) // McMaster University. Hamilton, Ontario, Canada, 2006.
- [8] *Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting // *Journal of Machine Learning Research*. 2014, Toronto, Ontario, Canada
- [9] *Dario I.* Revisiting Lambert’s Problem // *Journal of Guidance, Control, and Dynamics*. Vol. 29, No. 5 (2006), pp. 1242-1245.
- [10] *Levenberg K.* A Method for the Solution of Certain Problems in Last Squares // *Quart. Appl. Math.* 1944. Vol. 2. P. 164–168.

Оглавление

Введение	3
1. Постановка задачи	4
2. Введение в нейронные сети	6
3. Решение задачи	9
4. Моделирование	14
Заключение	26
Список литературы	27