САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

### Буньков Евгений Федорович

Выпускная квалификационная работа

**Применение генетического алгоритма в задаче оптимизации динамики пучка**

Уровень образования:

Направление *03.04.01*«Прикладные математика и физика» Основная образовательная программа *ВМ.5521.2019*

*«Математические и информационные технологии»*

|  |
| --- |
| Научный руководитель: доцент каф. ТСУЭФА, кандидат физ.-мат. наук, доцент  Рубцова И. Д. |
| Рецензент:  начальник отдела медицинской радиологии  НИЦ «Курчатовский институт» – ПИЯФ  Гранин Д.И. |

Санкт-Петербург 2021

Оглавление

[Введение 3](#_Toc71652545)

[Обзор литературы 7](#_Toc71652551)

[Постановка задачи 9](#_Toc71652552)

[Глава 1.Математическая модель продольного движение частиц 10](#_Toc71652553)

[1.1. Уравнения продольного движения 10](#_Toc71652554)

[1.2. Параметризация управляющих функций 12](#_Toc71652555)

[1.3. Численное моделирование динамики пучка 16](#_Toc71652556)

[Глава 2. Оптимизация динамики пучка 18](#_Toc71652557)

[2.1. Постановка задачи оптимизации 18](#_Toc71652558)

[2.2. Генетический стохастический алгоритм 19](#_Toc71652559)

[2.3. Результаты оптимизации 21](#_Toc71652560)

[Сравнительный анализ динамики пучка до и после оптимизации 25](#_Toc71652562)

[Заключение 25](#_Toc71652564)

[Список использованной литературы 27](#_Toc71652565)

### Введение

Настоящая работа посвящена исследованию и оптимизации динамики пучка электронов в ускорителе. Ускорители заряженных частиц широко применяются в различных областях [20]: в медицине, промышленности, сельском хозяйстве и в фундаментальных исследованиях. Однако сфера их использования постоянно расширяется; кроме того, возникает необходимость модернизации этих устройств в традиционных областях их применения. Этим обусловлена актуальность задач исследования и оптимизации динамики пучков в ускорителях.

Необходимым условием ускорения заряженных частиц является наличие элек­трического поля. Магнитное поле лишь меняет форму траектории частицы. Для осуществления ускорения частиц до высоких энергий необходимо либо создать сильное электрическое поле, либо обеспечить возможность частицам длительное время находиться под воздействием электрического поля (или проходить область воздействия многократно). Соответственно ускорители по способу создания электрических полей можно разделить на три большие группы [21]:

1.электростатические (однократное прохождение разности потенциалов)

2 индукционные (эл. поле создается переменным во времени магнитным полем)

3. резонансные (используется высокочастотное электрическое поле)

В зависимости от формы траектории частиц различают циклические и линейные ускорители. В циклических ускорителях частицы имеют круговые или спиральные орбиты, в линейных движутся по траекториям, близким к прямой.

В данной работе рассматривается движение пучка электронов в линейном волноводном ускорителе. Ускоритель такого типа представляет собой волновод, нагруженный диафрагмами [4,6,7,22]. В волноводе возбуждается бегущая электромагнитная волна. Продольная составляющая первой гармоники электрического поля имеет вид [4,10,11]:

### и называется ускоряющей волной. Здесь E0(z), υф(z) - амплитуда напряженности и фазовая скорость ускоряющей волны, изменяющиеся вдоль оси волновода; ω = 2πf = 2πc/λ - угловая частота; f и λ - соответственно частота, и длина волны в свободном пространстве; φ 0 - начальная фаза; t - время.

### Ясно, что ускорение электронов происходит при (-π + 2kπ,2kπ) (k = ±0,±1,±2,...)  при этом говорят, что частица находится в ускоряющей полуволне.

Основным принципом группировки и ускорения электронов является принцип автофазировки. В ускорителе созданы условия, обеспечивающие фазовую скорость ускоряющей волны меньше скорости света [8], причем обеспечивается текущее равенство фазовой скорости волны и продольной компоненты скорости электрона. Таким образом, в пучке всегда имеется синхронная частица, продольная скорость которой совпадает с фазовой скоростью ускоряющей волны. (Все характеристики этой частицы снабжают индексом с). Ускоритель обеспечивает постоянное возрастание фазовой скорости бегущей волны, поэтому синхронная частица все время ускоряется. Если , то частицы вблизи синхронной вовлекаются в колебательный процесс вокруг нее (то отставая, то опережая), причем вся совокупность таких частиц будет ускоряться вместе с синхронной [4,6,7,22]. Исследуем поведение других частиц, находящихся вблизи синхронной частицы.

Пусть фазе φс соответствует передний склон ускоряющей волны

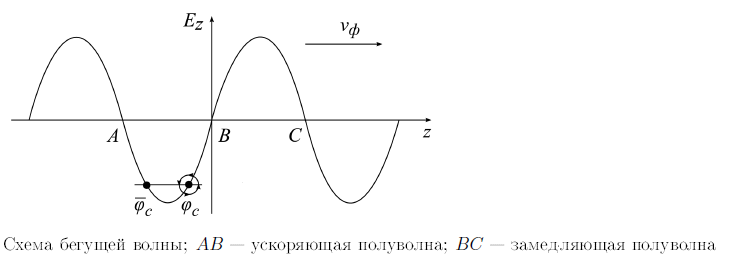


Рис.1

Тогда частицы, имеющие фазы φ < φс, получат большее ускорение, чем равновесная частица, и со временем опередят ее. Поэтому они перейдут в область пониженной напряженности ускоряющей волны, т.е. будут иметь фазы φ > φс. В этой области частицы будут получать меньшее приращение в скорости, чем равновесная частица. Следовательно, со временем они отстанут от нее, т.е. опять попадут в область повышенной напряженности поля, где будут получать ускорение, большее по сравнению с равновесной частицей.

Настоящая работа посвящена моделированию динамики частиц в данном ускорителе и оптимизации параметров ускорителя с целью получения требуемого качества пучка на выходе прибора. Работа опирается на опыт исследований в области анализа и оптимизации динамики заряженных пучков [10-16], проводимых на кафедре ТСУЭФА под руководством проф. Д.А. Овсянникова. Следует отметить, что данная тематика привлекает внимание многих исследователей [17-18], однако эта сфера исследования не исчерпана. Часто необходим более точный учет различных физических эффектов [19], адаптация устройства к конкретным условиям и целям эксплуатации и т.д.; при этом бывают востребованы новые математические модели и методы (примером может служить идея совместной оптимизации программного и возмущенных движений [7]). С учетом сложности задач управления ансамблями траекторий актуально внедрение в эту сферу современных методов оптимизации, в том числе стохастических. Именно этому направлению посвящена данная работа.

### Следует отметить, что наряду с непосредственной практической пользой повышения качества пучка, актуален сам опыт использования математических и численных алгоритмов оптимизации в рассматриваемых задачах. Этот опыт важен как для физиков, позволяя им правильно ориентироваться в море оптимизационных методов, так и для математиков, давая возможность апробации методов на практической задаче большой размерности.

### Задача оптимизации продольного движения электронов в линейном ускорителе на бегущей волне решается с применением генетического стохастического алгоритма, предложенного С.М. Ермаковым [33]. Основную сложность в использовании этого метода в задачах такого типа представляет большая размерность задачи [23]. Чтобы уменьшить размерность, в данной работе проводится параметризация амплитуды и фазовой скорости ускоряющей волны, и в рамках этой модели осуществляется применение алгоритма Ермакова, чего ранее не делалось.

В главе 1 работы введена параметризация амплитуды и фазовой скорости бегущей волны и осуществлен выбор начальных значений параметров. При данных параметрах выполнено численное моделирование продольной динамики пучка, представлены результаты, в том числе графически. Делается вывод о необходимости улучшения качества пучка, т.е. более удачного выбора параметров.

Во второй главе вводятся критерии качества, характеризующие среднюю выходную энергию частиц, энергетическую и фазовую неоднородность пучка, а также захват частиц в режим ускорения. Результирующий критерий качества представляет собой связку этих функционалов.

В разделе 2.2. дана идея генетического стохастического алгоритма [33] и представлен сам алгоритм. Программная реализация выполнена на языке PYTHON (код представлен в приложении).

Результаты численной оптимизации динамики пучка представлены в разделе 2.3. Даны в сравнении результаты численного моделирования динамики пучка при исходных и оптимизированных управляющих параметрах. Оптимизация позволила существенно повысить качество управляемого процесса.

### Обзор литературы

Исследованию динамики частиц в ускорителях посвящена обширная литература.

Прежде всего, следует отметить литературу физического характера, например [6,28]

В числе исследователей, разрабатывающих математические методы моделирования и оптимизации динамики заряженных пучков, следует отметить проф. Д.А. Овсянникова и его учеников. Именно представители данной школы рассматривают траектории частиц как ансамбль траекторий динамического управляемого процесса. Предложенный проф. Д.А. Овсянниковаым подход позволяет описать качество этого управляемого процесса функционалами различных типов и построить соответствующие математические методы оптимизации; во многих случаях получены аналитические выражения для градиента функционала качества. [7,10,11,13,15,24] Разработанные модели и методы реализованы в виде программных комплексов и были использованы в различных сферах науки, техники и хозяйства [15,17,25].

Оптимизация процессов в ускоряющих структурах может осуществляться и при использовании методов Монте-Карло [5] Среди них следует отметить многокритериальную оптимизацию [14, 18, 26, 27] и различные генетические стохастические алгоритмы[28-35]

Идея генетических стохастических алгоритмов следующая: моделируются поколения точек в пространстве управляющих параметров так, чтобы последовательность поколений с вероятностью 1 сходилась к точке глобального экстремума. Методы различаются способом перехода к следующему поколению. Случайный поиск с «памятью» [31,32] использует для первого поколения точек равноменое распределение в заданной области, а для последующих поколений – нормальное, причем центр и матрица ковариаций рассчитываются через положения точек предыдущего поколения. Алгоритм, предложенный С.М. Ермаковым (33), также использует нормальное распределение, параметры которого меняеются от поколения к поколению, но вычисления матрицы ковариаций не требуется. Поэтому данный алгоритм является простым в реализации и быстродействующим, что особенно важно в задачах с большой размерностью пространства параметров. В настоящей работе проводится апробация этого метода в задаче оптимизации параметров ускорителя при использовании параметризации амплитуды и фазовой скорости бегущей волны функцией арктангенс. (Заметим, что в работах [34,35] данный алгоритм использовался для другой математической модели пучка и при другой параметризации управляющих функций).

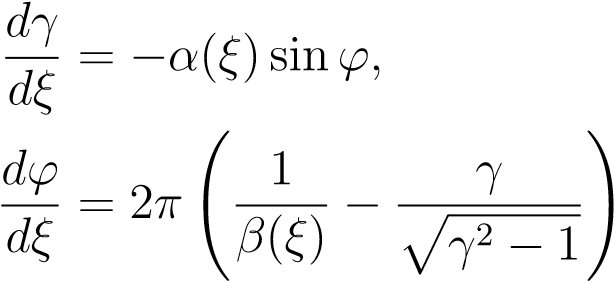
### Постановка задачи

1. Представить математическую модель, описывающую продольную динамику пучка частиц в линейном волноводном ускорителе;
2. Разработать программное обеспечение с целью реализации модели, а также проведения оптимизации динамики пучка;
3. Выполнить численное моделирование продольного движения и дать его анализ;
4. Поставить задачу оптимизации продольного движения электронов с целью получения требуемого качества пучка на выходе ускорителя;
5. Выполнить оптимизацию продольного движения, используя генетический стохасический алгоритм;
6. Провести сравнительный анализ динамики пучка до и после оптимизации

### Глава 1.Математическая модель продольного движение частиц

### Уравнения продольного движения

Уравнения продольного движения частиц имеет вид[2,4,10,11].



 — безразмерный параметр амплитуды напряженности ускоряющей волны. Фаза, частицы здесь отсчитывается от нуля ускоряющего поля; β (ξ) - приведенная фазовая скорость ускоряющей волны.

Уравнения, описывающие продольное движение электронов, используются при расчете линейных ускорителей электронов (ЛУЭ) и достаточно хорошо отражают реальное их движение при малой интенсивности пучка. Однако в силу существенной нелинейности этих уравнений исследование динамики продольного движения аналитическими методами ограничено.

Данная модель хорошо себя зарекомендовала при низкой интенсивности пучка. Обычно на входе распределение частиц по фазам считается равномерно, разбросом начальных энергий можно пренебречь. Впоследствии при взаимодействие с ускоряющей волной происходит формирование сгустков электронов; сгустки следуют друг за другом интервалом равным длине ускоряющей волны.

“Чтобы ускорить электроны наиболее эффективно, их следует поместить

в максимум ускоряющего поля. Но прежде их нужно хорошо сгруппировать

по фазам, так как вследствие того, что сгусток занимает по фазе некоторый

интервал, электроны с разными фазами получат разное приращение энергии. Поэтому если фазовый интервал, занимаемый сгустком, имеет большую

ширину, то на выходе ускорителя мы имеем электроны с разной энергией,

причем интервал этих энергий может быть очень большим. Но обычно на вы-

ходе ускорителя стремятся получить пучок частиц с узким энергетическим

спектром, который необходим для целого ряда физических исследований и

промышленных применений”[2,4,10,11].

Итак, обеспечение требуемых параметров динамики пучка, это довольно сложная задача. Управление динамикой пучка осуществляется посредством функций β (ξ), α (ξ), где (β (ξ) - приведенная фазовая скорость ускоряющей волны, а (ξ) - безразмерный параметр амплитуды). Для задания этих функций в работе[2,4,10,11] используется 84 параметра, чтобы уменьшить число параметров и тем самым облегчить процедуру оптимизации введем параметризацию управляющих функций посредством функций арктангенса.

### 1.2. Параметризация управляющих функций

Будем исследовать динамику пучка в ускорителе со следующими основными параметрами – W0=40кэв, λ = 0,1 м, l=8м

Прежде чем проводить численное исследование динамики пучка, следует задать функции β (ξ), α (ξ), характеризующие безразмерный параметр амплитуды поля ускоряющей волны и фазовую скорость этой волны. Было бы естественно привести значение этих функций через небольшие расстояния вдоль ускорителя, как это сделано в книге Овсянникова Д. А., однако понятно, что в этом случае данные функции будут зависеть от очень большого числа параметров, например в указанной книге число параметров 84, поскольку мы не используем направленные методы оптимизации, а намерены использовать методы поиска, при предполагаемой динамике оптимизации пучка, нам важно уменьшить число параметров оптимизации, т.е. снизить размерность пространства параметров. Поэтому используем другой метод параметризации, подспорьем в этом деле будет опыт в предыдущем исследование и физические исследования, из опыта исследований [2,4,10,11] известно, что обе указанные функции после некоторых колебаний в начале ускорителя, затем монотонно возрастают и в ускоряющей части прибора, когда пучок уже сгруппирован, сохраняют примерно постоянные значения. Примерно такими же свойствами обладает функция арктангенса, за исключением некоторых колебаний в начале ускорителя, но мы пренебрегаем этой возможностью, оставив на дальнейшие исследования, учет таких колебаний. Удобно выбрать для указанных функций следующую форму параметризации:

α(ξ, **ϴ**)= ϴ1arctg(ϴ2(ξ – ϴ3))+ ϴ4 (1)

β(ξ, **ϴ**)= ϴ1arctg(ϴ2(ξ – ϴ3))+ϴ4 (2)

Здесь **ϴ** =

Необходимо выбрать начальные значения параметров , чтобы осуществить численное моделирование динамики пучка, для того чтобы данная точка послужила отправной точкой для оптимизации. При подборе начальных параметров мы будем опираться на описанный выше опыт исследований и выбирать параметры так, чтобы полученные функции как можно меньше отличались от функций β (ξ), α (ξ), приведенных в работе Овсянникова Д. А.. Указанные функции β (ξ), α (ξ), приведены в таблице 1. Первая прикидка для получения параметров функций β (ξ)(2), α (ξ)(1), может быть осуществлена из анализа таблицы и графика из книги Овсянникова Д. А. с учетом свойств функции арктангенса.

Далее нам следует уточнить эти параметры, мы будем осуществлять приближение функций 1 и 2 к функциям β (ξ), α (ξ), приведенных в книге Овсянникова по методу наименьших квадратов.

min

ϴi ∈ Si, i=

min

ϴi ∈ Si, i=

Здесь , - управления из работы Овсянникова Д. А., , - функции (1) , (2), , - характерные точки структуры, в которых осуществляется сравнение функций

Итак, решалась следующая задача проводилась минимизация квадрата разности функций 1 и 2 и функций книги Овсянникова. Заметим что, функции 1 и 2 не могут совпадать с функциями β (ξ), α (ξ), использованными в книге Овсянникова, в силу характера параметризации, также полного совпадения сравниваемых функций быть не может, и мы проводили лишь с той целью, чтобы получить адекватную динамику пучка и начать оптимизацию с неплохой точки в пространстве параметров.

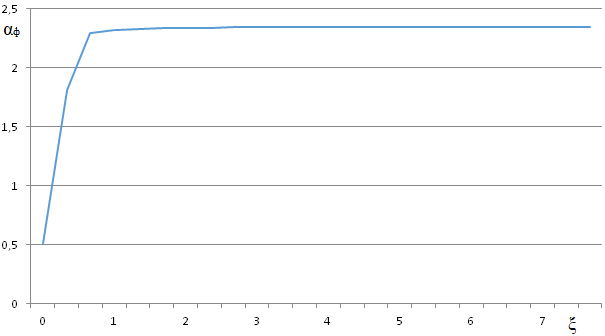
****

Рис.2 Исходное управление α(ξ)

В результате вычислений получаем α(ξ)= ϴ1arctg(ϴ2(ξ – ϴ3))+ ϴ4

ϴ1=0.605

ϴ2=9.64

ϴ3=1.25

ϴ4=1.4

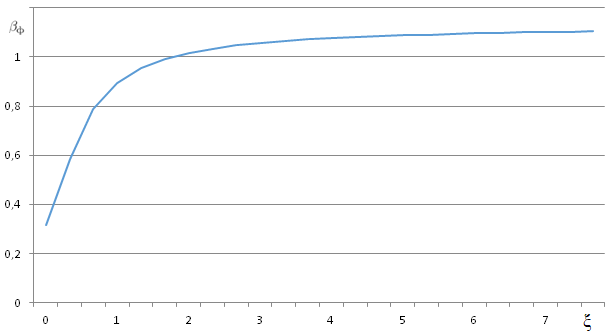
****

Рис.3 Исходное управление β(ξ)

β(ξ)= ϴ1arctg(ϴ2(ξ – ϴ3))+ϴ4

ϴ1=0.422

ϴ2=0.654

ϴ3=0.91

ϴ4=0.4

### Численное моделирование динамики пучка

**Параметры расчета**

Разработано программное обеспечение на языке python, осуществляющее численное моделирование динамики пучка, код представлен в приложении

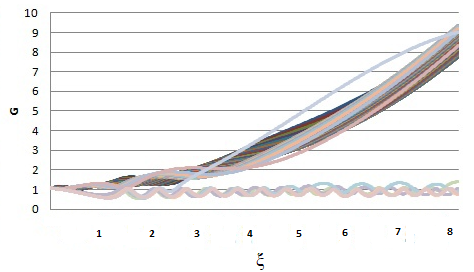
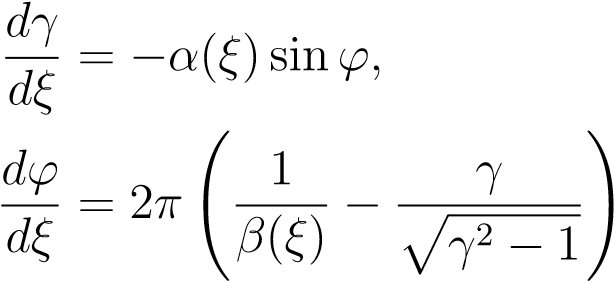


Рис.4 Изменение энергии модельных частиц вдоль ускорителя

Из графика видно, что часть частиц выпадает из режима ускорения и не набирает энергию. В результате расчета получена средняя энергия пучка на выходе =8,547 , разброс по энергиям =0,9 , количество частиц выпавших из режима ускорения =10,

Численно моделирование динамики пучка осуществляется в соответствие с уравнениями движениями:



Начальными условиями:

0=1.078

Начальная фаза частиц равномерно заполняет промежуток от – π до π

Численно интегрирование осуществляется с шагом h = 0,001

Количество модельных частиц = 50

Коэффициент захвата определяется соотношением

k = ∆φ /(2π)

k=5.026/2π=0,8

где ∆φ — интервал фаз в промежутке длиной 2π на входе ускорителя, занимаемый частицами, захваченными в режим ускорения.

δW = ∆ W/W

δW= 0,9/8,547 = 0,1

где ∆ W — интервал энергий, которые

имеют на выходе ускоренные электроны

W — средняя энергия электронов на выходе ускорителя.

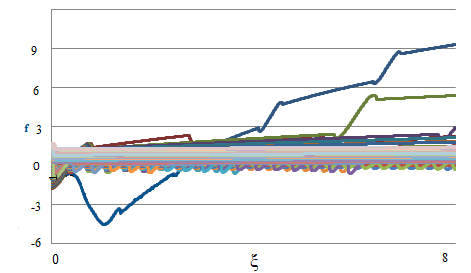


Рис. 5 Фазовые колебания электронов

Из графика так же видно, что много частиц выпало из режима ускорения, для остальных частиц разброс по фазе = 3,3

Вывод – данная динамика частиц нуждается в оптимизации. Поэтому переходим к разделу оптимизация.

### Глава 2. Оптимизация динамики пучка

### Постановка задачи оптимизации

Рассмотрим задачу оптимизации динамики пучка по управляющим параметрам u для обеспечения высокого качества группирования и ускорения пучка частиц. Обозначим через *u* вектор управляющих параметров:

Имеются несколько целей оптимизации; в соответствии с каждой целью предложим критерий качества, а затем сформируем связку критериев.

1. Первая цель состоит в том, чтобы максимизировать выходную энергию пучка. Соответствующий критерий качества задается формулой

Здесь A-соответствующая константа, выбираемая так, чтобы обеспечить неотрицательное значение критерия ; L-приведенная координата выхода устройства;

где – множество номеров модельных частиц, захваченных в режим ускорения, | | - общее число захваченных частиц, n - номер модельной частицы. - средняя приведенная энергия, рассчитывается по захваченным в режим ускорения частицам.

2.Цель минимизации разброса энергии пучка на выходе ускорителя может быть достигнута путем минимизации критерия

,

L – длина ускорителя

3. Для минимизации разброса фаз на выходе устройства вводится следующий критерий:

4. Оптимизация захвата частиц в режиме ускорения может быть сведена к минимизации следующего функционала:

Здесь, - поперечные сечения ускорителя.

Результирующий критерий выглядит следующим образом:

где - константы.

Принимаем коэффициенты

=1

=1

=0,01

*=0,1*

Таким образом, задача оптимизации динамики пучка формулируется как задача минимизации критерия (5) по управлениям u.

При данных параметрах, на основе численного моделирования получены следующие значения критериев

=23,83

=7,82

=563,01

=163,33

Результирующий критерий

K=53,6131

Далее применим генетический стохастический алгоритм [33]

### 

### Генетический стохастический алгоритм

Идея генетических стохастических алгоритмов состоит в следующем. Моделируются поколения точек в пространстве управляющих параметров так, чтобы последовательность поколений с вероятностью 1 сходилась к точке глобального экстремума. Методы различаются способом перехода к следующему поколению. Случайный поиск с «памятью» [34,35] использует для первого поколения точек равномерное распределение в заданной области, а для последующих поколений – нормальное, причем центр и матрица ковариаций рассчитываются через положения точек предыдущего поколения. Алгоритм, предложенный С.М. Ермаковым [33], также использует нормальное распределение, параметры которого меняются от поколения к поколению, но вычисления матрицы ковариаций не требуется. Поэтому данный алгоритм является простым в реализации и быстродействующим, что особенно важно в задачах с большой размерностью пространства параметров. Рассматриваемый алгоритм сходится к глобальному экстремуму для унимодальной функции.

**Алгоритм**

1. Обозначим через *u* вектор управляющих параметров: (значок Т обозначает транспонирование).
2. Нам надо сначала определить область поиска. Для этого определим промежуток изменения каждого параметра: . Здесь - это те значения параметров, которые мы подобрали. Эти промежутки выберем сначала интуитивно, а потом проверим, как будут вести себя функции и какова будет динамика пучка для управлений, соответствующих концам промежутков . Если будет получаться что-то совсем плохое – сузим промежутки. Окончательно получаем область поиска
3. Нулевое поколение: . Получение точек нулевого поколения.

Выбираем значение каждой компоненты как значение случайной величины, равномерно распределенной в . Получаем вектор . Ему соответствует точка в 8-мерной области *D.* Итак, получили случайную точку *.*

Повторяем процедурураз. Полученыслучайных точек в :  *.*

Получаем соответствующих значений критерия: *.*

Находим минимальное (из ) значение критерияи точку  *,* в которой оно достигается.

1. Подготовка к следующему поколению. Выбираем среди  *m* наилучших точек: *.* Заметим, что точкавходит вэти  *m* наилучших точек. (Ясно, что из двух точек лучше та, которой соответствует меньшее значение критерия). Готовим точки следующего поколения.

.

Здесь – реализация стандартной нормальной случайной величины

Другими словами:

точка ,

точка , ... ,

точка

При этом мы следим, чтобы каждая введенная точка принадлежала области . (проверяем каждую -ю компоненту точки на принадлежность промежутку . ) Если не принадлежит, отбрасываем такую точку и моделируем снова (т.е. берем новые реализации стандартной нормальной случайной величины ).

Пусть точек получено. Проверяем, все ли точки лучше . Если не все, то заменим наихудшую из на . (Тогда лучшая точка нового поколения будет гарантированно не хуже, чем .)

1. (увеличивам на 1 номер поколения)

Точки этого поколения: . Получим значений критерия: *.*

Находим минимальное (из ) значение критерияи точку *,* в которой оно достигается.

1. Проверка окончания: ( – заданная точность)

Если да, то перейти на ВЫХОД

Если нет, то перейти к п. 4.

1. ВЫХОД

Разработана программа, реализующая данный алгоритм

### Результаты оптимизации

Проведена численная оптимизация параметров ускоряющей структуры. Для достижения точности е=0,01, требуемая точность достигнута на 5ом поколении точек, примерное время счета – 163 минуты, M= 100, m=10, , =5

Получаем значения критериев после оптимизации:

=11

=0,3258

=3,41

=10,33

Результирующий критерий

K=12,36

Так же рассчитаем коэффициент захвата

k = ∆φ /(2π)

k=5.9/2π=0,92

Таблица 1.

|  |  |  |
| --- | --- | --- |
|  | До оптимизации | После оптимизации |
| Kзх | 0,8 | 0,92 |
| K1 | 23,83 | 11 |
| K2 | 7,82 | 0,3258 |
| K3 | 563,01 | 0,341 |
| K4 | 163,33 | 10,33 |
| Ksum | 53,6131 | 12,36 |
| W | 8,547 | 8,672 |
| ϴ1 | 0,605 | 0,768 |
| ϴ2 | 9,64 | 10,08 |
| ϴ3 | 1,25 | 1,12 |
| ϴ4 | 1,4 | 1,134 |
| ϴ5 | 0,422 | 0,567 |
| ϴ6 | 0,654 | 0,622 |
| ϴ7 | 0,91 | 0,789 |
| ϴ8 | 0,4 | 0,423 |
| ∆ W | 1,5 | 0,9 |
| ∆φ | 3,3 | 0,55 |

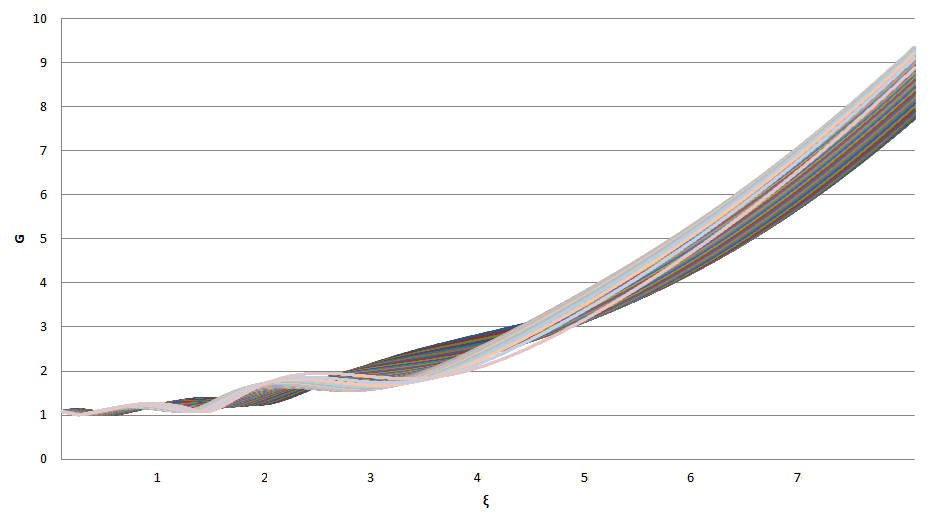
****

Рис. 6 График изменения энергии электронов в приведенных единицах

### 

Рис. 7 График фазовых колебаний электронов

### 

Рис. 8 Управление после оптимизации α(ξ)

### 

### Рис. 9 Управление после оптимизации β(ξ)

### Сравнительный анализ динамики пучка до и после оптимизации

### В результате оптимизации уменьшено количество частиц выпавших из режима ускорения с 10 до 4, средняя энергия возросла на 0,1, разброс по энергиям уменьшился с 1,5 до 0,9, разброс по фазам уменьшился с 3,3 до 0,55 , также произошло уменьшение каждого критерия.

### Заключение

Данная работа посвящена исследованию продольной динамики интенсивного квазипериодического пучка электронов в линейном волноводном ускорителе. Исследование проводится в два этапа: на первом осуществляется моделирование продольного движения частиц (без учета поперечного), без проведения оптимизации. На втором этапе выполняется оптимизация динамики пучка с применением генетического алгоритма.

Разработано программное обеспечение для исследования и оптимизации динамики электронов в ускорителе. Комплекс программ на языке python обеспечил возможность получения численных результатов и их графического представления.

Проведенная оптимизация является успешной апробацией генетического стохастического алгоритма в задаче глобального поиска управляющих параметров, обеспечивающих требуемое качество пучка на выходе ускорителя.

Работа по оптимизации динамики пучка может быть продолжена в следующих направлениях: модификация кода программы с целью повышения быстродействия; надлежащий выбор области поиска и параметров метода; сравнение с другими методами оптимизации; комбинация с направленными методами оптимзации.

### Список использованной литературы

1. Капчинский И.М. Динамика частиц в линейных резонансных ускорителях. – М. , 1966;
2. А. Д. Овсянников , А. Ю. Широколобов статья:

«Математическая модель оптимизации динамики пучка на бегущей волне».

1. Владимирова Л.В., Рубцова И.Д., Овсянников Д.А. Методы Монте-Карло в прикладных задачах. – СПб.: Изд-во ВВМ, 2015.
2. Овсянников Д.А., Егоров Н.В. Математическое моделирование систем формирования электронных и ионных пучков. СПб: Изд-во СПбГУ, 1998.
3. Владимирова Л.В., Овсянников Д.А., Рубцова И.Д. Методы МонтеКарло в прикладных задачах. – СПб.: Изд-во ВВМ, 2015. – 167 с.
4. Капчинский И.М. Теория линейных резонансных ускорителей.  
   Динамика частиц. – М.: Энергоиздат, 1982. 240с.
5. Овсянников А.Д. Математические модели оптимизации динамики  
   пучков. — СПб.: Изд-во ВВМ, 2014. — 181 с.
6. Вальднер О.А., Шальнов А.В. Основы радиотехники ускорителей. Генераторы и усилители сверхвысоких частот
7. Рубцова И.Д., Ломоносова Н.В., Чупрынина Т.А.. Исследование  
   продольной динамики интенсивного квазипериодического пучка в  
   линейном волноводном ускорителе // Вестник Санкт-Петербургского  
   государственного универсиета технологии и дизайна. Серия 1:  
   Естественные и технические науки. №3, 2017. С. 15-23
8. Овсянников Д.А. Моделирование и оптимизация динамики пучков заряженных частиц
9. Овсянников Д.А., Егоров Н.В. Математическое моделирование систем формирования электронных и ионных пучков
10. Овсянников Д.А., Рубцова И.Д., Козынченко В.А. Некоторые проблемы моделирования интенсивных пучков заряженных частиц в линейных ускорителях
11. Овсянников Д.А., Рубцова И.Д., Ломоносова Н.В. Исследование динамики интенсивного квазипериодического пучка в ускоряющей системе. IV Международная конференция «Лазерные, плазменные исследования и технологии» Лаплаз-2018: Сборник научных трудов. М.: НИЯУ МИФИ, 2018. С. 371-372.
12. Рубцова И.Д., Ломоносова Н.В., Чупрынина Т.А.. Исследование продольной динамики интенсивного квазипериодического пучка в линейном волноводном ускорителе // Вестник Санкт-Петербургского государственного универсиета технологии и дизайна. Серия 1: Естественные и технические науки. №3, 2017. С. 15-23.
13. Ovsyannikov D.A., Ovsyannikov A.D., Antropov I.V., Kozynchenko V.A. Software complex BDO-RFQ //Proceedings of III International Conference “Stability and Control Processes” in Memory of V.I.Zubov (SCP), IEEE, 2015. Pp. 335-337. DOI: 10.1109/SCP.2015.7342132
14. Vladimirova L V 2016 Multicriterial Approach to Beam Dynamics Optimization Problem II Conference on Plasma&Laser Research and Technologies (2016), Journal of Physics: Conference Series 747 No 1 012070 <http://iopscience.iop.org/1742-6596/747/1/012070>
15. Altsybeyev V. V., Ovsyannikov D. A. Optimization of beam parameters in APF channel // 27rd International Linear Accelerator Conference, LINAC 2014 - Proceedings. –– 2014. –– P. 722–725.
16. Vladimirova L V 2016 Multicriterial Approach to Beam Dynamics Optimization Problem II Conference on Plasma&Laser Research and Technologies (2016), Journal of Physics: Conference Series 747 No 1 012070 <http://iopscience.iop.org/1742-6596/747/1/012070>
17. Rubtsova I.D., Ovsyannikov D.A. Intense quasiperiodic beam dynamics in accelerating system: mathematical model and optimization method // III International Conference on Laser and Plasma Researches and Technologies (2017), Journal of Physics: Conference Series, Vol. 941, No 1, 012092 (2018); [http://iopscience.iop.org/article/10.1088/1742-6596/941/1/012092](http://iopscience.iop.org/issue/1742-6596/941/1)
18. Гольдин Л.Л. Физика ускорителей. М.: Наука, 1983. 144с.
19. Вальднер О.А., Власов А.Д., Шальнов А. В. Линейные ускорители. — М., 1969. — 248 с.
20. Овсянников Д.А. Моделирование и оптимизация динамики пучков заряженных частиц. — Л., 1990. —312 с.
21. Zhdanova, A.Y., Rubtsova, I.D.: Modeling and optimization of intense beam dynamics in traveling-wave field. In: Proc. of V International Conference on Laser&Plasma researches and technologies (LaPlas-2019), part 2, Moscow, National Research Nuclear University MEPhI, 160–161, (2019))
22. Rubtsova I.D., Ovsyannikov D.A. Intense quasiperiodic beam dynamics in accelerating system: mathematical model and optimization method // III International Conference on Laser and Plasma Researches and Technologies (2017), Journal of Physics: Conference Series, Vol. 941, No 1, 012092 (2018); [http://iopscience.iop.org/article/10.1088/1742-6596/941/1/012092](http://iopscience.iop.org/issue/1742-6596/941/1)
23. Владимирова Л. В., Овсянников Д.А., Свистунов Ю.А. Оптимизация захвата частиц в ускорение при больших токах в ЛУЭ //Вопросы атомной науки и техники. Сер. Электрофизическая аппаратура. 1993.Вып. 26. С.54-60.
24. Владимирова Л.В. Многокритериальная оптимизация динамики пучков. Известия Иркутского государственного университета, 2014. Серия: Математика, 2014, №7, с.3-18.
25. Соболь И.М., Статников Р.Б. Выбор оптимальных параметров в задачах со многими критериями. М.: Наука, 1981. 112 с.
26. Власов А.Д. Теория линейных ускорителей. — М., 1965. — 308 с.
27. Kennedy, J.: Particle swarm optimization. In: Proc. of IEEE Int. Conf. on Neural Networks  
    IV, 1942–1948 (1995)
28. Balabanov, M.Yu.: On initial control choice in charged particles beams dynamic optimization  
    problems. Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science.  
    Control Processes, **3**, 93–99 (2010).
29. Ermakov, S.M., Mitioglova, L.V.: On extreme search method based on the estimation of the covariance matrix. Automation and Computer Engineering, **5**, 38–41 (1977)
30. Vladimirova, L.V., Ermakov, S.M.: Random Search Method with a “Memory” for  
    Global Extremum of a Function. In: Proc. of 10th International Workshop on Simulation and Statistics. Universitat Salzburg, Salzburg, Workshop booklet, 89, (2019) <https://datascience.sbg.ac.at/SimStatSalzburg2019/SimStat2019BoA.pdf>
31. Ermakov, S.M., Semenchikov, D.N.: Genetic global optimization algorithms.  
    Communications in Statistics, Part B: Simulation and Computation (2019),  
    <https://doi.org/10.1080/03610918.2019.1672739>
32. Vladimirova, L.V., Zhdanova, A.Y., Rubtsova, I.D.: Application of the genetic global search algorithm in beam dynamics optimization problem. In: Proc. of VI International Conference on Laser&Plasma researches and technologies (LaPlas-2020), part 1, Moscow, National Research Nuclear University MEPhI, 91–92 (2020)

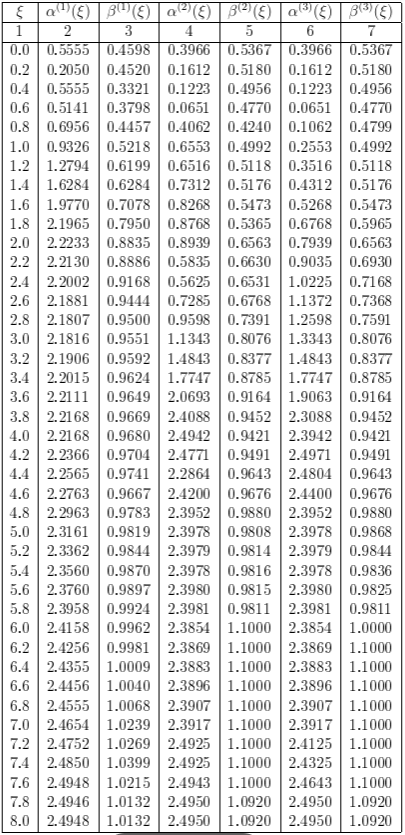
## L.V. Vladimirova, A.Y. Zhdanova, I.D.Rubtsova, N.S. Edamenko. Genetic Stochastic Algorithm Application in Beam Dynamics Optimization Problem (IV Stability and Control Processes Conference in memory of Prof. Vladimir Zubov, Saint Petersburg, Russia, 5-9

## October 2020; is being published)

**Приложение**

Приложение А

Табл.1

****

Приложение Б

from scipy.stats import norm

import matplotlib.pyplot as plt

import numpy as np

import math

import random

def FuncGen (IdolList):

rez = []

for i in range (0,8):

rez.append(random.uniform(IdolList[i] - 0.3, IdolList[i] + 0.3))

return rez

def FuncArctG (i, q1 , q2, q3, q4): #Управляющая функция G

rez = q1 \* math.atan(q2 \* (i - q3)) + q4

return rez

def FuncArctF (i, q1 , q2, q3, q4): #Управляющая функция F

rez = q1 \* math.atan(q2 \* (i - q3)) + q4

return rez

def FuncG (i, f, q1 , q2, q3, q4): #Функция расчета очередного G

rez = (-1) \* FuncArctG(i, q1 , q2, q3, q4)\*math.sin(f)

return rez

def FuncF (i, g, q1 , q2, q3, q4): #Функция расчета очередного F

if (g < 1): #При потере частицей энергии сообщаем ей дополнительную

g = 1.078

rez = 2 \* math.pi \* (1 / FuncArctF(i, q1 , q2, q3, q4) - g / ((g\*\*2 - 1)\*\*0.5))

return rez

def nextGenKritCalculate (NextGenList):

F = []

#список F

G = []

#список

CurentList = []

#Список тет на текущем круге проверок

Glast = []

Flast = []

Ksumm = []

TetaList = []

NextGenCalcFinish = []

for check in range (0, 100):

CurentList = NextGenList[check]

f0 = -3

#Задание начального значения F

f0delta = 0.12

h = 0

#Задание первого шага

Glast.clear()

Flast.clear()

Gsumm = 0

Gaver = 0

Fsumm = 0

Faver = 0

K1 = 0

K2 = 0

K3 = 0

K4 = 0

for iter in range (0, 51):

#Цикл для создания 50 проверок на меняющиеся значения f0

F.clear()

#Обнуление списка F

G.clear()

#Обнуление списка G

F.append(f0)

G.append(1.078)

for delta in range (0, 8001):

#Цикл симулирующий 8000 шагов

if F[delta] < -math.pi:

K4 += (F[delta] + math.pi)\*\*2

if F[delta] > math.pi:

K4 += (F[delta] - math.pi)\*\*2

G.append(G[delta] + FuncG(h, F[delta], CurentList[0], CurentList[1], CurentList[2], CurentList[3]) \* 0.001)

F.append(F[delta] + FuncF(h, G[delta], CurentList[4], CurentList[5], CurentList[6], CurentList[7]) \* 0.001)

h += 0.001

#Наращивание шага

Gsumm += G[delta]

Fsumm += F[delta]

Glast.append(G[delta])

Flast.append(F[delta])

f0 += f0delta

#Изменение стартового F

Gaver = Gsumm / iter

Faver = Fsumm / iter

K1 = A\*\*2 - Gaver\*\*2

for iter in range (0, 51):

K2 += (Glast[iter] - Gaver)\*\*2

K3 += (Flast[iter] - Faver)\*\*2

K2 = K2 / 50

K3 = K3 / 50

K4 = K4 / 400\_000

Ksumm.append(K1 + K2 + K3 + K4)

print("Iteration", check, "complete. Ksumm =", Ksumm[check])

return Ksumm

def NextGenGenerator (Ksumm, TetaList):

KsummMinList = []

ListNumberMin = []

BestTetas = []

SBTG = []

IterableList = []

IterableListNull = []

NextGenGeneratorFinish = []

NewGenTetas = []

MinTetasPreviosGeneration = []

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

for i in range (0, 100):

NewGenTetas.append(i)

for iter in range (0,10):

i = 0

max\_count = 101

KsummMinList.append(min(Ksumm))

while i < max\_count:

if KsummMinList[iter] == Ksumm[i]:

Ksumm[i] = 99999

max\_count = 0

ListNumberMin.append(i)

BestTetas = TetaList[i]

MinTetasPreviosGeneration.append(TetaList[i])

print("Tetas list number", iter+1, "is: q1 =", BestTetas[0], "q2 =", BestTetas[1], "q3 =", BestTetas[2], "q4 =", BestTetas[3], "q5 =", BestTetas[4], "q6 =", BestTetas[5], "q7 =", BestTetas[6], "q8 =", BestTetas[7])

i += 1

IterableListNull = TetaList[ListNumberMin[0]]

for NextGenForm in range (0, 100):

for iter in range (1,10):

IterableList = TetaList[ListNumberMin[iter]]

for i in range (0, 8):

a=(random.normalvariate(0, 1))

SBTG[i] += (IterableList[i] - IterableListNull[i]) \* a

for i in range (0, 8):

SBTG[i] = SBTG[i] / 10 + IterableListNull[i]

NewGenTetas[NextGenForm] = SBTG.copy()

NextGenGeneratorFinish.append(NewGenTetas)

NextGenGeneratorFinish.append(KsummMinList[0])

NextGenGeneratorFinish.append(MinTetasPreviosGeneration[0])

NextGenGeneratorFinish.append(TetaList[ListNumberMin[0]])

print("")

return NextGenGeneratorFinish

F = []

#список F

G = []

#список G

IdolList = []

#Задание списка эталонных тет

CurentList = []

#Список тет на текущем круге проверок

Glast = []

Flast = []

Ksumm = []

TetaList = []

KsummMinList = []

ListNumberMin = []

BestTetas = []

SBTG = []

IterableList = []

IterableListNull = []

NextGenGeneratorFinish = []

NewGenTetas = []

MinTetasPreviosGeneration = []

IdolList.append(0.605)

IdolList.append(9.64)

IdolList.append(1.25)

IdolList.append(1.4)

IdolList.append(0.422)

IdolList.append(0.654)

IdolList.append(0.91)

IdolList.append(0.47)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

SBTG.append(0)

A = 15

for check in range (0, 100):

CurentList = FuncGen(IdolList)

TetaList.append(CurentList)

f0 = -3

#Задание начального значения F

f0delta = 0.12

h = 0

#Задание первого шага

Glast.clear()

Flast.clear()

Gsumm = 0

Gaver = 0

Fsumm = 0

Faver = 0

K1 = 0

K2 = 0

K3 = 0

K4 = 0

for iter in range (0, 50):

#Цикл для создания 50 проверок на меняющиеся значения f0

F.clear()

#Обнуление списка F

G.clear()

#Обнуление списка G

F.append(f0)

G.append(1.078)

for delta in range (0, 8000):

#Цикл симулирующий 8000 шагов

if F[delta] < -math.pi:

K4 += (F[delta] + math.pi)\*\*2

if F[delta] > math.pi:

K4 += (F[delta] - math.pi)\*\*2

G.append(G[delta] + FuncG(h, F[delta], CurentList[0], CurentList[1], CurentList[2], CurentList[3]) \* 0.001)

F.append(F[delta] + FuncF(h, G[delta], CurentList[4], CurentList[5], CurentList[6], CurentList[7]) \* 0.001)

h += 0.001

#Наращивание шага

Gsumm += G[delta]

Fsumm += F[delta]

Glast.append(G[delta])

Flast.append(F[delta])

f0 += f0delta

#Изменение стартового F

Gaver = Gsumm / iter

Faver = Fsumm / iter

K1 = A\*\*2 - Gaver\*\*2

for iter in range (0, 50):

K2 += (Glast[iter] - Gaver)\*\*2

K3 += (Flast[iter] - Faver)\*\*2

K2 = K2 / 50

K3 = K3 / 50

K4 = K4 / 400\_000

Ksumm.append(K1 + K2 + K3 + K4)

print("Iteration", check, "complete. Ksumm =", Ksumm[check])

NextGenGeneratorFinish = NextGenGenerator(Ksumm, TetaList)

KminPreviosGeneration = NextGenGeneratorFinish[1]

MinTetasPreviosGeneration = NextGenGeneratorFinish[2]

NewGenTetas = NextGenGeneratorFinish[0]

Flag = False

Generation = 0

while Flag != True:

Generation += 1

Ksumm = nextGenKritCalculate(NewGenTetas)

NextGenGeneratorFinish = NextGenGenerator(Ksumm, NewGenTetas)

NewGenTetas = NextGenGeneratorFinish[0]

KminCurrentGeneration = NextGenGeneratorFinish[1]

BestTetasCurrentGen = NextGenGeneratorFinish[3]

rez = abs(KminCurrentGeneration - KminPreviosGeneration) / KminPreviosGeneration

if KminCurrentGeneration > KminPreviosGeneration:

NewGenTetas[0] = BestTetasCurrentGen

print("We have worst Tetas then previous Generation")

if rez < 0.01:

Flag = True

print("Finally success! Generation number", Generation, "has less defference then 0.01! It's", rez)

print("Optimal Tetas:", BestTetasCurrentGen)

else:

print("Unforunately, generation", Generation, "too bad, because criterion difference is", rez, ". Starting new circle...")

KminPreviosGeneration = KminCurrentGeneration

MinTetasPreviosGeneration = NextGenGeneratorFinish[2]