

Санкт–Петербургский государственный университет

ОБУХОВ Кирилл Андреевич

Выпускная квалификационная работа

*Разработка интеллектуальной системы управления
корректирующими магнитами комплекса NICA*

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2015 «Прикладная математика, фундаментальная информатика и программирование»

Профиль «Технологии программирования»

Научный руководитель:

кандидат физико-математических наук, доцент
кафедры информационных и ядерных технологий
Сидорин Анатолий Олегович

Рецензент:

ведущий инженер Лаборатории физики высоких энергий
Объединённого института ядерных исследований
Седых Георгий Сергеевич

Санкт-Петербург

2022 г.

Содержание

Введение	3
Постановка задачи	5
Обзор литературы	6
Глава 1. Общие положения	7
1.1. Интеллектуальные системы управления	7
1.2. Машинное обучение	8
1.3. Бустер	8
Глава 2. Расчет положения оптики	9
2.1. Математическая модель	9
2.2. Коррекция орбиты	10
2.3. Алгоритм коррекции	12
Глава 3. Машинное обучение	14
Глава 4. База данных	15
4.1. MAD-X	15
4.2. Сбор данных	15
4.3. Вариации базы данных	16
Глава 5. Модель машинного обучения	18
5.1. Описание	18
5.2. Обучение	18
Глава 6. Результаты	20
6.1. Главная цель	20
6.2. Сравнение с другими алгоритмами коррекции	21
Заключение	23
Дальнейшие планы	24
Список литературы	25

Введение

Объединенный институт ядерных исследований (ОИЯИ) — международная межправительственная организация, созданная на основе соглашения, подписанного одиннадцатью странами-учредителями. Основными направлениями научных исследований Института являются теоретическая физика, физика элементарных частиц, ядерная физика и физика конденсированных сред. Не менее важная роль в ОИЯИ отводится прикладным исследованиям, а также развитию компьютерных сетей и распределенных вычислений. В 2013 году началось строительство ускорительного комплекса NICA (Nuclotron based Ion Collider fAcility) на базе ОИЯИ. Проект NICA ставит перед собой задачу воспроизведения и изучения свойств ядерной материи в условиях, соответствующих ранним стадиям развития Вселенной (кварк-глюонная плазма).

В настоящее время одним из перспективных и многообещающих направлений в разработке робастных систем управления сложными физическими установками класса «Mega-science» таких, как ускорительный комплекс NICA, является построение интеллектуальных регуляторов на базе моделей, построенных с помощью машинного обучения [6, 7]. Применение интеллектуальных систем управления позволит повысить скорость настройки и надежность работы установок, в частности, в нештатных и непредвиденных ситуациях.

На ускорителе Нуклотрон, входящем в состав комплекса NICA, время настройки рабочего режима занимало довольно длительное время, в течение которого операторам приходилось вручную рассчитывать или подбирать параметры режима (например, токи обмоток корректирующих магнитов и др.). Разработка алгоритмов интеллектуальной системы управления (ИСУ) могла бы позволить операторам ускорителей Бустер и Нуклотрон производить настройку в автоматическом режиме с записью и накоплением знаний о режимах работы и особенностях ускорительного комплекса.

Особенностью предлагаемой технологии реализации ИСУ является использование не только математических и физических моделей, но и непосредственно данных, получаемых от физической установки для формирования и

выработки обучающих массивов данных управления установкой.

В управлении сложными техническими комплексами (АЭС, промышленные предприятия и т.д.) хорошо зарекомендовали себя ИСУ на базе моделей, построенных на машинном обучении.

Самой сложной частью в проектировании модели машинного обучения – является процедура обучения и формирования базы данных, в частности, для управления токами питания обмоток корректирующих магнитов (далее корректоров). Для обучения модели необходимо подготовить большой массив данных, что затруднительно (в году проводится около 2х сеансов работы установки) и несет за собой большие экономические издержки (для охлаждения установки используется жидкий гелий, являющийся очень дорогим материалом). Эту проблему можно решить, используя вместо экспериментальных результатов данные, полученные математическим моделированием.

К программным пакетам, позволяющих строить матмодели движения заряженных частиц в различных элементах ускорителей можно отнести MAD-X[5]. Создание математической модели установки в MAD-X позволяет генерировать необходимый набор данных для машинного обучения путем варьирования различными параметрами элементов установки. В дальнейшем эти данные будут обрабатываться и использоваться для обучения нашей модели.

Данная работа демонстрирует способность интеллектуальной системы управления находить наилучшие комбинации токов питания обмоток корректоров. Также ИСУ обладает большим быстродействием для коррекции замкнутой орбиты (ЗО) по сравнению с использованием классических алгоритмов, которые применяются на Бустере и Нуклотроне в настоящее время[8]. Модель написана при помощи языка Python и сторонних библиотек.

Постановка задачи

Целью данной работы является создание интеллектуальной системы коррекции орбиты, с помощью инструментов MAD-X и Python. Для достижения цели были решены следующие задачи:

- Изучить устройство оптической системы
- Изучить алгоритмы коррекции орбиты
- Изучить систему MAD-X
- Интегрировать MAD-X в Python
- Получить и обработать требуемый набор данных для обучения модели
- Обучить модель машинного обучения
- Протестировать модель на экспериментальных результатах, полученных на установке

Обзор литературы

Проект NICA[1], реализуемый на базе ОИЯИ[2], изучает свойства ядерной материи. Подробную информацию можно найти на сайте проекта или, например, в [3].

В начале каждого экспериментального сеанса необходимо проводить настройку оборудования, а также параметров пучка, например, коррекция замкнутой орбиты [4]. Сейчас такие задачи решаются "вручную" с помощью операторов, что занимает достаточно большое время. Использование пакета MAD-X позволяет ускорить процесс настройки, но требует информации о всех параметрах элементов установки и ошибках, многие из которых носят случайный характер.

Пакет MAD-X - это инструмент общего назначения для проектирования оптики заряженных частиц и исследований в ускорителях с переменным градиентом и линиями транспортировки пучка[5]. Он позволяет производить сложные физические расчеты за короткое время.

В настоящее время появляется большой спрос на создание интеллектуальной системы управления (ИСУ) большими физическими установками[6, 7]. Одним из методов реализации ИСУ является машинное обучение. Машинное обучение позволяет сократить время настройки оборудования и мгновенно реагировать на внештатные ситуации, но для такого метода реализации ИСУ необходим большой набор данных, на которых будет производиться обучение.

Коррекция орбиты происходит несколько раз в год (во время работы установки), поэтому реальных данных недостаточно. С помощью же MAD-X и алгоритмов коррекции[8, 9] можно собрать необходимый набор данных путем исследования математической модели установки, требуемый для обучения.

Сейчас существует много инструментов и библиотек для построения модели машинного обучения в Python. Одной из самых популярных является scikit-learn[10]

Глава 1. Общие положения

1.1 Интеллектуальные системы управления

Процесс функционирования ИСУ в эксплуатационном режиме элемента физической установки обобщённо выглядит следующим образом: сенсорная система установки (например, датчики положения пучка, токов корректоров, сигнал с датчиков тока пучка, магнитного поля и др.) выдает информацию в цифровом виде в вычислительные устройства для последующей регистрации всей информации в базе данных (БД) и выработки соответствующего управляющего воздействия на объект управления.

Приняв входные данные от сенсорной системы (которая относится к системе диагностики параметров установки и состояния ее аппаратуры), контроллер выдает результат нечёткого логического вывода, который поступает на объект управления – ОУ (исполняющие устройства системы управления установкой) и осуществляет регулировку и настройку режима работы. Таким образом, система интеллектуального управления, обработав входные данные, постоянно вырабатывает управляющие воздействия на системы управления установкой, компенсируя воздействие ошибки от непредвиденной ситуации управления.

Структурно ИСУ представляет собой двухуровневую систему, где на нижнем уровне происходит сбор, обработка и хранение данных, а на верхнем уровне происходит формирование управляющего воздействия на механизмы системы.

Для эффективного внедрения ИСУ должна создаваться как надстройка к существующей системе контроля и управления, не нарушающая уже существующую систему, и требующая минимальной исходной информации для выработки управляющих воздействий. Разработка иерархической ИСУ позволит решить задачи гарантированного достижения цели управления т.е., соответственно, ожидаемого режима работы установки.

1.2 Машинное обучение

Одним из способов построить интеллектуальную систему является машинное обучение.

Машинное обучение - это класс методов искусственного интеллекта, которые позволяют не напрямую решать задачу, а принимать решения за счет примеров множества сходных задач.

Сейчас многие задачи на ускорителях могут быть решены с помощью машинного обучения[6, 7].

1.3 Бустер

Наше исследование основано на работе Бустера, магнитооптическая структура которого представляет собой последовательное расположение составных элементов: магнитов(М), корректоров(ДК), датчиков положения пучка и т.д. (рис. 1)

Для того чтобы измерить положение пучка по всему периметру Бустера расположены датчики. Они позволяют узнать насколько отклонился пучок от равновесной орбиты.

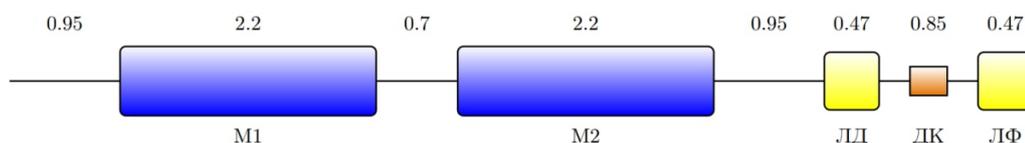


Рис. 1: Часть устройства бустера.

Глава 2. Расчет положения оптики

2.1 Математическая модель

Исследуемый ускоритель и/или канал транспортировки пучка описываются как последовательность оптических элементов (дипольных и квадрупольных магнитов, корректоров и т.п.) и устройств диагностики, расположенных вдоль сопровождающей системы равновесной частицы. Сопровождающая система равновесной частицы – это траектория заряженной частицы с расчетным значением импульса, проходящей через идеальные магниты, установленные без погрешностей (рис. 2)

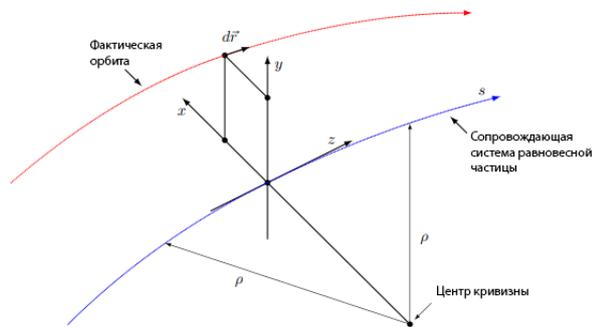


Рис. 2: Система координат, используемая для описания положения орбиты.

Движение частиц описывается в локальной системе координат, связанной с фактической орбитой ускорителя, которая отклоняется от сопровождающей системы равновесной частицы из-за погрешностей электромагнитных полей и установки оптических элементов. Одна из задач коррекции оптики заключается в том, чтобы минимизировать отклонение фактической орбиты сопровождающей системы пучка от сопровождающей системы равновесной частицы. Измерение положения фактической орбиты осуществляется с помощью датчиков положения пучка, а ее коррекция - с помощью дипольных корректирующих магнитов (корректоров). Для коррекции фактической орбиты используется несколько методов, например, метод гармонического анализа, метод локальных искажений орбиты[9]. Данная работа посвящена автоматизации процедуры коррекции, основанной на измерении матрицы отклика ускорителя[8].

2.2 Коррекция орбиты

Расчет положения орбиты в точке s_i при наличии возмущения θ_z , расположенного в точке s_0 продольной координаты, описывается с помощью уравнения Хилла[8]:

$$z(s_i) = \theta_z(s_0) \frac{\sqrt{\beta_z(s_i)\beta_z(s_0)}}{2 \sin(\pi Q_z)} \cos(Q_z \pi - |\mu_z(s_i) - \mu_z(s_0)|), \quad (1)$$

где s_0, s_i - продольная координата положения возмущения и точки наблюдения соответственно; z - поперечная координата в горизонтальной или вертикальной плоскости (x или y соответственно); β - бета-функция; μ - набег бета-тронной фазы; Q_z - когерентная бетатронная частота в горизонтальной или вертикальной плоскости. Уравнение (1) можно записать в матричной форме с использованием матрицы отклика орбиты:

$$\begin{pmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{pmatrix} \begin{pmatrix} \theta_x \\ \theta_y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \text{ или } \vec{z} = \mathbf{R}\vec{\Theta}, \quad (2)$$

где $\vec{\Theta}$ - вектор состояние корректора; \vec{z} - вектор координат в точке расположения датчика положения пучка; \mathbf{R} - матрица отклика орбиты.

Отклик орбиты - изменение положения орбиты (\vec{z}) в результате действия возмущений ($\vec{\Theta}$). Элементами матрицы \mathbf{R} являются коэффициенты отклика в датчиках i под действием возмущения в точках расположения корректоров j :

$$R_{zz,ij} = \frac{\Delta z_i}{\theta_{z;j}}, R_{xy,ij} = \frac{\Delta x_i}{\theta_{y;j}}, R_{yx,ij} = \frac{\Delta y_i}{\theta_{x;j}}.$$

Здесь элементы R_{zz} показывают отклик в вертикальной или горизонтальной плоскости, а $R_{xy,ij}$ и $R_{yx,ij}$ показывают наличие связи вертикального и горизонтального движений.

Рассмотрим Бустер с возмущением $\vec{q} = (x, x', y, y') = const$ в точке j . Вектор координат в четырехмерном фазовом пространстве $\vec{x} = (x, x', y, y')$ тогда определяется как:

$$\vec{x}_j = \mathbf{R}_{jj}\vec{x}_j + \vec{q}_j, \quad (3)$$

Выражение (3) можно преобразовать к виду:

$$\vec{x}_j = (1 - \mathbf{R}_{jj})^{-1}\vec{q}_j, \quad (4)$$

тогда в точке x_i (отличной x_j) имеем:

$$\vec{x}_i = \mathbf{R}_{ij}\vec{x}_j = \mathbf{R}_{ij}(1 - \mathbf{R}_{jj})^{-1}\vec{q}_j = \mathbf{A}_{ij}\vec{q}_j, \quad \mathbf{A}_{ij} = \mathbf{R}_{ij}(1 - \mathbf{R}_{jj})^{-1}. \quad (5)$$

Не исключая общности, предположим отсутствие связи горизонтального и вертикального бетатронных движений, далее будем рассматривать только горизонтальные (x). В случае Бустера с n датчиками и m корректирующими магнитами уравнение (2) с учетом преобразований (3), (4), (5) примет вид:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & A_{22} & \dots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nm} \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{pmatrix} \quad (6)$$

Таким образом, задача коррекции орбиты заключается в определении сил корректоров, минимизирующих возмущения орбиты, измеряемые в местах установки датчиков и вызванные наличием дополнительных возмущений, что представимо в виде:

$$\vec{\Theta} = \mathbf{R}(\mathbf{A})^{-1}\vec{z}_{ref}, \vec{z}_{ref} \rightarrow \min$$

Элементы матрицы отклика (6) определяются экспериментально по следующему алгоритму:

- Фиксируем начальное положение орбиты \vec{x}_0
- Передаем небольшое воздействие в j -й корректор - $\Delta\vec{\theta}_j$
- Записываем измененное положение орбиты \vec{x}_j

- Вычисляем соответствующий элемент матрицы отклика, нормализовав полученные значения:

$$\mathbf{A} = \left(\frac{\vec{x}_1 - \vec{x}_0}{\Delta\theta_1}, \frac{\vec{x}_2 - \vec{x}_0}{\Delta\theta_2}, \dots \right)$$

2.3 Алгоритм коррекции

После получения матрицы отклика необходимо найти обратную матрицу, для вычисления необходимого корректирующего воздействия:

$$\vec{\Theta} = -\mathbf{A}^{-1}\vec{x} \quad (7)$$

В общем случае количество корректоров не совпадает с количеством датчиков, и матрица отклика является прямоугольной. Так как понятие обратной матрицы существует только для квадратных матриц, то необходимо воспользоваться сингулярным разложением матриц (SVD)[11], которое позволяет найти псевдообратную матрицу.

Алгоритм сингулярного разложения матрицы \mathbf{A} :

1. Находим произведение матриц $\mathbf{A}\mathbf{A}^T$
2. Находим собственные числа и собственные вектора $\mathbf{A}\mathbf{A}^T$
3. Получаем матрицу E , размерность которой совпадает с размерностью матрицы \mathbf{A} , вида:

$$\begin{pmatrix} \lambda_1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \lambda_2 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & \lambda_{n-1} & 0 & 0 \\ 0 & 0 & \dots & 0 & \lambda_n & 0 \end{pmatrix},$$

где λ_j - квадратные корни из собственных чисел матрицы $\mathbf{A}\mathbf{A}^T$

4. Из собственных векторов составляем матрицу U , в предположении, что $UU^T = I$, где I - единичная матрица

5. Находим произведение матриц $\mathbf{A}^T \mathbf{A}$
6. Находим собственные вектора матриц $\mathbf{A}^T \mathbf{A}$
7. Из собственных векторов составляем матрицу V , в предположении, что $V^T V = I$, где I - единичная матрица

В итоге получаем сингулярное разложение вида $\mathbf{A} = U E V^T$. Псевдо-обратная матрица находится с помощью формулы $\mathbf{A}^{-1} = V E^{-1} U^T$.

После нахождения обратной или псевдообратной матрицы можем воспользоваться формулой (7), где \vec{x} - изначальное положение орбиты. Заметим, что присутствует линейная связь между корректирующим воздействием напряжением и смещением орбиты.

Глава 3. Машинное обучение

Методы машинного обучения нацелены на создание компьютерных программ, которые со временем автоматически улучшаются в зависимости от данных, на которых эти программы будут обучаться. В зависимости от проблемы и вида обучающих примеров выбираются различные подходы. Одним из таких подходов является обучение с учителем[12].

Обучение с учителем идеально подходит к тем моделям, обучающие примеры которых содержат пары 'стимул-реакция'. Во время обучения прогнозы делаются на основе попадающих на вход 'стимулов', а затем происходит сравнение их с эталонной соответствующей им 'реакцией'. Основная идея заключается в том, чтобы минимизировать разницу между прогнозами, сделанными во время обучения, и истинными результатами. Разница определяется как функция потерь (в данной работе, например, средняя абсолютная ошибка), которая минимизируется с помощью одного из алгоритмов минимизации.

В зависимости от задачи сложность модели может варьироваться от одномерной линейной регрессии до глубоких нейронных сетей. Цель прогнозирования, т.е. категория или количество, позволяет отнести задачу к задаче классификации или регрессии. В данной работе исследуется регрессионная модель, так как задача состоит в том, чтобы количественно измерить корректирующее воздействие.

Глава 4. База данных

Как уже отмечалось ранее, самым важным аспектом для создания модели машинного обучения является создание базы данных. Процедура получения данных на работающем ускорителе достаточно затратная по времени и стоимости, и, как правило, таких данных недостаточно. Эта проблема решается за счет использования математической модели ускорителя с помощью инструментов MAD-X. В идеальную оптическую структуру добавляются погрешности установки элементов и с помощью математической модели вычисляется положение фактической орбиты в точках расположения датчиков.

4.1 MAD-X

MAD-X — это инструмент общего назначения для проектирования оптики заряженных частиц и исследований в ускорителях с переменным градиентом и пучками. Он может работать с ускорителями среднего и очень большого размера и решает различные проблемы на таких машинах.

MAD-X является преемником MAD-8 и был специально адаптирован к потребностям конструкции Большого адронного коллайдера.

MAD-X основан на языке C++.

Для разных языков программирования существуют библиотеки, которые позволяют интегрировать пакет MAD-X. Для Python библиотека называется `crumad(1.9.3)`[13]. Интегрирование MAD-X в Python позволяет в одной среде работать с библиотекой машинного обучения и с MAD-X и напрямую передавать данные между ними без промежуточного хранения файлов.

4.2 Сбор данных

Для обучения будущей модели необходим большой набор данных. На периметре Бустера комплекса NICA установлено 24 корректора и 24 датчика, поэтому на вход мы подаем вектор X размерности 1×24 - вектор разницы смещения, полученных с датчиков, на выходе же имеем вектор Y размерности 1×24 - вектор корректирующего воздействия.

Имеющиеся экспериментальные и расчетные данные показывают, что

основной вклад в возмущения ЗО вносят ошибки положения магнитооптических элементов установки. Поскольку в задачи работы входит только алгоритм разработки ИСУ, то рассматриваются только ошибки такого типа.

Пользуемся модулем MAD-X - ealign, который позволяет внести в систему ошибки пространственного положения магнитного элемента. Добавим случайные ошибки в параметр $d\psi$, который отвечает за скручивание, в диапазоне от -10^{-11} до 10^{-11} радиан. Такие маленькие значения выбраны неслучайно, эмпирическим путем было выявлено, что маленькие величины позволяют модели более точно подбирать коэффициенты и более точно предсказывать корректирующее воздействие.

После того как ошибки были внесены в модель, пользуемся алгоритмом матрицы отклика. Получаем вектор корректирующих воздействий, который поочередно передаем в корректоры. Если новое положение орбиты, находится в окрестности $\pm 10^{-5}$ м, то такие данные нас устраивают и мы записываем их в наш набор.

База данных была собрана из приблизительно 1000 примеров, в каждом примере находится пара векторов размерности 1×24 . С помощью Pandas (1.4.2) база данных была записана в формате .csv, тем самым обеспечивая быстрый доступ к данным для обучения.

4.3 Вариации базы данных

Было рассмотрено несколько вариантов базы данных перед обучением. Первым из дополнительных вариантов сбора данных является случай, когда мы вносим ошибки не только в параметр $d\psi$, но и в другие, которые предоставляет модуль ealign. В этом наборе данных представлены ошибки для:

- dx - смещение в направлении x (горизонтальной оси);
- dy - смещение в направлении y (вертикальной оси);
- ds - смещение в направлении s (продольной оси);
- $d\phi$ - скручивание вокруг оси x ;
- $d\theta$ - скручивание вокруг оси y ;

- m_{rex} - горизонтальная ошибка чтения для датчика;
- m_{rey} - вертикальная ошибка чтения для датчика;
- m_{scalx} - Относительная ошибка масштабирования по горизонтали для датчика;
- m_{scaly} - Относительная ошибка масштабирования по вертикали для датчика.

Путем тестирования этого набора данных было выявлено, что модель, которая обучается для одного вида ошибок, относящихся к смещению, будет показывать неплохие результаты и для другого.

Второй из дополнительных наборов данных нацелен на будущую работу с более сложной архитектурой сети. Суть в том, что в реальных условиях не все корректоры могут исправно работать, поэтому необходимо учитывать такие внештатные ситуации для корректной работы ИСУ. По этой причине был разработан набор, который на вход подает вектор X размерности 1×48 , в отличие от предыдущих наборов данных этот входной набор увеличивается в два раза. На первой половине входного вектора находятся разница показаний датчиков, как и в вышеупомянутых случаях, на второй половине находится вектор из 0 и 1. 0 показывает неработающий корректор, 1 показывает работающий корректор. На выходе имеем тот же вектор Y размерности 1×24 , но на местах неработающих корректоров находятся значения 0.0. Данная работа не имела под собой задачи рассмотреть этот случай, поэтому набор данных собран для будущих исследований.

Также существовали наборы, в которых ошибки, вносимые в параметр $d\psi$, брались из большего диапазона (-10^{-5} ; 10^{-5}), но такие наборы показали результаты хуже, как уже выше упоминалось.

Глава 5. Модель машинного обучения

5.1 Описание

Общая идея состоит в том, чтобы создать регрессионную модель, которая автоматически изучит корреляцию между передаваемым воздействием и изменившимся положением орбиты. Как было видно из Главы 2, у нас имеется линейная связь между корректирующим воздействием и положением орбиты. Из линейной зависимости можно сделать вывод о том, что лучшей моделью является линейная регрессия. Линейная регрессия - это регрессионная модель зависимости одной переменной от другой или множества других с линейной функцией зависимости[14]. Мы используем обучение с учителем, т.е. обучение модели будет производиться по парам “вход-выход”, на вход у нас подается разница между изначальным положением орбиты и скорректированным, на выходе имеем воздействие, которое привело к такому изменению положения.

5.2 Обучение

Машинное обучение было реализована с помощью библиотеки `scikit-learn` (1.1.0)[10]. `Scikit-learn` - это один из самых популярных пакетов Python для машинного обучения. Он позволяет выполнять множество задач, связанных с этой темой:

- предварительная обработка данных
- выбор модели
- регрессии
- классификации
- кластерный анализ

В частности, в данной библиотеке присутствует класс для модели построенной на линейной регрессии - `LinearRegression()`.

Для того чтобы сделать вывод об эффективности обучения, набор был разделен на обучающий (80 процентов) и тестовый (20 процентов). Показателями качества для задач регрессии обычно являются средняя абсолютная ошибка (MAE) и коэффициент детерминации (R^2), определяемые как:

$$MAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|,$$

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{Var(\mathbf{y} - \hat{\mathbf{y}})}{Var(\mathbf{y})},$$

где \hat{y} - прогнозируемый результат, \hat{y}_i - прогнозируемое значение i -й выборки, y - соответствующие истинные значения для n общих выборок, а Var - дисперсия, квадрат стандартного отклонения.

На 1000 примерах модель показала отличные результаты $R^2 = 0.99$, $MAE = 12 * 10^{-5}$. Увеличение количества обучающих выборок не приводит к значительному улучшению результата. Модель достаточно простая и не требует большого хранилища данных.

Глава 6. Результаты

6.1 Главная цель

По итогу работы основная цель была выполнена - была разработана ИСУ корректирующими магнитами. ИСУ показала хорошие результаты коррекции, которые представлены на Рис. 3, Рис. 4, Рис. 5.

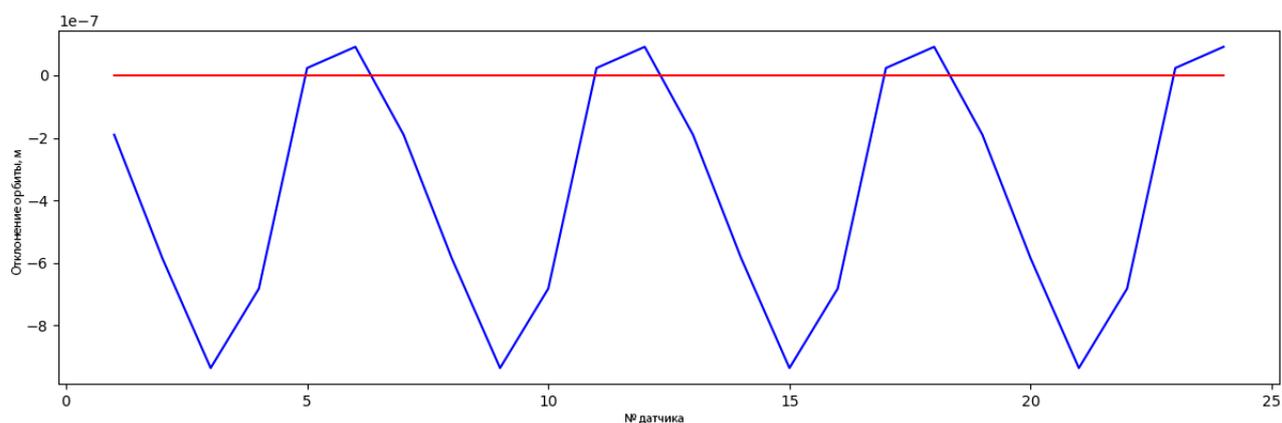


Рис. 3: Отклонение фактической орбиты ускорителя от идеального в метрах до коррекции (синяя кривая) и после применения корректирующего воздействия (красная линия). Для ошибок в диапазоне $(-10^{-4}; 10^{-4})$

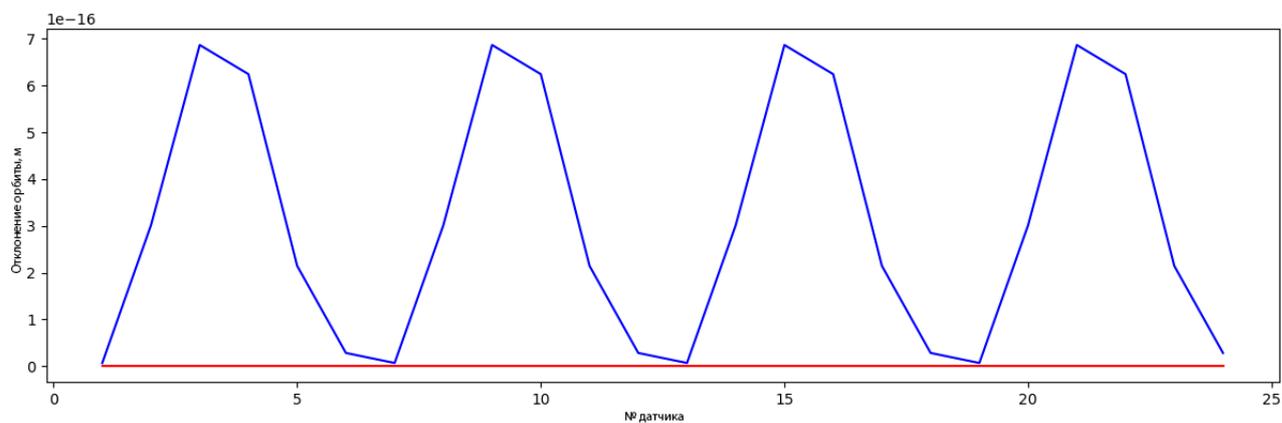


Рис. 4: Отклонение фактической орбиты ускорителя от идеального в метрах до коррекции (синяя кривая) и после применения корректирующего воздействия (красная линия). Для ошибок в диапазоне $(-10^{-9}; 10^{-9})$

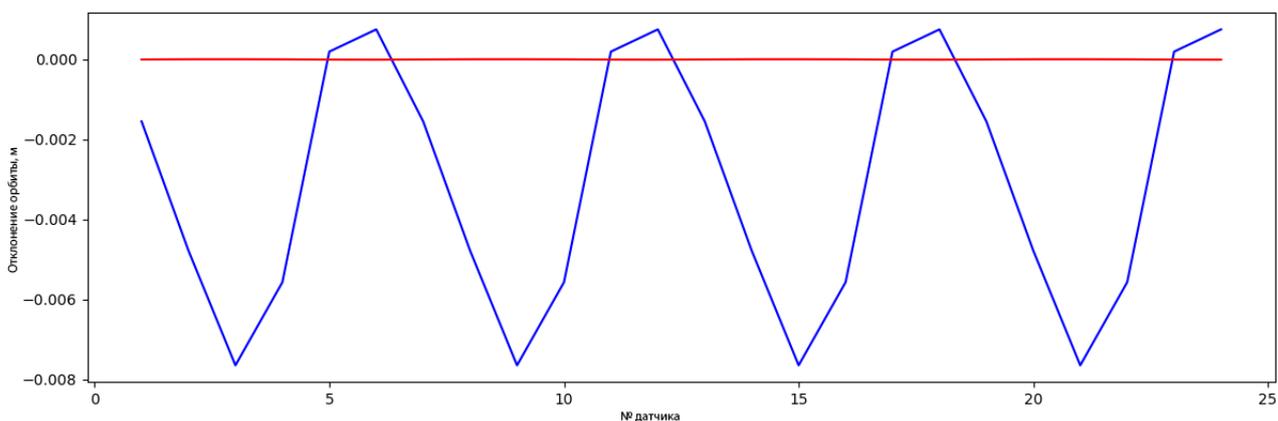


Рис. 5: Отклонение фактической орбиты ускорителя от идеального в метрах до коррекции (синяя кривая) и после применения корректирующего воздействия (красная линия). Для ошибок в диапазоне $(-10^{-2}; 10^{-2})$

6.2 Сравнение с другими алгоритмами коррекции

Модель сравнивалась с уже упомянутым алгоритмом матрицы отклика и алгоритмом LSQ. Алгоритм LSQ был использован с помощью внутренних инструментов MAD-X, подробное описание метода и инструментов можно найти в [5, 15].

Полученное время вычисления корректирующих воздействий с помощью разработанной модели машинного обучения в три раза меньше, чем с применением аналитических расчетов и встроенных в MAD-X алгоритмов (Таблица 1). Аналогичный результат был получен для набора данных с последнего экспериментального сеанса работы установки. Также стоит отметить, что полученные с помощью разработанной ИСУ требуемые корректирующие воздействия показывают более равномерную нагрузку на систему корректоров, что позволяет улучшить условия их эксплуатации и надежность системы коррекции, в целом (Рис. 6).

Таблица 1: Сравнение алгоритмов для ошибок из диапазона $(-10^{-4}; 10^{-4})$

Алгоритм	Затраченное время, с	Отклонение от идеального положения орбиты, м
Матрица отклика	3	$\pm 10^{-7}$
LSQ	2	$\pm 10^{-6}$
Модель	1	$\pm 10^{-6}$

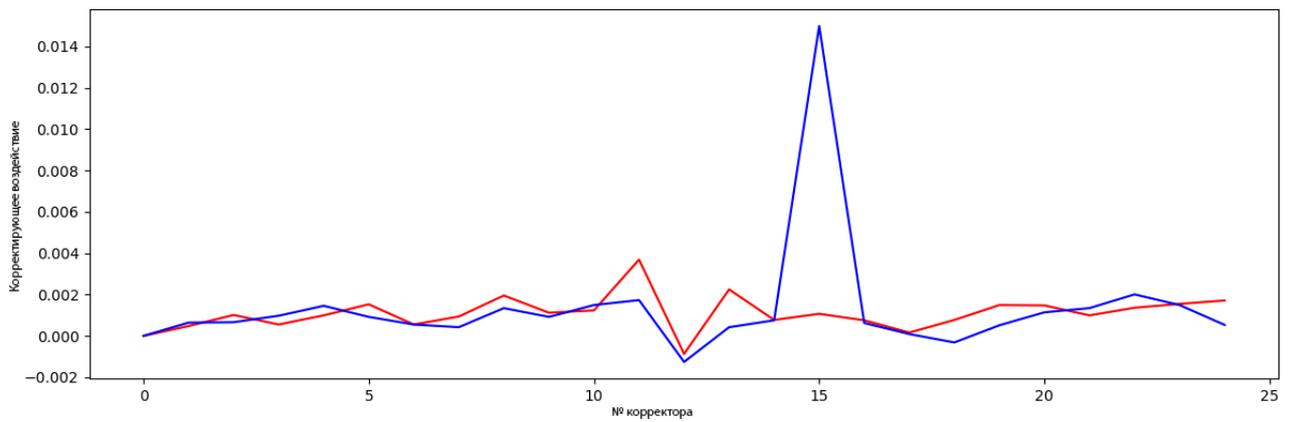


Рис. 6: Нагрузка на систему корректоров. Синим - воздействие на корректора, полученные матрицей отклика, красным - с помощью модели.

Заключение

Изучена структура и функции интеллектуальной системы управления физическими установками, основным достоинством ИСУ является ее быстрое действие, которое может обеспечить значительную экономию временных ресурсов и затрат на эксплуатацию установки.

Объектом управления разрабатываемой ИСУ является положение фактической орбиты ускорителя относительно равновесной, коррекция которой производится корректирующими магнитами.

Изучены алгоритмы создания базы данных для управления корректирующими магнитами. Для обучения модели разработан алгоритм, основанный на использовании математической модели ускорителя в пакете MAD-X. Изучены необходимые для использования этого пакета модули.

Создана база данных для обучения модели, состоящая из 1000 примеров, весь набор был разделен на обучающий (80 процентов) и тестовый (20 процентов).

Обученная модель показала следующие результаты $R^2 = 0.99$, $MAE = 12 * 10^{-5}$. Увеличение количества обучающих выборок не приводит к значительному улучшению результата.

Полученное время вычисления корректирующих воздействий с помощью модели в три раза меньше, чем с применением аналитических расчетов.

Таким образом, все поставленные задачи успешно выполнены.

Исходный код, разработанной модели доступен в GitHub по ссылке[16].

Дальнейшие планы

В процессе выполнения работы были намечены дальнейшие шаги исследования:

- Построить нейронную сеть, которая будет учитывать неработающие корректоры и на основе работающих минимизировать отклонение орбиты от идеального положения;
- Разработать алгоритм для корректировки других ошибок, например, индивидуальных ошибок магнитов;
- Разработать нейронную сеть по классификации и вычислению индивидуальных ошибок в магнитах.

Список литературы

- [1] Официальный сайт проекта NICA, <https://nica.jinr.ru>.
- [2] Официальный сайт ОИЯИ, <https://jinr.ru>.
- [3] V.D.Kekelidze, A.D.Kovalenko, R.Lednicky, V.A. Matveev, I.N.Meshkov, A.S.Sorin, G.V.Trubnikov, Project NICA at JINR, Nuclear Physics A, 2013, V. 904-905
- [4] О. С. Козлов, И. Н. Мешков, А. О. Сидорин, Г. В. Трубников, Динамика интенсивного ионного пучка в коллайдере NICA, С. 1321-1332
- [5] Документация пакета MAD-X,
<http://mad.web.cern.ch/mad/releases/5.04.01/madxguide.pdf>.
- [6] F.F. Van der Veken, G. Azzopardi, F.H. Blanc, L.T.D. Coyle, E. Fol, M. Giovannozzi, T. Pieloni, S. Redaelli, B.M. Salvachua Ferrando, M. Schenk, R. Tomas Garcia, G. Valentino, Machine learning in accelerator physics: Applications at the CERN Large Hadron Collider.
- [7] E. Fol, R. Tomás, G. Franchetti, Supervised learning-based reconstruction of magnet errors in circular accelerators.
- [8] М. М. Шандов, С. А. Костромин, Алгоритмы коррекции орбиты Бустера NICA.
- [9] V. Ziemann, Uppsala University, Imperfections and Correction.
- [10] Документация scikit-learn, <https://scikit-learn.org>.
- [11] Афанасьева А. А., Вычисление сингулярного разложения матриц.
- [12] Qiong Liu, Ying Wu, Supervised Learning.
- [13] Документация библиотеки csumad, <http://hibtc.github.io/csumad/>.
- [14] Shrikant I. Bangdiwala, Regression: simple linear.

[15] USPAS, Least Squares Fitting

[16] Ссылка на проект, <https://github.com/rewqqx/OpticsCorrections>