РГГМУ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра высшей математики и физики

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Бакалаврская работа

На тему: Компьютерное моделирование процесса синхронизации механических и электромагнитных колебаний

Исполнитель: Литуновский Игорь Николаевич
(фамилия, имя, отчество)

Руководитель: кандидат физико-математических наук, доцент кафедры высшей математики и физики

Бобровский Анатолий Петрович

«К защите допускаю»

Заведующий кафедрой

____
(подпись)

кандидат физико-математических наук, доцент кафедры высшей математики и физики

Зайцева Ирина Владимировна

«01» июня 2023г

Санкт- Петербург

2023 г.

Оглавление

Введение

Темой данной работы является компьютерное моделирование процессов синхронизации. В этой работе рассмотрена синхронизация как механических, так и электромагнитных колебаний. Однако, впоследствии станет ясно, что математическое описание как механических, так и электромагнитных колебаний схоже и между процессами синхронизации этих видов колебаний нет существенной разницы. В теоретической части основательно рассмотрено явление синхронизации колебаний – начиная с того, что такое колебания вообще, какие существуют виды колебаний. Так же, рассмотрены условия, в которых возможна синхронизация. Затем, описано, как различные параметры колебательных систем влияют на протекание и результат их синхронизации.

В практической части работы представлена написанная автором работы программа, позволяющая смоделировать синхронизацию колебательных систем с заданными параметрами при определенных условиях среды. Эта программа является основным результатом данной исследовательской работы. Она написана на языке Python с использованием пакета для визуализации данных Matplotlib, а так же пакетов SciPy и NumPy. Язык Python выбран как популярный язык программирования, простой в освоении и в то же время достаточно гибкий, что бы позволить получить подходящий результат. Пакет Matplotlib так же является одним из самых популярных и простых пакетов для визуализации данных на базе языка Python. В то же время, его возможностей вполне достаточно для целей данной работы.

Тема данной работы является актуальной, так как явление синхронизации колебаний активно используется в современной технике, особенно радиопередающей и принимающей аппаратуре, как для расшифровки сигнала, так и для его усиления [1]. Явление синхронизации лежит, в том числе, в основе работы осциллографа. Написание же программы, позволяющей быстро смоделировать большое количество синхронизаций различных колебательных систем имеющих разные параметры колебаний, позволит наглядно изучить этот процесс. Это поможет качественнее его изучить и лучше понять процессы, происходящие при синхронизации колебаний.

1.1. Колебания

Колебательные и волновые процессы в природе и технике встречаются повсеместно. Без понимания волновых и колебательных законов были бы невозможны многие достижения науки и техники. К таковым относятся, например, радиосвязь, акустические приспособления, использование переменного тока. Работу различных часовых механизмов – как механических, так и электрических, так же следует считать колебательным процессом. И, хотя природа сил, вынуждающих различные системы колебаться, различна, нетрудно заметить, что математическое описание всех колебательных систем, похоже [2]. Более того, уравнения, описывающие механические или электромагнитные колебания абсолютно идентичны. Уравнения колебательных процессов, как показано далее, имеют различную природу лишь в зависимости от видов колебаний – гармонические они или нет, линейны или нелинейны, вынужденные или свободные и так далее. Однако не все из них имеют свойство синхронизироваться между собой. Колебательные системы так же могут иметь различное количество степеней свободы. И, поскольку в рамках этой работы мы говорим о синхронизации, описывая колебания важно так же знать, какое количество осцилляторов присутствует в описываемой системе.

Таким образом, колебания – это разнообразные по своей природе процессы, которые объединяет определенная повторяемость во времени. Эта повторяемость, зачастую, неабсолютная. Вообще, колебание не всегда является повторяемым изменением. По этому признаку колебания разделяют на детерминированные, хаотические и случайные. Для детерминированных колебаний верно равенство:

$$\Phi(t) = \Phi(t + nT) \tag{1.1.1}$$

Где:

Ф – Фаза колебаний.

t – Переменная, произвольный момент времени.

T – Период колебаний

n – Целое число

Это простое уравнение (1.1) поможет ввести основные термины, встречающиеся в теории колебаний. Так, смысл этого уравнения состоит в строгой периодичности детерминированных колебаний – фактически, оно гласит, что фаза таких колебаний повторяется каждый промежуток времени, называемый периодом. Примеры таких колебаний представлены ниже:

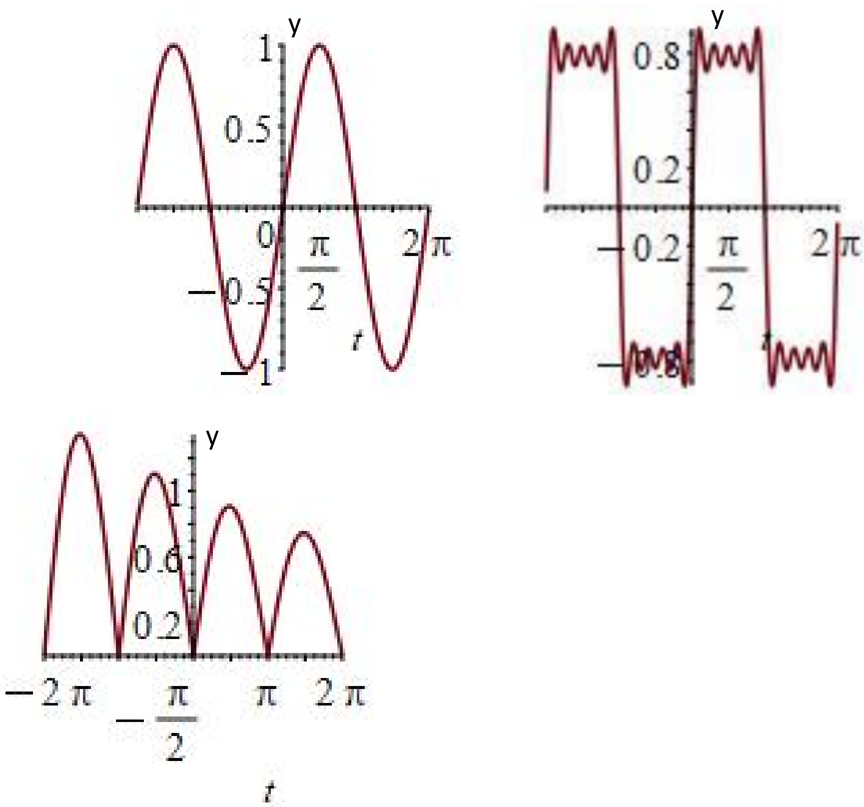

Рисунок 1: графики различных детерминированных колебаний

Фаза колебаний – это безразмерная величина, которая позволяет описать состояние колебательной системы в произвольный момент времени. На самом деле, детерминированные колебания не всегда гармонические. Как видно из графиков, для любых детерминированных колебаний – гармонических или нет - фаза имеет свойство повторяться во времени с определенной, присущей данной колебательной системе, периодичностью. Таким образом, период – это время, за которое колебательная система возвращается в некоторую начальную фазу. Так как период имеет размерность времени, легко ввести понятие частоты колебаний:

$$\upsilon = \frac{1}{T} = [\frac{1}{\text{с}}] \quad (1.1.2)$$

Физический смысл частоты – это число раз, которое колебательная система пройдет одинаковые состояния (фазы) за единицу времени. Гармонические системы, поскольку обладают постоянным периодом, так же имеют постоянную частоту. Когда мы говорим, что колебательные системы синхронизировались, мы как раз подразумеваем то, что между ними установилась одинаковая частота.

Случайными колебаниями называются колебания, в которых какие-либо параметры осцилляции являются случайной, зависящей от времени величиной. Этими параметрами могут быть, например, частота (период) или амплитуда. Внешний вид графиков таких колебаний может быть самым разнообразным. Однако, поскольку их строгое математическое описание, как правило, невозможно, такие колебания интересуют нас в меньшей степени. В данной работе рассмотрена синхронизация только детерминированных колебаний, хотя и синхронизация хаотических колебаний так же возможна и этой теме посвящены некоторые работы.

Суммируя выше сказанное, можно сказать, что детерминированные осцилляторы это такие системы, состояние которых можно определить в любой момент времени с абсолютной точностью, зная закон, по которым их колебания происходят.

Другой важной характеристикой колебательной системы является амплитуда колебаний. Амплитуда – это максимальное отклонение системы от положения равновесия. Амплитуда колебаний характеризует энергию системы. Например, если мы говорим о механических колебаниях маятника, то амплитудой будет являться максимальный угол его отклонения от отвесной линии. Говоря об электромагнитных колебаниях, под амплитудой следует понимать максимальное значение силы тока в цепи (положительное или отрицательное). С точки зрения зависимости амплитуды от времени, колебательные системы разделяют на консервативные и диссипативные.

Консервативные системы сохраняют механическую энергию, и лишь представляют собой процесс вечного перехода кинетической энергии в потенциальную и наоборот. Очевидно, что таких систем не существует в природе, однако, их математическая модель может быть вполне пригодна для решения определенного спектра задач.

Диссипативные системы – это системы, в которых происходит диссипация, то есть рассеяние энергии. Амплитуда, выражающая собой запасенную системой энергию, так же постепенно уменьшается. Такая математическая модель колебаний наиболее похожа на те, что встречаются в природе и технике.

Представленный ниже график демонстрирует разницу между отклонением системы от состояния равновесия и ее текущей амплитудой. В данном случае, амплитуда убывает по экспоненте. На этом графике так же нетрудно заметить еще один факт – частота линейных колебаний не зависит от их амплитуды [3]. Этот факт долгое время считался основополагающим во всей теории колебаний, однако, позднее было установлено, что это правило распространяется лишь на линейные колебания, которые в чистом виде в природе практически не встречаются.

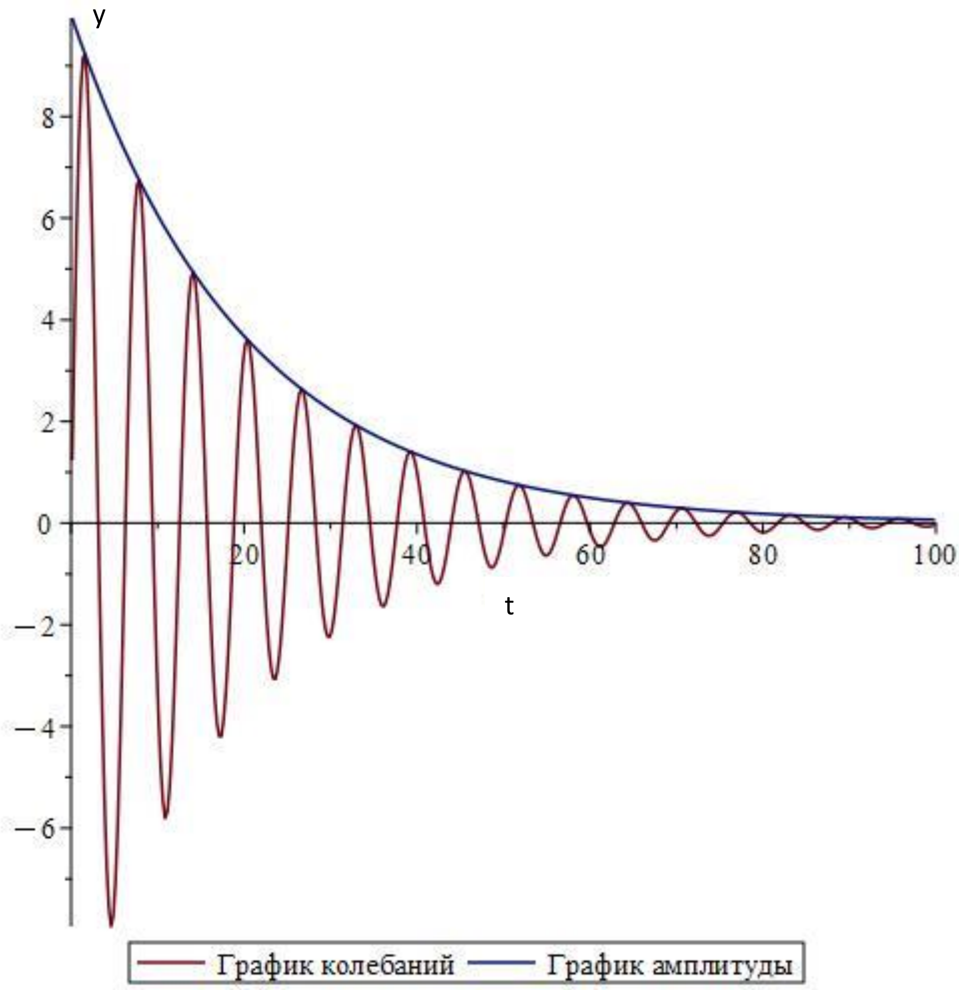

Рисунок 2: график линейных затухающих колебаний

Другим важным для нас видом колебаний являются автоколебания. Автоколебательный процесс по своей сути это процесс, в котором диссипация энергии в колебательной системе постоянно компенсируется подпиткой энергией извне. Такие колебания могут продолжаться без существенного изменения амплитуды продолжительное время, до тех пор, пока запасы энергии источника и самой системы не иссякнут, либо пока передача энергии не будет прекращена (например, при выключении автогенератора, прекращении радиопередачи и так далее) [4].

Автоколебательные процессы, как правило, являются вынужденным колебаниями – то есть, происходящими под воздействием внешней силы. Примером таких колебаний может служить качание листвы на ветру, если мы говорим о механических колебаниях (такие колебания так же являются хаотическими). Если речь идет об электромагнитных колебаниях, то можно привести в пример колебания тока, возникающие в радиоприемники, когда он вступает в резонанс с радиопередатчиком. Источником энергии в данном случае служит принятая волна, то есть, радиопередатчик. Таким образом, для приёма радиосообщений не нужна энергия – её переносят сами принятые электромагнитные волны. Радиоприём может продолжаться до тех пор, пока идет передача или пока приёмник не будет выключен (или его колебательный контур не будет изменен иным образом, что бы он больше не вступал в резонанс с передатчиком).

Колебательные законы различны для различных типов колебаний. Законы, по которым происходят колебания, обычно выводятся из дифференциальных уравнений. Для различных колебаний различна природа полученных дифференциальных уравнений. Например, описывая колебания маятника – математического или физического, или колебания грузика на пружинке или другой механической системы, дифференциальное колебательное уравнение получают обычно из второго закона Ньютона. В то же время, уравнения, описывающие электромагнитные колебания, возникающие в колебательном контуре (последовательно соединенные конденсатор, индуктивность и сопротивление) находят из уравнения токов. Однако, как уже говорилось ранее, несмотря на то, что природа колебаний может быть разнообразной, все они подчиняются схожим законам. Так, все описанные выше колебания подчиняются следующему дифференциальному уравнению:

$$\ddot{x} + 2\beta\dot{x} + \omega\sin(x) = 0 \tag{1.1.3}$$

Где x – отклонение от положения равновесия.

ω – циклическая частота. Величина, отличающаяся от частоты колебаний на 2π.

β – коэффициент затухания, определяемый физическими свойствами колебательной системы.

Полученное дифференциальное уравнение является нелинейным, поскольку содержит в себе член F(x) = sin(x). Зачастую, это уравнение сводят к линейному, пользуясь тем, что sin(x) ≈ x при малых значениях y. Тогда уравнение преобразуется к виду:

$$\ddot{x} + 2\beta\dot{x} + \omega x = 0 \tag{1.1.4}$$

Это уравнение является линейным. Его решением является обычная синусоида (или косинусоида). Однако, оно нас не удовлетворяет, поскольку процессу синхронизации подвержены только нелинейные осцилляторы. Уравнение же нелинейного осциллятора, в свою очередь, в общем виде выглядит так:

$$\ddot{x} + F(x) = 0 \tag{1.1.5}$$

Член, зависящий от первой производной координаты, может присутствовать, а может отсутствовать. В первом случае речь идет о диссипативной системе, то есть такой, в которой энергия постепенно рассеивается, а во втором – о консервативной. Это значит, что его наличие отвечает за постепенное затухание колебаний, то есть уменьшение их амплитуды. В рамках данной работы, этому не следует придавать большого значения, так как наличие или отсутствие данного члена не влияет на то, будет ли осциллятор линейным или нелинейным.

1.2. Колебания Ван-дер-Поля

Интересными и важными как с научной, так и технической точки зрения являются колебания Ван-дер-Поля. Уравнение этих колебаний было получено голландским физиком Бальтазаром Ван-дер-Полем и его коллегами в 1927 году. По своей сути, это затухающие нелинейные колебания. Примечательной особенностью этой системы является то, что она всегда стремится к предельному циклу. Форма и положение этого цикла на фазовом пространстве $x - \dot{x}$ (координата и ее первая производная) зависит исключительно от свойств самой колебательной системы, и никак не зависит от начального состояния осциллятора. Эти колебания были названы Ван-дер-Полем релаксационными.

Нелинейное дифференциальное уравнение, описывающие осциллятор Ван-дер-Поля, называется уравнением Ван-дер Поля. Рассмотрим его подробнее [5].

$$\ddot{x} - \mu(1 - x^2)\dot{x} + x = 0 \tag{1.2.1}$$

Где x – координата, функция одной переменной t

μ – коэффициент нелинейности (и затухания)

Именно коэффициент нелинейности определяет форму как предельного цикла, так и релаксационных колебаний. Ниже приведены примеры графиков колебаний Ван-дер-Поля при различных значениях μ и соответствующих им фазовых портретов.

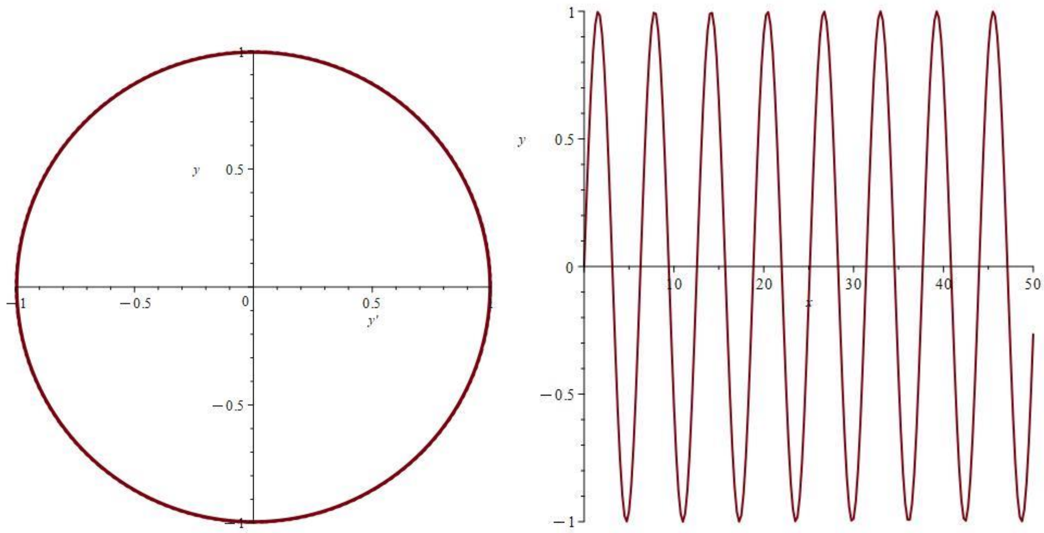

Рисунок 3: осциллятор Ван-дер-Поля, μ=0.

Полученные графики и фазовый портрет для случая μ=0 идентичны таковым для случая линейных гармонических колебаний. Очевидно, это происходит потому, что при μ=0 уравнение Ван-дер-Поля как раз сводится к уравнению гармонических колебаний. В дальнейшем, нас этот случай интересовать не будет. Характерно, однако, что предельный цикл гармонических колебаний проходит через начальное положение.

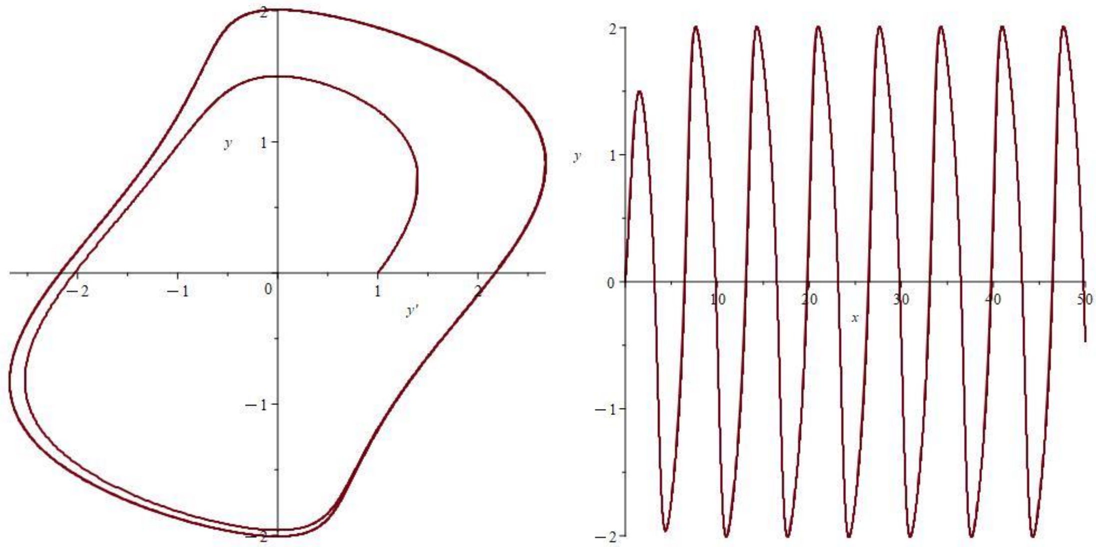

Рисунок 4: осциллятор Ван-дер-Поля, μ=1.

Эти колебания очевидно нелинейны. Начальное положение – $x = 0, \dot{x} = 1$, однако фазовая траектория сразу устремляется к предельному циклу, которому соответствует бóльшая энергия колебаний.

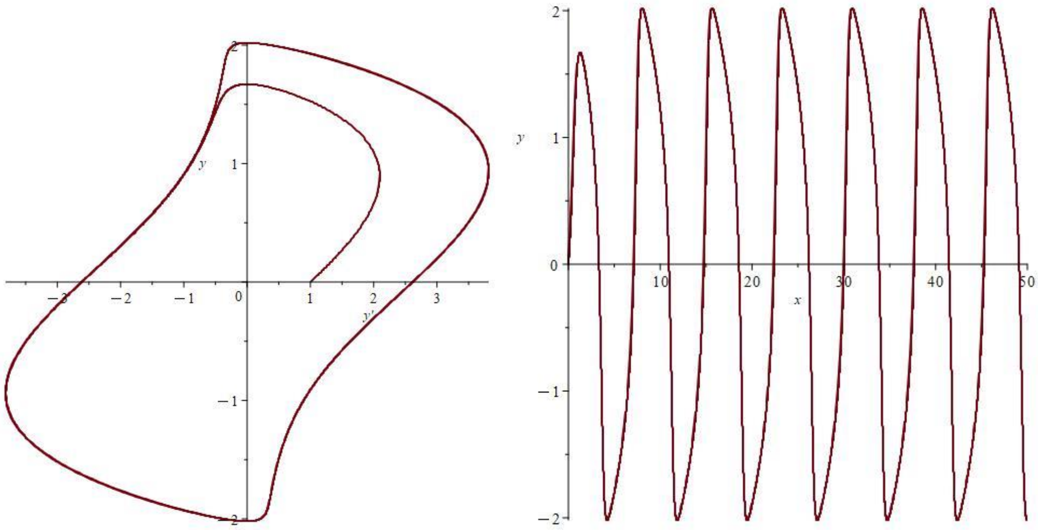

Рисунок 5: осциллятор Ван-дер-Поля, μ=2

Форма колебаний несколько изменилась, фазовая траектория устремилась к предельному циклу заметно быстрее.

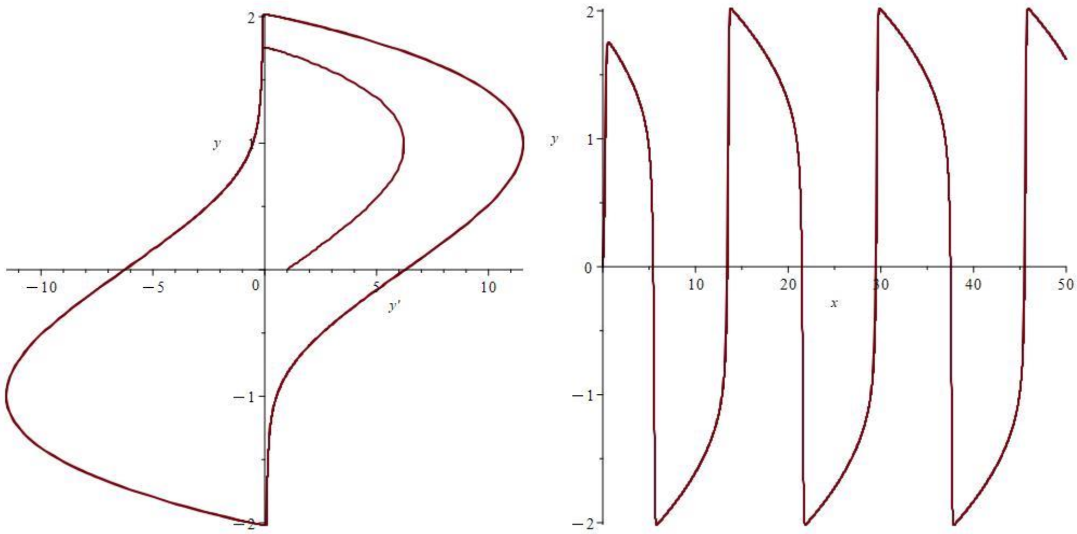

Рисунок 6: осциллятор Ван-дер-Поля, μ=8

Если последовательно сравнить все полученные изображения, видно, что форма фазового портрета стремится растянуться по горизонтали (Это видно, если обратить внимание на изменение масштаба по оси производной). При этом в верхних точках перегиб становится все более и более крутым.

Ниже продемонстрировано свойство колебаний Ван-дер-Поля стремиться к предельному циклу независимо от начальных условий:

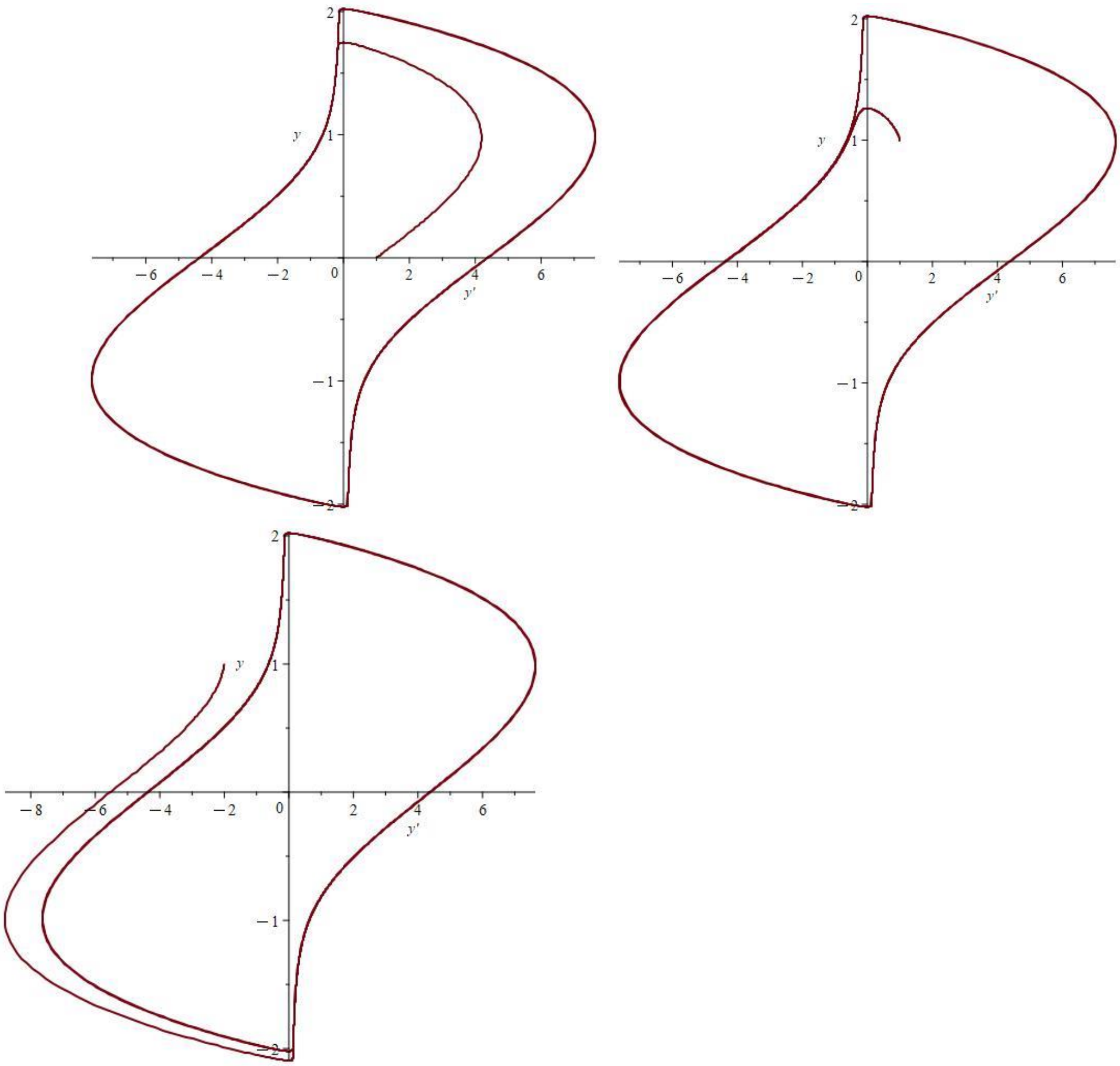

Рисунок 7: осциллятор Ван-дер-Поля, μ=5 при разных начальных условиях. Сверху слева по часовой стрелке: $x(0) = 0, \dot{x}(0) = 1$;

$x(0) = 1, \dot{x}(0) = 1$; $x(0) = 1, \dot{x}(0) = -2$

Как видно, фазовые траектории устремляются к предельному циклу независимо от начальных условий – будь начальная фаза внутри или за пределами этого цикла. Как было показано ранее, форма самого предельного цикла зависит только от параметров самого осциллятора.

Осциллятор Ван-дер-Поля является одной из простейших моделей автоколебательной системы. Она может быть реализована на практике в устройстве, называемом автогенератором. В дальнейшем, в практической части работы, проведено моделирование синхронизации автогенератора Ван-дер-Поля внешним сигналом, а так же взаимной синхронизации двух таких генераторов.

1.3. Автогенератор Ван-дер-Поля

Как уже было отмечено в предыдущих главах, автоколебания требуют постоянной подпитки энергией извне. Приведенные тогда примеры – листья, качающиеся на ветру и радиоприём – действительно могут поддерживать колебательный режим неограниченно долго. Однако, эти колебания никак нельзя назвать детерминированными. И хотя синхронизация хаотических колебаний, в определенном смысле, так же возможна, в рамках данной работы рассмотрена исключительно синхронизация детерминированных колебаний. Существует много способов поддержания таких колебаний. Примером могут послужить самые разнообразные часовые механизмы. Однако, в настоящий момент, мы остановимся на электромагнитных автоколебаниях в автогенераторе Ван-дер-Поля.

Итак, перед нами стоит задача неограниченно долго поддерживать режим электромагнитных колебаний в колебательном контуре. Ясно, что для этого следует подводить к системе дополнительную энергию. Но как именно следует это делать? Ведь если непрерывно подводить к системе большую энергию, то ее колебательный режим может оказаться нарушен.

Существует два условия корректной работы автогенератора.

1. Вновь подводимая энергия равна расходам системы. Очевидно, что если подводимая энергия окажется в среднем меньше расходов системы за тот же промежуток времени, то колебания будут затухать, а если подводимой энергии окажется слишком много, то колебания будут неограниченно нарастать [6].

2. Энергия подается в правильные фазы колебательной системы. В противном случае, подведение энергии извне может наоборот привести к затуханию колебаний, вместо их поддержания [6].

Оба этих условия соблюдаются в классическом автогенераторе, использующем нелинейный усилитель. Его принципиальная схема приведена ниже:

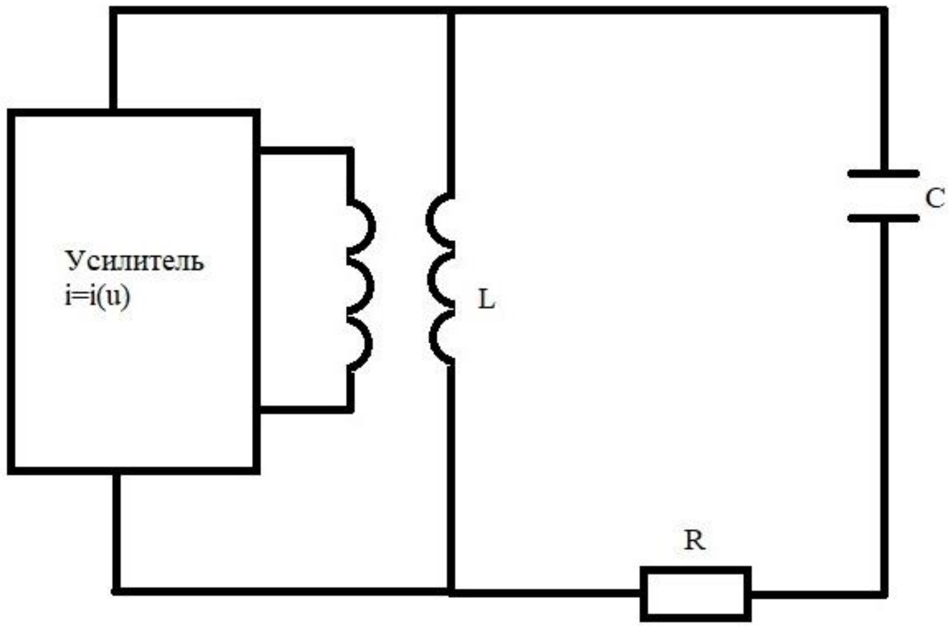

Рисунок 8: принципиальная схема автогенератора [7]

Напряжение из RLC контура подается на вход усилителя. Не вдаваясь в техническое устройство самого усилителя, скажем лишь, что это устройство, преобразующее напряжение на входе (подающиеся из колебательного контура) в ток на выходе. Принципиально важно, что этот усилитель нелинейный. Это значит, что закон, по которому он преобразует подающийся на него ток в напряжение, нелинейный:

$$i = i(u) = g_0 u + g_1 u^2 - g_2 u^3 + \cdots \qquad (1.3.1)$$

Где g_0, g_1, g_2 – это характеристики нелинейности, которые будем считать известными. Они так же должны являться положительными.

Выход усилителя подан на свою катушку, которая индуктивно связана с катушкой колебательного контура. Таким образом, в цепи создана обратная связь. Принципиально важно, что коэффициент обратной связи - нелинейная функция. В противном случае, колебания бы неограниченно нарастали, так как в ответ на увеличение входного напряжения усилителем он же непрерывно отвечал бы еще большим увеличением выходного тока, который через индуктивную связь еще больше увеличивал бы входной ток на усилителе. Однако, так как коэффициент обратной связи убывает по мере роста напряжения, выходной ток так же будет мал при большом напряжении (амплитуде колебаний в RLC цепи). Таким образом, удается сколь угодно долго поддерживать колебания, близкие к автоколебаниям.

Напишем уравнение автоколебаний. Сделаем это, записав уравнение токов (первый закон Кирхгоффа) [8]:

$$i(u) = C\frac{du}{dt} + \frac{1}{R}u + \frac{1}{L}\int_0^t u dt \qquad (1.3.2)$$

Где u – напряжение в колебательном контуре (q/C)

M – коэффициент индуктивной связи между индуктором контура и усилителя

Решение этого уравнения явно зависит от коэффициентов нелинейности g_o, g_1 и g_2 а так же наличия других коэффициентов нелинейности. Тем не менее, для случая с тремя коэффициентами нелинейности и проведя замены (1.3.3) и (1.3.4), можно получить результат (1.3.5).

$$\tau = \omega t \qquad (1.3.3)$$

$$\omega = \frac{1}{\sqrt{LC}} \qquad (1.3.4)$$

ω – собственная частота RLC контура

$$\frac{d^2u}{d\tau^2} - \mu(1 + g_0u - g_1u^2 - g_2u^4)\frac{du}{d\tau} + u = 0 \qquad (1.3.5)$$

Если положить все коэффициенты g, кроме g_1, равными нулю, а его самого равным единице, то полученное уравнение, по сути, будет представлять собой уравнение колебаний Ван-дер-Поля. В дальнейших главах рассмотрена синхронизация колебаний автогенераторов, как внешним сигналом, так и взаимная нескольких генераторов.

1.4. Синхронизация

В обобщенном смысле, под синхронизацией колебаний следует понимать явление, при котором колебательные системы, имеющие различные частоты колебаний, вследствие различных процессов, устанавливают единый ритм колебаний. Обязательным требованием для синхронизации колебаний является наличие обратных связей между колеблющимися системами. Эти связи могут быть различной степени «жесткости» [9].

Приведем пример жесткий связей. Так, если мы будем раскачивать нетяжелый маятник, крепко обхватив его рукой, то частота его колебаний будет совпадать с частотой, с которой мы покачиваем его рукой. Даже если его собственная частота колебаний отлична от выбранной нами для его раскачки, фактически, мы можем задать ему любую частоту колебаний, если приложим для этого достаточно усилий. В этом нет ничего удивительного – приведенный пример, по сути, является колебаниями под действием внешней силы.

Но что будет, если мы приложим недостаточно усилий для искусственного раскачивания маятника? Здесь возможны два варианта. Если мы выберем частоту внешнего воздействия, близкую к собственной частоте колебаний маятника, то произойдет синхронизация – то есть, колебания маятника примут некоторую среднюю частоту между его собственной частотой и частотой внешней периодической силы. Это значит, что выбранная нами частота попала в полосу синхронизации маятника. Полоса синхронизации – это диапазон частот, в пределах которого связанные колебания могут прийти к единому ритму. Очевидно, что полоса синхронизации зависит как от физических свойств самой колебательной системы, так и от амплитуды (мощности) внешнего воздействия. Как уже было отмечено, имея возможность приложить к осциллятору неограниченно мощное внешнее воздействие, можно вынудить его осциллировать с любой частотой. Потому, полоса синхронизации тем шире (охватывает больший диапазон частот), чем выше амплитуда внешнего воздействия.

Однако, если же мы подберем частоту внешнего воздействия, не попадающую в полосу синхронизации данного осциллятора при данной амплитуде воздействия, то система, перейдет в режим биений. Биение – это особенный тип колебаний, амплитуда которых периодически возрастает и убывает. Это свидетельствует о том, что в спектре полученных колебаний присутствуют обе рассмотренные частоты, то есть, мощности вынуждающей силы оказалось недостаточно для «захватывания» частоты осциллятора. Ниже приведен пример таких биений, возникающих при наложении двух гармонических колебаний близкой частоты [10]:

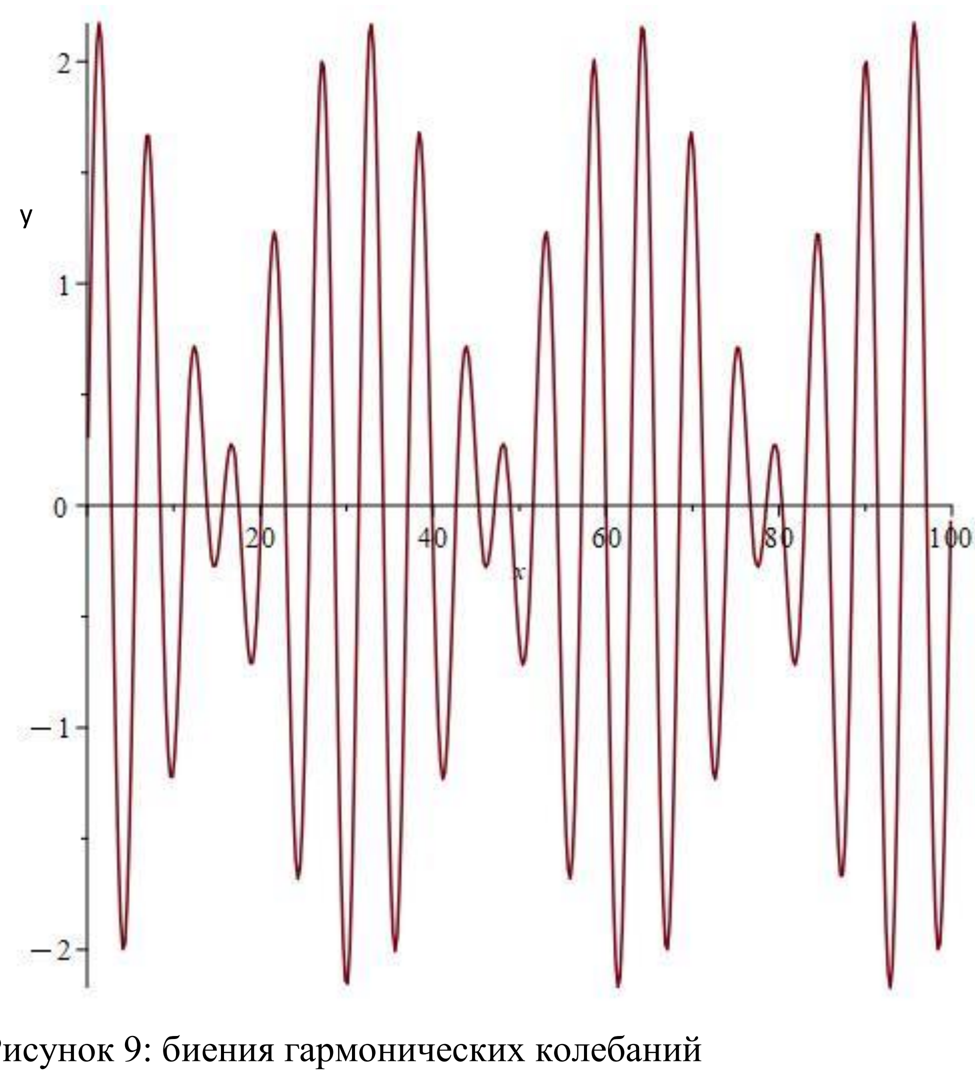

Рисунок 9: биения гармонических колебаний

Похожая картина так же наблюдается и при описанной ситуации неудачного захвата частоты нелинейных колебаний.

В 1655 году, голландский ученый Христиан Гюйгенс, описал наблюдавшуюся им «мягкую» синхронизацию. Он заметил, как двое часов, висевших на одной стене (имеются ввиду маятниковые часы) поддерживали одинаковый ритм. Искусственно расстраивая их ритм, он обнаружил, что в течение получаса они вновь пришли к режиму синхронизации.

Гюйгенс сразу догадался, что причина наблюдавшегося им явления состоит в некоторой связи между часами. Проведя ряд простых экспериментов, он установил, что источником этой обратной связи как раз является балка, на которой были закреплены часы. Таким образом, от одного маятника балке передаются колебания в определенной фазе. Второй маятник, если его фаза не совпадала с фазой колебаний первого, будет как бы подогнан им. Это будет происходить до тех пор, пока маятники не достигнут синхронизма.

Маятники могут синхронизироваться двумя способами – в синфазе и противофазе [11]. В первом случае балка так же будет колебаться вместе с маятниками в одной фазе, а во втором случае она, напротив, будет неподвижна. Именно синхронизацию в противофазе в 1655 году наблюдал Гюйгенс.

Таким образом, существует два основных типа синхронизации:

1. Жесткая синхронизация, или захват частоты – когда осцилляторы подстраиваются под одну из колебательных систем.
2. Мягкая синхронизация – когда все осцилляторы равноправны

Следует отметить, что разница между этими типами синхронизации условна – в действительности, будет ли синхронизации жесткой или мягкой зависит лишь от отношения мощностей источников.

1.5. Модели синхронизации

Рассмотрим математическую модель синхронизации автогенератора внешним сигналом. Для этого возьмем уравнение (1.3.5) и приравняем его левую часть некоторому внешнему гармоническому воздействию:

$$\frac{d^2u}{d\tau^2} - \mu(1 + g_0u - g_1u^2 - g_2u^4)\frac{du}{d\tau} + u = Asin(\Omega\tau) \qquad (1.5.1)$$

Где A – амплитуда внешнего сигнала

Ω – отношение частоты автогенератора к частоте внешнего сигнала.

Амплитуду A следует считать достаточно большой, либо отношение частот Ω достаточно близким к единице. В противном случае, система перейдет в режим биений, что продемонстрировано в практической части работы.

Уравнение (1.5.1) можно превратить в систему уравнений, описывающих синхронизацию нескольких генераторов внешним сигналом.

$$\begin{cases} \frac{d^2u_1}{d\tau^2} - \mu(1 + g_0u_1 - g_1{u_1}^2 - g_2{u_1}^4)\frac{du_1}{d\tau} + u_1 = Asin(\Omega_1\tau) \\ \frac{d^2u_2}{d\tau^2} - \mu(1 + g_0u_2 - g_1{u_2}^2 - g_2{u_2}^4)\frac{du_2}{d\tau} + u_2 = Asin(\Omega_2\tau) \end{cases} \qquad (1.5.2)$$

Где u_1 – отклонение первого автогенератора

u_2 – отклонение второго автогенератора

Ω_1, Ω_2 – отношения частот первого и второго генератора к частоте внешнего сигнала соответственно

Нетрудно заметить, что уравнения в этой системе никак не связаны между собой и могут решаться по отдельности. Это значит, что такая модель не учитывает взаимного влияния автогенераторов друг на друга. Схема такой синхронизации может быть представлена следующим образом:

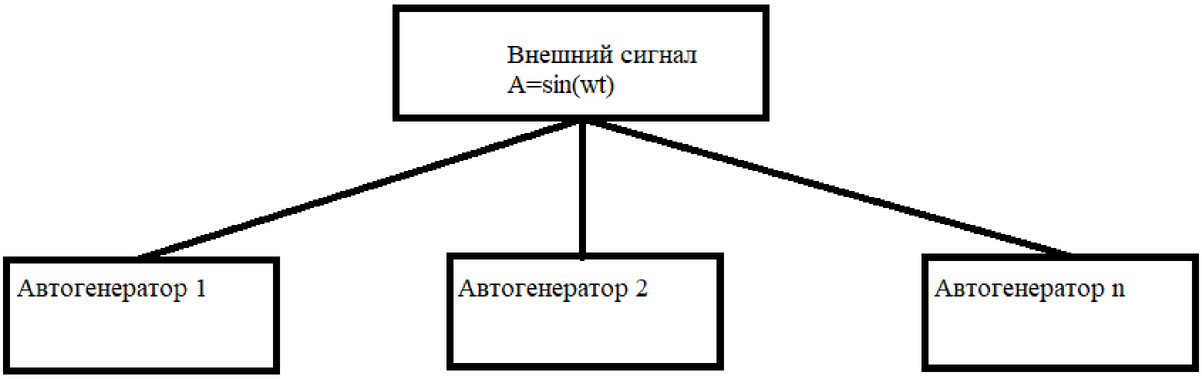

Рисунок 10: схема синхронизации n генераторов внешним сигналом

Как было указано ранее, возможен и другой тип синхронизации, при котором генераторы оказывают друг на друга взаимное влияние:

Рисунок 11: схема взаимной синхронизации автогенераторов

Для того, что бы построить математическую модель такой синхронизации, необходимо учесть степень взаимного влияния генераторов друг на друга. Для этого введем коэффициенты связи автогенераторов k_i.

$$\begin{cases} \frac{d^2u_1}{d\tau^2} - \mu(1 + g_0u_1 - g_1{u_1}^2 - g_2{u_1}^4)\frac{du_1}{d\tau} + \Omega_1u_1 + \sum k_iu_j = 0 \\ \frac{d^2u_2}{d\tau^2} - \mu(1 + g_0u_2 - g_1{u_2}^2 - g_2{u_2}^4)\frac{du_2}{d\tau} + \Omega_2u_2 + \sum k_iu_j = 0 \\ \frac{d^2u_n}{d\tau^2} - \mu(1 + g_0u_n - g_1{u_n}^2 - g_2{u_n}^4)\frac{du_n}{d\tau} + \Omega_2u_n + \sum k_iu_j = 0 \end{cases} \quad (1.5.3)$$

Данная система уравнений описывает взаимную синхронизацию n штук автогенераторов.

Аналогичным образом можно построить модель синхронизации колебаний механических маятников [12]:

$$\begin{cases} \ddot{\varphi} + \beta sin\varphi + \mu\dot{\varphi} = dsin(\alpha - \varphi) \\ \ddot{\alpha} + \beta sin\alpha + \mu\dot{\alpha} = dsin(\varphi - \alpha) \end{cases} \tag{1.5.4}$$

Где φ и α – отклонения первого и второго маятников соответственно.

Численному решению при помощи компьютера посвящена практическая часть данной работы

2.1. Python

Языком программирования для моделирования процесса синхронизации колебаний мною был выбран Python. Этот язык прост в освоении и активно развивается, но главным его достоинством является наличие большого количества дополнительных, написанных пользователями библиотек, позволяющих решать широкий спектр задач, в том числе инженерного и научного толка.

Язык Python был написан в 1991 году голландским программистом Гвидо ван Россумом. С тех пор язык активно совершенствовался как его разработчиком, так и его крупным и активным сообществом пользователей. Актуальной в настоящий момент является версия 3.8, именно на ней и были написаны моделирующие программы.

Существует множество интерпретаторов данного языка. Мною был выбран PyCharm из-за удобного, интуитивно понятного интерфейса и его функционала, в частности, возможности устанавливать библиотеки без обращения к командной строке.

Python – это высокоуровневый язык программирования. Его узнаваемым особенностями являются динамическая типизация, отсутствие точек с запятой в конце каждой строчки кода, автоматическое управление памятью, простота синтаксиса. Все это делает Python удобным языком программирования для программиста как на этапе создания программ, так и на этапе их дальнейшего развития за счет хорошей читаемости кода.

В то же время, основным недостатком языка является меньшая производительность написанных на нем программ по причине особенностей управления памятью. Однако, ввиду того, что написанная в рамках данной работы модель не потребляет большого количества памяти и не требует от компьютера серьезной производительности, этот недостаток был расценен как несущественный в рамках работы.

Конечной целью данной работы было решение системы нелинейных дифференциальных уравнений и представление полученных результатов в виде, позволяющим пользователю определить, произошла ли при введенных им параметрах синхронизация колебаний или нет.

Численное решение дифференциальных уравнений – это задача, уже давно решенная сообществом Python. В этом нам помогут библиотеки команд – SciPy и NumPy. Для визуализации полученных решений использовался пакет Matplotlib.

Рассмотрим подробнее использованные дополнительные пакеты и их возможности.

2.2. NumPy

Библиотека NumPy в Python служит для работы со множествами и массивами данных. Она позволяет создавать множества различных размерностей (одномерные, двумерные и так далее) и производить над ними различные математические операции. Эти операции могут включать в себя:

- Умножение на число
- Прибавление числа (к каждому элементу)
- Сложение множеств (одинаковых размерностей)
- Перемножение множеств (соответствующих размерностей)
- Проведение над элементами множества различных математических операций (таких, как взятие синуса, косинуса и так далее)
- Транспонирование
- Нахождение определителя матрицы
- Некоторые другие

Хотя множества существуют и в самом языке Python, пакет NumPy существенно упрощает работу с ними и расширяет базовый функционал языка.

Непосредственно в данной работе применялась возможность создать большой ряд значений, играющих роль независимой переменной одной командой (linspace(min, max, numpoints)). Была применена возможность взять синус каждого элемента массива данных для набора дифференциальных уравнений. Так же, решения дифференциальных уравнений записывались для ряда значений в многомерный массив, что упрощало работу с ними. Это общепринятая практика работы с дифференциальными уравнениями на языке Python.

Разработчиками была составлена подробная документация, позволяющая легко освоить пакет новому пользователю.

2.3. SciPy

Данная библиотека содержит в себе большое количество команд, позволяющий решать широкий спектр научных и прикладных задач. В данной работе, непосредственный интерес предоставляет возможность численного решения дифференциальных уравнений. Данный пакет позволяет решать обычные дифференциальные уравнения любого порядка, как линейные так и нелинейные.

Команда odeint решает дифференциальные уравнения методом Рунге-Кутты-Фельберга 4-5 порядка. Это численный метод, применяемый для решения обыкновенных дифференциальных уравнений. Этот метод является итерационным, то есть приблизительное значение уравнение последовательно высчитывается в каждой точке, начиная от начальных условий. Метод Рунге-Кутты-Фельберга – это улучшенная версия классического метода Рунге-Кутты. Главное их отличие в том, что метод Рунге-Кутты-Фельберга автоматически изменяет шаг интегрирования, что бы достичь большей точности.

Стоит отметить, что сам по себе методы Рунге-Кутты-Фельберга пригоден для численного решения дифференциальных уравнений только лишь первого порядка. Однако, любое уравнение второго порядка:

$$\frac{d^2x}{dt^2} + f(x,t) = 0 \qquad (2.3.1)$$

Можно свести к системе из двух уравнений первого порядка:

$$\begin{cases} \frac{d\dot{x}}{dt} + f(x,t) = 0 \\ \frac{dx}{dt} = \dot{x} \end{cases} \qquad (2.3.2)$$

Аналогичным образом можно решать уравнения более высокого порядка. При необходимости решить систему из двух дифференциальных уравнений второго порядка (которая возникает в данной работе), следует представить их в виде системы из четырех дифференциальных уравнений первого порядка.

Минусом реализации метода Рунге-Кутты-Фельберга в пакете SciPy является тот факт, что сводить уравнения высоких порядков к системе необходимо вручную. Например, в таких программах, как Matlab и Maple достаточно ввести дифференциальное уравнение в любом удобном виде. Кроме того, команда odeint имеет неудобный синтаксис, требующий разделять производную в левой части и остальное уравнение в правой. Такой вид отличается от общепринятого, когда сумма всех функций, участвующих в уравнении, приравнивается к нулю.

2.4. Matplotlib

Эта библиотека используется для визуализации данных. Она способна строить различные диаграммы – графики, в том числе неявные, гистограммы и прочее. В данной работе пакет использовался для построения графиков колебаний и их фазовых портретов. Для этого необходимо ввести в качестве аргументов два вектора, один из которых будет отвечать за значения по оси X, а другой за значения по оси Y. Кроме того, в качестве аргументов можно указать дополнительные параметры графиков – их желаемый цвет, внешний вид, наличие сетки и многое другое.

Данная библиотека поддерживает вывод данных в разнообразных форматах, будь то PNG, JPEG или PDF и так далее.

С помощью библиотеки можно так же создавать анимированные и интерактивные графики. Те графики, что были выведены через команду «show» можно масштабировать без потери качества изображения.

Благодаря своей гибкости и простоте использования, Matplotlib остается одной из самых популярных библиотек для визуализации данных в Python сообществе. Ее удобно комбинировать с другими пакетами, в том числе использовавшимися в данной работе пакетами SciPy и NumPy.

2.5. Полученные программы

В результате выполнения данной работы, были написаны три программы. Эти программы решают определенные дифференциальные уравнения (и их системы), моделирующие различные виды синхронизации механических и электромагнитных колебаний. Эти уравнения (4.1), (4.2) и (4.3) были представлены в конце главы «Модели синхронизации». Однако, как было сказано ранее, библиотека SciPy накладывает определенные требования на формат вводимых дифференциальных уравнений. Перепишем эти уравнения в соответствии с этими требованиями:

$$\begin{cases} \frac{du_1}{d\tau} = v \\ \frac{dv}{dt} = -\mu(1 + g_0 u_1 - g_1 {u_1}^2 - g_2 {u_1}^4)v - u_1 - Asin(\omega\tau) \end{cases} \tag{2.5.1}$$

Эта система уравнений описывает синхронизацию автогенератора внешним сигналом (1.5.1). За ее моделирование отвечает программа VDPSincsig (код программы представлен в приложении).

Управляющими параметрами являются:

- μ – коэффициент нелинейности колебаний
- g_0 – первый коэффициент нелинейности генератора
- g_1 – второй коэффициент нелинейности генератора
- g_2 – третий коэффициент нелинейности генератора
- A – амплитуда внешнего сигнала
- ω – отношение частоты внешнего сигнала к собственной частоте системы

На выходе программа предоставляет графики колебаний автогенератора и внешнего сигнала, а так же фазовый портрет колебаний генератора. График внешнего сигнала – оранжевый.

$$
\begin{cases}
\frac{d^2u_1}{d\tau^2} = -\mu(1 + g_0u_1 - g_1{u_1}^2 - g_2{u_1}^4)v_1 - u_1 - k_1u_2 \\
\frac{du_1}{dt} = v_1 \\
\frac{d^2u_2}{d\tau^2} = -\mu(1+g_0u_2-g_1{u_2}^2-g_2{u_2}^4)v_2-\Omega_1u_2-k_1u_1 \\
\frac{du_1}{dt} = v_2
\end{cases}
\qquad (2.5.2)
$$

Система уравнений (2.5.2) описывает взаимную синхронизацию двух одинаковых автогенераторов (1.5.3). Она приведена в вид, удовлетворяющий требованиям команды odeint и реализована в программе VDPsincself.

Управляющими параметрами являются:

- μ – коэффициент нелинейности колебаний
- g_0 – первый коэффициент нелинейности генератора
- g_1 – второй коэффициент нелинейности генератора
- g_2 – третий коэффициент нелинейности генератора
- Ω_1 – отношение частоты второго генератора к частоте первого
- k_1 – коэффициент связи первого генератора со вторым
- k_2 – коэффициент связи второго генератора с первым

На выходе программа предоставляет три графика – график колебаний первого генератора, график колебаний второго генератора и график разности их колебаний в тех же точках. Произошла синхронизация или нет можно определить по характеру третьего графика. Его приблизительно гармоническая форма указывает на то, что разность колебаний зависит лишь от соотношения амплитуд колебаний, но разность фаз постоянна. Кроме того, программа предоставит два фазовых портрета – первого и второго генераторов соответственно.

Наконец, рассмотрим систему уравнений (1.5.4) реализованную в последней программе Msincself в качестве системы уравнений (2.5.3):

$$\begin{cases} \dot{\varphi} - \beta_1 sin\varphi - \mu\vartheta_1 + dsin(\alpha - \varphi) \\ \vartheta_1 = \dot{\varphi} \\ \dot{\alpha} - \beta_2 sin\alpha - \mu\vartheta_2 + dsin(\varphi - \alpha) \\ \vartheta_2 = \dot{\alpha} \end{cases} \qquad (2.5.3)$$

Управляющие параметры:

- β_1 и β_2 – отвечают за частоты маятников
- d – коэффициент связи между маятниками
- μ – показатель затухания

Эта программа на выходе предоставляет три графика и три фазовых портрета. Первые два фазовых портрета так же представляют обе системы в отдельности, а третий одновременно показывает оба фазовых портрета. Определить, произошла ли синхронизация можно по третьему графику и третьему фазовому портрету.

2.6. Анализ результатов. Захват частоты.

Проанализируем результаты, полученные в результате работы программ. Начнем с наиболее простой программы – VDPsincsig.

Характеристики нелинейности были взяты 0.1, 3 и 0.2 соответственно. Хотя программа позволяет указать любые, эти позволяют смоделировать автогенератор с единственным предельным циклом, что удобно

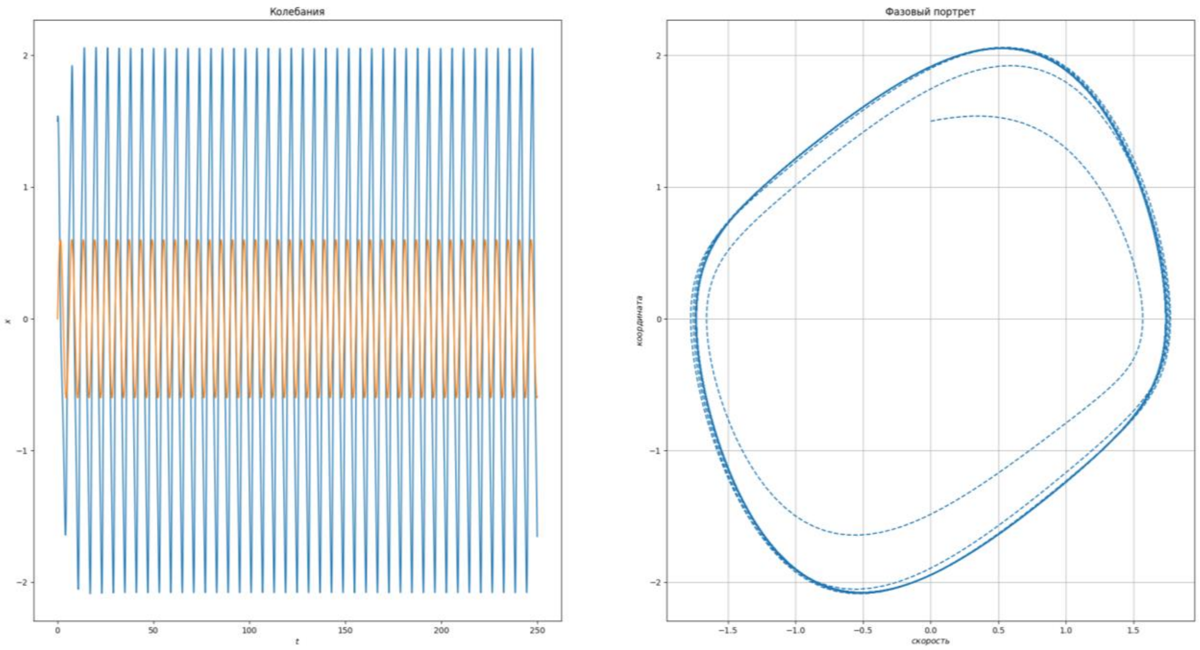

Рисунок 13: VPDSincsig при A=0.6, ω=1.05

Хорошо видно, что при выбранных параметрах автогенератор хорошо синхронизирован со внешним сигналом. То есть, произошел захват частоты. В этом так же можно убедиться, увеличив изображение:

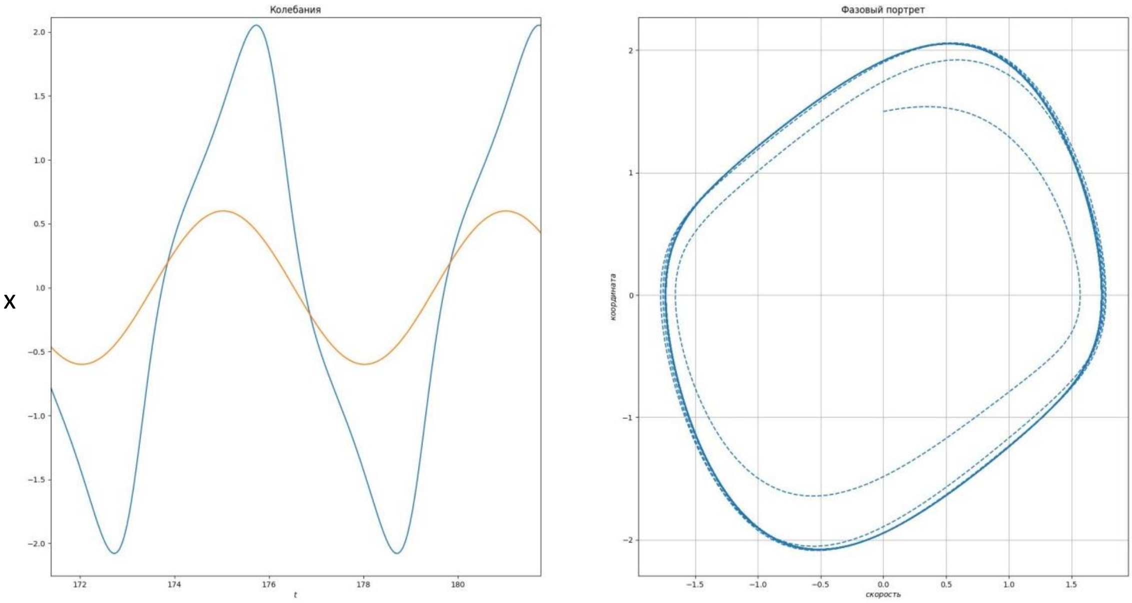

Рисунок 14: предыдущее изображение увеличенное

В данном случае частоты различались всего на 5 процентов. Попробуем увеличить частоту внешнего сигнала до 110%:

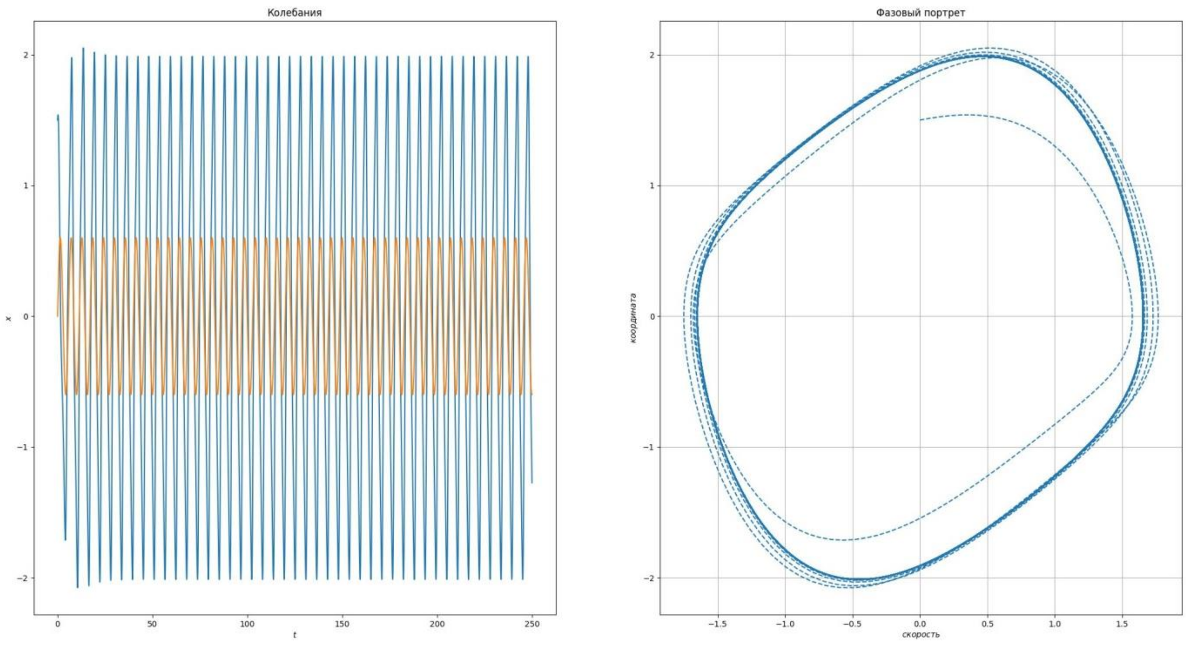

Рисунок 15: VPDSincsig при A=0.6, ω=1.1

Как видно, синхронизация по прежнему происходит, но разность фаз стала заметно больше:

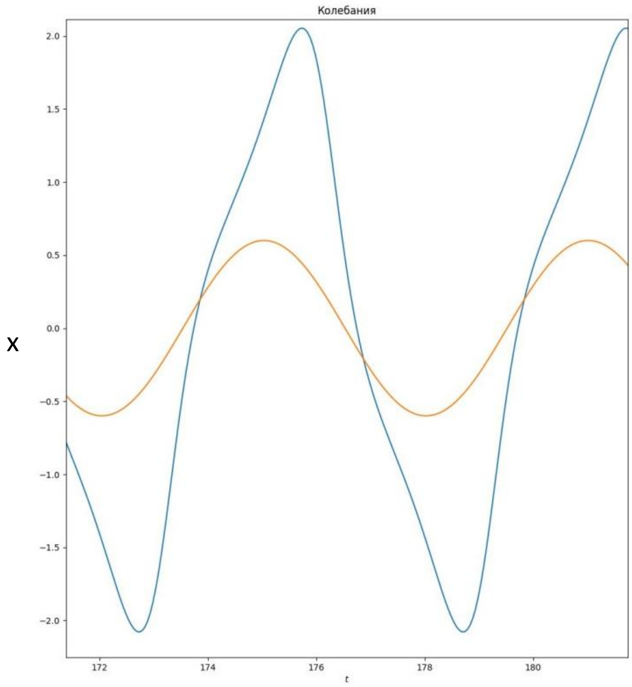

Рисунок 16: предыдущее изображение увеличенное

При дальнейшем увеличении частоты внешнего сигнал режим синхронизации нарушается:

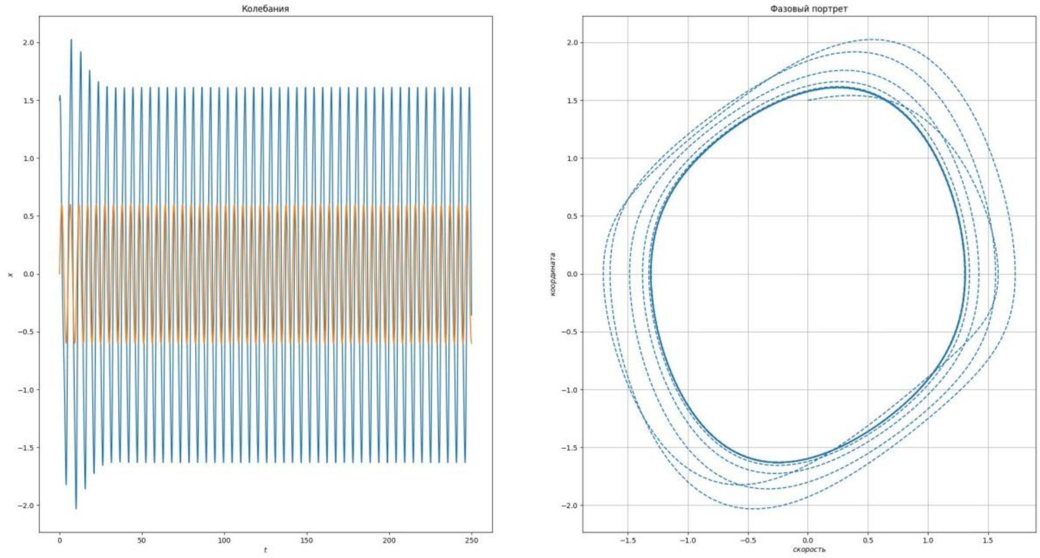

Рисунок 17: VPDSincsig при A=0.6, ω=1.2

А затем явно переходит в режим биений:

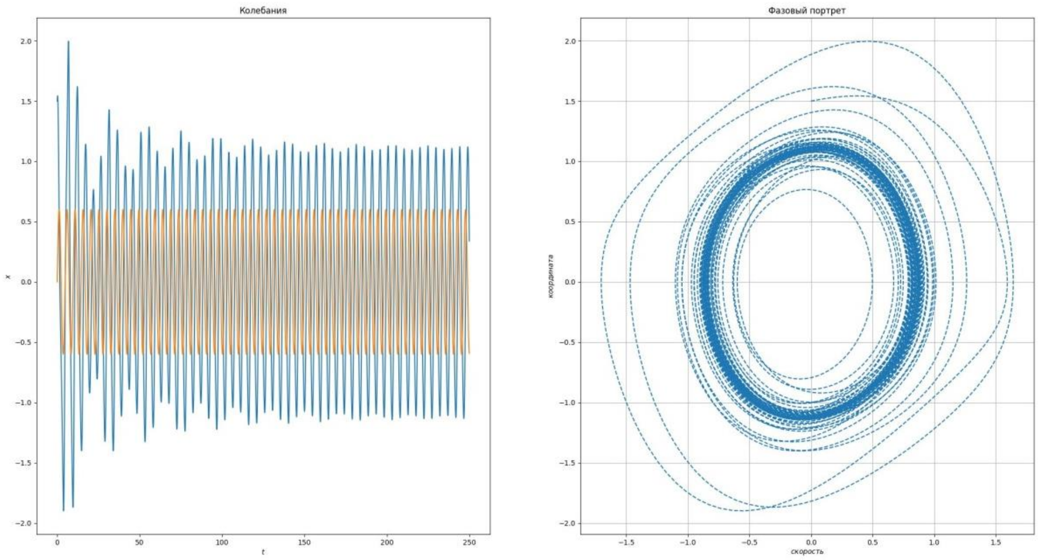

Рисунок 18: VDPSincsig при A=0.6, ω=1.3

Режим биений можно определить по периодически меняющейся амплитуде генератора, если посмотерть на левый график, или по наличию нескольких окружностей, по которым последовательно перемещается фазовая траектория, если посмотреть на правый.

Тем не менее, режим синхронизации можно восстановить, увеличив амплитуду внешнего сигнала.

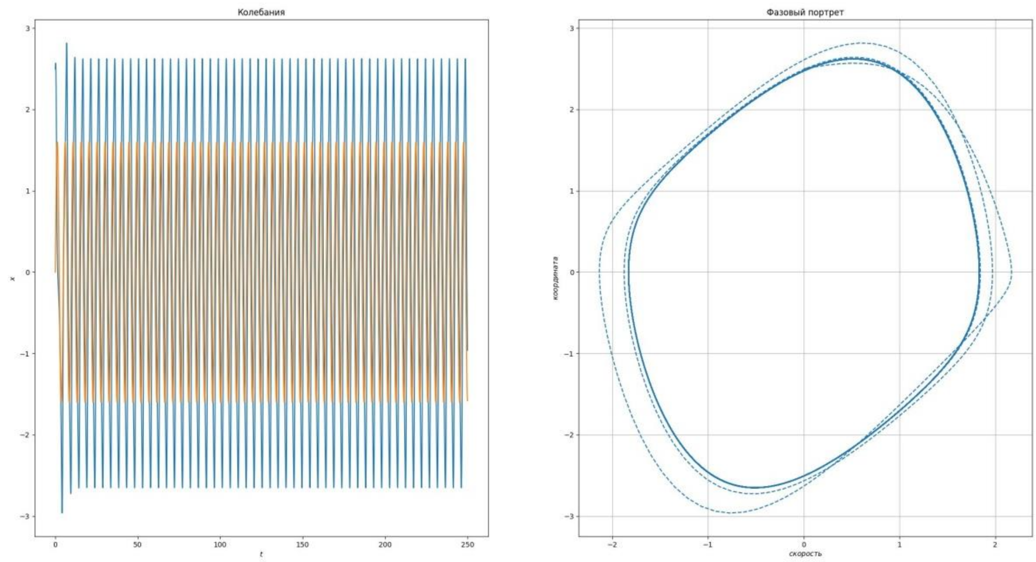

Рисунок 19: VDPSincsig при A=1.6, ω=1.3

Таким образом, ширина полосы синхронизации действительно прямо пропорциональна амплитуды внешнего сигнала. Это подтверждает предположения, выдвинутые в предыдущих главах теоретической части. Проведя множество подобных моделирований, можно было бы построить график зависимости максимальной частоты, при которой возможна синхронизация, от амплитуды внешнего воздействия.

Перейдем к анализу результатов моделирования взаимной синхронизации двух автогенераторов, то есть, полученных при помощи программы VDPSincself.

2.7. Анализ результатов. Взаимная синхронизация.

Возьмем те же параметры μ, g_0, g_1 и g_2 для двух генераторов, как и для генератора из предыдущей программы. Коэффициенты k_1 и k_2 (влияние второго генератора на первый и первого на второй) возьмем равными 0,4.

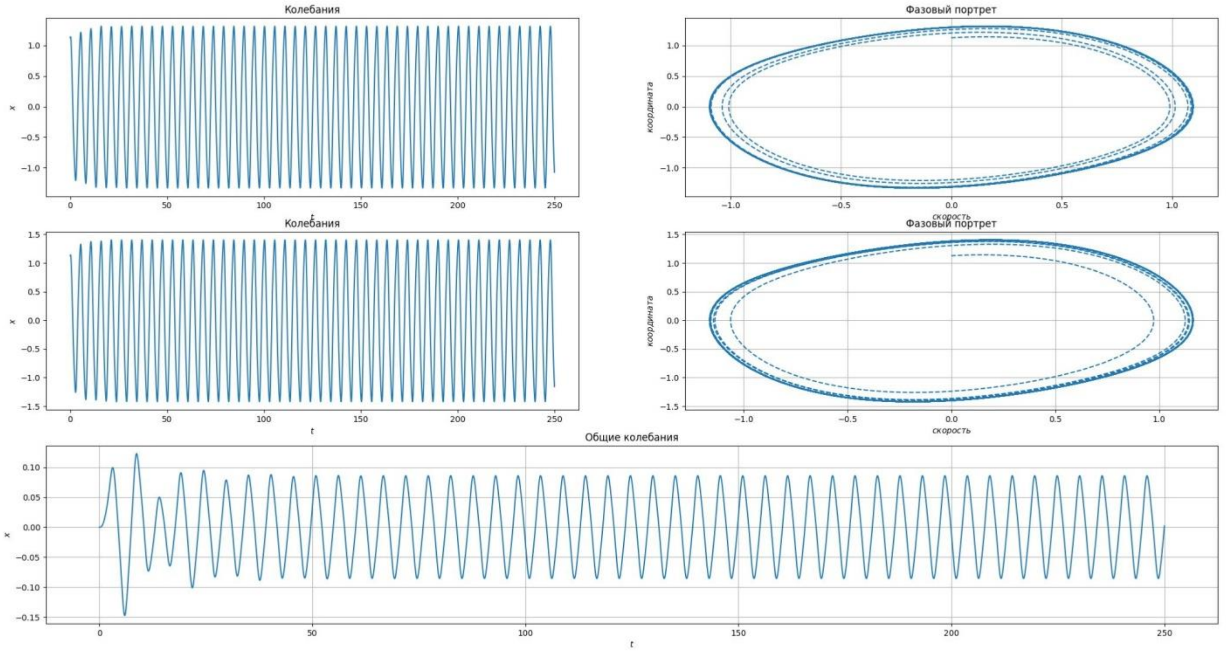

Рисунок 20: VDPsincself при k_1=0,4 k_2=0,4 ω=1,05

Масштаб верхних двух графиков примерно равен 1,2 единицы, в то время как нижнего – около 0,1, что примерно на порядок меньше. Это значит, что амплитуды колебаний генераторов примерно равны. Видно так же, что в начале третьего графика колебания имеют явно негармоническую форму. Вскоре, приблизительно после t=35 форма разности колебаний принимает приближенно гармонический вид. Это значит, что генераторы пришли к режиму синхронизации.

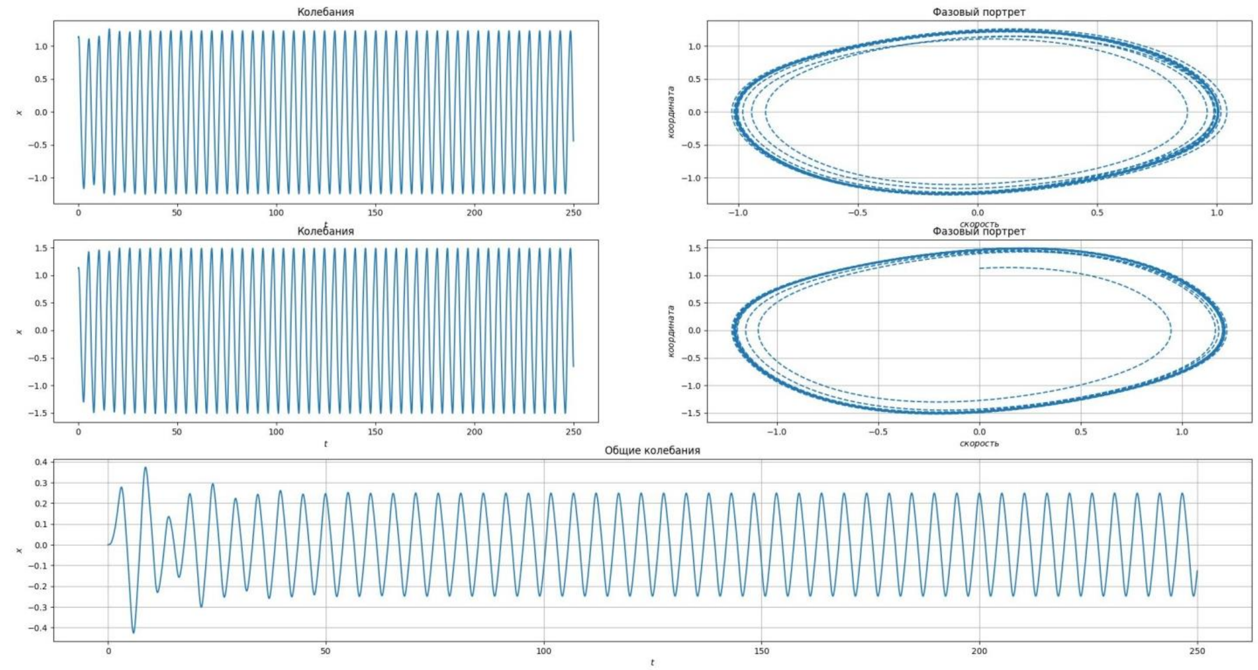

Рисунок 21: VDPsincself при k_1=0,4 k_2=0,4 ω=1,15

Масштаб третьего график вырос до 0,3 – 0,4, что уже составляет порядка 20-25% процентов от амплитуды синхронизируемых колебаний. Режим синхронизации между ними так же наступает приблизительно на t=35.

При дальнейшем росте разницы частот режим синхронизации наступает позже:

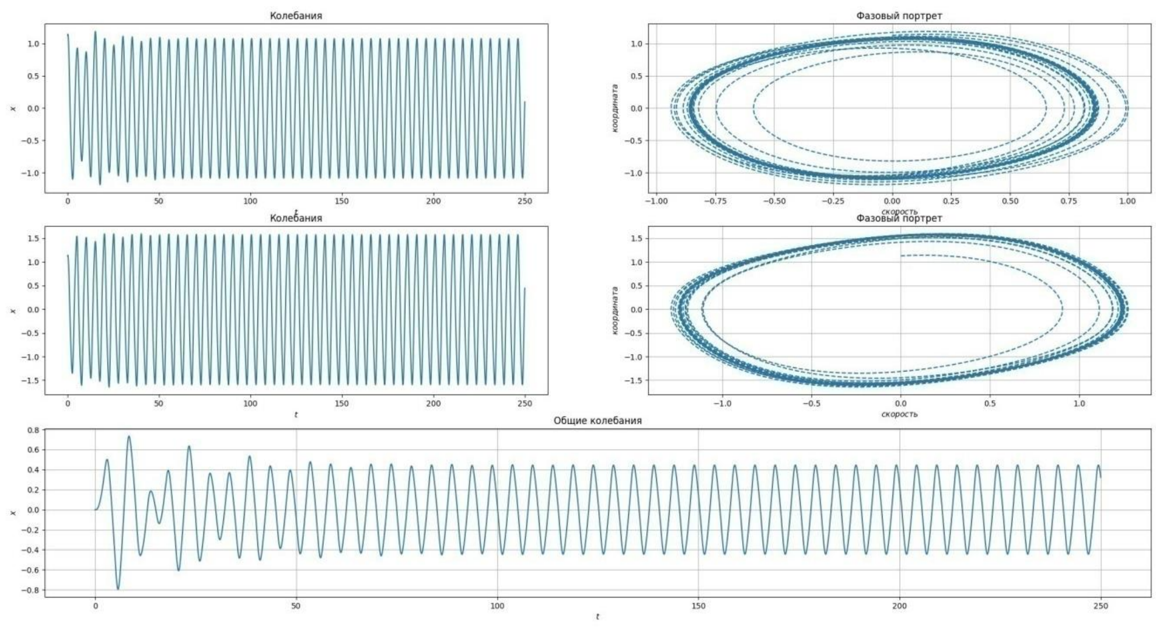

Рисунок 22: VDPsincself при k_1=0,4 k_2=0,4 ω=1,3

Амплитуда разницы колебаний составляет уже порядка одной трети от амплитуды самих колебаний. Затем, при дальнейшем увеличении частоты второго генератора еще на 20% система переходит в режим биений:

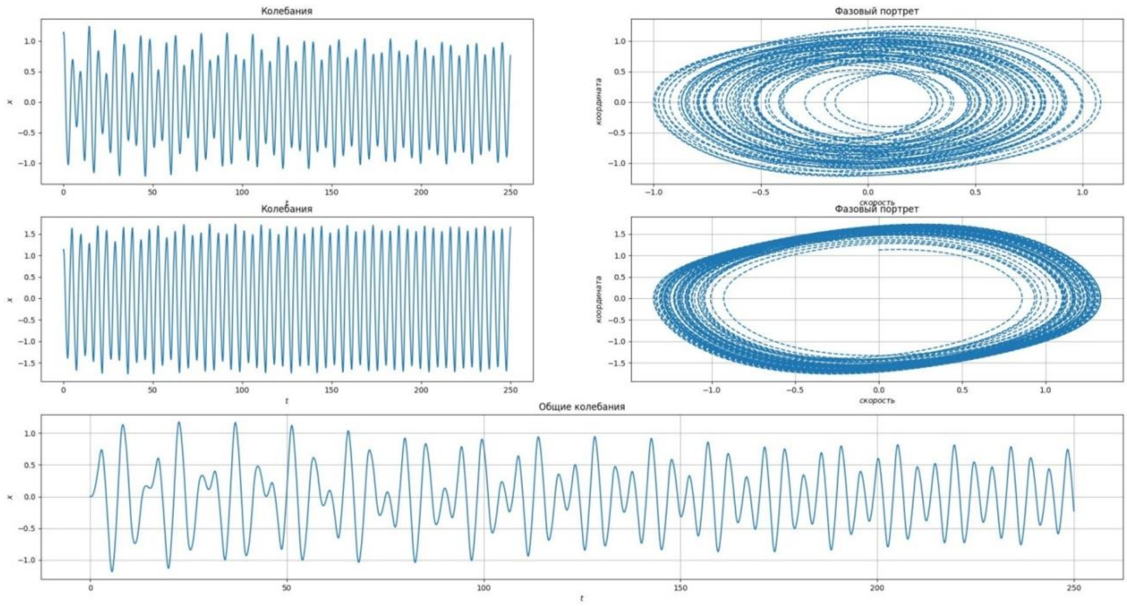

Рисунок 23: VDPsincself при k_1=0,4 k_2=0,4 ω=1,5

Режим биений хорошо идентифицируется по любому из пяти представленных графиков. Однако, режим синхронизма генераторов по прежнему можно восстановить, увеличив коэффициенты обратной связи:

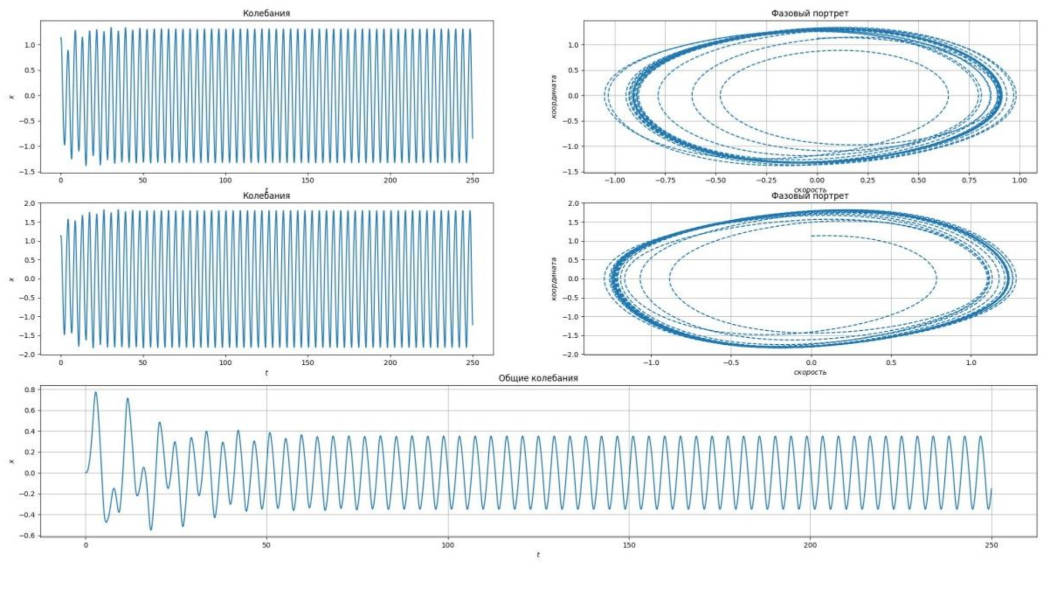

Рисунок 24: VDPsincself при k_1=0,8 k_2=0,8 ω=1,5

Попробуем воссоздать при помощи VDPSincself картину, полученную при работе VDPSincsig. Для этого укажем значения коэффициентов k_2 и μ равными нулю, тем самым превратив схему из рисунка 10 в таковую из рисунка 9.

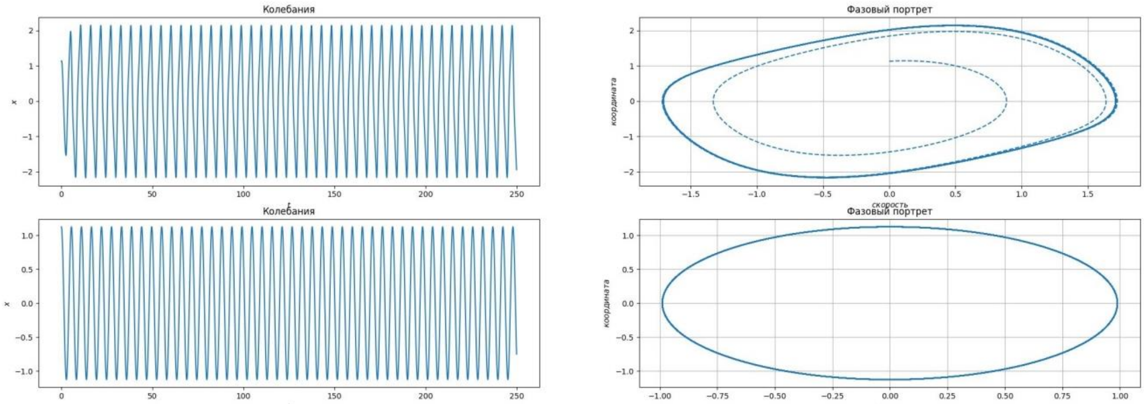

Рисунок 25: VDPsincself при k_1=0,8 k_2=0 ω=1,3

Как видно, фазовый портрет вторых колебаний представляет собой замкнутый эллипс, что соответствует периодическим колебаниям, на которые не оказывается внешнего воздействия.

Полученные результаты наглядно показывают, что между явлениями захватывания частоты и взаимной синхронизации нет принципиальной разницы. По сути, это одно и то же явление, разница лишь в том, в какой степени осцилляторы способны оказывать влияние друг на друга.

Это так же значит, что программа VDPSincself способна выполнять функции программы VDPSincsig. Однако, последняя имеет более понятный синтаксис для целей моделирования процесса захватывания частоты.

Перейдем к рассмотрению программы Msincself, моделирующей синхронизацию затухающих колебаний двух маятников.

2.8. Анализ результатов. Затухающие колебания

По аналогии с предыдущими программами, будем считать коэффициент β_1 равным единице, а изменять будем только параметр β_2. Это позволит рассматривать коэффициент β_2 как его же отношение к коэффициенту β_1, что упростит сравнение полученных результатов с таковыми из предыдущих программ. Коэффициент μ возьмем равным 0.05, чтобы колебания затухали не слишком быстро. Коэффициент d = 0.1

Возьмем для первого случая β_2 равным 1,01. Получим следующие результаты:

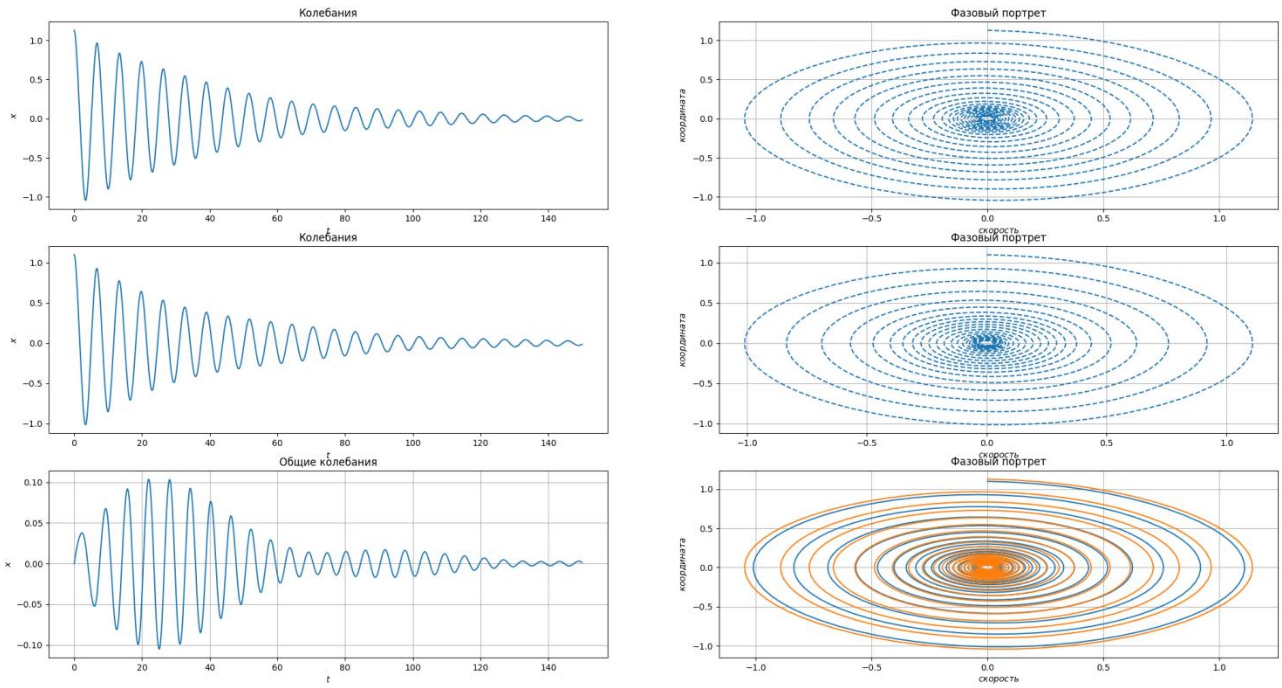

Рисунок 26: MSincself при β_2 = 1,01 и d = 0,1

График разницы колебаний имеет очень малый масштаб. Сперва его амплитуда нарастает, а затем убывает, как и амплитуда колебаний самих осцилляторов. Причина такого поведения станет ясна позднее. Видно, что система быстро пришла в режим синхронизма.

Увеличим разницу частот:

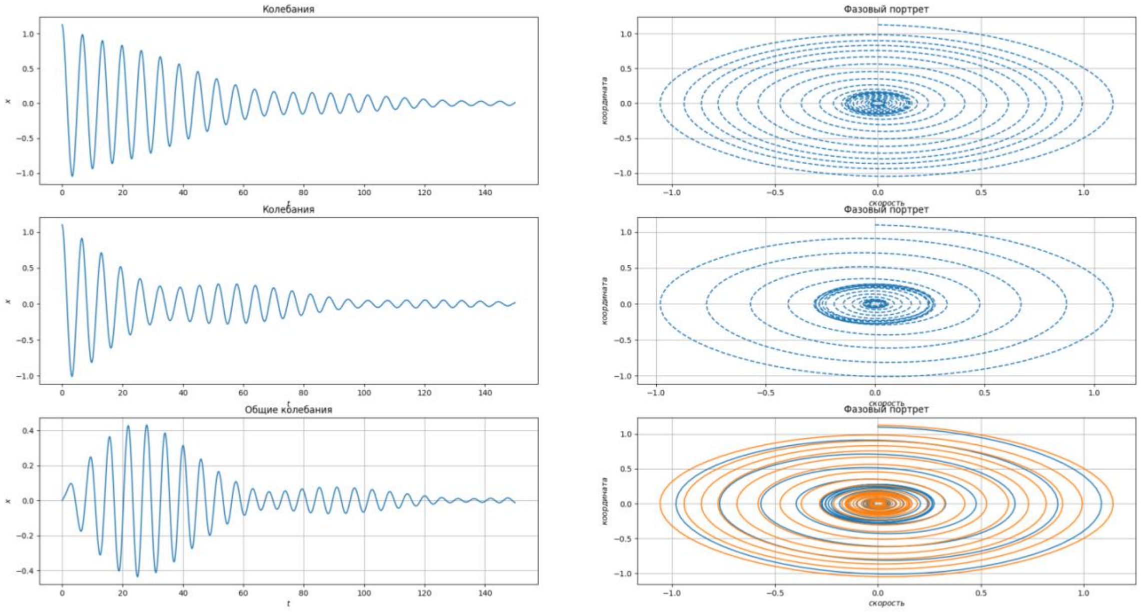

Рисунок 27: MSincself при $\beta_2 = 1{,}05$ и $d = 0{,}1$

Колебания с большей частотой затухали быстрее (Так как колебания нелинейные, это нормально). Однако, из-за обратной связи, после некоторого значения их амплитуда практически перестала убывать – напротив, стала несколько нарастать. Лишь после того, как их амплитуды сравнялись, затухание продолжилось. Режим синхронизации, если судить по количеству оборотов фазовых портретов, примерно соответствует t = 25. Как раз в этот момент амплитуда разницы колебаний начала убывать.

При увеличении разницы частот до 10% система явно перешла в режим биений.

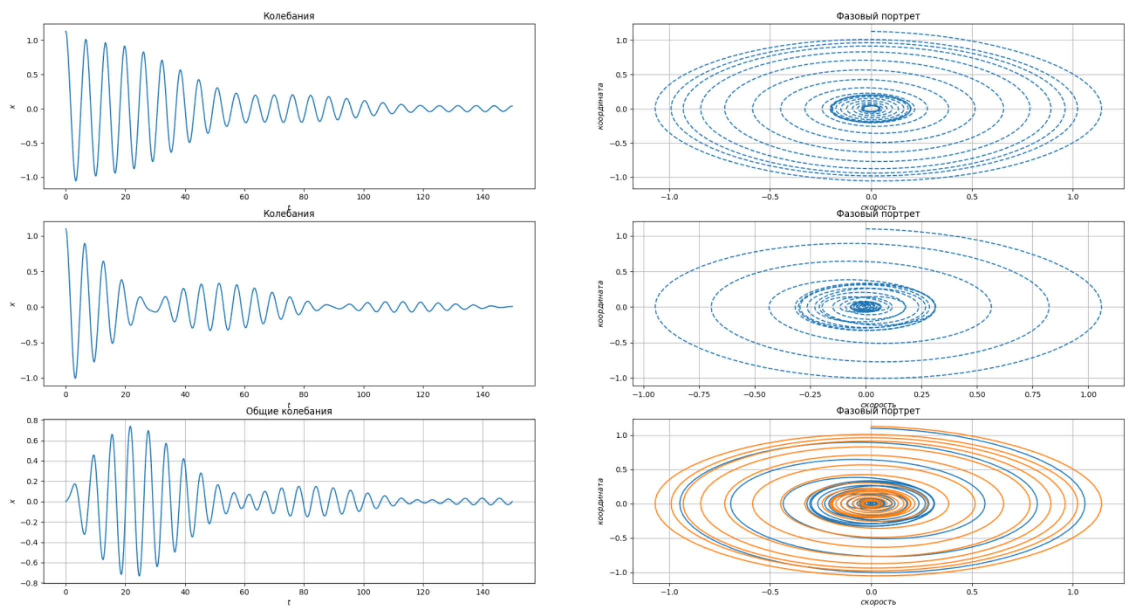

Рисунок 28: MSincself при $\beta_2 = 1,1$ и $d = 0,1$

Однако, аналогично предыдущей модели, ее можно вернуть в режим синхронизма увеличением коэффициента связи.

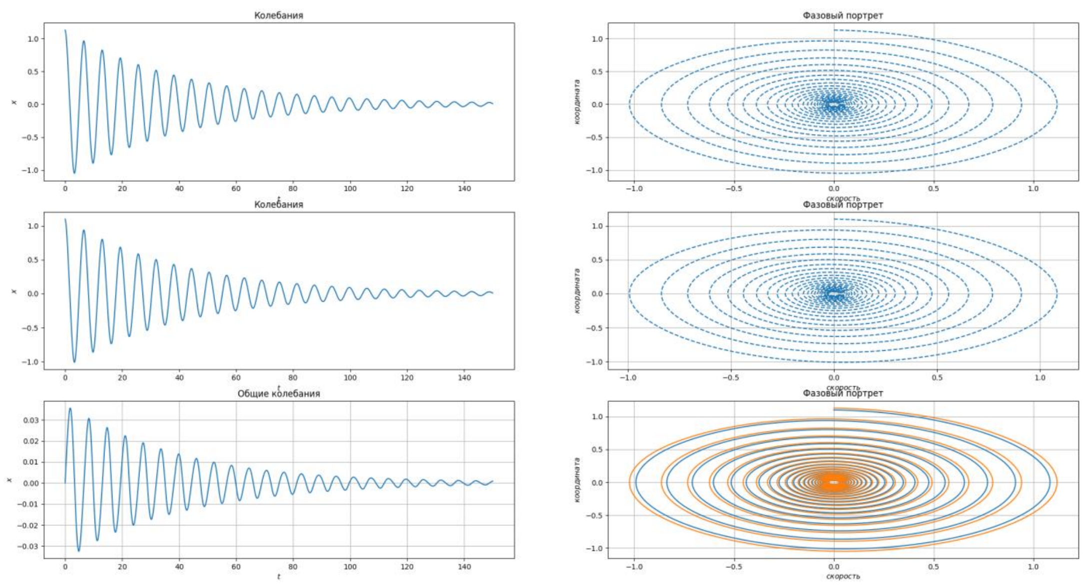

Рисунок 29: MSincself при $\beta_2 = 1,1$ и $d = 1,5$

Масштаб графика разницы колебаний по-прежнему пренебрежимо мало. Видно, что при таком большом коэффициенте обратной связи система практически сразу перешла в режим синхронизации.

Таким образом, при затухании системы из связанных осцилляторов наблюдается некоторое перераспределение энергии, в результате которого синхронизированные осцилляторы не только колеблются, но и затухают в едином темпе. Программа Msincself так же могла бы воссоздать захват частоты одного маятника другим:

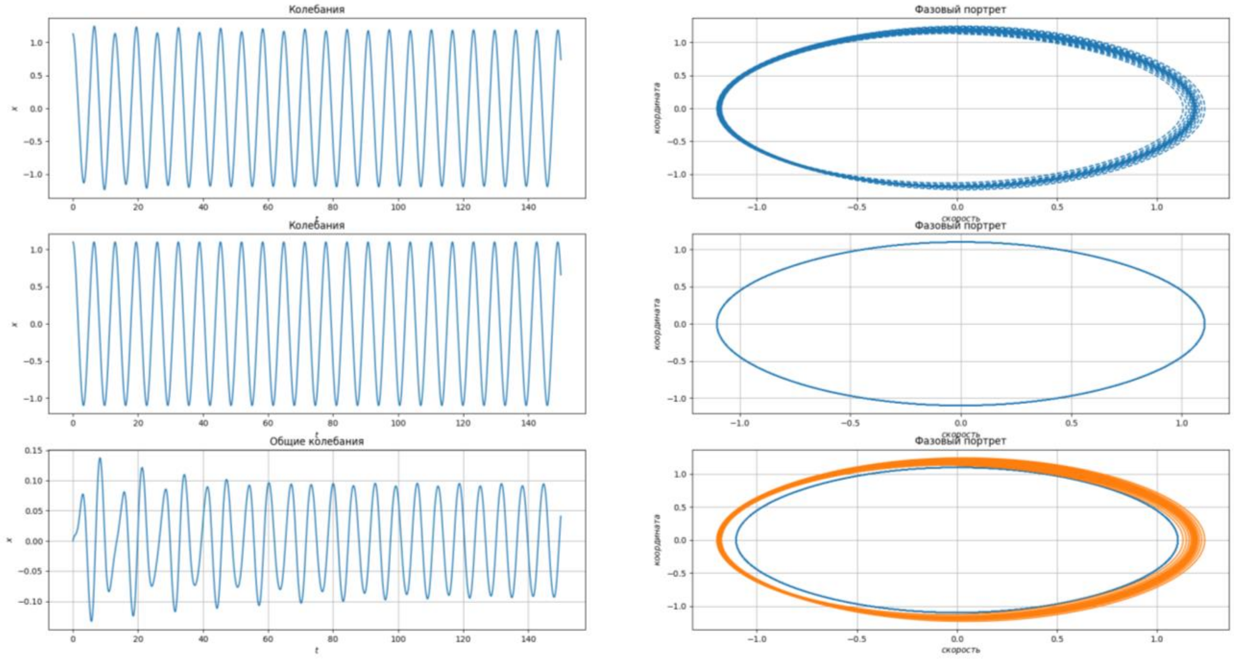

Рисунок 30: MSincself при $\beta_2 = 1{,}1$ и $d = 1{,}5$

Для этого коэффициенты связи и затухания во втором уравнении положим равными нулю. Таким образом, первый осциллятор не оказывает никакого влияния на второй, который, в свою очередь, не затухает. По фазовому портрету первого осциллятора видно, что он по прежнему медленно затухает. Вероятно, это вызвано линейностью обратной связи и наличием изначально небольшой разности фаз.

Заключение

В результате выполнения данной работы были написаны три программы – VDPSincsig, VDPsincself и Msincself. Эти программы моделируют различные процессы синхронизации – внешним сигналом, взаимной синхронизации и синхронизации затухающих колебаний.

Анализ результатов, полученных при помощи данных программ позволил подтвердить описанные в теоретической части работы свойства синхронизации. В частности, на примере данных программ видно, что связанные осцилляторы действительно могут находиться как в режиме синхронизма, так и в режиме биений. То, к какому виду колебаний придут осциллирующие системы, зависит от соотношения частот их колебаний а так же наличия и силы обратных связей между ними. Было продемонстрировано, что полоса синхронизации, то есть та область частот, при которых системы будут приходить к режиму синхронизации вместо биений, зависит от силы обратных связей. Чем эти связи, соединяющие между собой осцилляторы, сильнее, тем полоса синхронизации шире.

Так же было установлено, что принципиальной разницы между видами синхронизации – захват частоты и взаимная синхронизация – нет. Один режим синхронизации может переходить в другой, их различает лишь соотношения коэффициентов обратных связей.

Полученные программы позволяют моделировать процессы синхронизации. Однако их можно улучшить. В частности, полезной для практических целей была бы возможность рассматривать взаимную синхронизацию нескольких колебательных систем. Так же, для ее частого использования пригодился бы пользовательский интерфейс, ведь сейчас программы получают данные исключительно через командную строку. Это вынуждает пользователя часто вводить повторяющиеся значения коэффициентов, когда он хотел бы лишь немного изменить параметры систем.

Список литературы

1. Тимошенко С. П., Янг Д. Х., Уивер У. Колебания в инженерном деле/Пер. с англ. Л. Г. Корнейчука; Под ред. Э. И. Григолюка. – М.: Машиностроение 1985. – 472 с.
2. Горелик Г. С. Колебания и волны. Введение в акустику, радиофизику и оптику. — 3-е изд.: под ред. С.М. Рытова. — М.: ФИЗМАТЛИТ, 2007. — 656 с.
3. Трубецков Д. И., Рожнёв А. Г. Линейные колебания и волны. М.: Физматлит, 2001. 416 с.
4. Ланда П. С. Автоколебания в системах с конечным числом степеней свободы. М.: Наука. Физматлит, 1980. 360 с.
5. Ю. П. Емельянова, А. П. Кузнецов, Синхронизация связанных автогенераторов Ван-дер-Поля и Кислова-Дмитрова, Журнал технической физики, 2011, том 81, вып. 4
6. Каганов В. И. Колебания и волны в природе и технике. Компьютеризированный курс: Учебное пособие для вузов. – М.: Горячая линия-Телеком, 2008 – 336 с.:
7. Кузнецов А. П., Кузнецов С. П., Рыскин Н. М. Нелинейные колебания: Учеб. Пособие для вузов. – М.: Издательство физико-математической литературы, 2002. – 292 с. (серия Современная теория колебаний и волн).
8. Конторович М. И. Нелинейные колебания в радиотехнике. М., Советское радио, 1973, 320 с.
9. Блехман И. И. Синхронизация в природе и технике. – М.: Наука. Главная редакция физико-математической литературы, 1981. - 352 с
10. Нелинейные механические колебания [Электронный ресурс]: электрон. курс лекций / М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (нац. исслед. ун-т); авт.-сост: Л. М. Савельев, С.А. Чернякин. - Электрон. текстовые и граф. дан. - Самара, 2013. – 1 эл. опт. диск (CD-ROM)

11. Пиковский А., Розенбаум М., Куртс Ю. Синхронизация: Фундаментальное нелинейное явление. Москва: Техносфреа, 2003, 494 с.
12. Динамика системы двух нелинейно связанных маятников, С. О. Хрисанфова, Е. Ю Кадина, Е. В. Губина, Л. В. Коган, Г. В. Осипов, Институт информационных технологий, математики и механики, нижегородский государственный университет им. Н. И. Лобачевского, 2016.

Приложение

Код программы VDPsincsig

```
from scipy.integrate import odeint

import numpy as np

from matplotlib import pyplot as plt


print('d2x/dt2-mu(1+g0x-g1x2-g2x4)dx/dt+x=Asin(wt)')


mu=float(input('Введите значение коэффициента нелинейности mu '))

A=float(input('Введите значение амплитуды внешнего сигнала A '))

g0=float(input('Введите значение первого коэффициента нелинейности автогенератора g0 '))

g1=float(input('Введите значение второго коэффициента нелинейности автогенератора g1 '))

g2=float(input('Введите значение третьего коэффициента нелинейности автогенератора g2 '))

w=float(input('Введите значение частоты внешнего сигнала w '))


def f(y, t):

   v=y[0]

   x=y[1]

   f0 = mu*v*(1+g0*x-g1*pow(x,2)-g2*pow(x,4))-x+A*np.sin(w*t)

   f1 = v
```

```
    return [f0, f1]

t = np.linspace(0, 250, 5000)

x0=2.5
v0=0.0
y0 = [x0, v0]

sol = odeint(f, y0, t)

fig = plt.figure()

ax1 = fig.add_subplot(121)
ax1.set_xlabel(r'$t$')
ax1.set_ylabel(r'$x$')
ax1.set_title( 'Колебания')
ax1.plot(t, sol[:,0], t, A*np.sin(w*t))

ax2 = fig.add_subplot(122)
plt.grid(True)
ax2.set_xlabel(r'$скорость$')
```

```
ax2.set_ylabel(r'$координата$')

ax2.set_title( 'Фазовый портрет')

ax2.plot(sol[:,1], sol[:,0], '--',)


plt.show()
```

Код программы VDPSincself

```
from scipy.integrate import odeint

import numpy as np

from matplotlib import pyplot as plt


print('d2(x1)/d(t)2-mu(1+g0x1-g1x1^2-g2x1^4)dx/dt+x1+k1x2')

print('d2(x2)/d(t)2-mu(1+g0x1-g1x1^2-g2x1^4)dx/dt+wx2+k2x1')


mu1=float(input('введите значение коэффициента нелинейности mu перового генератора'))

mu2=float(input('введите значение коэффициента нелинейности mu второго генератора'))

g01=float(input('введите значение первого коэффициента нелинейности g первого генератора'))

g11=float(input('введите значение второго коэффициента нелинейности g первого генератора'))

g21=float(input('введите значение третьего коэффициента нелинейности g первого генератора'))

g02=float(input('введите значение первого коэффициента нелинейности g второго генератора'))

g12=float(input('введите значение второго коэффициента нелинейности g второго генератора'))

g22=float(input('введите значение третьего коэффициента нелинейности g второго генератора'))

w=float(input('введите значение'))

k1=float(input('введите значение'))

k2=float(input('введите значение'))
```

```
def f(y, t):
    v1=y[0]
    x1=y[1]
    v2=y[2]
    x2=y[3]
    f0 = mu1*v1*(1+g01*x1-g11*pow(x1,2)-g21*pow(x1,4))-x1-k1*x2
    f1 = v1
    f2 = mu2*v2*(1+g02*x2-g12*pow(x2,2)-g22*pow(x2,4))-w*x2-k2*x1
    f3 = v2
    return [f0, f1, f2, f3]

t = np.linspace(0, 250, 5000)


x10=1.128
v10=0.0
x20=1.128
v20=0.0
y0 = (x10, v10, x20, v20)

sol = odeint(f, y0, t)

fig = plt.figure()

ax1 = fig.add_subplot(321)
ax1.set_xlabel(r'$t$')
```

```
ax1.set_ylabel(r'$x$')
ax1.set_title( 'Колебания')
ax1.plot(t, sol[:,0])

ax2 = fig.add_subplot(322)
plt.grid(True)
ax2.set_xlabel(r'$скорость$')
ax2.set_ylabel(r'$координата$')
ax2.set_title( 'Фазовый портрет')
ax2.plot(sol[:,1], sol[:,0], '--',)

ax3 = fig.add_subplot(323)
ax3.set_xlabel(r'$t$')
ax3.set_ylabel(r'$x$')
ax3.set_title( 'Колебания')
ax3.plot(t, sol[:,2])

ax4 = fig.add_subplot(324)
plt.grid(True)
ax4.set_xlabel(r'$скорость$')
ax4.set_ylabel(r'$координата$')
ax4.set_title( 'Фазовый портрет')
ax4.plot(sol[:,3], sol[:,2], '--',)

Sol0 = sol[:,1]-sol[:,3]
```

```
ax5 = fig.add_subplot(313)
ax5.set_xlabel(r'$t$')
ax5.set_ylabel(r'$x$')
ax5.set_title( 'Общие колебания')
plt.grid(True)
ax5.plot(t, Sol0)

plt.show()
```

Код программы Msincself

```
from scipy.integrate import odeint

import numpy as np

from matplotlib import pyplot as plt


print('d2(x1)/d(t)2+mu1*(d(x1)/d(t)+a1sin(x1)=d1*sin(x2-x1')

print('d2(x2)/d(t)2+mu2*(d(x2)/d(t)+a2sin(x2)=d2*sin(x1-x2')


a1=1

a2=float(input('введите значение относительной частоты второго маятника'))

mu1=float(input('введите значение коэффициента затухания первого маятника'))

mu2=float(input('введите значение коэффициента затухания второго маятника'))

d1=float(input('введите значение коэффициента связи первого маятника со вторым'))

d2=float(input('введите значение коэффициента связи второго маятника со первым'))


def f(y, t):

    v1=y[0]

    x1=y[1]

    v2=y[2]

    x2=y[3]

    f0 = d1*np.sin(x2-x1)-a1*np.sin(x1)-mu1*v1

    f1 = v1

    f2 = d2*np.sin(x1-x2)-a2*np.sin(x2)-mu2*v2

    f3 = v2
```

```
    return [f0, f1, f2, f3]

t = np.linspace(0, 150, 5000)

x10=1.128
v10=0.0
x20=1.1
v20=0.0
y0 = (x10, v10, x20, v20)

sol = odeint(f, y0, t)

fig = plt.figure()

ax1 = fig.add_subplot(321)
ax1.set_xlabel(r'$t$')
ax1.set_ylabel(r'$x$')
ax1.set_title( 'Колебания')
ax1.plot(t, sol[:,0])

ax2 = fig.add_subplot(322)
plt.grid(True)
ax2.set_xlabel(r'$скорость$')
ax2.set_ylabel(r'$координата$')
ax2.set_title( 'Фазовый портрет')
```

```
ax2.plot(sol[:,1], sol[:,0], '--',)

ax3 = fig.add_subplot(323)
ax3.set_xlabel(r'$t$')
ax3.set_ylabel(r'$x$')
ax3.set_title( 'Колебания')
ax3.plot(t, sol[:,2])

ax4 = fig.add_subplot(324)
plt.grid(True)
ax4.set_xlabel(r'$скорость$')
ax4.set_ylabel(r'$координата$')
ax4.set_title( 'Фазовый портрет')
ax4.plot(sol[:,3], sol[:,2], '--',)

Sol0 = sol[:,1]-sol[:,3]

ax5 = fig.add_subplot(325)
ax5.set_xlabel(r'$t$')
ax5.set_ylabel(r'$x$')
ax5.set_title( 'Общие колебания')
plt.grid(True)
ax5.plot(t, Sol0)

ax4 = fig.add_subplot(326)
```

```
plt.grid(True)
ax4.set_xlabel(r'$скорость$')
ax4.set_ylabel(r'$координата$')
ax4.set_title( 'Фазовый портрет')
ax4.plot(sol[:,3], sol[:,2], "-", sol[:,1], sol[:,0], "-")

plt.show()
```