

Санкт–Петербургский государственный университет

***САВЕЛЬЕВ Игорь Вадимович***

**Выпускная квалификационная работа**

***Аналитическое и численное исследование  
автоматических систем управления с гистерезисом***

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

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

Профиль «Прикладная математика, информатика и процессы управления»

Научный руководитель:

профессор, кафедра высшей математики,  
д.ф. - м.н. Камачкин Александр Михайлович

Рецензент:

доцент, кафедра космических технологий  
и прикладной астродинамики,  
к.ф. - м.н. Степенко Николай Анатольевич

Санкт-Петербург

2020 г.

# Содержание

<b>Введение</b> . . . . .	3
<b>Постановка задачи</b> . . . . .	4
<b>Обзор литературы</b> . . . . .	5
<b>Глава 1. Обзор альтернатив</b> . . . . .	6
1.1. Альтернативы . . . . .	6
1.2. Simulink . . . . .	6
1.3. Симинтех . . . . .	7
1.4. Winset . . . . .	7
1.5. Итог . . . . .	8
<b>Глава 2. Исследуемая модель</b> . . . . .	9
2.1. Математическая модель системы . . . . .	9
2.2. Алгоритм численного моделирования . . . . .	10
<b>Глава 3. Программная реализация</b> . . . . .	13
3.1. Обоснование выбора языка программирования . . . . .	13
3.2. Архитектура программного продукта . . . . .	14
3.3. Реализация архитектуры . . . . .	16
<b>Глава 4. Моделирование динамических систем</b> . . . . .	22
<b>Выводы</b> . . . . .	31
<b>Заключение</b> . . . . .	32
<b>Список литературы</b> . . . . .	33

## Введение

Анализ периодических режимов в нелинейных системах управления является ключевой задачей в теории нелинейных колебаний [1]. На данный момент нет универсальных методов, позволяющих аналитически изучать такие системы. Данная область исследований остается актуальной в связи с развитием космической отрасли, морских систем управления, в которых зачастую математическая модель системы содержит нелинейные функции вида идеального и неидеального реле, испытывающих колебательное возмущение [2]. Наиболее перспективным и эффективным методом анализа таких систем является метод фазового пространства (плоскости), который позволяет исследовать системы, которые затруднительно анализировать обычными методами, то есть частотными и временными. Метод пространства состояний делает динамику системы наглядной.

Цель моделирования заключается в получении фазового пространства, которое может содержать периодическое решение в виде предельного цикла. Существование таких решений и описание соответствующих необходимых условий производились в ряде исследований [1][2][3]. Используя метод декомпозиции, возможно представить фазовое пространство порядка, большего чем три, в виде совокупности фазовых пространств второго и третьего порядка в условиях некоторых сечений пространства параметров исходной системы. Это позволяет иметь достоверный набор эталонных подсистем для контроля за динамикой исходной системы, учитывая, как правило, очень сложный характер движений, возникающий в многомерных системах [4]. В связи с этим, рассматриваются фазовые пространства не выше третьего порядка.

## **Постановка задачи**

Целью данной работы является разработка методов для исследования автоматических систем управления (АСУ) с гистерезисом и их программная реализация. Для достижения данной цели были поставлены следующие задачи:

- Выполнить обзор существующих методов.
- Выбрать метод исследования АСУ с гистерезисом.
- Разработать архитектуру программы.
- Исследовать требуемую систему.
- Показать практические результаты работы программы.

## Обзор литературы

Изучение САУ и дифференциальных уравнений с нелинейной правой частью имеет длинную историю [5]. Исследования в области совершенствования методов оптимизации производились в первую очередь такими учеными как Л.С. Понтрягин и А.А. Фельдбаум. В работе Понтрягина [6] был разработан «принцип максимума», говорящий о оптимальном по быстродействию управлении при использовании нелинейности в виде реле в управляющем сигнале САУ. Точные методы исследования нелинейных систем управления получили значительное развития благодаря работам отечественных ученых А.М. Ляпунова, А.А. Андропова, Н.Г. Четаева, Н.Н. Красовского и других [7]. Развитием для многомерных фазовых пространств нелинейных систем можно назвать метод сечений пространства параметров. Исследованиями в этой области занимались ученые Р.А. Нелепин, Е.П. Попов и другие [8]. В современные дни интерес представляет исследование нелинейных систем под воздействием периодического возмущающего воздействия. Такие системы описываются дифференциальными уравнениями с нелинейной правой частью. Ряд исследователей, в том числе А.М. Камачкин, В.Н. Шамберов, В.В. Евстафьева, как доказали существование периодических решений таких систем и уравнений [1][3][5], так и предложили аналитические методы нахождения такого решения [2][9][10].

# Глава 1. Обзор альтернатив

## 1.1 Альтернативы

В сфере практической реализации моделирования доминируют в основном проприетарные коммерческие программы, такие как пакет «Simulink» - часть пакета компьютерной математики «Matlab», отечественная разработка среды моделирования технических систем «Симинтех», а также отдельные специализированные программы, например «Winset» [11]. Кроме этих программ, существует ряд других, но принципиальными отличиями от представленных они не обладают.

## 1.2 Simulink

Данный программный комплекс является графической средой для моделирования, симуляции и анализа различных динамических систем. Графический интерфейс представляет собой ряд блок-диаграмм и ряд настраиваемых библиотек. Simulink предлагает плотную интеграцию с остальным окружением Matlab и поставляется вместе с ним. К преимуществам Simulink можно отнести:

- Широкий спектр готовых блоков и библиотек, который можно использовать после настройки.
- Интеграция в Matlab позволяет использовать внутренний высокоуровневый язык программирования, адаптированный для работы с матрицами и упрощенный для ученых-математиков.
- Большое количество учебных пособий и документации, позволяющих быстро и эффективно обучиться работе с Simulink.

К недостаткам можно отнести:

- Проприетарная лицензия, стоимость которой довольно высока.
- Последние версии программы требовательны к мощности оборудования, на котором запущены.

- Закрытость экосистемы в целом усложняет переход от использования Simulink к другим системам.

### **1.3 Симинтех**

«Симинтех» – отечественный аналог Simulink, представляющий собой графическую среду, позволяющую производить моделирование различных технических систем с помощью блок-диаграмм. К преимуществам можно отнести:

- Скорость моделирования. По заявлению разработчиков, в ряде случаев при моделировании нелинейных систем, «Симинтех» в несколько раз производительней аналогов.
- Стоимость. Цена лицензии «Симинтех» зависит от количества услуг, предоставляемых компанией, а для учебных целей в высших учебных заведениях лицензия бесплатна.
- Отечественная разработка.

К недостаткам можно отнести:

- Малое количество учебных пособий и материалов. Программа обладает справочной информацией, но она не всегда легкоусвояемая.
- Отсутствие простой интеграции языков программирования, что несколько осложняет работу с полученными данными.
- Проприетарность.

### **1.4 Winset**

Winset – авторская программа, разработанная для анализа набора задач из физики, механики, биологии, химии, экономики и других областей путем построения инвариантных множеств соответствующих динамических систем (фазовых кривых, резонансных структур, странных аттракторов, фракталов, паттернов и др.). К преимуществам можно отнести:

- Малый объём и производительность.
- Бесплатное распространение.

К недостаткам можно отнести:

- Узкая область применения.
- Проприетарность.
- Малое количество справочной информации и учебных пособий.

## 1.5 Итог

В результате анализа различных программ, предназначенных для моделирования, в том числе, нелинейных автоматических систем управления, в связи с их недостатками, создадим свою программу, главными преимуществами которой будут являться:

- Открытость исходного кода, а потому возможность изучения и улучшения программы множеством людей.
- Отсутствие других элементов, не рассчитанных на моделирование нелинейных автоматических систем управления, и, как следствие, упрощение как интерфейса, так и внутренней логики программы.

Данные преимущества возможно реализовать, поскольку правая часть систем ОДУ рассматриваемых систем дает фазовую траекторию, состоящую из кусков разных фазовых кривых. Это позволяет упростить моделирование. К недостаткам можно отнести:

- Малый функционал по сравнению с аналогами.
- Узкую направленность.

Однако для целей и задач данной работы они представляются несущественными.

## Глава 2. Исследуемая модель

### 2.1 Математическая модель системы

Обобщенную модель динамической системы с нелинейностью вида реле с гистерезисом можно представить как [1]:

$$\dot{X} = AX + Bu(\sigma) + Kf(t), \quad \sigma = (C, X) \quad (1)$$

Здесь  $A[n \times n]$  – вещественная матрица, описывающая объект управления;  $B[n \times 1]$ ,  $K[n \times 1]$ ,  $C[n \times 1]$  – постоянные вещественные векторы, описывающие коэффициенты усиления перед соответствующими компонентами системы;  $X[n \times 1]$  – вектор состояний системы ( $X \in E^n$ ), причем  $E^n$  –  $n$ -мерное евклидово пространство.  $u(\sigma(t))$  – скалярная функция, описывающая нелинейность типа неидеального реле с пороговыми числами  $l_1, l_2$  и выходными числами  $m_1, m_2$ . Примем, что гистерезисная нелинейность является запаздывающей, а значит  $l_1 < l_2$  и  $m_1 < m_2$ . Функция  $u(\sigma(t))$  определена для  $t \geq 0$  в классе непрерывных функций, может принимать только два значения  $m_1$  и  $m_2$  и задается следующим образом. При  $\sigma(t) \leq l_1$  выполнено равенство  $u(\sigma(t)) = m_1$ , а при  $\sigma(t) \geq l_2$  – равенство  $u(\sigma(t)) = m_2$ . Если же  $l_1 < \sigma(t) < l_2$  при всех  $t_1 < t < t_2$  и при этом  $\sigma(t_1) = l_1$  или  $\sigma(t_1) = l_2$ , то положим  $u(\sigma(t)) = u(\sigma(t_1))$ . Наконец, если  $l_1 < \sigma(t) < l_2$  при всех  $0 \leq t < t_2$ , то положим  $u(\sigma(t)) = u_0$ , где  $u_0$  — одно из чисел  $m_1$  и  $m_2$ . В последнем случае, динамика системы зависит от начального положения реле, соответствующего значению  $u_0$ .  $f(t)$  – функция, описывающее внешнее воздействие на систему. В общем виде она рассматривается как непрерывная периодическая функция. Представленную выше нелинейную зависимость в виде реле с гистерезисом можно представить графически:

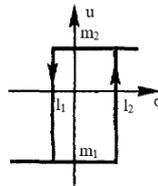


Рис. 1: Статическая характеристика реле с гистерезисом

Как видно на рисунке 1, отрезки прямых, соответствующих  $m_1$  и  $m_2$ , представляют собой состояние реле, определенное неоднозначно, что в целом осложняет моделирование системы. Получение аналитического решения такой системы возможно лишь для частных случаев. Численными методами осуществимо моделирование при помощи разбиения системы на дифференциальные уравнения, которые возможно проинтегрировать по отдельности. Хотя численное решение и подразумевает неточность полученных данных, на сегодняшний день вычислительные мощности технической аппаратуры позволяют производить расчет такой модели с определенной точностью (определяемой шагом интегрирования) за приемлемое время. Можно представить, что все части системы представляют собой дифференциальные уравнения. Тогда их можно свести к системе дифференциальных уравнений в форме Коши, разрешенных относительно производной. Численный метод интегрирования позволяет обращаться с нелинейностью таким же образом, как и с линейным уравнением, достаточно лишь, чтобы модель нелинейного объекта была задана адекватно.

## 2.2 Алгоритм численного моделирования

На данный момент наиболее распространенным методом численного интегрирования является метод Рунге-Кутты четвертого порядка [12]. Он позволяет проинтегрировать сложные дифференциальные уравнения за приемлемое время с определенной точностью. При численном решении задачи Коши

$$\frac{du(t)}{dt} = f(t, u), \quad t > 0 \quad (2)$$

$$u(0) = u^0 \quad (3)$$

по начальному значению в точке  $t = 0$  из уравнения (2) необходимо найти решение при других  $t$ . При численном решении задачи (2), (3) для простоты будем использовать равномерную сетку по переменной  $t$  с шагом  $\tau > 0$ . Приближенное решение задачи (2), (3) в точке  $t = t_n$  обозначим  $y^n$ . Метод сходится в точке  $t_n$ , если  $|y^n - u(t_n)| \rightarrow 0$  при  $\tau \rightarrow 0$ . Метод имеет  $p$ -й порядок точности, если  $|y^n - u(t_n)| = O(\tau^p)$ ,  $p > 0$  при  $\tau \rightarrow 0$ . Простейшая

разностная схема для приближенного решения задачи (2), (3) есть

$$\frac{y^{n+1} - y^n}{\tau} = \sigma f(t_{n+1}, y^{n+1}) + (1 - \sigma)f(t_n, y^n), \quad n = 0, 1, \dots \quad (4)$$

При  $\sigma = 0$  имеем явный метод и в этом случае разностная схема аппроксимирует уравнение (2) с первым порядком. Симметричная схема  $\sigma = \frac{1}{2}$  в (4) имеет второй порядок аппроксимации. Эта схема относится к классу неявных — для определения приближенного решения на новом слое необходимо решать нелинейную задачу. Явные схемы второго и более высокого порядка аппроксимации удобно строить, ориентируясь на метод предиктор-корректор. На этапе предиктора (предсказания) используется явная схема

$$\frac{y^{n+1} - y^n}{\tau} = f(t_n, y^n),$$

а на этапе корректора (уточнения) — схема

$$\frac{y^{n+1} - y^n}{\tau} = \frac{1}{2}(f(t_{n+1}, y^{n+1}) + f(t_n, y^n)), \quad n = 0, 1, \dots$$

В одношаговом методе Рунге—Кутта идеи предиктора-корректора реализуются наиболее полно. Этот метод записывается в общем виде:

$$\frac{y^{n+1} - y^n}{\tau} = \sum_{i=1}^s b_i k_i, \quad (5)$$

где

$$k_i = f(t_n + c_i \tau, y^n + \tau \sum_{j=1}^s a_{ij} k_j), \quad i = 1, 2, \dots, s$$

Формула 5 основана на  $s$  вычислениях функции  $f$  и называется  $s$ -стадийной. Если  $a_{ij} = 0$  при  $j \geq i$ , имеем явный метод Рунге—Кутта. Если  $a_{ij} = 0$  при  $j > 1$  и  $a_{ii} \neq 0$ , то  $k_i$  определяется неявно из уравнения:

$$k_i = f(t_n + c_i \tau, y^n + \tau \sum_{j=1}^{i-1} a_{ij} k_j + \tau a_{ii} k_i)$$

О таком методе Рунге—Кутта говорят как о диагонально-явном. Параметры  $b_i, c_i, a_{ij}$  определяют вариант метода Рунге—Кутта. Используется следующее представление метода (таблица Бутчера)

$$\frac{[c \mid A]}{b^*} = \begin{array}{cccccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline b_1 & b_2 & \cdots & b_s & \end{array}$$

Одним из наиболее распространенных является явный метод Рунге—Кутта четвертого порядка

$$\frac{y^{n+1} - y^n}{\tau} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = f(t_n, y^n), \quad k_2 = f\left(t_n + \frac{\tau}{2}, y^n + \tau \frac{k_1}{2}\right),$$

$$k_3 = f\left(t_n + \frac{\tau}{2}, y^n + \tau \frac{k_2}{2}\right), \quad k_4 = f(t_n + \tau, y^n + \tau k_3).$$

Таким образом достаточно создать программный продукт, в котором можно задавать и изменять параметры системы, составлять систему уравнений из уравнений объектов, приводить ее к форме Коши, после этого находить решение уравнений (интегрировать методом Рунге-Кутта) и сводить эти решения в общем фазовом пространстве системы.

## Глава 3. Программная реализация

### 3.1 Обоснование выбора языка программирования

Разработка программного продукта выполнена [13] на языке программирования Python с использованием таких внешних библиотек:

- NumPy – реализация быстрых математических операций и массивов.
- SciPy – интегрирование дифференциальных уравнений.
- Tkinter – построение графического интерфейса пользователя.
- Matplotlib – отрисовка графиков.

Среди преимуществ Python по сравнению с другими языками программирования в рамках разработки данного программного продукта можно выделить [14]:

- Простота синтаксиса. Данный язык программирования является высокоуровневым и имеет более простой и универсальный синтаксис по сравнению с другими специализированными языками программирования.
- Поддержка ООП (объектно-ориентированное программирование). Парадигма ООП зарекомендовала себя как индустриальный стандарт в разработке программного обеспечения во всех отраслях, и имеет такие преимущества, как структурированность программного кода, уменьшение количества строк за счет повторного использования функций и сущностей, упрощение поддержки и расширения функционала программного продукта.
- Широкий функционал и испытанный временем набор стандартных библиотек, который позволяет ускорить разработку программного продукта за счет использования готовых решений, а также быстро расширять функционал программы.

К главным недостаткам Python следует отнести архитектурную ограниченность в использовании нескольких потоков, а также меньшее быстродействие

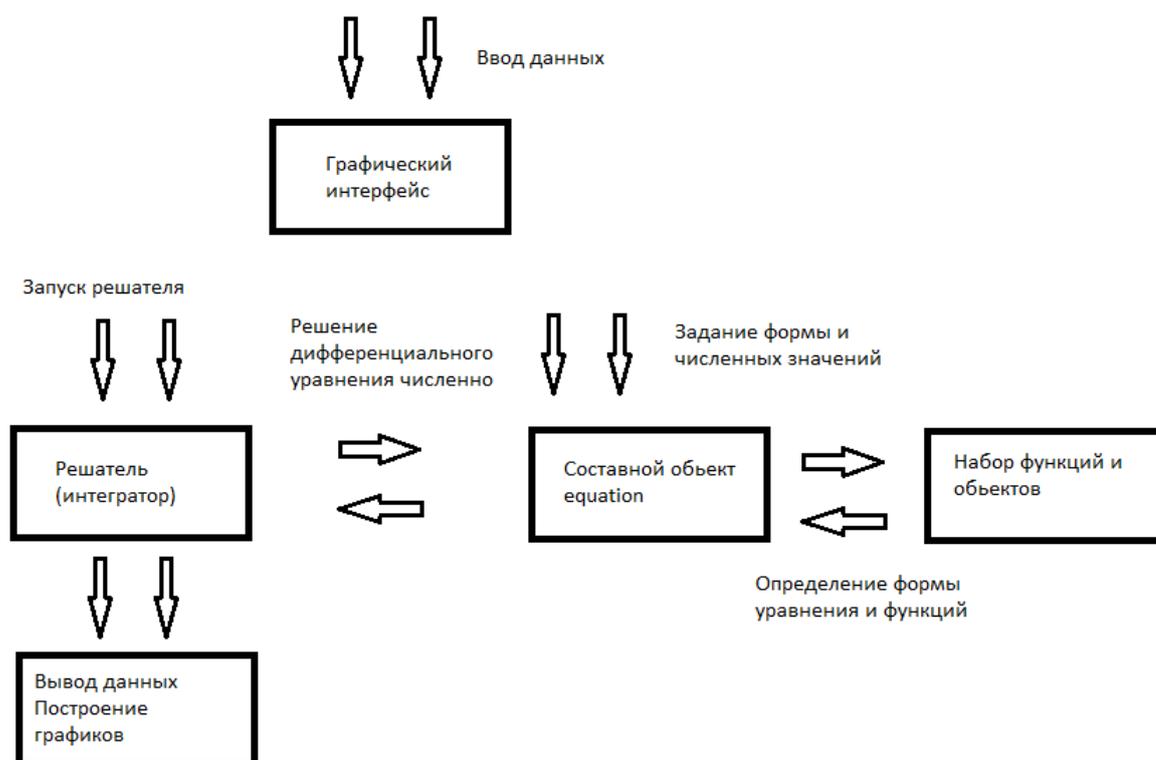
по сравнению с другими языками программирования, особенно более низкоуровневыми. Задача, решаемая данным программным продуктом, включает в себя работу с нелинейными объектами, таких как реле с гистерезисом. Расчеты с участием нелинейных объектов слабо поддаются распараллеливанию, поэтому недостаток в виде однопоточности вычислений можно не учитывать. В частности, состояние реле с гистерезисом на каждом шаге моделирования зависит от его предыстории, поэтому параллельное вычисление нескольких шагов в данном случае не даст выигрыша в производительности ввиду необходимости ожидания результата вычисления предыдущих шагов. Для преодоления второго недостатка – сравнительной медлительности языка программирования - ряд ресурсоемких вычислений был реализован с использованием библиотеки NumPy, которая использует для вычисления и реализации типов данных низкоуровневый язык программирования C, являющийся одним из наиболее быстродействующих среди языков программирования.

### **3.2 Архитектура программного продукта**

Как было указано выше, основной парадигмой программирования для данного программного продукта было выбрано ООП, как зарекомендовавший себя временем индустриальный стандарт. Поскольку программа оперирует малым количеством категорий сущностей, для упрощения архитектуры наследование заменено на композицию. Архитектурно, программный продукт состоит из пяти частей:

- Графический интерфейс, который также отвечает за ввод данных.
- Блок данных, хранящий информацию о параметрах моделирования, форме численной модели и конкретной ее реализации. Данные функции выполняет класс `equation`, чьи параметры и методы возможно составлять модульно (из различных объектов и функций).
- Набор функций и объектов, которые реализуют нелинейности.
- Решатель (интегратор), который производит расчет составленной модели.

- Вывод данных в виде графиков и координат точек переключения реле.



**Рис. 2:** Принципиальная схема работы программного продукта

Опишем процесс, происходящий на рисунке 2, отображающий принцип работы программного продукта:

- Ввод данных. Пользователь задает и выбирает коэффициенты модели динамического объекта, параметры расчета, вид нелинейных функций и численное значение их параметров.
- Программа считывает данные из графического интерфейса и задает параметры объекта equation. При этом происходит реализация принципа композиции, при котором объект состоит из других объектов (реализаций нелинейных функций, имеющих зависимость от состояния во времени), которые описаны в блоке «набор функций и объектов»
- Запуск решателя, который производит численное решение дифференциального уравнения, составленного в объекте equation на прошлом шаге.

- Численное решение находится в результате вычислительного цикла, в процессе которого объект `equation` может обращаться к набору функций и объектов, в зависимости от заданных параметров (наличие и вид возмущающего воздействия, форма и параметры гистерезиса и т.д.)
- Результат расчета визуализируется – происходит вывод данных.

После этого пользователь может изменить параметры модели, если его интересуют другие условия, или же изучить и проанализировать результат.

### **3.3 Реализация архитектуры**

Графический интерфейс и ввод данных реализован в файле `main.py`, на основе стандартной библиотеки Python `tkinter`. Составление графического интерфейса при помощи `tkinter` основывается на декларативном принципе и ООП [15]. Таким образом, главное окно представляет собой объект, которому наследуют все остальные второстепенные графические окна и «виджеты». Главное окно представлено в виде набора виджетов класса «`label`» для подписи, «`entry`» для ввода данных, «`combo`» для выбора ввода данных из предложенных вариантов, «`button`» для запуска действий.

Построение фазовых траекторий

Выбрать пример:  Размеры матрицы A:

a00:  a01:  a02:   
 a10:  a11:  a12:   
 a20:  a21:  a22:

V=

b0:  b1:  b2:

Г=

r0:  r1:  r2:

Реле с гистерезисом:

Переключение вверх I2:  Переключение вниз I1:  Верхнее значение m2:  Нижнее значение m1:

Начальные условия

x0\_0:  x1\_0:  x2\_0:

Параметры расчета:

Начало расчета:  Конец расчета:  Шаг:   
 Тип гистерезиса:  Начальное положение реле:

$X' = AX + Bu(X, \Gamma) - f(t)$

Возмущающее воздействие:

f1:  f2:  f3:

Тип возмущающего воздействия:

v0:  alpha:  k:  phi:  w:

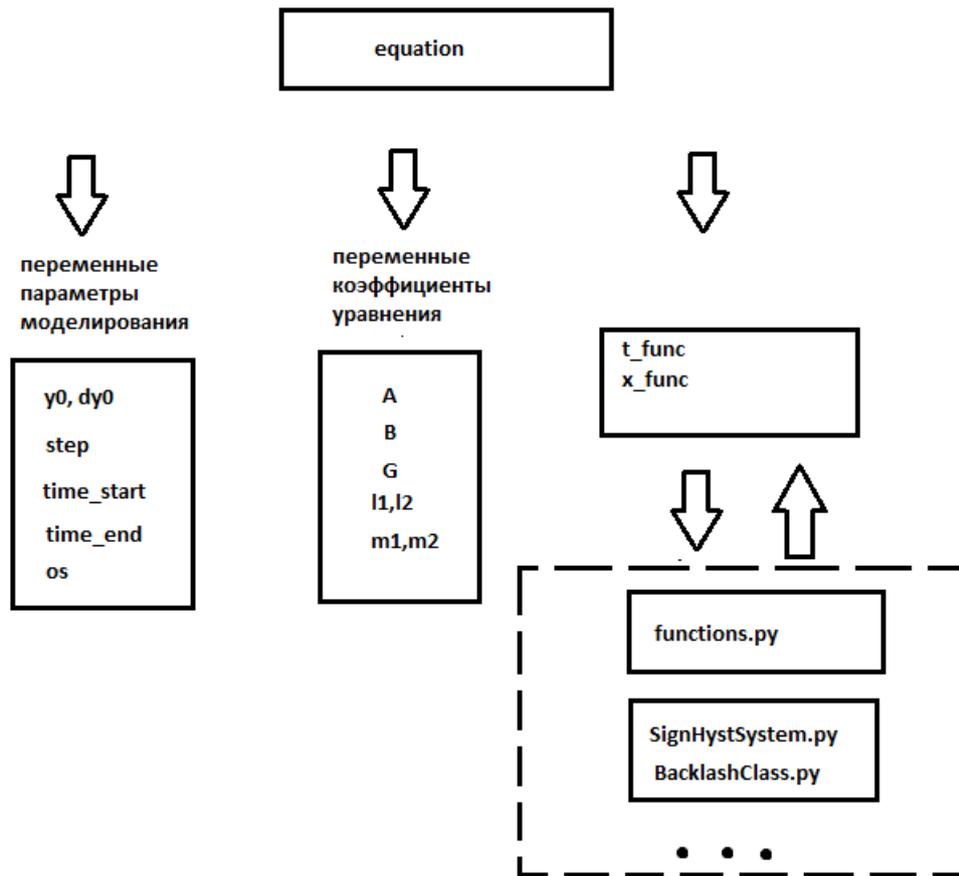
**Рис. 3:** Главное графическое окно программы

На рисунке 3 представлены такие параметры для настройки:

- Выбор примера – заранее заготовленные примеры с заранее известными результатами для проверки адекватности модели (опционально).
- Размер матрицы A – выбор размера матрицы модели объекта.
- Матрица a(00-22) – соответствующий коэффициент матрицы модели объекта.
- B(0-2) – матрица-вектор коэффициентов усиления выхода реле.

- $\Gamma(0-2)$  – матрица-вектор коэффициентов усиления входа реле.
- $I_2$  и  $I_1$  – абсциссы линии переключения реле вверх и вниз.
- $m_2$  и  $m_1$  – значения выхода реле в положении вверх и вниз.
- $x(0-2)_0$  – начальные условия для системы.
- Параметры расчета – начальное и конечное время моделирования, шаг интегрирования.
- Тип гистерезиса – выбор гистерезисной характеристики с запаздыванием или с опережением.
- Начальное положение реле – данный параметр влияет на расчет при условии, что в момент начала расчета изображающая точка на фазовой плоскости находится в зоне гистерезисной характеристики, и, таким образом, состояние реле не определено однозначно.
- $f(1-3)$  – коэффициент усиления возмущающего воздействия для каждой строчки матрицы коэффициентов объекта управления.
- Тип возмущающего воздействия – выбор формы возмущающего воздействия, а ниже - его параметры.

После запуска расчета, программа вызывает функцию, которая считывает все введенные данные пользователем и задает их как параметры объекта `equation`. После составления композиции `equation` вызывается функция `main()` файла `integrate.py`, которая запускает численное интегрирование дифференциального уравнения, заданного в объекте `equation`. Детальнее рассмотрим структуру этого объекта:



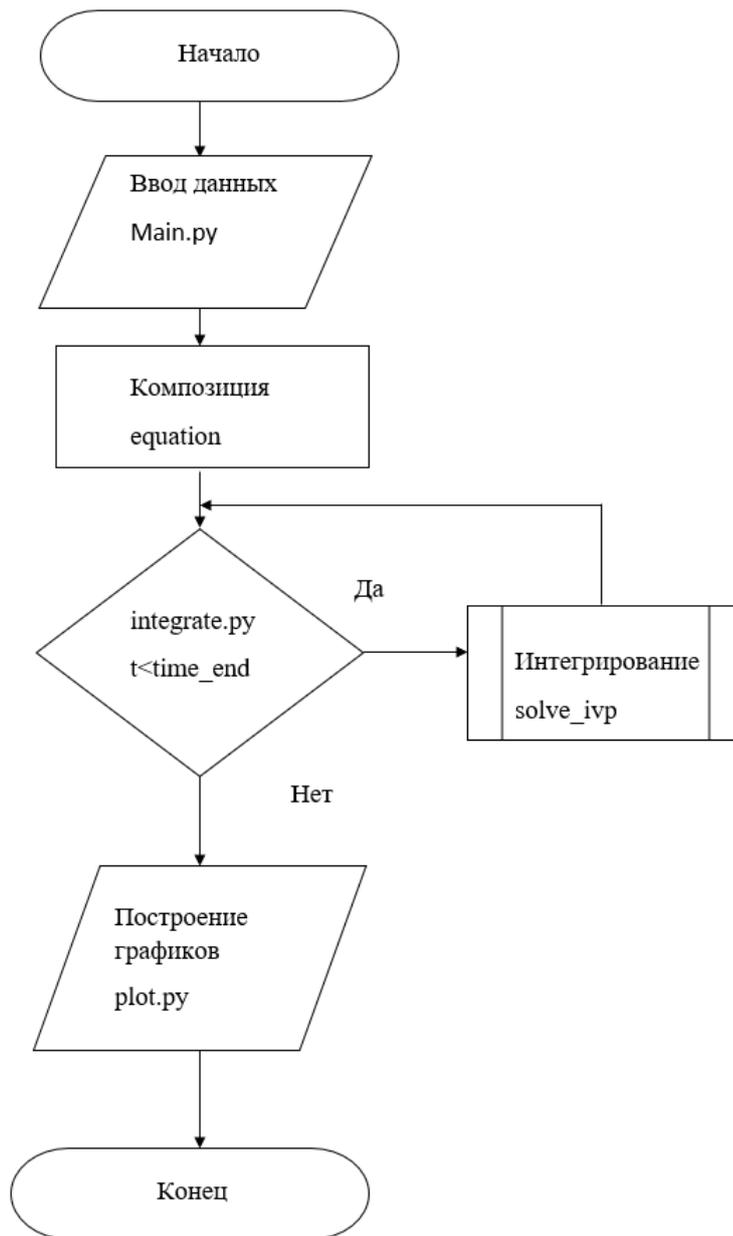
**Рис. 4:** Структура класса «equation»

Как видно из рисунка 4 объект класса «equation» состоит условно из трех частей:

- Переменные параметры процесса моделирования, такие как размер шага, начало и конец расчета, обратная связь, начальные условия.
- Переменные параметры, описывающие модель динамического объекта и нелинейности.
- Композиция из настраиваемых функций и объектов.

Применение композиции в последнем пункте обеспечивает модульность объекта. Это позволяет легко расширять функционал программы, если в этом находится необходимость. В самой же программе это позволяет гибко настраивать вид и структуру моделируемой динамической системы. Численное

интегрирование реализовано в функции `solve_ivp()` библиотеки `scipy`. Интегратор использует алгоритм Рунге-Кутты четвертого порядка с модификацией Дорманда-Принса, который обладает достаточной точностью интегрирования для данного исследования. В результате решения дифференциального уравнения на всем интервале заданного времени вызывается соответствующая функция для построения заданного графика из `plot.py`. В процессе расчета для реле фиксируются точки переключения, которые также отображаются в отдельном окне вместе с графиком.



**Рис. 5:** Блок-схема основного алгоритма программы

По данному описанию можно построить общий алгоритм (блок-схему) работы данного программного продукта, представленный на рисунке 5.

## Глава 4. Моделирование динамических систем

Рассмотрим частный случай уравнения 1. Динамическая система описывается системой дифференциальных уравнений:

$$\begin{cases} \dot{x}_1 = -4x_1 - [F(\sigma) + \phi(t)] \\ \dot{x}_2 = x_1 - 4x_2 \end{cases}$$

Причем,  $\phi(t) = F_0 + k\sin(\omega t + \varphi)$ , где  $F_0 = 1, k = 2, \omega = 2, \varphi = 0$ ;  $F(\sigma)$  - функциональная нелинейность в виде реле с гистерезисом, которая описывается такими параметрами:  $m_1 = -5.8, m_2 = 2.2, l_1 = -1, l_2 = 1$ . Гистерезис движется против часовой стрелки, то есть является гистерезисом с запаздыванием. Данная система является двумерной, соответственно, фазовое пространство в данном случае «сожмется» до фазовой плоскости. Построим фазовую траекторию данной системы на промежутке от  $t_1 = 0$  с до  $t_2 = 20$  с, с начальными условиями  $x_{10} = x_{20} = 1$ . Для этого зададим эти параметры в главном окне программы.

Выбор пример: 8    Размеры матрицы A: 2x2

a00: -4    a01: 0  
a10: 1    a11: -4

B=

b0: -1    b1: 0

Г=

r0: 1    r1: 1

Реле с гистерезисом:

Переключение вверх l2: 1    Переключение вниз l1: -1    Верхнее значение m2: 2.2    Нижнее значение m1: -5.8

Начальные условия

x0\_0: 1    x1\_0: 1

Параметры расчета:

Начало расчета: 0    Конец расчета: 20    Шаг: 0.001

Тип гистерезиса: С запаздыванием    Начальное положение реле: Верх

$X' = AX + Bu(X, \Gamma) - f(t)$

Возмущающее воздействие:

f1: 1    f2: 0    f3: 0

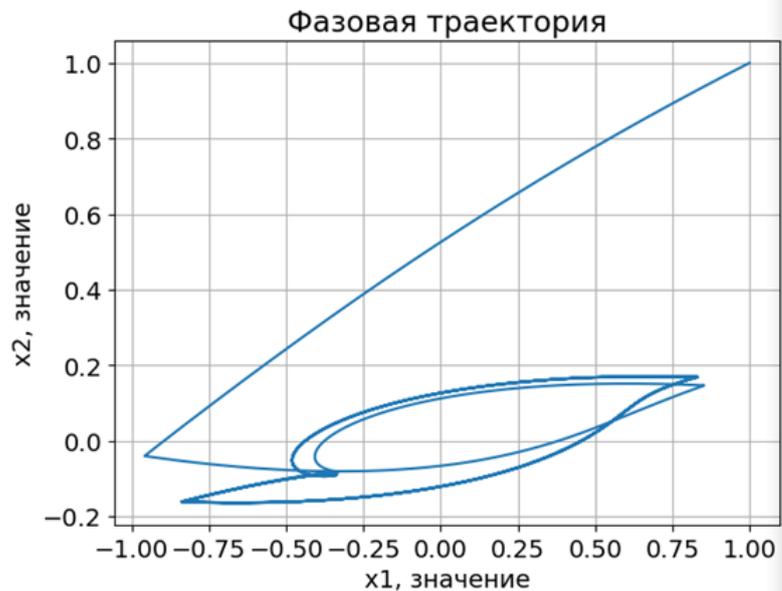
Тип возмущающего воздействия:  $f(t) = v_0 + \exp(\alpha \cdot t) \cdot k \cdot \sin(\phi + \omega t)$

v0: 1    alpha: 0    k: 2    phi: 0    w: 2

Построить фазовую плоскость    Выход

Рис. 6: Главное графическое окно программы, настроенное для моделирования первой заданной системы

В результате моделирования получен график:



**Рис. 7:** Фазовая траектория заданной системы

Как видно из рисунка 7, фазовая траектория, выходя из точки (1;1), начала движение в сторону отрицательных значений  $x_1$  и  $x_2$ , но, попав в зону гистерезиса, стала периодически колебаться.

Точки переключения

t	x1	x2	dt
0.6033	-0.95927909	-0.04087708	0.6033
1.457	0.85390365	0.14614622	0.8537
3.4173	-0.83843826	-0.16162501	1.9603
4.5073	0.83155532	0.16851875	1.09
6.5583	-0.83813251	-0.1619463	2.051
7.6488	0.83155029	0.16856144	1.0905
9.69988889	-0.83812945	-0.16194618	2.05108889
10.7908	0.83177401	0.16856912	1.09091111
12.8418	-0.83841302	-0.16200502	2.051
13.932	0.83153332	0.16853443	1.0902
15.9832	-0.83824152	-0.1619695	2.0512
17.0738	0.83167777	0.16857169	1.0906
19.1248	-0.8382479	-0.16197055	2.051

**Рис. 8:** Точки переключения реле

Анализируя точки переключения реле, из рисунка 8 можно прийти к выводу, что траектория вышла на периодическое колебание, характеризующееся двумя стационарными точками (0.831; 0.168) и (-0.838; -0.161). Шаг расчета при моделировании задан равным 0.001 с, поэтому полученные численные результаты имеют три значащих цифры после запятой. Следует заметить, что фазовая траектория перемещается от первой стационарной точки ко второй за 1.09 с, а от второй к первой за 2.051 с. Отличия координат стационарных точек и различное время перехода между ними объясняются наличием постоянного периодического возмущающего воздействия.

Рассмотрим еще одну систему дифференциальных уравнений:

$$\begin{cases} \dot{x}_1 = -2x_1 - F(\sigma) + \phi(t) \\ \dot{x}_2 = -3x_2 - F(\sigma) \end{cases}$$

Здесь  $A = \begin{pmatrix} -2 & 0 \\ 0 & -3 \end{pmatrix}$ ,  $B = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$ ,  $\phi(t) = e^{\alpha t} k \sin(\omega t + \varphi)$ , где  $\alpha = -4$ ,  $k = 1$ ,  $\omega = 2$ ,  $\varphi = -\frac{\pi}{4}$ ;  $F(\sigma)$  - функциональная нелинейность в виде реле с гистерезисом, которая описывается параметрами как:  $m_1 = -2$ ,  $m_2 = 2$ ,  $l_1 = -1$ ,  $l_2 = 1$ . Причем,  $\sigma = (\Gamma, X)$ , где  $\Gamma = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$ . Гистерезис движется против часовой стрелки, то есть является гистерезисом с запаздыванием. Зададим данные параметры в программе.

Выбрать пример: 9    Размеры матрицы A: 2x2

$a_{00}$ : -2     $a_{01}$ : 0  
 $a_{10}$ : 0     $a_{11}$ : -3

$B =$   
 $b_0$ : -1     $b_1$ : -1

$\Gamma =$   
 $r_0$ : 2     $r_1$ : 0

Реле с гистерезисом:

Переключение вверх I2: 1    Переключение вниз I1: -1    Верхнее значение m2: 2    Нижнее значение m1: -2

Начальные условия:

$x_{0,0}$ : 1     $x_{1,0}$ : 1

Параметры расчета:

Начало расчета: 0    Конец расчета: 20    Шаг: 0.001

Тип гистерезиса: С запаздыванием    Начальное положение реле: Верх

$X' = AX + Bu(X, \Gamma) - f(t)$

Возмущающее воздействие:

$f_1$ : -1     $f_2$ : 0     $f_3$ : 0

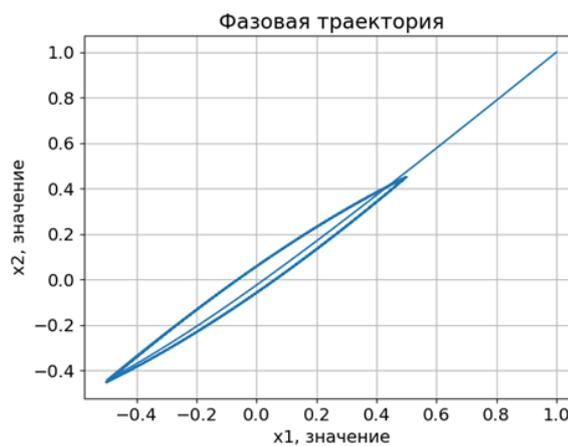
Тип возмущающего воздействия:  $f(t) = v_0 + \exp(\alpha t) * k * \sin(\phi + \omega t)$

$v_0$ : 0     $\alpha$ : -4     $k$ : 1     $\phi$ : -0.785     $\omega$ : 2

**Рис. 9:** Главное графическое окно программы, настроенное для моделирования второй заданной системы

В результате моделирования получен график:



**Рис. 10:** Фазовая траектория заданной системы

И точки переключения:

t	x1	x2	dt
0.6702	-0.50006482	-0.4434862	0.6702
1.2133	0.50003168	0.44908795	0.5431
1.7632	-0.50014425	-0.45248229	0.5499
2.3128	0.50038631	0.45155986	0.5496
2.86188889	-0.50000939	-0.4515235	0.54908889
3.4118	0.50019772	0.45162862	0.54991111
3.96088889	-0.50006298	-0.45151025	0.54908889
4.5108	0.50018078	0.45163117	0.54991111
5.05988889	-0.50006866	-0.45150977	0.54908889
5.6098	0.50017887	0.45163126	0.54991111
6.15888889	-0.5000693	-0.45150975	0.54908889
6.7088	0.50017866	0.45163127	0.54991111
7.25788889	-0.50006937	-0.45150975	0.54908889
7.8078	0.50017864	0.45163127	0.54991111
8.35688889	-0.50006937	-0.45150975	0.54908889

**Рис. 11:** Точки переключения полученной фазовой траектории

Проанализировав полученные данные, можно прийти к выводу, что система вышла на стационарную орбиту практически сразу после первого переключения реле. Стационарными точками являются  $(-0.5; -0.449)$  и  $(0.5; 0.449)$ . Время перехода между этими точками одинаково и равняется 0.549 секунд. Как и в прошлом моделировании, шаг расчета был принят равным 0.001 с. Отличие поведения фазовой траектории в этой системе от прошлой можно объяснить тем, что в данной системе возмущающее воздействие со временем затухает, поэтому траектория выходит на стационарную орбиту быстро. Рассмотрим еще одну систему дифференциальных уравнений, с тремя независимыми переменными:

$$\begin{cases} \dot{x}_1 = -x_1 + x_2 + 2F(\sigma) - \phi(t) \\ \dot{x}_2 = -x_1 - F(\sigma) - \phi(t) \\ \dot{x}_3 = -x_3 - 2F(\sigma) - \phi(t) \end{cases}$$

Причем,  $\phi(t) = k \sin(\omega t + \varphi)$ , где  $k = 4, \omega = 1, \varphi = 1$ ;  $F(\sigma)$  - функциональная нелинейность в виде реле с гистерезисом, которая описывается параметрами как:  $m_1 = -1.4024, m_2 = 1.4024, l_1 = -3.5, l_2 = 3.5$ . Причем,  $\sigma = (\Gamma, X)$ , где

$$\Gamma = \begin{pmatrix} 0 \\ -0.13 \\ 2.9 \end{pmatrix}$$
 Гистерезис движется против часовой стрелки, или является гистерезисом с запаздыванием. Зададим данные параметры в программе.

Выбрать пример: 5    Размеры матрицы A: 3x3

a00: -1    a01: 1    a02: 0  
 a10: -1    a11: 0    a12: 0  
 a20: 0    a21: 0    a22: -2

B =  
 b0: 2    b1: -1    b2: -2

Г =  
 r0: 0    r1: -0.13    r2: 2.9

Реле с гистерезисом:  
 Переключение вверх I2: 3.5    Переключение вниз I1: -3.5    Верхнее значение m2: .4024    Нижнее значение m1: -1.402

Начальные условия  
 x0\_0: 1    x1\_0: 1    x2\_0: 1

Параметры расчета:  
 Начало расчета: 0    Конец расчета: 100    Шаг: 0.001

Тип гистерезиса: С запаздыванием    Начальное положение реле: Верх

$X' = AX + Bu(X, \Gamma) - f(t)$

Возмущающее воздействие:  
 f1: 1    f2: 1    f3: 1

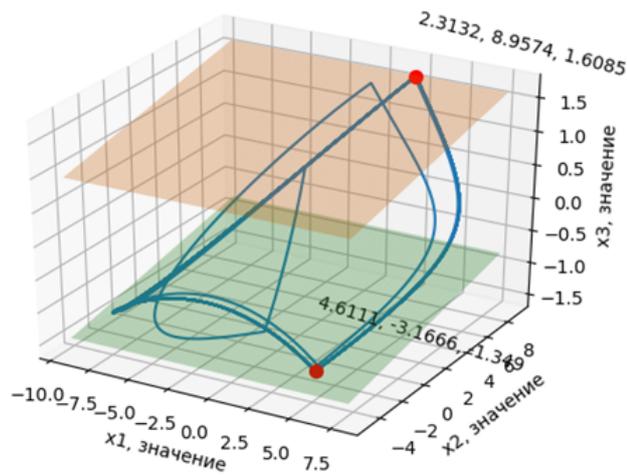
Тип возмущающего воздействия:  $f(t) = v_0 + \exp(\alpha t) * k * \sin(\phi + \omega t)$

v0: 2    alpha: 0    k: 4    phi: 1    w: 1

Построить фазовую плоскость    Выход

**Рис. 12:** Главное графическое окно программы, настроенное для моделирования третьей заданной системы

В результате моделирования получен график:



**Рис. 13:** Фазовая траектория заданной системы

И точки переключения:

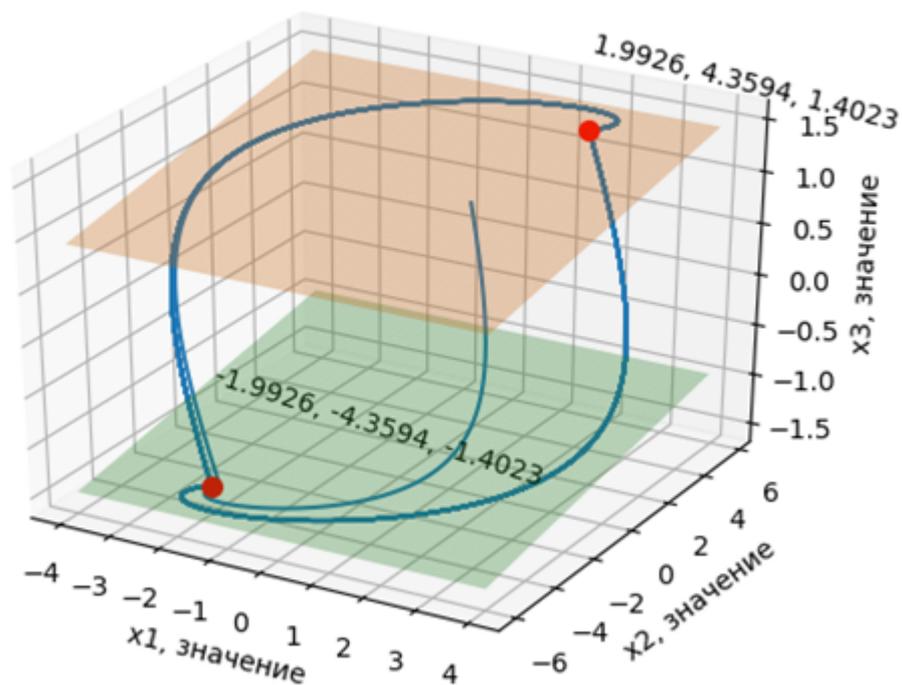
t	x1	x2	x3	dt
0.283	0.02247119	-1.14432971	-1.25864168	0.283
3.302	0.43266661	7.64118861	1.54952948	3.019
5.1508	4.41154535	-2.7122753	-1.32872971	1.8488
9.62588889	2.28198665	8.87688166	1.6048934	4.47508889
11.4508	4.5686459	-3.18893538	-1.35028728	1.82491111
15.9118	2.31619138	8.95670465	1.60848841	4.461
17.7338	4.60700964	-3.17164965	-1.34966662	1.822
22.195	2.31349484	8.95768185	1.608508	4.4612
24.0168	4.60969296	-3.16827383	-1.34934105	1.8218
28.4782	2.31335466	8.95742614	1.60852758	4.4614
30.2998	4.61092349	-3.16671362	-1.34903113	1.8216
34.7618	2.31564967	8.95760082	1.6090756	4.462
36.58288889	4.61216922	-3.16515899	-1.34883394	1.82108889
41.0448	2.3145287	8.95777193	1.60883045	4.46191111
42.8662	4.61103584	-3.16659565	-1.34906787	1.8214
47.3278	2.31354612	8.95742413	1.60858581	4.4616
49.1493	4.61100291	-3.16662164	-1.3489549	1.8215
53.61088889	2.31308556	8.95722959	1.60845853	4.46158889
55.4328	4.61010326	-3.16754674	-1.34940371	1.82191111
59.8942	2.31362433	8.95746241	1.60862462	4.4614
61.7158	4.61065939	-3.16697026	-1.34914107	1.8216

**Рис. 14:** Точки переключения реле полученной фазовой траектории

Проанализировав полученные данные можно прийти к выводу, что система вышла на стационарную орбиту после третьего переключения реле. Стационарными точками являются (2.313; 8.957; 1.609) и (4.611; -3.166; -1.349). Время перехода между этими точками равняется 1.82 с и 4.46 с. Как и в

прошлом моделировании, шаг расчета был принят равным 0.001 с. Периодическое возмущение вносит смещение в траекторию движения периодического колебания.

Произведем моделирование той же системы, без возмущающего воздействия:



**Рис. 15:** Фазовая траектория заданной системы без возмущающего воздействия

t	x1	x2	x3	dt
5.0608	-1.81282338	-4.35890912	-1.40230342	5.0608
10.374	1.98777128	4.35966919	1.40233197	5.3132
15.6638	-1.99248734	-4.35937777	-1.4023287	5.2898
20.9528	1.99261432	4.35937518	1.4023286	5.289
26.2418	-1.99261013	-4.35936473	-1.4023286	5.289
31.5308	1.99260941	4.35936454	1.4023286	5.289
36.8198	-1.99260942	-4.35936459	-1.4023286	5.289
42.1088	1.99260942	4.35936459	1.4023286	5.289

**Рис. 16:** Точки переключения реле полученной фазовой траектории

Как видно из результатов моделирования, та же система без возмущающего воздействия выходит на стационарную орбиту после первого же переключения, координаты точек переключения равны по значению (1.992; 4.359;1.402) и противоположны по знаку, а время перехода одинаково и равно 5.289 с.

## **Выводы**

Фазовая траектория динамических систем с гистерезисом под воздействием незатухающего гармонического периодического возмущения смещается периодически относительно стационарной точки на определенную величину. Стоит заметить, что периодическое возмущающее воздействие ускорило движение фазовой траектории между точками переключения, тем самым реле под возмущающим воздействием переключается чаще, чем без него. В реальных объектах это может увеличивать расход топлива, или же приводить к более быстрому износу аппаратуры. В результате проверки работы разработанной программы сделан вывод, что расчеты соответствуют действительности, а их достоверность напрямую зависит от адекватности построения математической модели. Точность же напрямую зависит от выбранного шага расчета, который определяет необходимую вычислительную мощность для расчета, и того, каковы фазовые портреты системы между поверхностями переключения. Из этого следует вывод, что шаг расчета необходимо выбирать из соображений необходимой точности для конкретной задачи.

## Заключение

В результате анализа нелинейных систем с гистерезисом, получено средство для численного моделирования сложных динамических систем, содержащих реле с гистерезисом. Это значительно упрощает анализ таких систем, сложность которого обозначена в ряде работ [1][3]. В отличие от аналитического подхода, численное моделирование позволяет обобщить различные виды систем, полагаясь лишь на верное редуцирование физических явлений до адекватной математической модели. Это позволяет применять такие численные методы унифицировано для ряда задач, различных по виду. Благодаря выбранным принципам и парадигме разработки (модульность, ООП), функционал программы расширять относительно легко. В перспективе возможно как добавление математических моделей нелинейностей другого типа, так и расширение функционала до поиска периодических решений в автоматизированном порядке. Эта задача обозначена как приоритетная в данной области очень давно, но полностью не решена до сих пор [8].

## Список литературы

- [1] Евстафьева В. В. О необходимых условиях существования периодических решений в динамической системе с разрывной нелинейностью и внешним периодическим воздействием // Уфимск. матем. журн. 2011. Т. 3, № 2. С. 20–27.
- [2] Камачкин А.М., Шамберов В.Н. Отыскание периодических решений в нелинейных динамических системах. СПб.: Изд-во СПбГУ, 2002. 88 с.
- [3] Камачкин А. М., Шамберов В. Н. Существование периодических движений в неавтономных многомерных нелинейных системах // Системы управления и информационные технологии. Воронеж: Изд-во «Научная книга», 2015. № 1(59). С. 16–19.
- [4] Камачкин А. М., Хитров Г. М., Шамберов В. Н. Нормальные формы матриц в задачах декомпозиции и управления многомерных систем // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2017. Т. 13. Вып. 4. С. 417–430.
- [5] Kamachkin A. M., Potapov D. K., Yevstafyeva V. V. Existence of solutions for second-order differential equations with discontinuous right-hand side // Electron. Journal of Differential Equations, 2016. N 124. P. 1–9
- [6] Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов. 4-е изд., стереотипное. М.: Наука, 1983. 393 с.
- [7] Точные методы исследования нелинейных систем автоматического управления / Р. А. Нелепин [и др.] // М.:Изд-во «Машиностроение», 1971. – серия «Нелинейные системы автоматического управления». С. 7-8
- [8] Нелепин Р. А. Точные аналитические методы в теории нелинейных автоматических систем (с примерами из судовой автоматики). Л.: «Судостроение», 1967. 447 с.

- [9] Нелепин Р. А., Камачкин А. М., Туркин И. И., Шамберов В. Н. Алгоритмический синтез нелинейных систем управления. Л.: Изд-во Ленингр. ун-та, 1990. 240 с.
- [10] Kamachkin A. M., Potapov D. K., Yevstafyeva V. V. Solution to second-order differential equations with discontinuous right-hand side // Electron. Journal of Differential Equations. 2014. N 221. P. 1–6.
- [11] Морозов А. Д. Драгунов Т. Н. Визуализация и анализ инвариантных множеств динамических систем. Москва-Ижевск: Институт компьютерных исследований, 2003, 304 с
- [12] Галкин А. В. Дятчина Д. В. Численное решение математических моделей объектов, заданных составными системами дифференциальных уравнений // Современные проблемы науки и образования. Пенза: Издательский Дом «Академия Естествознания», 2011. № 6. С.127
- [13] Исходный код. URL: <https://github.com/IgorSavelyev404/phasesnonlinear>
- [14] Python documentation. URL: <https://docs.python.org/3/>
- [15] Полякова Н. М., Ширяева Е. В. Python 3. Создание графического интерфейса пользователя (на примере решения методом пристрелки краевой задачи для линейных обыкновенных дифференциальных уравнений). Ростов-на-Дону, 2017