Лабораторная работа №8.

Моделирование в среде *Simulink*.

Цель работы
1 Получить практические навыки разработки моделей в среде Simulink.

Теоретическая часть

Описание Simulink.

Simulink — интерактивный инструмент для моделирования, имитации и анализа динамических систем. Он дает возможность строить графические блок-диаграммы, имитировать динамические системы, исследовать работоспособность систем и совершенствовать проекты. *Simulink* полностью интегрирован с *MatLab*, обеспечивая немедленным доступом к широкому спектру инструментов анализа и проектирования. *Simulink* также интегрируется с *Stateflow* для моделирования поведения, вызванного событиями.

Simulink содержит в себе:

- Обширную библиотеку блоков для построения графических моделей систем.
- Графический отладчик *Simulink* для интерактивной локализации и диагноза ошибок в модели.
- Браузер моделей (*Model Browser*) для иерархической навигации.
- Диалоговое окно поиска для поиска моделей и библиотек.
- Настраиваемые блоки, которые могут инкорпорировать имеющиеся пользовательские *C, Ada, MatLab,* и *Fortran* коды.

При разработке больших моделей, можно использовать:

- Линейные, нелинейные, непрерывные и дискретные во времени, с переменной скоростью передачи сигналов, с заданными условиями выполнения и гибридные системы.
- Модели могут быть сгруппированы иерархически для эффективного просмотра компонент или подсистем.
- Объекты данных *Simulink* позволяют создавать специфические для приложений типы данных *MatLab* для *Simulink* моделей.
- Специальный графический пользовательский интерфейс *Simulink* для наблюдения и редактирования объектов данных *Simulink*.
- Браузер библиотек для удобного выбора блоков.
- Имитации могут быть запущены из командной линии *MatLab*, как интерактивно, так и в пакетном режиме.

Так же в *Simulink* присутствует:

- Поддержка матричных сигналов и операций.
- Поразрядный логический оператор логически маскирует, обращает, или расщепляет на биты целочисленный сигнал.

Simulink обеспечивает полный набор инструментов моделирования, которые могут использоваться для быстрого проектирования детальных блок-диаграмм систем. Такие инструменты, как объекты данных *Simulink* , библиотеки блоков, интерактивное моделирование, обозначения сигналов и настройка подсистем обеспечивают мощеный набор возможностей для создания, моделирования и управления блок-диаграммами. Эти инструменты моделирования, вместе с исчерпывающим набором поставляемых с *Simulink's* блоков, упрощает создание выразительных макетов систем, независимо от их сложности.

Simulink поступает к пользователям с более 100 встроенных блоков, в состав которых входят наиболее необходимые функции моделирования. Блоки сгруппированы в библиотеки в соответствии с их назначением: источники сигнала, приемники, дискретные, непрерывные, нелинейные, математика, функции и таблицы, сигналы и системы.

В дополнение к обширному набору встроенных блоков, *Simulink* имеет функции создания пользовательских блоков и библиотек. Можно настраивать не только функциональность блоков, но также пользовательский интерфейс, используя иконки и диалоги.

Например, можно создать блоки для моделирования поведения специальных механических, электрических и программных компонент. Однажды созданные, пользовательские блоки могут быть сохранены в библиотеке блоков для будущего использования.

После построения блочной диаграммы в программе *Simulink*, можно отладить ее, используя интерактивный *Simulink* отладчик. Затем можно включить интерактивную имитацию и наблюдать результат. Мощный набор алгоритмов программы *Simulink* позволяет получать чрезвычайно точную имитацию систем.

Процессор имитации *Simulink* предлагает множество функций для имитации больших, сложных систем. Лучшие среди них — набор алгоритмов интегрирования (называемых решателями) обыкновенных дифференциальных уравнений. Эти решатели хорошо подходят для непрерывных во времени (аналоговых), дискретных во времени, гибридных и многосигнальных имитаций любого размера.

С помощью *Simulink* можно создавать и исследовать модели с подсистемами, поведение которых определяется наложенными условиями, задаваемыми контрольными логическими сигналами. Сигналы могут включать или переключать выполнение подсистем.

Simulink включает много функций для детального анализа систем. Ключевые возможности включают: линеаризацию, определение точки равновесия, анимацию, оптимизацию параметров и параметрический анализ.

Благодаря тесной интеграции с *MatLab*, *Simulink* имеет непосредственный доступ к широкому диапазону средств проектирования и анализа. Традиционный подход к проектированию систем обычно заключается в создании прототипа, за которым следует всестороннее тестирование и внесение соответствующих изменений. Этот подход требует больших временных и финансовых затрат. Эффективной альтернативой является имитационное моделирование. *Simulink* — мощный инструмент для моделирования, обеспечивающий быстрое построение и тестирование виртуальных прототипов, и дающий доступ к любому уровню детализации проекта с минимальными усилиями. Используя *Simulink* для итеративного исправления проекта до построения прототипа, проект будет разработан быстрее и эффективнее.

Пакет моделирования динамических систем *Simulink* запускается кнопкой с одноименным названием в верхней панели инструментов *MatLab*. Также *Simulink* можно запустить в командном окне *MatLab*, набив команду >>*Simulink*.

Тем самым открывается библиотека *Simulink Library Browser*.

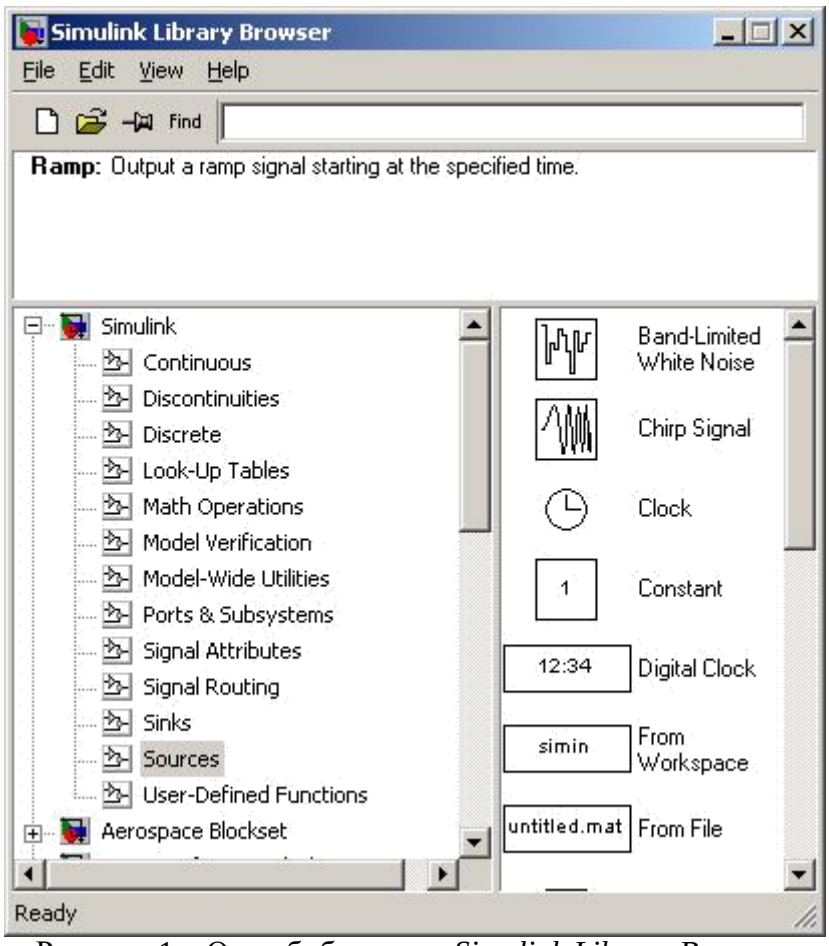


Рисунок 1 – Окно библиотеки *Simulink Library Browser*.

В ней находятся 20 наборов пакетов инструментов (*Toolbox*). Прежде всего, нас будет интересовать пакет инструментов *Simulink*. В нем несколько разделов. Рассмотрим основные.

1 *Continuous* — раздел линейных непрерывных звеньев. *Integrator* — интегрирующее звено; *Transfer Fcn* — звено произвольного порядка; *Zero-Pole* — звено, описываемое нулями и полюсами, *Transport Delay* — задержка по времени.

2 *Math Operations* — математические операции. *Abs* — взятие модуля; *Gain* — множитель, усилительное звено; *Sum* — сумматор.

3 *Sources* — источники сигналов. *Random number* — случайный сигнал; *Sine Wave* — синусоидальный сигнал; *Step* — скачок.

4 *Sinks* — регистрирующие устройства. *Scope* — осциллограф; *XY Graph* — графопостоpитель.

В *Simulink Library Browser* необходимо создать новую форму для сборки нужной модели. Для этого нажимается меню *File–>New–>Model*.

Меню новой модели будет содержать следующие пункты:

File — работа с файлами моделей и библиотек (их создание, сохранение, считывание и печать);

Edit — операции редактирования, работа с буфером обмена и создание подсистем;

View — управление отображением панели инструментов и строки состояния;

Simulation — управление процессом моделирования (старт, пауза, вывод окна настройки параметров моделирования);

Format — операции форматирования модели (смена шрифтов, редактирование надписей, повороты блоков, использование тени от блоков, операции с цветами линий блоков, их фоном и общим фоном);

Tools — управление видом анализа (в линейной области и в режиме реального времени RTW).

Любая динамическая система описывается системой дифференциальных уравнений. Решив такую систему, можно перейти к передаточной функции динамической системы. Передаточная функция представляет собой отношение изображений по Лапласу выходного параметра к входному при нулевых начальных условиях: $W(s) = \frac{Y(s)}{X(s)}$, где s — оператор Лапласа. Рассмотрим простое дифференциальное уравнение:

$$a_1 \dot{y}(t) + a_2 y(t) = b_1 x(t) .$$

Применим к нему преобразование Лапласа, то есть оператор дифференцирования заменится умножением на s. Получим:

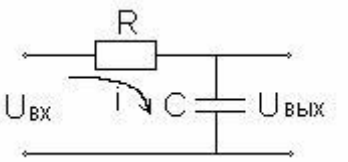
$$sa_1Y(s) + a_2Y(s) = b_1X(s).$$

После упрощения получим передаточную функцию $W(s) = \frac{Y(s)}{X(s)} = \frac{b_1}{a_1 s + a_2}$.

Simulink позволяет численно решать составленные системы дифференциальных уравнений.

Создание модели

Рассмотрим RC-цепочку и построим ее модель.



$$u_{вх}(t) = i(t)R + u_{вых}(t) ,$$

$$i(t) = C\frac{du_C(t)}{dt} = C\frac{du_{вых}(t)}{dt} ,$$

$$u_{вх}(t) = RC\frac{du_{вых}(t)}{dt} + u_{вых}(t) .$$

После применения преобразования Лапласа получим:

$$U_{вх}(s) = RCsU_{вых}(s) + U_{вых}(s) ,$$

$$W(s) = \frac{Y(s)}{X(s)} = \frac{1}{RCs + 1} .$$

Для создания модели в среде *Simulink* необходимо последовательно выполнить ряд действий:

1 Создать новый файл модели с помощью команды *File–>New–>Model*, или используя кнопку на панели инструментов (здесь и далее с помощью символа “/” указаны пункты меню программы, которые необходимо последовательно выбрать для выполнения указанного действия). Вновь созданное окно модели показано на рисунке 2.

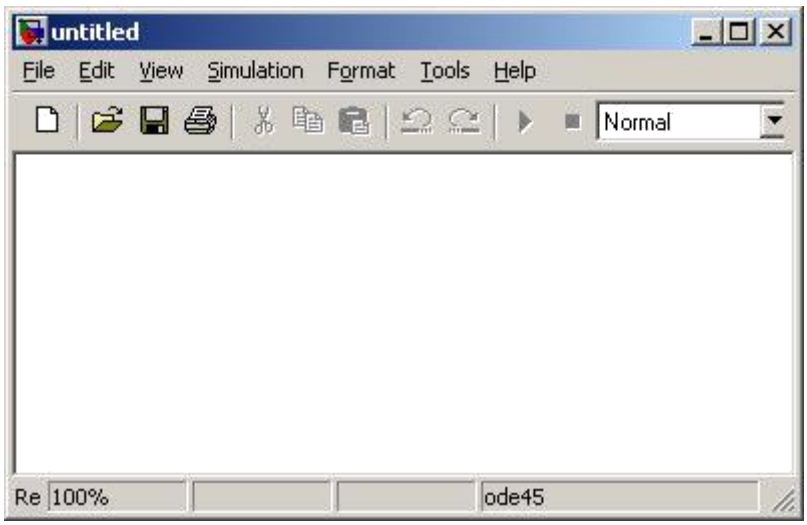


Рисунок 2 — Пустое окно модели.

2 Расположить блоки в окне модели . Для этого необходимо открыть соответствующий раздел библиотеки (например, *Sources* - источники). Далее, указав курсором на требуемый блок и нажав на левую клавишу “мыши”, — “перетащить” блок в созданное окно. Клавишу мыши нужно держать нажатой. На рисунке 3 показано окно модели, содержащее блоки.

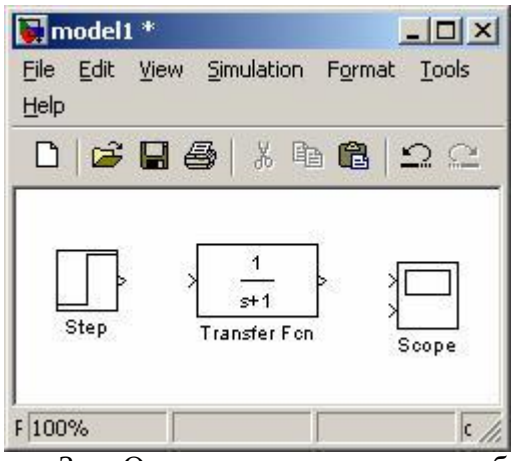


Рисунок 3 — Окно модели, содержащее блоки.

Для удаления блока необходимо выбрать блок (указать курсором на его изображение и нажать левую клавишу “мыши”), а затем нажать клавишу *Delete* на клавиатуре. Для изменения размеров блока требуется выбрать блок, установить курсор в один из углов блока и, нажав левую клавишу “мыши”, изменить размер блока (курсор при этом превратится в двухстороннюю стрелку). Если блок нужно повернуть (скажем, он стоит в цепи обратной связи), то необходимо выбрать этот блок и нажать *Rotate block* в меню *Format*.

3 Далее, если это требуется, нужно изменить параметры блока, установленные программой “по умолчанию”. Для этого необходимо дважды щелкнуть левой клавишей “мыши”, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой *OK*. На рисунке 4 в качестве примера показаны блок, моделирующий передаточную функцию, и окно редактирования параметров данного блока.

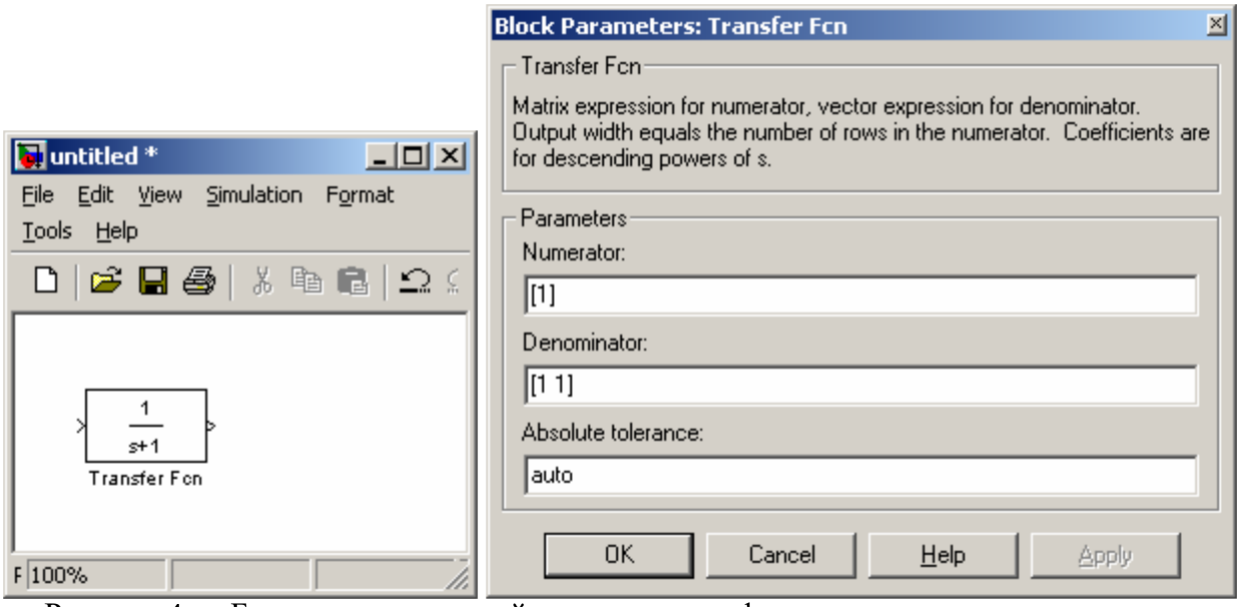


Рисунок 4 — Блок, моделирующий передаточную функцию, и окно редактирования параметров блока.

4 После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на “выход” блока, а затем нажать и, не отпуская левую клавишу “мыши”, провести линию ко входу другого блока. После чего отпустить клавишу. В случае правильного соединения изображение стрелки на входе блока изменяет цвет. Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав *правую* клавишу “мыши”, протянуть линию. Для удаления линии требуется выбрать линию (так же, как это выполняется для блока), а затем нажать клавишу *Delete* на клавиатуре. Схема модели, в которой выполнены соединения между блоками, показана на рисунке 5.

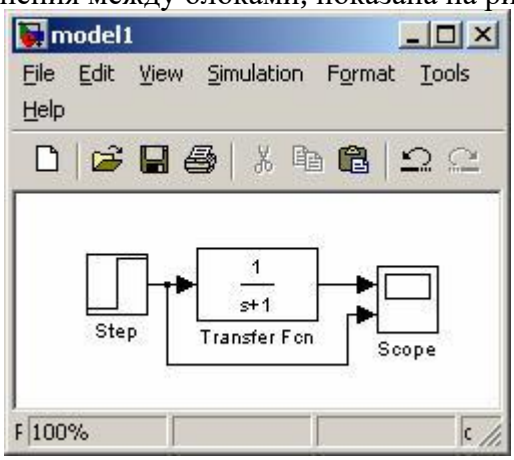


Рисунок 5 — Схема модели.

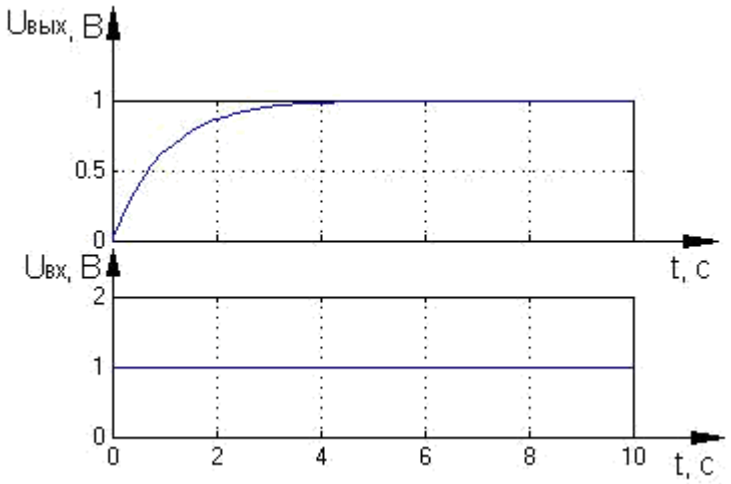


Рисунок 6 — Осциллограмма модели.

5 После составления расчетной схемы необходимо сохранить ее в виде файла на диске, выбрав пункт меню *File–>Save As* в окне схемы и указав папку и имя файла. Следует иметь в виду, что имя файла не должно превышать 32 символов, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. Это же требование относится и к пути файла (к тем папкам, в которых сохраняется файл). При последующем редактировании схемы можно пользоваться пунктом меню *Fille–>Save*. При повторных запусках программы *Simulink* загрузка схемы осуществляется с помощью меню *File–>Open* в окне обозревателя библиотеки или из основного окна *MatLab***.**

Рассмотрим параметры некоторых элементов. Во-первых, звено *Transfer Fcn*, его параметрами являются коэффициенты числителя (*numerator*) и знаменателя (*denominator*). При этом порядок числителя должен быть меньше порядка знаменателя, порядок знаменателя не ограничен. Во-вторых, сумматор, его параметрами являются знак суммирования и форма представления сумматора. В-третьих, осциллограф . При двойном щелчке на осциллографе открывается его осциллограмма. В панели инструментов необходимо выбрать пиктограмму с надписью *Parameters*. Надпись появляется, если навести курсор мыши на данную кнопку. При нажатии этой кнопки появляется следующее окно.

Рисунок 7 — Окно параметров осциллографа.

Number of axes — число осей (каналов) осциллографа;

Time range — пределы временного интервала;

Tick labels — вывод/скрытие отметок по осям;

Sampling — установка временных соотношений (*Decimation* — кратность вывода данных , по умолчанию 1, или *Simple Time* — в тактах эталонного времени, по умолчанию 0).

Пример: модель аттрактора Лоренца.

Аттрактор Лоренца является типичной замкнутой колебательной системой с положительной обратной связью, в которой развиваются колебания очень сложной формы, демонстрирующие явление хаоса. Модель содержит ряд вычислительных модулей, отражающих ее нелинейность и основные особенности структуры, а также два регистрирующих прибора — виртуальные осциллограф и графопостроитель.

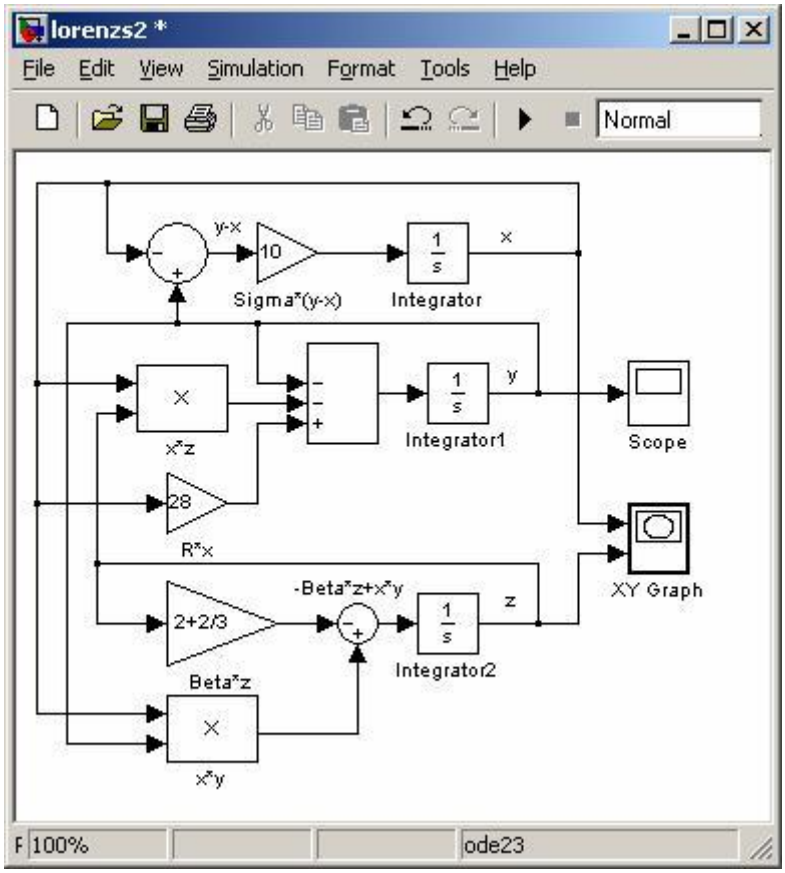


Рисунок 8 — Модель аттрактора Лоренца.

Изменить параметры моделирования (время моделирования, шаг и другие) можно в закладке *Simulation parameters* меню *Simulation*.

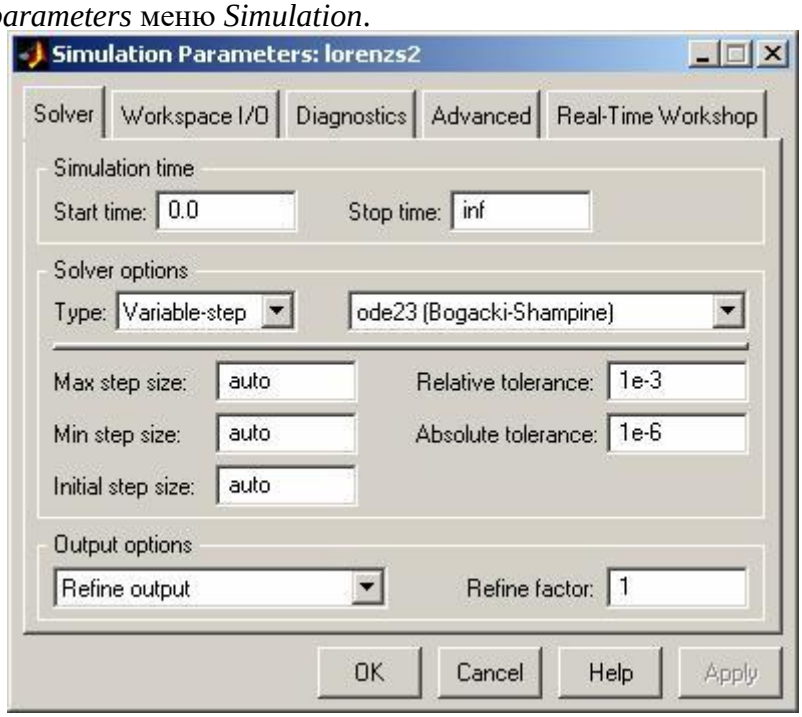


Рисунок 9 — Окно параметров моделирования (*Simulation parameters*).

Запуск модели осуществляется нажатием на *Start* в меню *Simulation*.

Результаты моделирования можно посмотреть несколькими способами. Первым является просмотр осциллограммы, для этого необходимо выполнить двойной щелчок на осциллографе. Вторым способом является проведение линейного анализа с помощью функции *Linear analysis* в меню *Tools*. Об этой функции будет рассказано в лабораторной работе №5.

Обратим внимание на некоторые параметры блоков схемы и параметры моделирования.

1) Параметр *initial condition* у интеграторов взят равным 1.

2) Время моделирования не ограничено (в этой строке стоит *inf*). При этом мы будем обязаны прервать моделирование.

3) Способ моделирования ode23 (*Bogacki-Shampine*).

4) Заданы также точностные коэффициенты: относительная погрешность интегрирования Relative tolerance = 10^{-3} и абсолютная погрешность интегрирования *Absolute tolerance* = 10^{-6}.

5) Тип шага интегрирования можно установить или переменным (*variable-step*), как в данном примере, или фиксированным (*fixed-step*).

Также в параметрах моделирования можно изменять следующие значения, которые в данном примере были заданы автоматически:

Max step size — максимальный шаг интегрирования системы однородных дифференциальных уравнений;

Min step size — минимальный шаг интегрирования;

Initial step size — начальный шаг интегрирования.

Если схема получается очень громоздкой и сложной, то отдельные части ее можно свернуть в подсистемы (*Subsystem*). Для этого на выделенном куске схемы надо нажать правую клавишу мыши, затем выбрать *Create subsystem*.

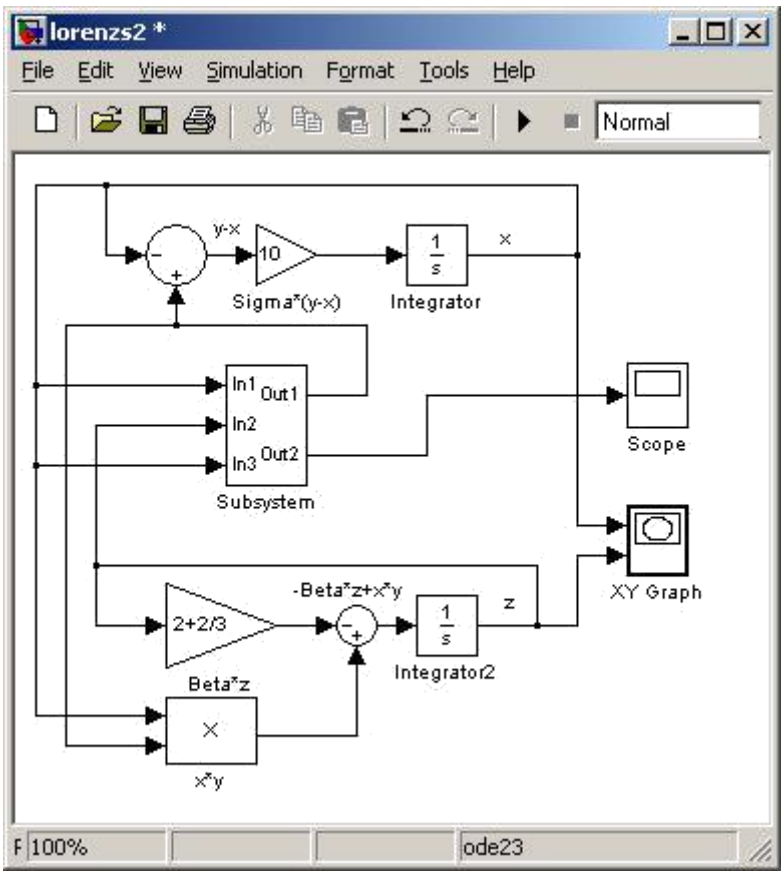


Рисунок 10 — Пример модели с использованием подсистемы.

Чтобы развернуть подсистему, необходимо два раза щелкнуть на ней мышью.

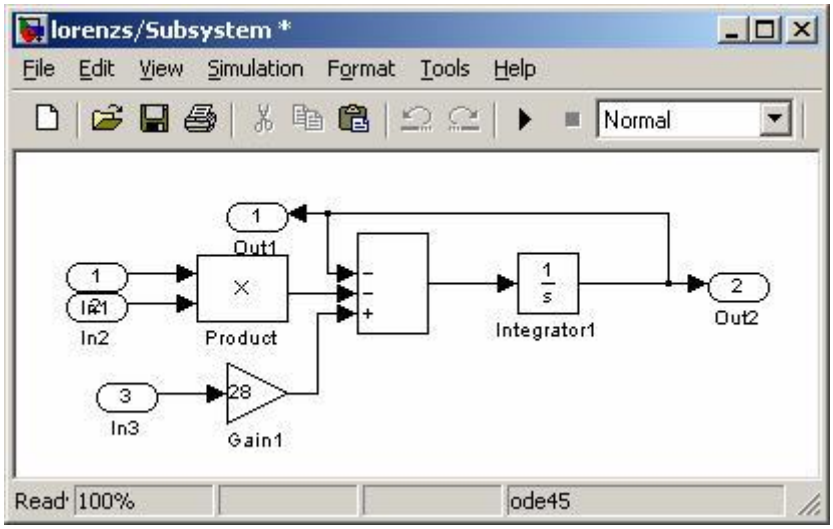


Рисунок 11 — Подсистема.

После проведения моделирования получена следующая картина на графопостроителе:

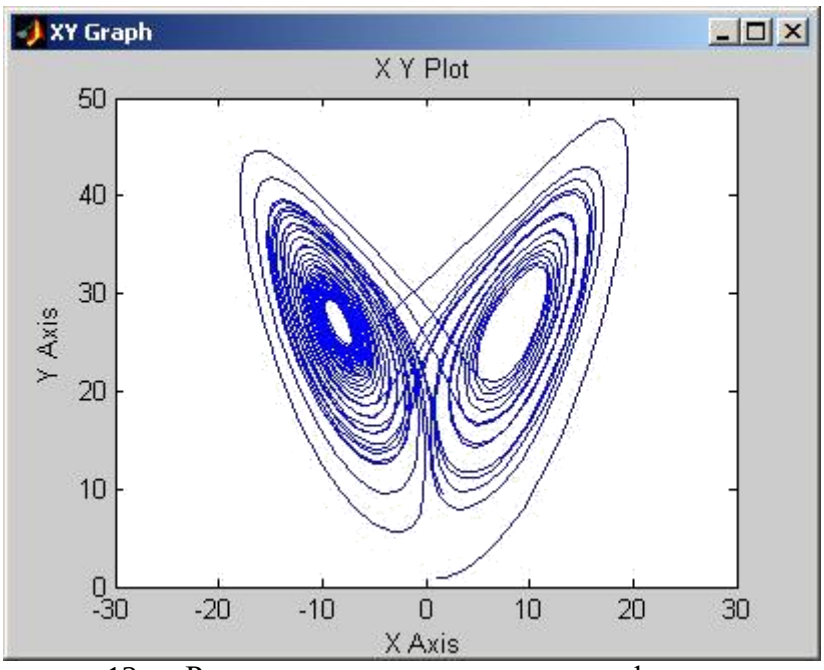


Рисунок 12 — Результат моделирования на графопостроителе.

Осциллограмма выглядит следующим образом:

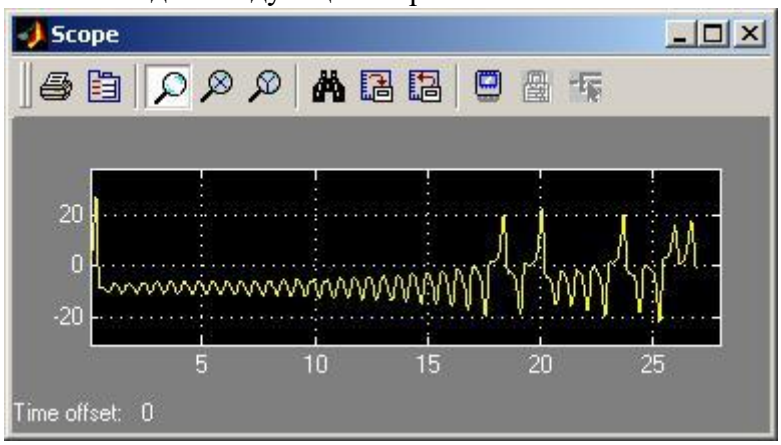


Рисунок 13 — Результат моделирования на осциллографе.

Рассмотрим пример интегрирования.

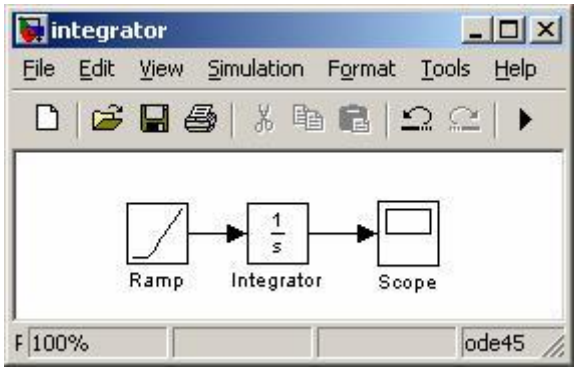


Рисунок 14 — Модель интегратора.

Simulink интегрирован с *MatLab*, поэтому позволяет параметры, полученные в процессе моделирования в *Simulink* переводить в область *MatLab*. Для этого необходимо в *Simulink Library Browser* в разделе приемников сигнала выбрать элемент *ToWorkspace* и соединить с ним нужный выходной сигнал.

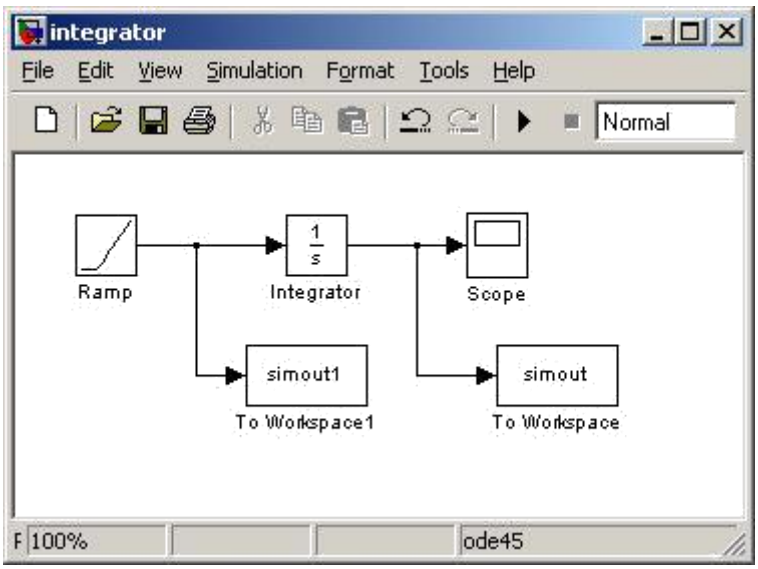


Рисунок 15 — Использования элемента *ToWorkspace*.

Для того чтобы убедиться в передаче данных в *MatLab*, необходимо совершить несколько действий.

Во-первых, надо посмотреть, какие переменные находятся в *MatLab*. Это осуществляется командой *whos*:

>> whos

Результат выполнения данной команды выдается в виде таблицы.

Name	*Size*	*Bytes*	*Class*
simout	*1x1*	*1214*	*struct array*
simout1	*1x1*	*1216*	*struct array*
tout	*52x1*	*416*	*double array*

Как видно из таблицы, в ней автоматически появляется переменная *tout* (время).

Рассмотрим выходную переменную *simout* подробнее. Выведем ее на экран.

>> simout

```
simout =

        time: []
     signals: [1x1 struct]
   blockName: 'integrator/To Workspace'
```

Отсюда видно, что переменная *simout* состоит из еще одной структуры — *signals*. Теперь, чтобы посмотреть, что содержит выходная переменная *simout*, необходимо набить команду, содержащую последовательно путь по структурам переменной, а в конце команду *values*:

>> Uvyh=simout.signals.values

Вектор *Uvyh* будет представлять собой столбец значений. Далее с ним можно проводить любые действия в рамках *MatLab*.

Постоим график зависимости выходного сигнала от входного.

>> plot(tout,Uvyh);grid;

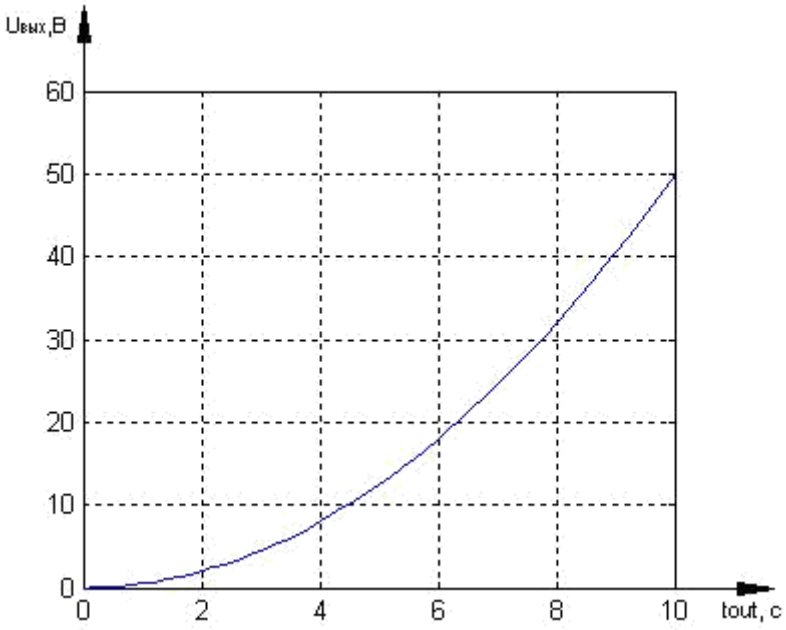


Рисунок 16 — Зависимость выходного сигнала от входного.

В результате, проинтегрировав входной линейный сигнал, получили параболу.

Рассмотрим применение графопостроителя для построения фигуры Лиссажу.

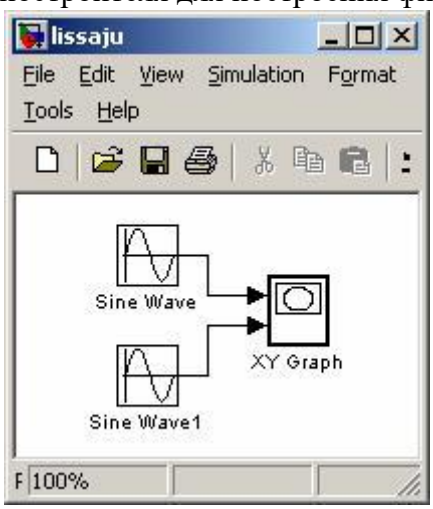


Рисунок 17 — Модель для построения фигуры Лиссажу.

Фигура Лиссажу образуется при подаче на вход графопостроителя двух синусоидальных сигналов с частотами 2 и 3 Гц.

Полученная фигура представлена на рисунке 18.

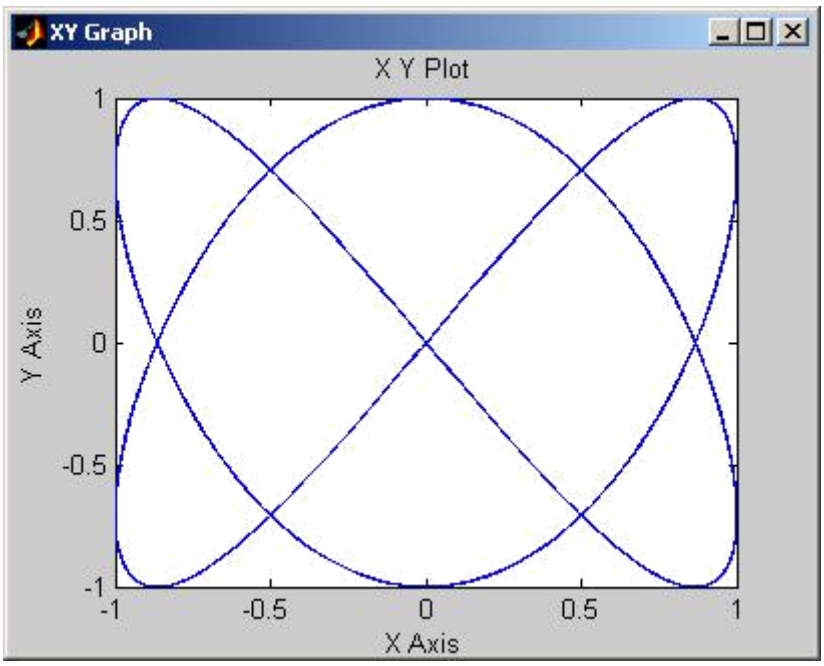


Рисунок 18 — Фигура Лиссажу.

Практическая часть

Задание 1

Исследовать модель аттрактора Лоренца. Изменять в ней параметры (время, шаг моделирования, метод интегрирования), изменять число каналов в осциллографе и подавать на них различные сигналы со схемы, накладывать графики, посмотреть, как меняются результаты моделирования.

Порядок работы.

1. открыть файл «lab3_lorenzs.mdl» в *MatLab*;
2. задать необходимые параметры: время, шаг, метод моделирования;
3. зафиксировать полученные результаты;
4. поменять значения параметров на новые;
5. зафиксировать новые результаты моделирования;
6. сделать выводы.

Задание 2

Собрать модель RLC-цепочки, промоделировать согласно варианту, подобрать время моделирования.

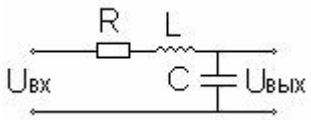


Рисунок 19 — Схема RLC-цепочки.

$$u_{вх}(t) = iR + L\frac{di(t)}{dt} + u_{вых}(t),$$

$$u_{вх}(t) = RC\frac{du_{вых}(t)}{dt} + LC\frac{d^2u_{вых}(t)}{dt^2} + u_{вых}(t),$$

$$U_{вх}(s) = RCsU_{вых}(s) + LCs^2U_{вых}(s) + U_{вых}(s),$$

$$W(s) = \frac{1}{LCs^2 + RCs + 1}.$$

Задание 3

Вычислить функцию в зависимости от варианта.

Помощь: использовать блоки *integrator, derivative, gain, product, sum, sin, constant* и другие по необходимости.

Задание 4.

Построить спираль Карно: $x(t) = \int_0^{t_{max}} a_i \cos(x^2)dx$ и $y(t) = \int_0^{t_{max}} b_i \sin(x^2)dx$. Использовать графопостроитель, блок *Ramp*. Нарастающий сигнал начинается с отрицательного уровня: промоделировать с –10 и с –20. Коэффициенты брать из таблицы вариантов.

Таблица вариантов

Вариант	Номер задачи				
	1	2	3	4	5
1, 13, 25		$R=1$ $L=2$ $C=3$	$F(x)=\int(3x^2-5x-12)dx$	$a=1$ $b=1$	
2, 14, 26		$R=2$ $L=1$ $C=5$	$F(x)=\dfrac{d(8x^3+7x^2-15x+4)}{dx}$	$a=1.5$ $b=1.2$	
3, 15, 27		$R=10$ $L=4$ $C=1$	$F(x)=5+\int(x-3)dx$	$a=1.4$ $b=1.8$	
4, 16, 28		$R=7$ $L=2$ $C=4$	$F(x)=(8x+1)'+\int(3x^2-7)dx$	$a=1.1$ $b=1.6$	
5, 17, 29		$R=8$ $L=6$ $C=2$	$F(x)=\frac{1}{2}\int(3x^2-4(8x-3))'dx$	$a=0.8$ $b=1.2$	
6, 18, 30		$R=3$ $L=9$ $C=7$	$F(x)=\frac{1}{4}\int(6x^3-8x^2+12x-2)dx$	$a=0.5$ $b=0.4$	
7, 19, 31		$R=6$ $L=5$ $C=9$	$F(x)=6(-x^4+23)''$	$a=0.9$ $b=1.5$	
8, 20, 32		$R=4$ $L=0.5$ $C=1.5$	$F(x)=(\sin(x^2+4x+7))'$	$a=1$ $b=0.7$	
9, 21, 33		$R=7.5$ $L=4.8$ $C=2.3$	$F(x)=\int(\sin(3x^2)-5x-12)dx$	$a=1.6$ $b=0.8$	
10, 22, 34		$R=1.5$ $L=10$ $C=10$	$F(x)=\int(tg(x)+\cos(x-2))'dx$	$a=1.4$ $b=0.4$	
11, 23, 35		$R=2$ $L=1.5$ $C=0.5$	$F(x)=9(\sin(x)\cos(6x))'$	$a=0.5$ $b=0.9$	
12, 24, 36		$R=0.5$ $L=5.5$ $C=3.5$	$F(x)=(5x+4)\int(x+\sin(x))dx$	$a=1.1$ $b=0.6$	