

Санкт-Петербургский государственный университет

БЕЛЯЕВА Александра Дмитриевна

Выпускная квалификационная работа

***Программное обеспечение системы измерения
положения пучка бустера NICA***

Уровень образования: магистратура
Направление 03.04.01 «Прикладная математика и физика»
Основная образовательная программа ВМ.5826.2019
«Процессы управления Мега-сайенс установками»

Научный руководитель:

доцент, кафедра
ИиЯТ ПМ-ПУ,
к.ф.-м.н.,
Сидорин А.О.

Рецензент:

гл. инженер Базовой установки
Нуклотрон ЛФВЭ ОИЯИ,
д.ф.-м.н., профессор
Сыресин Е.М.

Санкт-Петербург
2021

Оглавление

Введение.....	3
Постановка задачи.....	5
Обзор литературы.....	6
Глава 1. Процессор положения адронного пучка Libera Hadron.....	8
Глава 2. Расчёт параметров ускорителя в точке инжекции	9
2.1. Математическая модель ускорителя в окрестности точки инжекции.....	9
2.2. Расчёт фазовых координат центра тяжести пучка.....	11
2.3. Расчёт структурных функций	12
2.4. Расчёт положения замкнутой орбиты	14
Глава 3. Разработка приложения в системе Tango Controls.....	16
3.1. Распределённая система управления Tango Controls	16
3.2. Создание класса устройства Tango Controls	17
3.3. Реализация логики класса PhaseProfile	22
3.4. Конфигурация приложения.....	23
3.5. Утилита Astor	25
3.6. Дополнительные возможности	25
Глава 4. Разработка Web-клиентского приложения	26
4.1. Устройство Tango RestDS2	26
4.2. Средства разработки Web-клиентского приложения.....	27
4.3. Web-клиентское приложение.....	28
Глава 5. Тестирование	30
5.1. Тестирование на эмуляторах	30
5.2. Апробация ПО во время первого сеанса на Бустере NICA, набор данных для последующего анализа.....	30
5.3. Обработка результатов сеанса	31
Выводы	32
Заключение	33
Список литературы	34

Введение

Объединённый институт ядерных исследований (ОИЯИ) — крупный международный научный проект, учреждённый восемнадцатью странами-участницами в 1956 году. Цель его создания – экспериментальные и теоретические исследования в области ядерной физики, физики элементарных частиц и физики конденсированного состояния. На базе ОИЯИ в 2013 году было запущено строительство ускорительного комплекса NICA (Nuclotron based Ion Collider fAcility). Проект NICA нацелен на воспроизведение и изучение свойств ядерной материи в условиях, соответствующих ранним стадиям развития Вселенной (кварк-глюонная плазма).

В 2020 году был введён в эксплуатацию новый бустерный синхротрон тяжёлых ионов. Он служит первым ускоряющим кольцом комплекса NICA и расположен внутри ускорителя Синхрофазотрон. Ядро магнита Синхрофазотрона исполняет роль биологической защиты от ускоряемого пучка. Основными задачами Бустера являются накопление $2 \cdot 10^9$ ионов золота $^{197}\text{Au}^{31+}$ и ускорение их до энергии 578 МэВ/н, которой достаточно для последующей обдирки их до состояния $^{197}\text{Au}^{79+}$.

Большие научные установки, такие как ускорительный комплекс NICA, состоят из множества элементов оборудования различного типа и назначения. Каждое устройство или комплекс устройств имеет свою систему управления. Для слаженной работы всей системы необходимо корректное взаимодействие её элементов. Не важно, как связаны все элементы между собой: напрямую или через промежуточные звенья, главное, чтобы вся система работала как единое целое. В этом состоит задача общей системы управления. На ускорительных установках и, в частности, на NICA в качестве общей системы управления используется распределённая система Tango Controls (Taco Next Generation Objects).

Tango Controls является бесплатно и свободно распространяемым программным обеспечением. Его концепция была разработана в Европейском центре синхротронного излучения ESRF в Гренобле, Франция, около 20 лет

назад. На данный момент система имеет серьёзный набор инструментов и компонентов.

Система управления NICA состоит из распределённых локальных Tango систем, каждая из которых закреплена за определённой установкой, например, Бустером, Нуклотроном, инжекционным комплексом, а также на каждом канале. Для корректного взаимодействия всех систем, они связываются с помощью специальных шлюзов. В настоящей работе рассматривается Tango система на Бустере NICA.

Для минимизации потерь при инжекции и обеспечения высокого качества ускоренного пучка необходимо оптимизировать параметры пучка, поступающего на вход ускорителя. Для этого необходимо знать положение замкнутой орбиты ускорителя и параметры траектории частиц на фазовых плоскостях в точке инжекции. Эти параметры можно определить, используя данные о положении центра тяжести пучка на первых оборотах его циркуляции. Для измерения положения пучка на Бустере NICA используются ёмкостные датчики положения пучка – пикап-электроды, сигналы которых обрабатываются процессором положения адронного пучка Libera Hadron.

Данная работа посвящена созданию программного обеспечения системы измерения положения пучка Бустера NICA, его диагностики и расчёта необходимых параметров ускорителя в точке инжекции. Программное обеспечение включает в себя Tango сервер и Web-клиент для взаимодействия пользователя с системой управления.

Постановка задачи

Целью данной работы является создание программного обеспечения в системе Tango Controls, производящее диагностику циркуляции пучка в Бустере NICA на первых оборотах.

Для достижения цели были поставлены следующие задачи:

- Изучить методику диагностики циркуляции пучка на первых оборотах;
- Изучить систему управления Tango Controls;
- Изучить алгоритм написания приложения для Tango Controls с использованием внутренних утилит;
- Написать Tango класс устройства для диагностики пучка на языке программирования C++;
- Интегрировать класс в систему управления Tango Controls;
- Разработать Web-клиентское приложение для отображения характеристик пучка: фазового портрета, структурных функций, «нулевого» положения пучка на выходе септума;
- Протестировать программное обеспечение на эмуляторах и ввести в эксплуатацию на Бустере NICA.

Обзор литературы

Проект NICA [1], строящийся на базе ОИЯИ [2] с 2013 года, имеет своей целью изучение свойств ядерной материи. Подробную официальную информацию о проекте можно найти, например, в [3].

При проведении исследований на ускорительных установках важным этапом является диагностика пучков заряженных частиц. Основные аспекты и технические тонкости описываются как в учебной литературе [4, 5], так и в материалах различных школ по физике ускорителей, например, USPAS [6]. Этот этап не может быть проведён с максимально возможной эффективностью без использования систем управления, одной из которых является система Tango Controls.

Система управления Tango впервые была представлена на конференции ICALEPCS в 1999 году [7]. Её разработка началась в исследовательском ускорительном комплексе ESRF в Гренобле, Франция. Согласно официальному сайту [8], на сегодняшний день, система Tango Controls внедрена в сорок девять исследовательских центров и институтов, таких как SOLEIL (Франция), ALBA (Испания), DESY (Германия) и др. Среди российских институтов-пользователей Tango Controls числятся Курчатовский институт и ОИЯИ.

В июне 2018 года на 32-м съезде Tango сообщества сотрудниками ОИЯИ был представлен доклад “JINR NICA beam transfer channels CS” [9], который содержал сведения об использовании системы Tango Controls на ускорительном комплексе NICA. Также в докладе были отмечены преимущества данной системы, позволившие ей получить столь широкое распространение. Результаты внедрения Tango структур в некоторые подсистемы ускорителя представлены в работах [10, 11, 12, 13, 14]. Последнюю информацию об использовании Tango Controls в системе управления Бустера можно найти в недавней публикации [15].

Для удобного отображения информации и взаимодействия пользователей с устройствами системы управления Tango Controls

дополнительно создаются Web-клиентские приложения. Их взаимодействие с устройствами Tango основано на использовании устройства Tango RestDS2, созданного сотрудниками ОИЯИ [16].

При разработке Web-клиентских приложений использовались библиотека EXT JS 6.2.0 с оболочкой Sencha Cmd. Библиотека представляет собой набор виджетов для создания пользовательского интерфейса. Подробную информацию о ней можно найти в технической документации [17], а также на сайте компании-разработчика Sencha, Inc. [18].

Глава 1. Процессор положения адронного пучка Libera Hadron

Для диагностики замкнутой орбиты циркулирующего пучка в ускорительных установках необходимо измерение положения его центра тяжести. Для этого в комплексе NICA, и в частности, на Бустере, используются ёмкостные датчики, или пикапы, двух типов: ортогональные и диагональные. Пикап, используемый в Бустере NICA, представляет собой два последовательно расположенных горизонтального и вертикального электродов. По принципу действия он относится к пикапу ортогонального типа. На бустерном кольце расположено двадцать четыре таких датчика.

Процессор положения адронного пучка Libera Hadron разработан компанией Instrumentation Technologies [19]. Его задача состоит в точном и воспроизводимом измерении положения пучка и создании обратной связи. Процессор включает в себя 4 типа модулей:

- Модули процессора положения пучка, включающие в себя программируемую логическую интегральную схему, клиентскую сторону к аппаратной части и плату оперативной памяти.
- Межсоединительную плату, отвечающую за интерфейс системы управления.
- Модуль синхронизации, выполняющий функцию приема событий, а также распределение тактовых и триггерных сигналов.
- Модуль GDX (Gigabit data exchange), обеспечивающий передачу данных в режиме реального времени.

Данные сохраняются в выделенных буферах и становятся доступны для чтения и последующей обработки. Для работы с входными сигналами используется Tango Device Server, разработанный для управления устройством в среде Tango Controls.

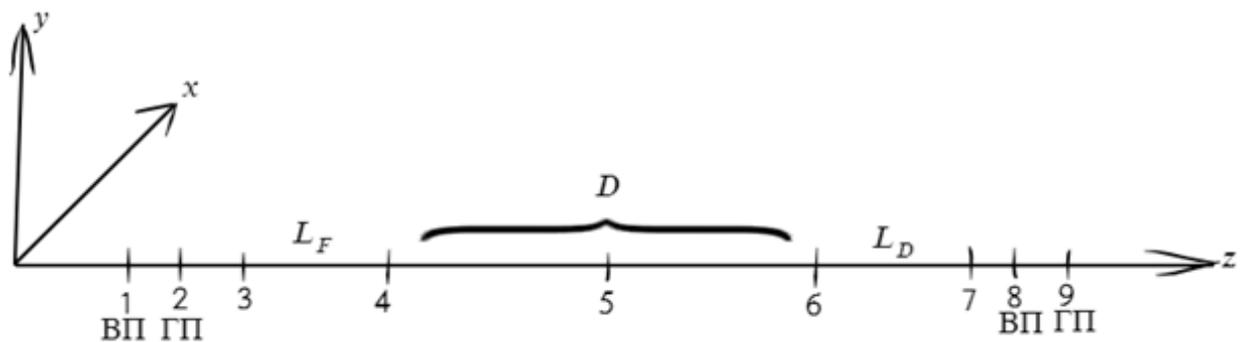
Глава 2. Расчёт параметров ускорителя в точке инжекции

2.1. Математическая модель ускорителя в окрестности точки инжекции

Датчики положения пучка предоставляют информацию только о координатах его центра тяжести. Для вычисления положения центра тяжести пучка на главных фазовых плоскостях необходимо дополнительно определить его угловые координаты. Угловые координаты можно вычислить по результатам измерения координат в двух точках ускорителя на основании математической модели участка ускорителя между этими точками. Математическая модель участка представляет собой матрицы преобразования всех структурных элементов, которые последовательно проходит пучок. Для расчёта параметров ускорителя в точке инжекции используются данные с двух ближайших к ней датчиков положения пучка – это пикапы под номером 1 и 24.

Математическая модель строится на основе схемы расположения элементов ускорителя, представленной на Рис. 1, пучок движется вдоль оси Oz . На расстоянии d_{y_1} и d_{x_1} от вертикального (т. 1, Рис. 1) и горизонтального (т. 2, Рис. 1) электродов пикапа №24 соответственно находится фокусирующая линза L_F . Далее пучок проходит участок дрейфа длиной D , после которого расположена дефокусирующая линза L_D . На расстоянии d_{x_2} и d_{y_2} от линзы находятся горизонтальный (т. 8, Рис. 1) и вертикальный (т. 9, Рис. 1) электроды пикапа №1 соответственно. Заметим, что фокусирующая линза L_F фокусирует пучок по оси Ox и одновременно дефокусирует его по оси Oy . Аналогично устроена дефокусирующая линза L_D .

Точка инжекции (т. 5, Рис.1) расположена в середине участка дрейфа между линзами. После каждого прохождения пучка по бустерному кольцу рассчитываются фазовые координаты центра тяжести в положении 5. Множество таких точек образует его фазовую траекторию.



- 1 и 2 – вертикальный и горизонтальный пикапы,
 1 – 3 – дрейф длиной d_{y_1} , 2 – 3 – дрейф длиной d_{x_1}
 3 – 4 – фокусирующая линза L_F ,
 4 – 6 – дрейф длиной D , 6 – 7 – дефокусирующая линза L_D ,
 8 и 9 – горизонтальный и вертикальный пикапы,
 7 – 8 – дрейф длиной d_{y_2} , 7 – 9 – дрейф длиной d_{x_2}

Рис. 1. Схема прохождения пучка между *Libera Hadron* №24 и №1.

Преобразование фазовых координат центра тяжести пучка может быть представлено с помощью матриц перехода [5]: для участков дрейфа (1), для фокусирующей линзы L_F в плоскостях xx' и yy' (2а) и (2б) соответственно, для дефокусирующей линзы L_D в плоскостях xx' и yy' (3а) и (3б) соответственно.

$$M_D = \begin{pmatrix} 1 & D \\ 0 & 1 \end{pmatrix}, \quad (1)$$

$$M_{L_F x} = \begin{pmatrix} \cos \varphi_1 & \frac{1}{\sqrt{|k_1|}} \sin \varphi_1 \\ -\sqrt{|k_1|} \sin \varphi_1 & \cos \varphi_1 \end{pmatrix}, \quad (2a)$$

$$M_{L_F y} = \begin{pmatrix} ch \varphi_1 & \frac{1}{\sqrt{|k_1|}} sh \varphi_1 \\ \sqrt{|k_1|} sh \varphi_1 & ch \varphi_1 \end{pmatrix}, \quad (2б)$$

$$M_{L_D x} = \begin{pmatrix} ch \varphi_2 & \frac{1}{\sqrt{|k_2|}} sh \varphi_2 \\ \sqrt{|k_2|} sh \varphi_2 & ch \varphi_2 \end{pmatrix}, \quad (3a)$$

$$M_{L_D y} = \begin{pmatrix} \cos \varphi_2 & \frac{1}{\sqrt{|k_2|}} \sin \varphi_2 \\ -\sqrt{|k_2|} \sin \varphi_2 & \cos \varphi_2 \end{pmatrix}. \quad (3б)$$

Параметры матриц перехода определяются токами питания линз и результатами измерения их длин и положений:

- $k_1 = 0.8334$ – жёсткость L_F линзы (квадрупольный коэффициент);
- $k_2 = 0.8522$ – жёсткость L_D линзы (квадрупольный коэффициент);
- $L = 0.47$ м – длина линз;
- $\varphi_1 = L\sqrt{|k_1|}$;
- $\varphi_2 = L\sqrt{|k_2|}$;
- $D = 7$ м;
- $d_{y_1} = 0.585$ м – расстояние между центром вертикального пикапа и краем линзы L_F (в начале участка инжекции);
- $d_{x_1} = 0.265$ м – расстояние между центром горизонтального пикапа и краем линзы L_F (в начале участка инжекции);
- $d_{y_2} = 0.265$ м – расстояние между краем линзы L_D (в конце участка инжекции) и центром вертикального пикапа;
- $d_{x_2} = 0.585$ м – расстояние между краем линзы L_D (в конце участка инжекции) и центром вертикального пикапа.

2.2. Расчёт фазовых координат центра тяжести пучка

В случае независимого движения пучка в главных плоскостях оптической системы $x x'$ и $y y'$ формулы для плоскостей $x x'$ и $y y'$ имеют одинаковый характер, и далее их вывод будет представлен только для плоскости $x x'$. Для плоскости $y y'$ все построения происходят аналогичным образом.

При переходе из точки 2 в точку 9 фазовые координаты преобразуются по следующему закону:

$$\begin{pmatrix} x_9 \\ x'_9 \end{pmatrix} = M_{29x} \begin{pmatrix} x_2 \\ x'_2 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} x_2 \\ x'_2 \end{pmatrix}, \quad (4)$$

где

$$M_{29x} = M_{d_{x_2}} M_{L_D x} M_D M_{L_F x} M_{d_{x_1}} = \begin{pmatrix} -2.7668 & 9.4128 \\ -1.1389 & 3.5131 \end{pmatrix}. \quad (5)$$

Положение пучка в точке 2 и в точке 9 известно из Tango устройства. Тогда фазовая переменная x'_2 вычисляется как:

$$x'_2 = \frac{x_9 - m_{11}x_2}{m_{12}} = \frac{x_9 + 2.7668x_2}{9.4128}. \quad (6)$$

Чтобы определить фазовые координаты в точке инъекции x_5 и x'_5 необходимо рассмотреть переход из точки 2 в точку 5. Закон преобразования имеет вид:

$$\begin{pmatrix} x_5 \\ x'_5 \end{pmatrix} = M_{25x} \begin{pmatrix} x_2 \\ x'_2 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} x_2 \\ x'_2 \end{pmatrix}, \quad (7)$$

где

$$M_{25x} = M_{\frac{D}{2}} M_{L_F x} M_{d_{x_1}} = \begin{pmatrix} -0.4199 & 3.5272 \\ -0.3798 & 0.8087 \end{pmatrix}. \quad (8)$$

Из формулы (7) можно получить следующие выражения:

$$x_5 = m_{11}x_2 + m_{12}x'_2 = -0.4199x_2 + 3.5272x'_2, \quad (9)$$

$$x'_5 = m_{21}x_2 + m_{22}x'_2 = -0.3798x_2 + 0.8087x'_2. \quad (10)$$

Фазовые координаты пучка y_5 и y'_5 :

$$y_5 = m_{11}y_1 + m_{12}y'_1 = 2.5069y_1 + 5.7782y'_1, \quad (11)$$

$$y'_5 = m_{21}y_1 + m_{22}y'_1 = 0.4038y_1 + 1.3297y'_1. \quad (12)$$

Множество всех пар точек $(x_5; x'_5)$ и $(y_5; y'_5)$ образует фазовую траекторию центра тяжести циркулирующего пучка за первые N оборотов.

2.3. Расчёт структурных функций

В процессе циркуляции центр тяжести пучка в точке инъекции движется по фазовой траектории, определяемой фокусирующими свойствами ускорителя, и называемой «эллипсом Флоке». Расчёт параметров эллипса Флоке, называемых структурными функциями, и является задачей данной работы. Набор точек, представляющих последовательные пересечения пучком участка инъекции, может интерпретироваться как набор частиц некоторого условного пучка, согласованного со структурными функциями канала. Параметры Твисса, описывающие фазовый портрет такого пучка и являются искомыми структурными функциями.

Для их расчёта необходимо вычисление моментов первого (13а) и (13б) и второго (14а), (14б) и (14в) порядков фазовых координат за первые N оборотов:

$$\langle x \rangle = \frac{1}{N} \sum_{j=1}^N x^j, \quad (13а)$$

$$\langle x' \rangle = \frac{1}{N} \sum_{j=1}^N x'^j, \quad (13б)$$

$$\langle x^2 \rangle = \frac{1}{N} \sum_{j=1}^N (x^j - \langle x \rangle)^2, \quad (14а)$$

$$\langle x'^2 \rangle = \frac{1}{N} \sum_{j=1}^N (x'^j - \langle x' \rangle)^2, \quad (14б)$$

$$\langle x \cdot x' \rangle = \frac{1}{N} \sum_{j=1}^N (x^j - \langle x \rangle)(x'^j - \langle x' \rangle). \quad (14в)$$

Тогда величина, аналогичная среднеквадратичному эмиттансу [4, 5], имеет вид:

$$I_x = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle x \cdot x' \rangle^2}. \quad (15)$$

Для исследования динамики преобразования фазового портрета пучка достаточно проследить изменения структурных функций:

$$\alpha_x = -\frac{\langle x \cdot x' \rangle}{I_x}, \quad (16а)$$

$$\beta_x = \frac{\langle x^2 \rangle}{I_x}, \quad (16б)$$

$$\gamma_x = \frac{\langle x'^2 \rangle}{I_x}. \quad (16в)$$

не заботясь о способе определения площади, занимаемой представляющими точками на фазовой плоскости, поскольку они описывают только форму эллипса: отношение полуосей и угол наклона, и не зависят от его площади.

Уравнение эллипса Флоке (Рис. 2) в терминах структурных функций имеет вид:

$$\gamma_x x^2 + 2\alpha_x x x' + \beta_x x'^2 = I_x. \quad (17)$$

В программе реализовано его параметрическое построение.

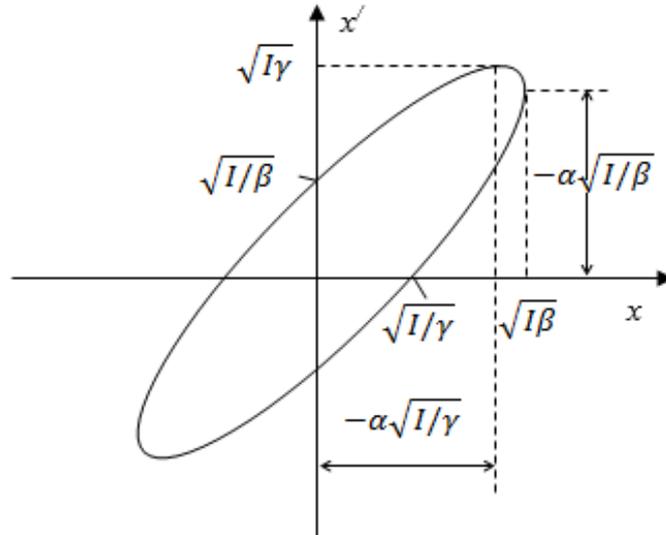


Рис. 2. Связь структурных функций и характерных точек эллипса Флоке.

2.4. Расчёт положения замкнутой орбиты

Чтобы выполнить настройку однократной однооборотной инжекции необходимо рассчитать положение замкнутой орбиты ускорителя для последующей инжекции пучка в эту точку. «Нулевое» положение пучка рассчитывается в несколько этапов.

Сперва производится нормировка фазовых координат пучка:

$$X^j = \frac{x^j - \langle x \rangle}{\sqrt{\beta_x}}, \quad (18)$$

$$X'^j = \frac{\alpha_x}{\sqrt{\beta_x}} (x^j - \langle x \rangle) + \sqrt{\beta_x} (x'^j - \langle x' \rangle). \quad (19)$$

Затем происходит вычисление положения R_x^j и угла движения Q_x^j циркулирующего пучка в точке инжекции на каждом из первых N оборотов:

$$R_x^j = \sqrt{X^{j^2} + X'^{j^2}}, \quad (20)$$

$$Q_x^j = \operatorname{arctg} \left(\frac{X'^j}{X^j} \right) \quad (21)$$

и дробной части частоты бетатронных колебаний:

$$Q_x = \frac{1}{2\pi} \frac{1}{N-1} \sum_{j=2}^N \left(\operatorname{arctg} \left(\frac{X'^j}{X^j} \right) - \operatorname{arctg} \left(\frac{X'^{j-1}}{X^{j-1}} \right) \right). \quad (22)$$

Вычисление замкнутой орбиты ускорителя в точке инжекции – мнимого положения инжектированного пучка, получаемого трассировкой в обратном направлении координат пучка на первом обороте, производится по формулам:

$$x^0 = \beta_x X^0 + \langle x \rangle, \quad (23)$$

$$x'^0 = \frac{X'^0 - \alpha_x X^0}{\sqrt{\beta_x}} + \langle x' \rangle, \quad (24)$$

где

$$X^0 = R_x^1 \cos \varphi^0, \quad (25)$$

$$X'^0 = R_x^1 \sin \varphi^0, \quad (26)$$

$$\varphi^0 = \operatorname{arctg} \left(\frac{X'^0}{X^0} \right) - Q_x. \quad (27)$$

Глава 3. Разработка приложения в системе Tango Controls

3.1. Распределённая система управления Tango Controls

Распределённая система Tango Controls [8] – это набор инструментов с открытым исходным кодом для управления любым оборудованием или программным обеспечением. Система Tango Controls основана на принципе распределённых устройств. Устройство Tango может представлять собой отображение реального устройства или какой-либо логической сущности. Например, источник питания, систему сбора данных с пикап-электродов, математический алгоритм обработки данных. В качестве транспортного протокола Tango использует CORBA и ZMQ. Tango Controls не зависит от операционной системы и поддерживает C++, Java и Python для всех компонентов.

Концепция Tango Controls основана на акторах (Actors) и микросервисах (MicroServices). Модель акторов – это математическая модель параллельных вычислений. Акторы имеют внутреннее состояние. Они не связаны между собой и могут взаимодействовать посредством стандартных сообщений. Микросервисы – концепция разработки программного обеспечения, как набора небольших сервисов, каждый из которых работает самостоятельно. Распределённые устройства в системе Tango являются акторами и, одновременно, представляют собой микросервис.

Каждое устройство Tango решает определённую задачу и обменивается с другими устройствами сообщениями со стандартным интерфейсом. Устройства Tango могут одновременно выполнять роль и сервера, и клиента. Для взаимодействия с устройством необходимо знать только его уникальный идентификатор вида: домен/семейство/имя. В системе управления комплексом NICA принята следующая иерархия:

- Домен – установка, например, Нуклотрон, Бустер и т.д.;
- Семейство – подсистема, например, вакуум, термометрия, ВЧ и т.д.;
- Имя – тип оборудования и его номер;

В Tango существуют три модели взаимодействия серверов и клиентов:

1) Синхронное взаимодействие – вызов инициируется клиентом. Клиент переходит в ожидание, пока сервер не отправит ответ или не истечёт время ожидания.

2) Асинхронное взаимодействие – вызов также инициирует клиент, продолжая при этом свою работу. При отсутствии ответа может быть отправлен повторный запрос.

3) Подписка на события – клиент регистрирует свою заинтересованность в событии один раз. После этого сервер информирует клиента о каждом событии. Это позволяет избежать многочисленных обращений клиента, освобождает его для выполнения других задач, работает быстро и эффективно. События можно настраивать только на атрибуты и пайпы. Есть несколько видов событий:

- События по изменению значения: генерируются тогда, когда значение интересующей величины изменяется больше, чем на предустановленное значение. Изменение может иметь абсолютный или относительный характер.
- Периодические события.
- Архивные события: объединение событий по изменению значения и периодических событий. События присылаются либо по изменению атрибута, либо по истечению заданного временного периода, если изменений не произошло.

3.2. Создание класса устройства Tango Controls

Каждое устройство Tango является экземпляром класса, поэтому для создания устройства необходимо разработать соответствующий ему класс. Для генерации шаблонов исходного кода используется утилита Pogo (Рис. 3). С её помощью можно задавать интерфейс будущего устройства Tango. В рамках этой работы был разработан класс устройств с названием PhaseProfile.

Каждое устройство имеет состояние (State) и статус (Status). Всего существует 14 предустановленных возможных состояний. Разработанный класс имеет три состояния:

- ON – устройство включено;
- OFF – устройство выключено;
- FAULT – устройство сломано.

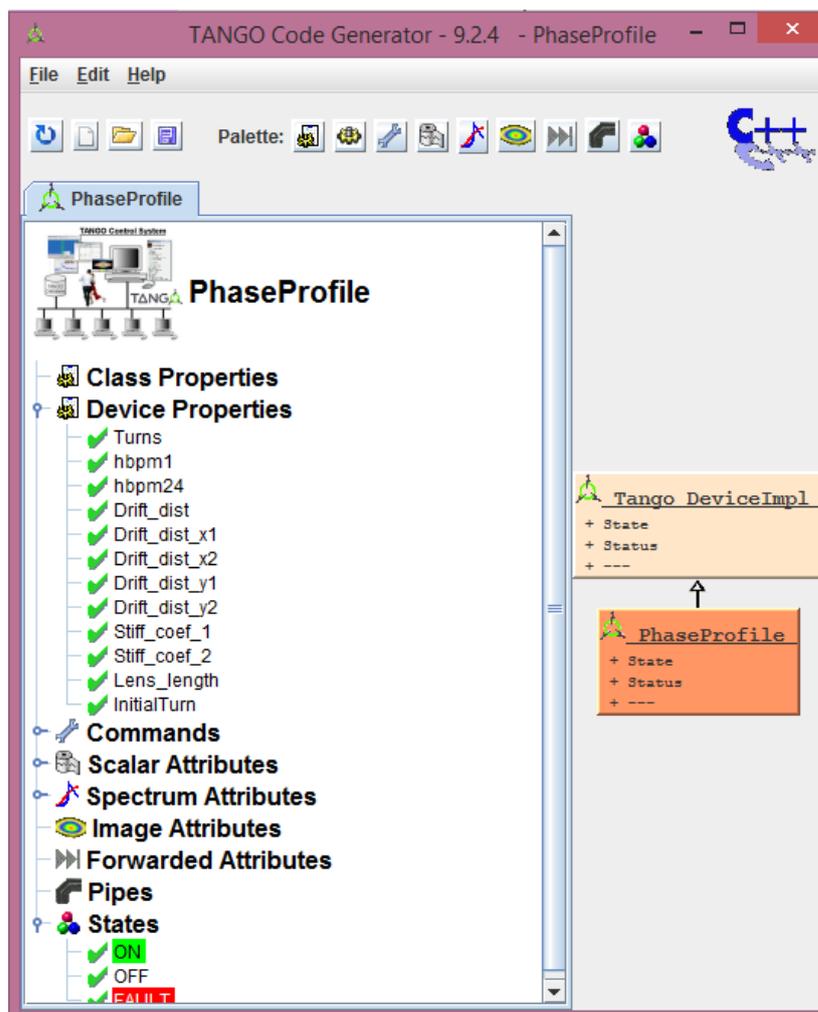


Рис. 3. Интерфейс утилиты Pogo.

Статус – это строка, которая произвольным образом описывает режим работы устройства. С помощью статуса система мониторинга может выявить ошибки и их причины.

Также в интерфейс входят атрибуты, команды, пайпы и свойства.

Атрибутами (Attributes) называются поля данных с определенным типом (логический, строковый, целое число, число с плавающей точкой, символы, перечисления и др.). Атрибуты обладают размерностью: скалярные (Scalar

Attributes), спектральные (Spectrum Attributes) – одномерные массивы и картинки (Image Attributes) – двумерные массивы. Также поля данных могут содержать дополнительные параметры: описание, единицы измерения, минимальное и максимальное значения, формат отображения. Разработанный класс PhaseProfile имеет порядка сорока скалярных и спектральных атрибутов типа DevDouble.

Скалярные атрибуты:

- TimeStamp – временная метка, показывает время (в секундах) последнего обновления данных;
- emit_X, emit_Y – площади эллипсов Флоке для плоскостей xx' и yy' соответственно, рассчитанные по формуле (15);
- alpha_X, alpha_Y – структурные функции α_x и α_y , рассчитанные по формуле (16а);
- beta_X, beta_Y – структурные функции β_x и β_y , рассчитанные по формуле (16б);
- gamma_X, gamma_Y – структурные функции γ_x и γ_y , рассчитанные по формуле (16в);
- DX0_norm, DY0_norm – нормированные фазовые координаты x и y положения замкнутой орбиты, рассчитанные по формуле (25);
- DX0ang_norm, DY0ang_norm – нормированные фазовые координаты x' и y' положения замкнутой орбиты, рассчитанные по формуле (26);
- DX0_center, DY0_center – фазовые координаты x и y положения замкнутой орбиты, рассчитанные по формуле (23) без смещения на средние значения;
- DX0ang_center, DY0ang_center – фазовые координаты x' и y' положения замкнутой орбиты, рассчитанные по формуле (24) без смещения на средние значения;
- DX0, DY0 – фазовые координаты x и y положения замкнутой орбиты, рассчитанные по формуле (23);

- $DX0ang, DY0ang$ – фазовые координаты x' и y' положения замкнутой орбиты, рассчитанные по формуле (24);

- QX, QY – дробные части частот бетатронных колебаний для плоскостей xx' и yy' соответственно, рассчитанные по формуле (22).

Спектральные атрибуты:

- $XPlane, YPlane$ – фазовые координаты центра тяжести пучка x и y , рассчитанные по формулам (9) и (11);

- $XangPlane, YangPlane$ – фазовые координаты центра тяжести пучка x' и y' , рассчитанные по формулам (10) и (12);

- X_centre, Y_centre – фазовые координаты центра тяжести пучка x и y , смещённые на средние значения положения пучка, рассчитанные по формуле (13а);

- $Xang_centre, Yang_centre$ – фазовые координаты центра тяжести пучка x' и y' , смещённые на средние значения положения пучка, рассчитанные по формуле (13б);

- X_norm, Y_norm – нормированные фазовые координаты центра тяжести пучка x и y , рассчитанные по формуле (18);

- $Xang_norm, Yang_norm$ – фазовые координаты центра тяжести пучка x' и y' , рассчитанные по формуле (19);

- $ellips_X, ellips_Xang$ – фазовые координаты x и x' , удовлетворяющие уравнению эллипса Флоке (17);

- $ellips_Y, ellips_Yang$ – фазовые координаты y и y' , удовлетворяющие уравнению эллипса Флоке (17);

- Rx, Ry – длины радиус-векторов положений циркулирующего пучка в плоскостях xx' и yy' , рассчитанные по формуле (20);

- $ThetaX, ThetaY$ – полярные углы радиус-векторов положений циркулирующего пучка в плоскостях xx' и yy' , рассчитанные по формуле (21);

- $DThetaX, DThetaY$ – изменение полярных углов, при прохождении пучком одного оборота.

Команды (Commands) – это функции, которые могут изменять состояния устройства или совершать различные действия. В классе PhaseProfile реализовано четыре команды:

- On – включить устройство;
- Off – выключить устройство;
- Check_devices;
- Update.

Пайпы (Pipes) – это канал для обмена данных. Не рекомендовано использовать пайпы при отсутствии острой необходимости, поэтому в данном классе и в других классах систем управления NISA, они не реализованы.

Последним элементом интерфейса являются свойства (Properties). Это набор данных, которые хранятся в центральной базе данных. При включении устройство обращается к базе данных и только после этого конфигурируется определенным образом. Такой способ задания свойств имеет ряд важных преимуществ: конфигурационная информация не записывается в код, её легко изменить и следить за её актуальностью, также можно породить произвольное количество экземпляров с одинаковой логикой и разными значениями свойств. В классе PhaseProfile заданы свойства, отвечающие за уникальный идентификатор устройств, к которым обращается экземпляр данного класса, характеристики системы и количество оборотов, которое необходимо рассмотреть.

В классе Tango можно задать поведение устройства в зависимости от его текущего состояния. За это отвечает машина состояний (State Machine). В зависимости от состояния можно разрешить или запретить выполнение определённых команд или чтение или запись атрибутов. Этот механизм помогает защитить данные. Например, в случае поломки устройства, можно запретить чтение атрибутов, так как они могут выдавать некорректные значения.

Когда интерфейс настроен, создаётся шаблон будущего класса. При необходимости можно открыть в утилите Pogo файл с разрешением .xmi и внести изменения в сгенерированный класс.

3.3. Реализация логики класса PhaseProfile

Для разработки класса был выбран язык программирования C++ и среда разработки Visual Studio Community. Для того, чтобы написанный код не был утерян при необходимости изменить структуру шаблона в Pogo, код необходимо помещать в специальные защищённые участки, которые начинаются со слов «protected region id» и заканчиваются – «protected region end».

В классе PhaseProfile реализованы команды включения и выключения устройства, его инициализации, команды Check_devices и Update.

При инициализации устройства считываются свойства из базы данных. Также создается Tango группа – локальный набор устройств, присоединённых к определённой группе. Это иерархический объект: можно построить группу как из групп, так и из отдельных устройств. В классе создана Tango группа, состоящая из двух Tango устройств – драйверов процессоров положения адронного пучка Libera Hadron. В команду инициализации дополнительно включена процедура заполнения матриц перехода, так как вычисление матричных элементов производится только один раз на основе данных из свойств устройства.

Команда Check_devices обращается к Tango группе и проверяет наличие отклика от каждого устройства и обновления данных. Если отклик положительный и данные обновлены, то вызывается метод Update. Команда Check_devices вызывается каждые 500 мс.

При вызове команды Update считываются данные с двух Libera Hadron, а именно координаты положения центра тяжести пучка X и Y. С помощью метода read_attributes(), примененного к Tango группе, можно прочитать сразу

два атрибута с двух устройств. Далее производятся расчёты из главы 2 и запись соответствующих данных в атрибуты.

3.4. Конфигурация приложения

Для хранения общей структуры системы управления и конфигурационной информации для каждого устройства служит база данных MySQL, к которой присоединено устройство Tango – DatabaseDS. Таким образом, клиенты получают доступ к центральной базе через интерфейс устройства Tango. База данных имеет точку доступа, состоящую из хоста и порта. Для этого необходимо создать переменную окружения TANGO_HOST и задать ей нужное значение, например, localhost:10000. Центральная база данных ускорительного комплекса NICA состоит из трёх баз данных, синхронизированных между собой.

Для регистрации серверов и добавления новых классов, а также для настройки устройств служит утилита Jive (Рис. 4, Рис. 5). Прежде чем добавить новый класс, необходимо зарегистрировать сервер устройств. Для этого задаётся имя сервера (ServerName) – имя исполняемого файла, и объект (Instance) – запускаемая программа. После этого добавляется класс. В сервере устройств могут быть устройства как одного, так и нескольких классов.

Для добавления экземпляров класса необходимо каждому задать уникальный идентификатор в виде: домен/семейство/имя. Например, Tango устройства, с которых класс PhaseProfile читает данные имеют идентификаторы: «booster/orbita/hbpm.1» и «booster/orbita/hbpm.24».

После регистрации сервера и добавления устройства в утилите Jive можно изменять различные настройки и задавать свойства. Также Jive отображает основные характеристики устройства, значения атрибутов и текущий статус.

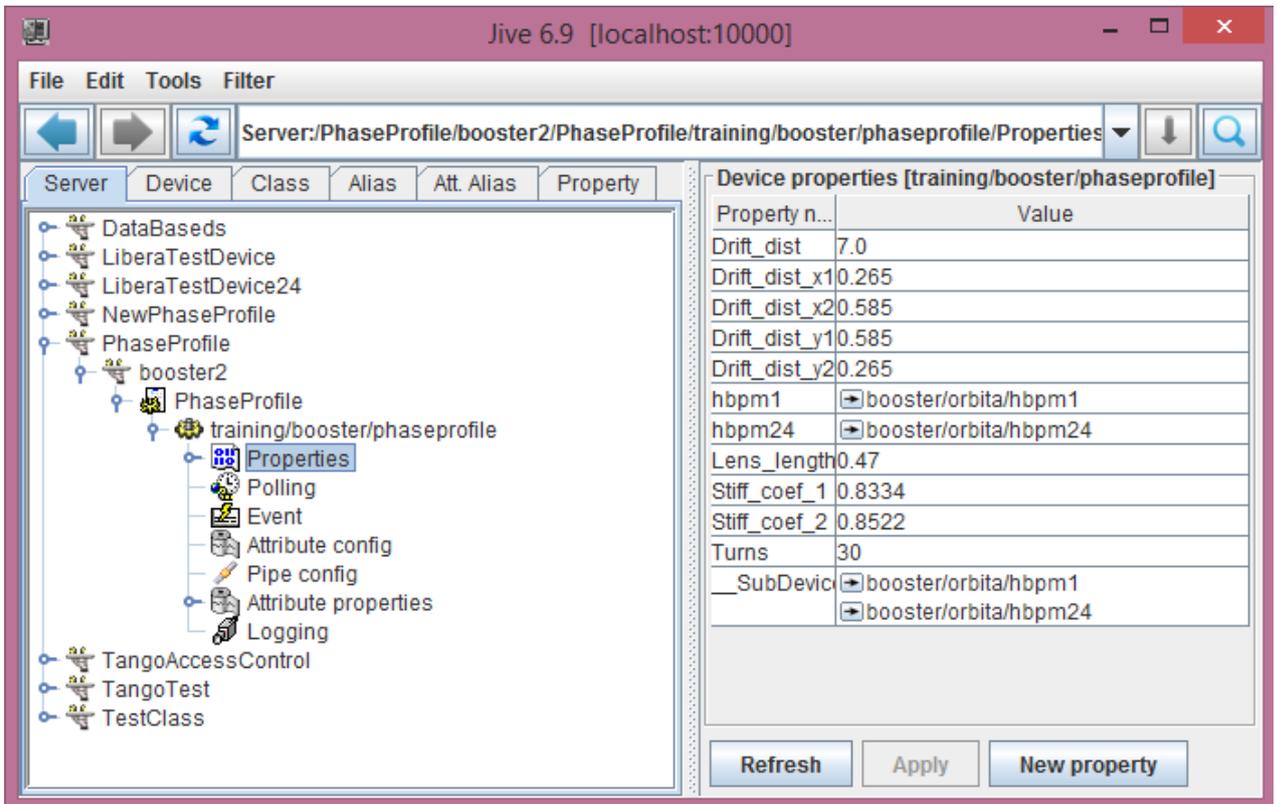


Рис. 4. Интерфейс утилиты Jive.

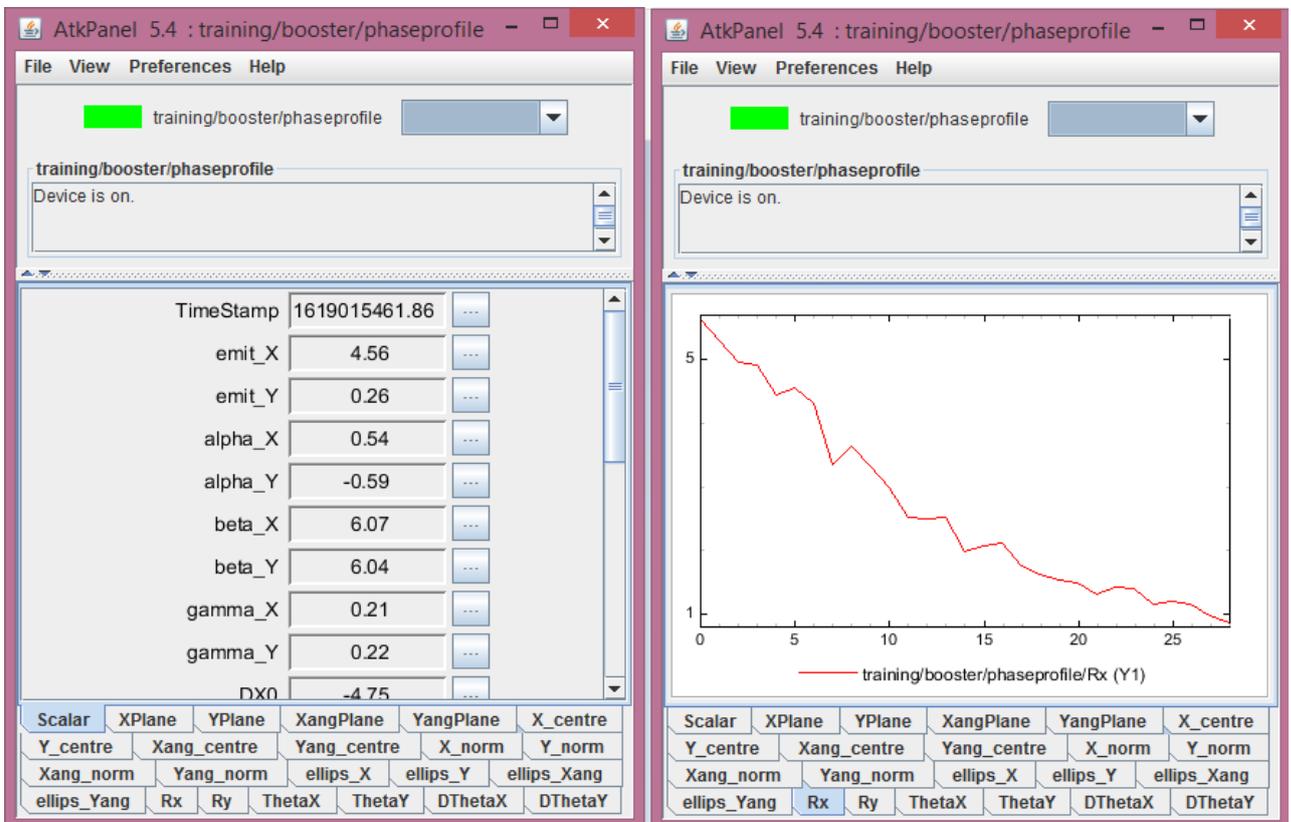


Рис. 5. Интерфейс устройства PhaseProfile в утилите Jive.

3.5. Утилита Astor

Ещё одним полезным компонентом системы управления является утилита Astor (Рис. 6). Она даёт возможность выполнить быструю диагностику системы и, в случае обнаружения проблемы, решить её. Наряду с этим можно настраивать контролируруемую систему и её компоненты, а также устанавливать последовательность запуска этих компонентов. Утилита Astor обладает необходимым функционалом для долгосрочного анализа (журналы, статистика и т.д.).

Astor отображает состояние системы управления и её компонентов (запуск, остановка, тестирование, настройка, отображение информации) с помощью пиктограмм различных цветов.



Рис. 6. Интерфейс утилиты Astor.

3.6. Дополнительные возможности

В созданном классе PhaseProfile используется синхронное взаимодействие с пикапами. Так как объем передаваемых данных небольшой (не превосходит 10 000 элементов), взаимодействие осуществляется быстро. Однако в классе присутствует возможность подписки на события в случае их добавления в Tango устройства процессоров Libera Hadron. Тогда будет возможен переход от синхронного взаимодействия к взаимодействию по подписке, что является более эффективным способом использования сети.

Глава 4. Разработка Web-клиентского приложения

4.1. Устройство Tango RestDS2

Для Web-доступа к автоматической системе управления Tango Controls в ЛФВЭ ОИЯИ было разработано Tango устройство RestDS2 [16], которое реализует интерфейс Tango REST (REpresentational State Transfer) API. RestDS2 принимает HTTP(S) запросы и, в зависимости от их типа, определённым образом взаимодействует с другими Tango устройствами. Приложение поддерживает чтение и запись атрибута или нескольких атрибутов, получение информации об устройствах, командах, атрибутах и истории их изменения.

В RestDS2 реализовано три вида запросов:

- GET – для чтения атрибутов. URL адрес запроса имеет вид:

/tango/rest/rc4/devices/{домен}/{семейство}/{имя}/attributes/{имя_атрибута}.

В случае успешного обращения сервер возвращает данные в формате JSON, содержащие имя атрибута, значение и время прочтения.

- POST – для записи атрибутов. URL адрес запроса имеет вид:

/tango/rest/rc4/devices/{домен}/{семейство}/{имя}/attributes/{имя_атрибута}

Дополнительно необходимо передать параметр: *argin={новое_значение}*.

- POST – для исполнения команд. URL адрес запроса имеет вид:

/tango/rest/rc4/devices/{домен}/{семейство}/{имя}/commands/{имя_команды}

При необходимости возможна передача параметра: *argin={аргумент}*.

В случае неудачного запроса возвращается строка, содержащая информацию о возникшей проблеме.

Взаимодействие Web-клиентского приложения с сервером может осуществляться по защищённому протоколу HTTPS и быть реализовано через базовый механизм авторизации. В данной работе Web-клиентское приложение является только средством отображения данных и не подразумевает изменения настроек устройства и его атрибутов, поэтому взаимодействие осуществляется по простому протоколу HTTP.

4.2. Средства разработки Web-клиентского приложения

Для разработки Web-клиентского приложения использовался язык JavaScript и библиотека EXT JS 6.2.0 с оболочкой Sencha Cmd [17, 18].

В данном приложении используется архитектура «Модель-Представление-Контроллер» (MVC), поддерживаемая библиотекой EXT JS. Такой подход к созданию Web-приложения разделяет код на логические части:

- Модель (Model) – компонент, отвечающий за данные, определяющий структуру приложения.
- Представление (View) – компонент, отвечающий за взаимодействие с пользователем.
- Контроллер (Controller) – компонент, отвечающий за связь между моделью и представлением.

Данная архитектура обеспечивает структуру и согласованность кода в программе.

Важной частью приложения является магазин (Store) – хранилище данных. В магазин данные поступают из внешнего источника, например, из базы данных или через REST API. Для динамического обращения к Tango устройству и обмена данными, используется AJAX технология (Asynchronous Javascript and XML): создаётся объект XMLHttpRequest, который отправляет запросы на сервер и получает ответы.

Технология AJAX реализует не только запросы к серверу, но и динамическое изменение содержания страницы: динамическая подгрузка данных, живой поиск, формы, кнопки и т.д.

Данные между пользователем и сервером передаются в формате JSON, основанном на JavaScript. Закодированный текст представляет собой либо запись – неупорядоченный набор пар «ключ:значение», либо упорядоченный набор значений. В формате JSON значения могут принадлежать следующим типам данных: строка, число, массив, булева переменная или запись. Ключ может быть представлен только строкой.

4.3. Web-клиентское приложение

Web-клиентское приложение отображает данные, полученные с устройства класса PhaseProfile.

В верхней части экрана отображено название приложения, дата и время последнего обновления данных.

Основную часть экрана занимает TabPanel, имеющая три вкладки. На каждой вкладке расположены по два графика в плоскостях xx' и yy' . На первой вкладке отображаются фазовые координаты пучка (XPlane, XangPlane, YPlane и YangPlane). Дополнительно на графиках красным цветом показано «нулевое» положение центра тяжести пучка (DX0, DX0ang, DY0, DY0ang). На второй вкладке отображаются центрированные фазовые координаты (X_centre, Xang_centre, Y_centre и Yang_centre), эллипсы Флоке (ellips_X, ellips_Xang, ellips_Y и ellips_Yang) и «центрированное» «нулевое» положение центра тяжести пучка (DX0_center, DY0_center, DX0ang_center и DY0ang_center). На третьей вкладке отображаются нормированные фазовые координаты (X_norm, Y_norm, Xang_norm и Yang_norm) и «нулевое положение» в нормированных фазовых координатах (DX0_norm, DY0_norm, DX0ang_norm и DY0ang_norm).

Под TabPanel представлены зависимости положений центра тяжести пучка (R_x , R_y), углов движения (Θ_X , Θ_Y) и дробных частей частоты бетатронных колебаний ($D\Theta_X$, $D\Theta_Y$) от номера оборота для двух фазовых плоскостей xx' и yy' , а также таблица со значениями структурных функций.

Итоговый интерфейс программы представлен на рисунках 7 и 8.

Данные обновлены: 2021-04-29 17:30:08

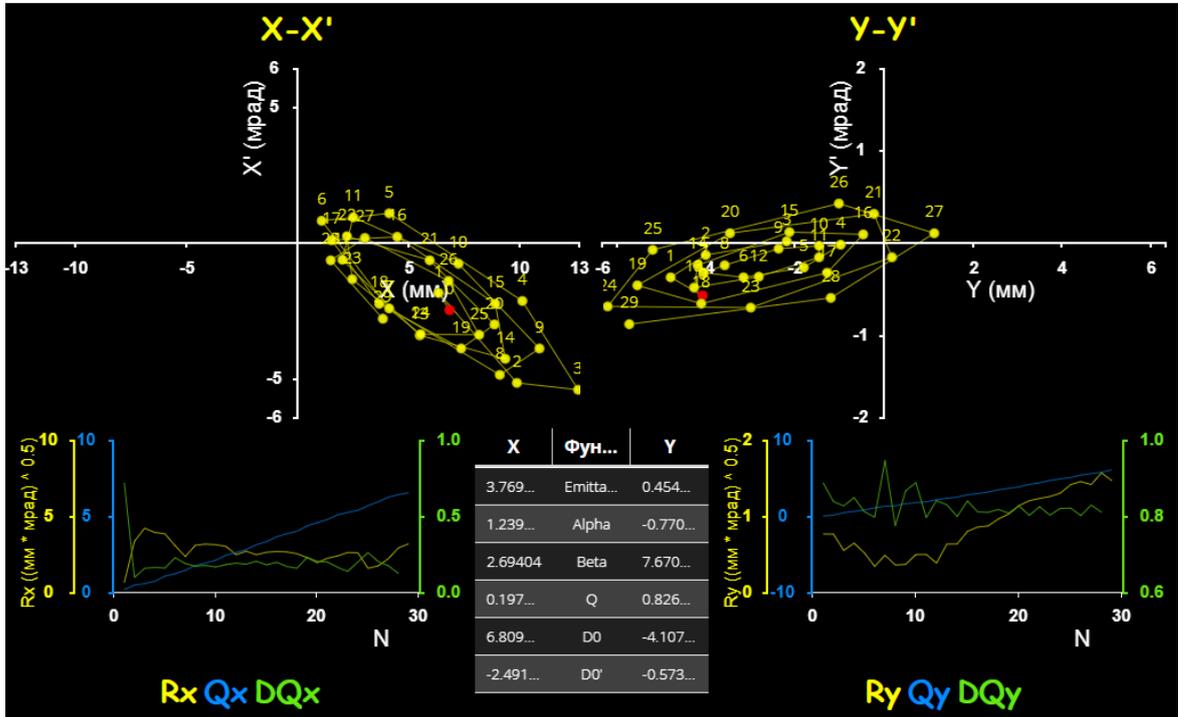


Рис. 7. Интерфейс Web-приложения PhaseProfile, вкладка «Фазовые координаты».

Данные обновлены: 2021-04-29 17:30:08

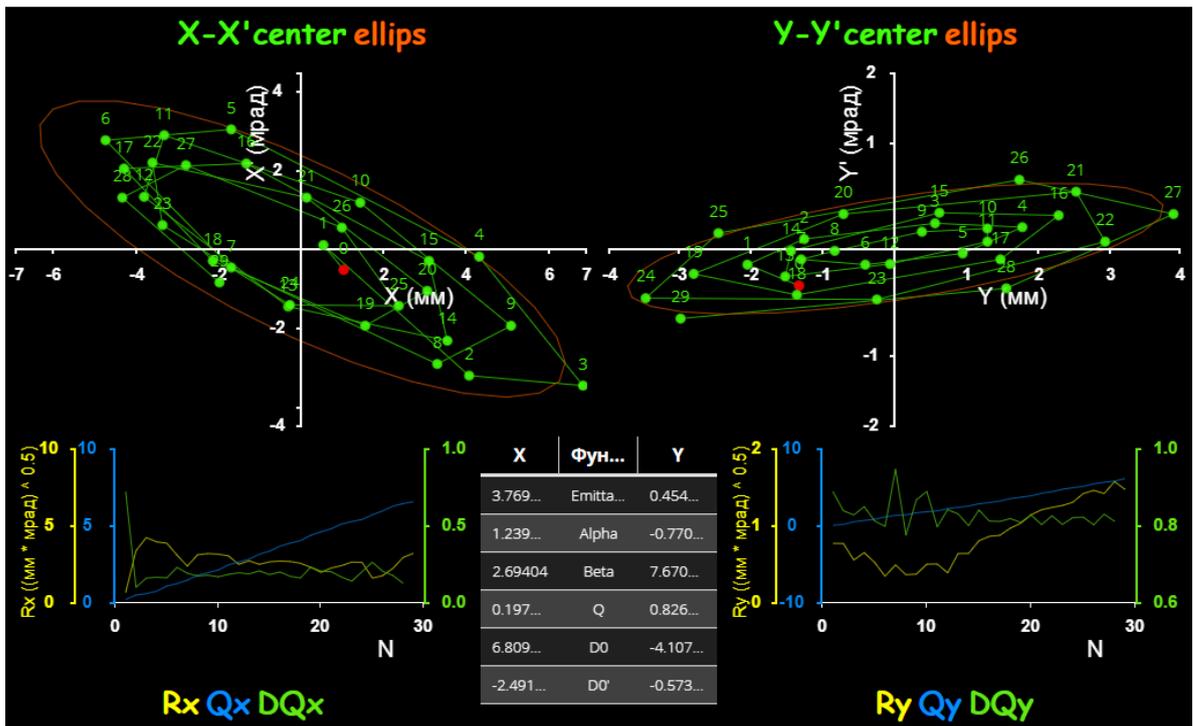


Рис. 8. Интерфейс Web-приложения PhaseProfile, вкладка «Центрированные фазовые координаты».

Глава 5. Тестирование

5.1. Тестирование на эмуляторах

Tango Controls является бесплатным программным обеспечением с открытым исходным кодом. Следовательно, пользователь самостоятельно определяет, какие компоненты он будет использовать, а значит, несет полную ответственность за качество создаваемого продукта. Тогда важным этапом разработки Tango устройств является их отладка и тестирование.

Для тестирования экземпляра класса PhaseProfile были созданы два устройства, имитирующие работу пикапов: LiberaTestDevice и LiberaTestDevice24. Первое устройство генерирует положение центра тяжести пучка в первом и последнем пикапах, записывает значения, отвечающие за первый пикап в свои атрибуты, а отвечающие за последний пикап – в атрибуты второго устройства. Таким образом, второе устройство выполняет роль «заглушки» – оно не генерирует собственных значений, а только отображает передаваемые ей данные. Такая логика взаимодействия исключает ошибки и асинхронную генерацию данных.

В результате тестирования была достигнута корректная работа созданного устройства, после чего оно было внедрено в систему управления ускорительного комплекса NICA.

5.2. Апробация ПО во время первого сеанса на Бустере NICA, набор данных для последующего анализа

В декабре 2020 года на Бустере NICA был проведён первый сеанс работы с циркулирующим пучком. Во время сеанса диагностика фазового портрета пучка использовалась со стартовой версией интерфейса и отработала несколько дней (Рис. 9). При этом тестировались различные режимы работы Бустера и основными задачами разработанной диагностики являлись проверка надёжности работы и сбор данных для последующей обработки. В результате разработанное программное обеспечение отработало весь сеанс без сбоев,

результаты измерения положения замкнутой орбиты, полученные с его помощью, позволили оптимизировать работу устройств системы инжекции пучка и коррекции замкнутой орбиты. Были записаны несколько десятков фазовых траекторий, отвечающих характерным режимам циркуляции.

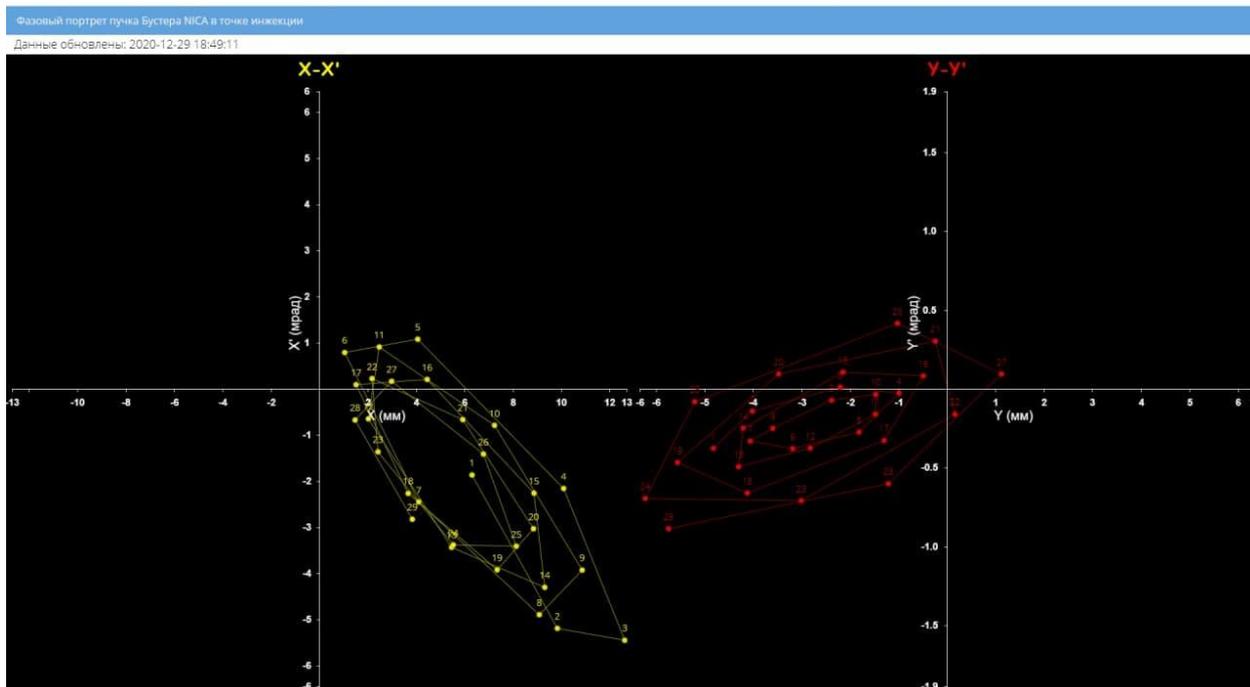


Рис. 9. Интерфейс Web-приложения PhaseProfile, во время первого сеанса на Бустере NICA.

5.3. Обработка результатов сеанса

По завершении сеанса сохранённые фазовые траектории были обработаны в соответствии с методикой, описанной в главе 2. Проанализированы возможные источники погрешностей при вычислении структурных функций ускорителя, одним из которых являются потери частиц на первых оборотах, приводящие к скачкообразному смещению центра тяжести пучка. Интерфейс был доработан до вида, удобного для оперативного анализа результатов в ходе следующего сеанса.

Выводы

В процессе выполнения данной работы была изучена методика диагностики замкнутой орбиты на участке инъекции на основе мнимого положения пучка и получены формулы для расчёта необходимых параметров.

Также была изучена система управления Tango Controls, введённая в эксплуатацию на ускорительном комплексе NICA. На примере написания устройства для диагностики циркулирующего пучка PhaseProfile был освоен процесс создания класса устройства в системе Tango Controls с использованием следующих утилит:

- Pogo – утилита, служащая для создания шаблона будущего класса;
- Jive – утилита, необходимая для работы с базой данных Tango, регистрации серверов и настройки устройств;
- Astor – утилита, необходимая для общего контроля за системой, например, задания последовательности активации подсистем в общей схеме.

В классе устройства PhaseProfile реализованы различные методы, отвечающие за корректное поведение устройства и расчёт фазового портрета, структурных функций и «нулевого» положения пучка на выходе септума.

Для отображения характеристик пучка и удобного наблюдения было разработано Web-клиентское приложение. Для этого были изучены библиотеки EXT JS 6.2.0, а также технология AJAX и формат данных типа JSON.

Для тестирования устройства были написаны эмуляторы процессоров положения адронного пучка Libera Hadron, являющиеся Tango устройствами. Разработанное программное обеспечение было протестировано на эмуляторах и отлажено.

Заключительным этапом стала апробация созданного программного обеспечения во время первого сеанса Бустера NICA в декабре 2020 года.

Заключение

В результате выполнения данной работы создано программное обеспечение в системе Tango Controls производящее диагностику циркуляции пучка в Бустере NICA на первых оборотах. Данное Tango устройство было протестировано, отлажено и апробировано во время первого сеанса на Бустере NICA.

Исходный код разработанного программного обеспечения в системе Tango Controls доступен в GitLab по ссылке [20].

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, ISSN:0375-9474, eISSN:1873-1554, Изд: Elsevier B.V., 2013, том 904-905
- [4] Смалюк В.В., Диагностика пучков заряженных частиц в ускорителях / Под ред. чл.-корр. РАН Н.С.Диканского. Новосибирск: Параллель, 2009. 294 с., ISBN 978-5-98901-063-9
- [5] Мешков И.Н., Сидорин А.О., Транспортировка пучков заряженных частиц: учебно-методическое пособие // Дубна, 2013, 192 стр.
- [6] Материалы школы физики ускорителей частиц U.S. Particle Accelerator School (USPAS), <https://uspas.fnal.gov/materials/materials-table.shtml>
- [7] J-M. Chaize *et al.*, TANGO – an object oriented control system based on corba // Proceedings of ICALEPCS1999. – Trieste, Italy, 1999, http://www.elettra.trieste.it/ICALEPCS99/proceedings/bklt_toc.html
- [8] Официальный сайт TANGO, <https://www.tango-controls.org>
- [9] 32nd TANGO Collaboration Meeting (5-7 June 2018), <https://indico.elibeams.eu/event/310/session/5/contribution/97>
- [10] Gorbachev E.V., Sedykh G.S. The equipment database for the control system of the NICA accelerator complex // Proceedings of ICALEPCS2013. – San Francisco, CA, USA, 2013, P. 1111-1113.
- [11] В. Андреев, В. Волков, Е. Горбачёв, А. Кириченко, Д. Монахов, С. Романов, Г. Седых, Т. Рукояткина, Развитие комплексной системы контроля Нуклотрона и ускорителей NICA, In proceedings of “NICA accelerating complex: problems and solutions — 2014”, Sozopol, Bulgaria

- [12] Gorbachev E. V., Andreev V. A., Kirichenko A. E., Monakhov D. V., Romanov S. V., Rukoyatkina T. V., Sedykh G. S., Volkov V. I. NUCLOTRON AND NICA CONTROL SYSTEM DEVELOPMENT STATUS // Proceedings of ICALEPCS2015, Melbourne, Australia, 2015, P. 437-440.
- [13] Andreev V. A., Volkov V. I., Gorbachev E. V., Isadov V. A., Kirichenko A. E., Romanov S. V., Sedykh G. S. TANGO standard software to control the Nuclotron beam slow extraction // Physics of Particles and Nuclei Letters, 2016. P. 605-608.
- [14] E. V. Gorbachev, V. A. Andreev, A. E. Kirichenko, D. V. Monakhov, S. V. Romanov, T. V. Rukoyatkina, G. S. Sedykh, V. I. Volkov, The Nuclotron and Nica control system development status., Physics of Particles and Nuclei Letters, ISSN:1547-4771, eISSN:1531-8567, Изд:МАИК Наука/Interperiodica distributed exclusively by Springer Science+Business Media LLC. 2016 том 13 журнал 5
- [15] В. Андреев, В. Волков, Е. Горбачев 1, В. Елкин, В. Исадов, А. Кириченко, Д. Монахов, Х. Назлев, С. Романов, Т. Рукояткина, Г. Седых, СИСТЕМА УПРАВЛЕНИЯ И ДИАГНОСТИКИ ПУЧКА БУСТЕРА NICA // Письма в ЭЧАЯ. 2020. Т. 17, № 4(229). С. 615–620
- [16] G. S. Sedykh, E. V. Gorbachev, and V. G. Elkin, RestDS2: the C++ TANGO REST API Implementation, Phys. Part. Nuclei Lett. 17, 604–606, 2020.
- [17] Документация библиотеки EXT JS 6.2.0, <https://docs.sencha.com/extjs/6.2.0>
- [18] Официальный сайт компании, Sencha, Inc., <https://www.sencha.com>
- [19] Официальный сайт продукта, Libera Hadron, <https://www.i-tech.si/products/libera-hadron>
- [20] Ссылка на репозиторий проекта, http://tangodevel.jinr.ru/git/beliaeva_ad/phaseprofile