Четырёхэлементная антенная решётка на базе SDR платформ с наносекундной синхронизацией

Н. В. Шеремет, Г. А. Фокин

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича

sidwavedata@gmail.com, grihafokin@gmail.com

Аннотация. **Данная работа посвящена проектированию системы четырехэлементной антенной решетки на основе программно-определяемого радио (SDR), сопутствующего разработанного программного обеспечения, ее настройке и оценке производительности. Для построения аппаратного комплекса SDR используются четыре платы NI USRP 2932. Для синхронизации применяются тактовые сигналы 1PPS и 10 МГц, источниками которых являются сервер точного времени Метроном-PTP-1U и транслятор сигналов Метроном-Т. Предложен подход и его программная реализация для конфигурирования и синхронизации плат SDR в среде USRP Hardware Driver (UHD), позволяющие управлять выравниванием временных меток и фаз нескольких плат SDR и учитывающая присущую собственным тактовым генераторам плат временную и фазовую нестабильность. Представленный подход обеспечивает временную и частотную синхронизацию с предсказуемыми и компенсируемыми фазовыми сдвигами между радиочастотными каналами нескольких SDR-плат. Проведена оценка наносекундной временной и частотной синхронизации в соответствии с предложенным подходом.**

Ключевые слова: антенная решётка, USRP, синхронизация, Python API UHD, Метроном-PTP-1U, Метроном-Т

I. Введение

Множественные входы и множественные выходы (MIMO) сегодня лежат в основе многих технологий мобильной связи [1]. MIMO обеспечивает увеличение пропускной способности, спектральной эффективности и мощности сигнала в таких технологиях, как Wi-Fi, LTE и 5G. Многоантенные системы позволяют использовать пространственное мультиплексирование для увеличения скорости передачи, разнесённые приём и передачу для улучшения соотношения сигнал/шум, а также алгоритмы оценки угла прихода и адаптивного формирования луча для позиционирования и пространственного разделения сигналов **Ошибка! Источник ссылки не найден.**.

Создание MIMO-систем с нуля – сложная и ресурсоемкая задача для разработчиков. Прототипирование и оценка многоантенных систем и их алгоритмов могут быть выполнены эффективно и быстрее с помощью программно-определяемого радио (SDR) [3]-[5]. Одна плата SDR может иметь несколько радиочастотных (РЧ) каналов, но при построении систем с высоким порядком MIMO, большим количеством антенн, рано или поздно возникает проблема масштабирования. Объединение нескольких плат в одно виртуальное устройство требует синхронизации их радиочастотных каналов как по времени, так и по частоте. Решение для синхронизации SDR-платформ, предложенное авторами в предыдущей работе [6], иллюстрирует подход к проектированию единого виртуального многоканального устройства.

Анализ литературы по проблеме разработки MIMO-систем с использованием SDR-платформ показывает, что задача построения макета четырехканальной системы антенных решеток на базе SDR-плат является весьма актуальной. В работе [7] рассматривается макет многоантенной системы на платформе Universal Software Radio Peripheral (USRP) NI 2953R [8]. Предложенная система подходит для записи реального LTE-сигнала, что позволяет качественно анализировать и изучать эфирные сигналы. Авторы предлагают использовать круговую антенную решетку, элементы которой равноудалены от передатчика, расположенного в центре, что позволяет устранить фазовый сдвиг. В работе [9] исследуется тема алгоритмов выбора антенны для массивного MIMO. Разработан прототип системы, в которой реализован алгоритм выбора антенны на основе SDR, позволяющий увеличить пропускную способность канала на 15 % по сравнению с обычной системой MIMO. В работе [10] описан тестовый стенд системы адаптивного формирования луча на базе SDR-платформы USRP N210 [11]. Рассматриваемая задача оценки угла прихода сигнала решается несколькими методами, использующими в качестве входных данных уровень принимаемого сигнала (RSS) луча. Приведено сравнение алгоритмов машинного обучения: Q-learning (quality-learning), double Q-learning и deep Q-learning.

В данной работе демонстрируется макет четырехканальной системы антенных решеток на основе четырех синхронизированных плат SDR. Раздел II посвящен описанию аппаратного обеспечения, используемого в системе, и построению макета. В разделе III описано программное обеспечение и подход к синхронизации плат SDR. В разделе IV демонстрируется процедура оценки степени синхронизации и результат временной и фазовой синхронизации радиочастотных каналов. Наконец, в разделе V сформулированы выводы.

II. Аппаратная часть макета

Предлагаемый макет основан на четырех SDR-платах NI USRP 2932 [12], основные характеристики которых [13] приведены в табл. 1. Каждая плата поддерживает частотный дуплекс и имеет радиочастотный канал Tx/Rx, который используется для приема, когда передача

не ведется, и канал Rx, который используется для приема, когда передача ведется.

ТАБЛИЦА I. ХАРАКТЕРИСТИКИ ПЛАТЫ NI 2932

Характеристика	Значение
Макс. частота дискретизации АЦП	100 миллионов отсчётов в секунду
Макс. частота дискретизации ЦАП	400 миллионов отсчётов в секунду
Разрядность АЦП	14 бит
Разрядность ЦАП	16 бит
Частотный диапазон	400 МГц – 4,4 ГГц

NI USRP 2932 подключается к хосту (который является ПК) с помощью интерфейса Gigabit Ethernet и использует в качестве идентификаторов адреса IPv4. Для корректной работы все порты Ethernet на хосте должны иметь разные третьи октеты адресов, чтобы сформировать отдельные подсети с подключенными платами. Плате с адресом 192.168.10.101 должен соответствовать порт 192.168.10.100, в то время как другие устройства и порты не могут занимать адрес 192.168.10. [...].

Для согласованной работы платы USRP должны быть синхронизированы. NI USRP 2932 поддерживает несколько методов синхронизации: MIMO-кабель, генератор с опорным сигналом глобальной системы позиционирования (GPSDO) и 1PPS/10MHz [14]. При использовании MIMO-кабеля только две платы могут быть соединены по принципу «ведущий-ведомый», что недостаточно для построения четырехканальной сетки. Использование синхронизации с GPSDO [15] требует, чтобы каждая плата принимала стабильный сигнал глобальной навигационной спутниковой системы с помощью антенны, расположенной на открытом пространстве. Синхронизация с использованием опорных сигналов 1PPS (1 импульс в секунду) и 10 МГц позволяет синхронизировать несколько плат (в том числе четыре, как в данном исследовании) и не требует привязки к сигналу GPS для работы. Прямоугольный сигнал 10 МГц используется для синхронизации устройств по частоте, заменяя сигнал от собственного генератора платы. Помимо согласования частот радиочастотных цепей, этот сигнал поддерживает внутреннее время устройств и определяет длительность операций потоков передачи/приема, обеспечивая синхронное выполнение одних и тех же операций на разных устройствах при условии, что они начинаются в одно и то же время. Для синхронизации времени используется прямоугольный сигнал 1PPS с частотой 1 Гц. Края его импульсов служат маркерами начала операций на устройствах.

Источниками 1PPS и 10 МГц могут служить генераторы тактовых сигналов, такие как OctoClock-G [16] и Метроном-PTP-1U-V [17], о которых пойдет речь далее. Сервер точного времени Метроном-PTP-1U-V2 поддерживает взаимодействие с аналогичными устройствами по модели «ведущий-ведомый», синхронизацию Ethernet-устройств, генерацию сигналов IRIG, 2,048 МГц, 1PPS и 10 МГц. Метроном-PTP-1U-V2 может генерировать сигналы 1PPS с длительностью импульса 1 мс и 270 мс, что позволяет использовать более широкий спектр платформ SDR. Сервер времени имеет по одному выходу для сигналов 1PPS и 10 МГц, поэтому для подачи сигналов на несколько SDR-плат используется транслятор Метроном-T [18], который распределяет сигнал от источника на четыре SDR-платы.

Подключение к Метроном-PTP-1U-V2 осуществляется через Ethernet-соединение и требуется только для настройки. Настроить Метроном-T можно только с помощью элементов управления на передней панели.

При подключении сервера точного времени к транслятору сигнал 1PPS подается на общий разъем A IN, 10 МГц – на разъем B IN. Платы получают опорные сигналы с соответствующих портов A OUT 1-4 и B OUT 1-4. Чтобы активировать этот режим работы портов транслятора, нужно перевести функциональные язычки на передней панели в положения, соответствующие таблице 2, где «0» – верхний язычок, а «1» – нижний.

ТАБЛИЦА II. КОНФИГУРАЦИЯ МЕТРОНОМ-T

Функция	Логический ввод					
	2M	A	B	SW	AIN FAULT	BIN FAULT
AOUT=AIN BOUT=BIN	1	0	0	1	0	0

Для обеспечения корректной работы системы необходимо использовать кабели одинаковой длины при подключении сервера точного времени Метроном-PTP-1U-V2 к транслятору Метроном-T, при подключении плат к выходам транслятора, а также между антенными портами и конечными устройствами, которыми могут быть как антенны, так и приемные порты других плат. Использование кабелей разной длины в этих случаях может привести к нарушению синхронизации. Схема разводки показана на рис. 1. Общий вид макета представлен на рис. 2.

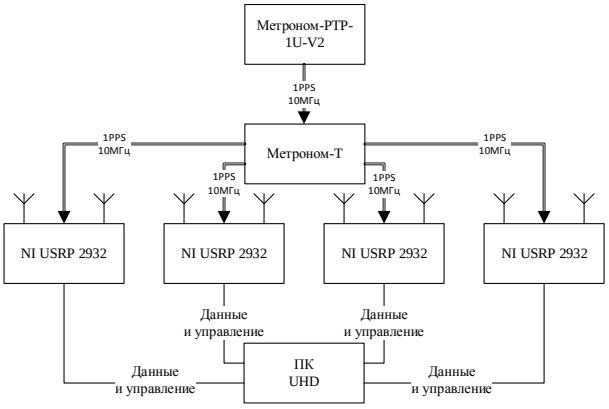

Рис. 1. Схема макета

Рис. 2. Основные элементы макета: платы, сервер точного времени и транслятор

III. Программмная часть макета

После подачи опорных сигналов на платы устройства должны быть настроены и синхронизированы программно. Далее будет рассмотрен интерфейс прикладного программирования USRP Hardware Driver Python [19], [20]. Для работы с SDR-платами в UHD используется объект MultiUSRP, взаимодействуя с которым можно получить доступ к устройству, отправлять команды, применять параметры и выполнять операции передачи/приема. MultiUSRP может автоматически находить подключенную плату без каких-либо аргументов, но только одну. Несколько плат USRP с интерфейсом Ethernet, например NI USRP 2932, можно объединить в один объект MultiUSRP, передав набор их адресов в качестве одного аргумента с индексами устройств.

Инициализация устройств в UHD Python API происходит следующим образом:

```
usrp = uhd.usrp.MultiUSRP('addr0=192.168.11.101, addr1=192.168.12.102,addr2=192.168.13.103, addr3=192.168.14.104')
```

Далее описывается принцип синхронизации в UHD. Во-первых, для каждого объявленного объекта (одного, если устройства Ethernet объединены) определяются источники опорного сигнала. Во-вторых, каждому объекту дается команда обнулить свой тактовый генератор по следующему фронту сигнала 1PPS, после чего следует секундная пауза для захвата фронта. В-третьих, флаг готовности к запуску `stream_cmd.stream_now` устанавливается в значение *False*, что необходимо для того, чтобы можно было ввести задержку перед началом приема/передачи и для работы с несколькими каналами. В-четвертых, вводится задержка перед стартом и передается объектам потока. Шаги, описанные в UHD Python API, выглядят следующим образом:

```
usrp.set_clock_source("external")

usrp.set_time_source("external")

time.sleep(0.1) # delay to stabilize oscillator

usrp.set_time_next_pps(uhd.libpyuhd.types.time_spec(0.0)) # reset internal clock

time.sleep(1) # delay to catch 1PPS

stream_cmd.stream_now = False

stream_cmd.time_spec = usrp.get_time_now() + uhd.libpyuhd.types.time_spec(2)

streamer.issue_stream_cmd(stream_cmd);
```

Описанный подход позволяет использовать внешние тактовые сигналы для конфигурирования и управления платами SDR, но не гарантирует их синхронизацию и корректный прием или передачу. Для объединения нескольких плат этот алгоритм необходимо усовершенствовать.

Платы NI USRP 2932 при отсутствии заявленного внешнего источника сигнала 1PPS выдают неуточненную ошибку приемника/передатчика, поэтому проверка регистрации сигналов 1PPS и 10 МГц важна для раннего обнаружения неисправностей. Для сигнала 10 МГц достаточно убедиться в его наличии с помощью функции `get_mboard_sensor`. Сигнал 1PPS должен присутствовать на всех платах в одно и то же время, это можно проверить, получив внутреннее время платы с помощью функции `get_time_now`. Время, возвращаемое средой Python, не будет полностью совпадать даже для синхронизированных плат из-за относительно большого временного интервала между вызовами функций для разных плат в Python, достаточно совпадения в сотые доли секунды. Этот метод подходит для обнаружения нестабильных плат. Функция `get_time_synchronised` выполняет аналогичную оценку и возвращает значение bool, где *True* означает, что время на платах совпадает с точностью до тысячных долей секунды. Эта проверка необходима, так как позволяет более точно оценить состояние всей системы после установки времени.

Функция `set_time_next_pps` обнуляет внутреннее время по следующему фронту 1PPS, но она работает некорректно, если вызывается слишком близко к следующему фронту опорного сигнала, поэтому, не зная времени последнего импульса, может потребоваться несколько вызовов. Для синхронизации нескольких плат больше подходит функция `set_time_unknown_pps`. Этот метод последовательно ожидает сигнал 1PPS, чтобы зафиксировать его как последний, и устанавливает время на следующем импульсе. Он также выводит информацию о разнице во времени между нулевой платой и остальными. Процесс синхронизации с помощью этого метода занимает до двух секунд. Время последнего импульса можно получить с помощью `get_time_last_pps`, что позволяет реализовать предыдущую функцию вручную. Результат оценивается функцией `get_time_synchronised`, и, если расхождения во времени остаются, синхронизацию необходимо повторить. Причиной расхождений может быть как возможная нестабильность платы генератора, так и переходные процессы источника опорного сигнала в течение нескольких минут после его запуска, однако синхронизация, полученная этим методом, сохраняется в течение длительного времени, и временные метки плат не расходятся.

Полная процедура синхронизации нескольких плат в UHD Python API выглядит следующим образом:

```
usrp.set_clock_source(sync_source)

usrp.set_time_source(sync_source)

time.sleep(0.1) # delay to stabilise oscillator

clock_sync = [False] * boards_num

while not all(clock_sync):

   for boards in boards_idx:

clock_status = usrps.get_mboard_sensor('ref_locked', board).to_bool()

        clock_sync[board] = clock_status

sync = False

while not sync:
```

```
usrps.set_time_unknown_pps(uhd.libpyuhd.types.time_spec(0.0))
time.sleep(2)
sync = usrps.get_time_synchronised()
```

Во время приема функция `streamer.recv` может получить метаданные, включая параметр `time_spec`, который является временной меткой первого образца, полученного платой. При успешной синхронизации временные метки всех плат должны совпадать.

IV. Оценка временной и частотной синхронизации

Оценка синхронизации плат заключается в приеме всеми платами известного сигнала и определении корреляции с ним. Сигналом может быть любая сложная последовательность, для большей наглядности полученной корреляционной характеристики с целью получения явных корреляционных пиков в конце последовательности при передаче следует добавить последовательность незначащих символов (нулей) той же длины. Передача сигнала должна осуществляться по кабелям с использованием делителя сигналов для обеспечения одновременного поступления сигнала на приемные порты плат. В рассматриваемом примере одна из плат макета выступает в роли передатчика для обеспечения полного совпадения параметров устройства при передаче и приеме, остальные платы осуществляют прием. Платы передачи и приема подключены к двум разным хостам, но все они соединены с источником тактового сигнала через транслятор. Сигнал передается в циклическом режиме, а на приеме регистрируется интервал времени, в несколько раз превышающий ожидаемую последовательность, чтобы убедиться в наличии нескольких пиков. Схема системы для проверки синхронизации плат показана на рис. 3.

Корреляционная характеристика системы показана на рис. 4. Незначительные различия в характеристиках каналов заметны только в начале характеристики. Корреляционные пики разных каналов совпадают с эталонными, что позволяет судить о том, достигнута ли временная синхронизация. Точность временной синхронизации в данной методике оценки определяется как обратная величина частоты дискретизации сигнала. Используемая частота 10 МГц дает точность 100 нс. Оценка синхронизации должна производиться не только во временной области. Для оценки частотной синхронизации необходимо сравнить фазовые сдвиги каналов относительно друг друга. Для начала необходимо определить фазовые углы между записанными выборками каждого канала и нулем, что в среде Python можно сделать с помощью библиотеки NumPy [21] следующим образом:

```
phases = np.angle(channel0.conj() * channel)
```

Во-вторых, полученные значения необходимо сгладить для удаления шумов и выбросов, а также для определения общей тенденции сигнала:

```
phases = np.convolve(phases, np.ones(window) / window, mode='same')
```

Относительная разность фаз каналов в радианах показана на рис. 5. Фазовые сдвиги каналов относительно нуля имеют практически постоянный уровень, что свидетельствует о синхронизации в частотной области, и малую дисперсию, что говорит о стабильности работы генераторов плат. Стабильность и предсказуемость сдвигов позволяет легко компенсировать их при обработке сигнала.

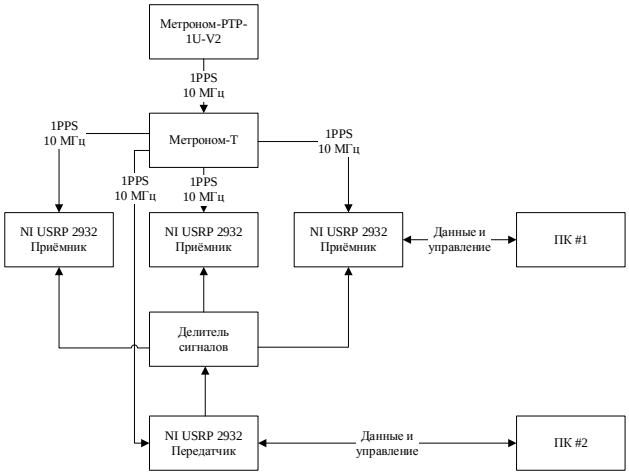

Рис. 3. Схема оценки синхронизации

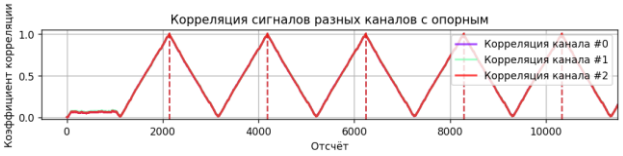

Рис. 4. График временной корреляции каналов

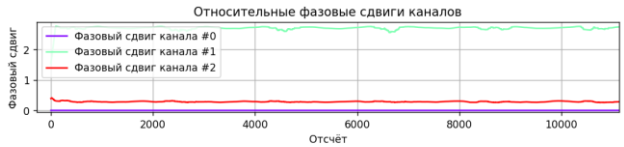

Рис. 5. График относительных фазовых сдвигов каналов

V. Заключение

В данной работе был разработан и протестирован макет четырехэлементной системы антенных решеток на основе плат SDR. Предложенный подход позволил выполнить синхронизацию со стабильными и предсказуемыми результатами во временной и частотной областях.

Была достигнута временная синхронизация с точностью до 100 нс, а также стабильные в течение длительных периодов времени и компенсируемые фазовые сдвиги между РЧ-каналами. Наличие фазовой синхронизации позволяет не только создавать прототипы MIMO-систем стандартов LTE и 5G, но и реализовывать алгоритмы адаптивного диаграммообразования, управления лучом и пространственного позиционирования. В следующих работах на базе этого макета планируется прототипирование системы оценки угла прихода на приёме и формирователя луча диаграммы направленности на передаче.

Список литературы

[1] G. Fokin, D. Volgushev, A. Kireev, D. Bulanov and V. Lavrukhin, "Designing the MIMO SDR-based LPD transceiver for long-range robot control applications," 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops

(ICUMT), St. Petersburg, Russia. 2014. pp. 456-461. doi: 10.1109/ICUMT.2014.7002144

[2] О.С. Барышев, Н.В. Шеремет. Использование методов определения угла прихода в задачах цифрового диаграммообразования сетей 5G // Сборник лучших докладов Всероссийской научно-технической и научно-методической конференции магистрантов и их руководителей. 2023. С. 114-118. EDN ODZGRE.

[3] Г.А. Фокин, Н.В. Шеремет, К.Е. Рютин, И.В. Гришин. Лабораторные испытания SDR демонстратора передачи и приема сигнала LTE с двух антенн // Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А.С. Попова, посвященная Дню радио. 2024. № 1(79). С. 170-173. EDN ZRYTXS.

[4] Н.В. Шеремет, Г.А. Фокин. Экспериментальная апробация технологии MIMO стандарта LTE средствами программно-конфигурируемого радио // Вестник СПбГУТ. 2024. Т. 2, № 3. EDN SCUWLY.

[5] G. Fokin and D. Volgushev, "Software-Defined Radio Network Positioning Technology Design. Problem Statement // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation. 2022. pp. 1-6. doi: 10.1109/IEEECONF53456.2022.9744391.

[6] N. Sheremet and G. Fokin, "Software-Defined Radio Wireless Communication Technology Design. MIMO 4×4 Mode Configuration // 2024 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED). Moscow, Russian Federation. 2024. pp. 1-4. doi: 10.1109/TIRVED63561.2024.10769833

[7] T. Izydorczyk, F. M. L. Tavares, G. Berardinelli, and P. Mogensen, "A USRP-Based Multi-Antenna Testbed for Reception of Multi-Site Cellular Signals // IEEE Access. vol. 7. pp. 162723-162734. 2019. doi: 10.1109/ACCESS.2019.2952094.

[8] USRP-2953 – NI. [Электронный ресурс] URL: https://www.ni.com/docs/en-US/bundle/usrp-295x-getting-started (Дата обращения 06.03.2025).

[9] M. Jean, M. Yuksel, and X. Gong, "ML-Enabled Millimeter-Wave Software-Defined Radio With Programmable Directionality // IEEE Trans. Machine Learning Commun. Networking. vol. 2. pp. 1159-1177. 2024. doi: 10.1109/TMLCN.2024.3449834.

[10] P. Zhang et al., "Design of Reconfigurable SDR Platform for Antenna Selection Aided MIMO Communication System // IEEE Access, vol. 7. pp. 169267-169280. 2019. doi: 10.1109/ACCESS.2019.2946720

[11] USRP N210 Software Defined Radio (SDR). [Электронный ресурс] URL: https://www.ettus.com/all-products/un210-kit/ (Дата обращения 06.03.2025).

[12] USRP-2932 Specifications. [Электронный ресурс] URL: https://www.farnell.com/datasheets/3752197.pdf (Дата обращения 06.03.2025).

[13] Getting started guide NI USRP-2930/2932 Universal Software Radio Peripheral. [Электронный ресурс] URL: https://www.apexwaves.com/pdfs/usrp-2932-user-manual.pdf (Дата обращения 06.03.2025).

[14] Synchronization and MIMO Capability with USRP Devices. [Электронный ресурс] URL: https://kb.ettus.com/Synchronization_and_MIMO_Capability_with_USRP_Devices (Дата обращения 06.03.2025).

[15] GPSDO Kit for USRP N200/N210. [Электронный ресурс] URL: https://www.ettus.com/all-products/gpsdo-kit/ (Дата обращения 06.03.2025).

[16] OctoClock Clock Distribution Module with GPSDO. [Электронный ресурс] URL: https://www.ettus.com/all-products/octoclock-g/ (Дата обращения 06.03.2025).

[17] Metrotek. [Электронный ресурс] URL: https://metrotek.ru/?p=4209 (Дата обращения 06.03.2025).

[18] Translators Radius-50-T, Metronome-T. [Электронный ресурс] URL: https://kbmetrotek.ru/metronom-t/ (Дата обращения 06.03.2025).

[19] UHD (USRP Hardware Driver). Ettus Research. [Электронный ресурс] URL: https://www.ettus.com/sdr-software/uhd-usrp-hardware-driver/ (Дата обращения 06.03.2025).

[20] UHD Python API. [Электронный ресурс] URL: https://kb.ettus.com/UHD_Python_API (Дата обращения 06.03.2025).

[21] NumPy Documentation. [Электронный ресурс] URL: https://numpy.org (Дата обращения 06.03.2025).