

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК УКРАИНЫ
Институт проблем моделирования в энергетике им. Г.Е. Пухова
Отделение гибридных моделирующих и управляющих систем в
энергетике

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Национальный авиационный университет
Кафедра электротехники и светотехники

В.В. Васильев, Л.А. Симак, А.М. Рыбникова

**МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ
ПРОЦЕССОВ И СИСТЕМ В СРЕДЕ MATLAB/SIMULINK.**

Учебное пособие

Киев – 2008

УДК 681.3

Авторы: В.В. Васильев, Л.А. Симак, А.М. Рыбникова

Рецензент: чл.-корр. НАН Украины, д.т.н., проф. С.Г. Таранов

Математическое и компьютерное моделирование процессов и систем в среде MATLAB/SIMULINK. Учебное пособие для студентов и аспирантов / В.В. Васильев, Л.А. Симак, А.М. Рыбникова. – К.: НАН Украины, 2008. – 91 с.

ISBN 978-966-02-4389-7

Учебное пособие посвящено инженерным методам моделирования процессов и систем с использованием методов графического (визуального) программирования в среде системы MATLAB/SIMULINK. Рассмотрены методы аппроксимации сигналов, построения структурных схем и моделей динамических и безинерционных объектов, решения задач математического программирования. Изложение материала сопровождается иллюстративными примерами и упражнениями для самостоятельной работы. Предназначено для студентов и аспирантов, занимающихся вопросами математического и компьютерного моделирования и их применениями в различных областях естествознания и техники.

The textbook deals with engineering methods for modeling and simulation of systems and processes with graphical (visual) programming methods. MATLAB / SIMULINK software environment is used mainly. Signal approximation methods, as well as structures for dynamical and without inertia objects simulation and mathematical programming problems solution have been considered. An illustrative examples and praxis accompany the theoretical results. The work is destined to the students and post-graduate dealing with modeling and simulation methods and their applications in various fields of research (science & technology).

© Отделение гибридных моделирующих и управляющих систем в энергетике Института проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, 2008

© В.В. Васильев, Л.А. Симак, А.М. Рыбникова, 2008

ISBN 978-966-02-4389-7

СОДЕРЖАНИЕ

Введение.....	4
1. Общие сведения о программном комплексе SIMULINK.....	7
1.1. Решающие элементы.....	9
1.2. Построение простейших моделей.....	11
2. Модели алгебраических объектов.....	18
3. Аппроксимация сигналов.....	23
3.1. Метод наименьших квадратов.....	23
3.2. Модифицированный метод равных площадей.....	26
3.3. Блочно-импульсная аппроксимация.....	35
4. Модели динамических объектов.....	41
5. Обратимость моделей и задачи оптимизации.....	47
5.1. Обратимость решающих элементов.....	48
5.2. Виртуальные решающие элементы системы MATLAB/Simulink/SimPowerSystem.....	49
5.3. Необратимый линейный преобразователь в системе Simulink.....	51
5.4. α -аналоговая модель системы линейных алгебраических уравнений.....	52
5.5. Виртуальный аналог операционного усилителя.....	54
5.6. Обратимый линейный преобразователь.....	54
5.7. Обратимая модель системы линейных алгебраических уравнений.....	55
5.8. Обратимая модель системы линейных неравенств.....	57
5.9. Обратимая модель задачи линейного программирования.....	59
5.10. Модель транспортной задачи линейного программирования.....	61
6. Упражнения для самостоятельной работы.....	67
7. Иллюстрации к упражнениям.....	71
8. Литература.....	89

ВВЕДЕНИЕ

Развитие методов и средств компьютерного моделирования всегда происходило в направлении упрощения диалога человека с вычислительной (моделирующей) системой, приближения языка программирования задачи к инженерному. В последнее время широкое распространение получил визуальный или графический подход к программированию. В некоторой степени этот подход напоминает программирование для аналоговых вычислительных машин, которые были широко распространены в 60–х годах прошлого столетия. Аналоговые решающие элементы, выполняющие определенные математические операции над переменными, соединялись между собой в соответствии со структурой уравнения или другого способа определения задачи, вводились сигналы, определяющие внешние воздействия на исследуемую систему и необходимые начальные условия для переменных. После этого регистрировался переходный процесс, происходящий в полученной электронной системе после ее включения, который являлся решением поставленной задачи. В современных программных комплексах моделирования, использующих визуальный (графический) подход к программированию, виртуальная модель решаемой задачи формируется на экране дисплея в виде структурной схемы из виртуальных решаемых элементов, имеющихся в библиотеке программной системы, или создаваемых пользователем, последующего соединения элементов между собой виртуальными проводниками. Виртуальные решающие элементы представляют собой фрагменты программ выполнения соответствующих математических операций, написанные на языке высокого уровня с использованием методов автоматизации программирования. Процедуре создания на экране дисплея структурной схемы решаемой задачи соответствует формирование полной программы решения задачи, которая в зависимости от сложности проблемы может насчитывать десятки и сотни тысяч команд. Пользователь освобождается от необходимости составления и отладки сложной программы и даже может не владеть в совершенстве методами программирования на языках высокого уровня.

Средства графического (визуального) программирования позволяют вводить описание моделируемой системы в естественной для пользователя, преимущественно графической форме, автоматически переводить это описание на язык компьютера и

представлять результаты моделирования опять же в графической форме, например в виде временных или фазовых диаграмм и анимированных картинок. Трудоемкость и время разработки модели и проведения вычислительных экспериментов в таких средах сокращаются в десятки раз по сравнению с традиционным способом, когда для каждой новой разработки создается индивидуальная программа. Относительная дешевизна графических сред визуального моделирования и простота их эксплуатации делают компьютерное моделирование доступным для каждого инженера, технолога и менеджера.

В настоящее время существует несколько десятков графических сред визуального моделирования, среди них можно выделить следующие:

- Приложение SIMULINK, работающее на базе программного комплекса MATLAB фирмы The MathWorks Inc.

Используется для проектирования систем управления, цифровой обработки сигналов, коммуникационных систем.

- Комплекс LabVIEW фирмы National Instruments.

Используется в системах сбора и обработки данных, а также для управления техническими объектами и технологическими процессами.

- Программный комплекс Electronics Workbench (Multisim) фирмы Interactive Image Technologies Ltd.

Используется для моделирования электронных схем и решения задач автоматизации проектирования.

- Программный комплекс FEMLAB (COMSOL MULTIPHYSICS).

Используется для моделирования систем с распределенными параметрами, которые описываются интегро-дифференциальными уравнениями в частных производных, с использованием метода конечных элементов. Может работать совместно с комплексом MATLAB или автономно.

Перечисленные пакеты ориентированы на разные классы задач, но могут быть полезны для читателей данного пособия. Более подробно с возможностями интегрированных комплексов моделирования с использованием методов графического (визуального) программирования можно познакомиться в публикациях [1, 9, 11 -17, 19, 27].

Пособие построено следующим образом. В первом разделе приведены краткие сведения о программном комплексе SIMULINK как приложении к системе MATLAB. Рассмотрены основные решающие элементы, используемые для построения структурных схем (графических моделей) динамических систем. Приведены простейшие иллюстративные примеры построения структурных схем виртуальных генераторов сигналов заданной формы и систем функций. Эти

примеры позволяют продемонстрировать основные правила работы в среде приложения SIMULINK. Второй раздел посвящен построению моделей безинерционных (алгебраических) объектов на примере систем линейных алгебраических уравнений. В третьем разделе рассмотрены ряд методов полиномиальной аппроксимации непрерывных сигналов. В четвертом разделе рассмотрены виртуальные модели простейших динамических систем (физического маятника, динамической системы 3 порядка и модель полета тела в поле тяготения Земли, брошенного под углом к горизонту). Построению обратимых моделей и моделированию задач оптимизации посвящен пятый раздел. Описан виртуальный аналог операционного усилителя и виртуальные обратимые линейные преобразователи. Приведены структурные схемы обратимых и необратимых моделей систем линейных алгебраических уравнений и неравенств, а также задач линейного программирования. Рассмотрены структурные схемы виртуальных элементов для построения моделей транспортных задач линейного программирования. На иллюстративных примерах показана методика построения моделей транспортных задач линейного программирования, использующих аналогию Денниса. В последующих разделах приведены упражнения для самостоятельной работы и пояснения к ним вместе со структурными схемами и иллюстрациями. Список литературы является избыточным и предназначен для более глубокого освоения материала.

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ КОМПЛЕКСЕ SIMULINK

Программа SIMULINK является приложением к пакету MATLAB. В определенном смысле SIMULINK можно рассматривать как самостоятельный продукт фирмы MathWorks, однако он работает только при наличии ядра MATLAB и использует многие функции, входящие в его состав.

Следует отметить, что пакет MATLAB ориентирован в первую очередь на обработку массивов данных (матриц, векторов и т.п.). Это позволяет существенно повысить эффективность процедур, работающих с указанными типами данных, по сравнению с языками программирования «общего назначения» (Pascal, C и т.п.), и существенно отличает MATLAB от других систем, таких, как MAPLE, MathCAD, Mathematica. Векторная обработка данных обеспечивает высокую скорость вычислений, в большинстве случаев избавляет пользователя от написания циклов и гарантирует необходимую точность.

Приложение SIMULINK является инструментом, с помощью которого можно объединять блоки, соответствующие отдельным элементам динамической системы в единое целое и изучать их поведение во времени.

Разработка моделей средствами SIMULINK (S-модели) основана на технологии drag-and-drop («перетаски и оставь»). Для построения S-модели используются модули (или блоки), хранящиеся в библиотеке SIMULINK.

Библиотека SIMULINK хороша тем, что, с одной стороны, обеспечивает пользователю доступ ко всем основным возможностям пакета MATLAB, а с другой – является достаточно самостоятельной его компонентой, в том смысле, что при работе с ней не обязательно иметь навыки в использовании других инструментов, входящих в состав пакета.

Блоки, включаемые в создаваемую модель, могут быть связаны друг с другом как по информации, так и по управлению. Вид связи зависит от типа блока и логики работы модели. Данные, которыми обмениваются блоки, могут быть скалярными величинами, векторами или матрицами произвольной размерности.

Любая S-модель может иметь иерархическую структуру, т.е. состоять из моделей более низкого уровня, причем число уровней иерархии практически не ограничено. Наряду с другими параметрами моделирования пользователь может задавать способ изменения модельного времени (с постоянным или переменным шагом), а также условия окончания моделирования.

В ходе моделирования имеется возможность наблюдать за процессами, происходящими в системе. Для этого используются специальные «смотровые окна», входящие в состав библиотеки SIMULINK. Интересующие пользователя характеристики могут быть представлены как в числовой, так и в графической форме.

Применение принципов структурного и модульного программирования позволяет представлять различные алгоритмы в виде набора унифицированных программных модулей, что улучшает обзорность программы, облегчает ее отладку и в конечном счете уменьшает общий объем программного обеспечения, подлежащего разработке. Кроме этого состав библиотеки SIMULINK может быть пополнен пользователем за счет разработки собственных блоков.

В последующих разделах будут рассмотрены типовые решающие элементы, приведены структурные схемы распространенных систем базисных функций, основные блоки, реализующие методы аппроксимации сигналов и моделирования динамических систем.

Следует отметить, что в составе MATLAB имеется множество других приложений, основанных на методах графического (визуального) программирования, допускающих совместную работу с приложением SIMULINK.

Aerospace Blockset – содержит специальные инструменты для моделирования авиационных, космических, реактивных и турбореактивных систем.

DSP Blockset – предназначен для проектирования систем и моделирования задач цифровой обработки сигналов (DSP). Данные библиотеки включают такие ключевые операции, как классическая, многоступенчатая и адаптивная фильтрация, преобразования, матричные операции и линейная алгебра, статистика и спектральный анализ.

Nonlinear Control Design Blockset – предоставляет в распоряжение пользователя графический интерфейс для настройки параметров динамических объектов.

SimPowerSystems – предназначен для моделирования электротехнических и электроэнергетических устройств и систем.

SimMechanics – позволяет моделировать системы управления с помощью ненаправленных сигнальных графов, объединять их с физическими моделями и моделями из других библиотек.

В настоящее время существует большое количество учебной литературы по приложению SIMULINK [1, 9, 11, 14, 27]. Кроме этого полезную информацию можно найти на сайтах:

<http://matlab.exponenta.ru/simulink/default.php>

<http://www.mathworks.com>

1.1. Решающие элементы

Пакет SIMULINK запускается из программной среды MatLab. Соответствующая пиктограмма расположена на панели инструментов.

Для начала работы необходимо создать новый файл *.mdl (Simulink model), в который можно с помощью мыши перетаскивать блоки из библиотеки SIMULINK. Необходимо отметить, что наборы инструментов в библиотеке выбираются пользователем при установке программы.

В главном меню находится вкладка Simulation/ Configuration Parameters, где можно менять параметры процесса моделирования, такие как время начала и конца моделирования, выбор решателя, шаг с которым происходит расчет и т.д.

Рассмотрим часто встречающиеся блоки SIMULINK, которые будут встречаться далее в пособии.

Для любого блока по двойному щелчку мыши открывается окно параметров Function Block Parameters, где можно менять установки.

Источники сигналов (Sources)

На рис. 1.1 представлены некоторые источники сигналов.

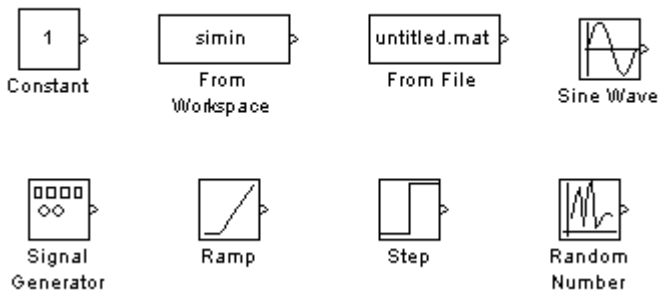


Рис. 1.1. Источники сигналов

Constant задает постоянный по уровню сигнал;

From Workspace – блок считывания данных из рабочей области;

From File – блок считывания данных из файла;

Sine Wave формирует синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением;

Signal Generator формирует один из четырех видов периодических сигналов (синусоидальный, прямоугольный, пилообразный, случайный сигналы);

Ramp формирует линейный сигнал;

Step формирует ступенчатый сигнал;

Random Number формирует случайный сигнал с нормальным распределением уровня сигнала.

Приемники сигналов (Sinks)

На рис. 1.2 представлены блоки-приемники сигналов

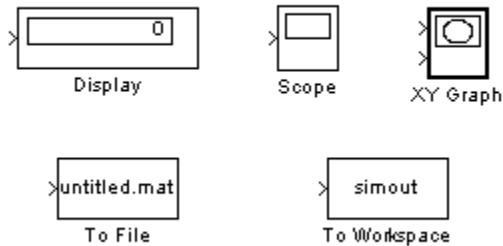


Рис. 1.2. Приемники сигналов

Цифровой дисплей (**Display**) отображает значение сигнала в виде числа;

Виртуальный осциллограф (**Scope**) строит график сигнала от функции времени;

Виртуальный графопостроитель (**XY Graph**) строит график одного сигнала в зависимости от другого (график вида $Y(X)$);

Блок «**To File**» записывает данные, поступающие на его вход, в файл;

Блок «**To Workspace**» записывает данные, поступающие на его вход, в рабочую область MATLAB.

Блоки математических операций

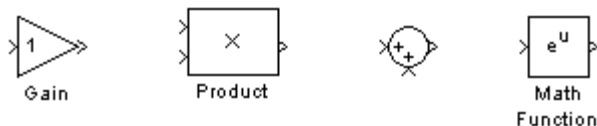


Рис. 1.3. Блоки математических операций

Блок **Gain** умножает входной сигнал на заданный коэффициент.

Блок **Product** перемножает два (или более) сигналов.

Блок **Sum** суммирует два (или более) сигналов.

Блок **Math Function** преобразует входной сигнал по заданной из списка функции (например, возведение в квадрат, корень квадратный, логарифм и т.д.).

Отметим еще некоторые часто встречающиеся блоки (рис. 1.4).

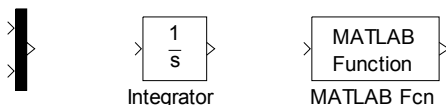


Рис. 1.4. Примеры часто встречающихся функций

Блок **Mux** (мультиплексор) объединяет сигналы. Пиктограмма мультиплексора показана первой слева на рис. 1.4.

Блок **Integrator** интегрирует сигнал.

Блок **MATLAB Fcn** формирует функцию, заданную пользователем на языке MatLab.

1.2. Построение простейших моделей

Рассмотрим пример, на котором покажем, как строится модель сигнала вида $x(t) = 0,5 * \sin \pi t + t^2$ на интервале $[0;2]$, и отобразим его на виртуальном осциллографе. На рис. 1.5 приведен один из вариантов построения сигнала.

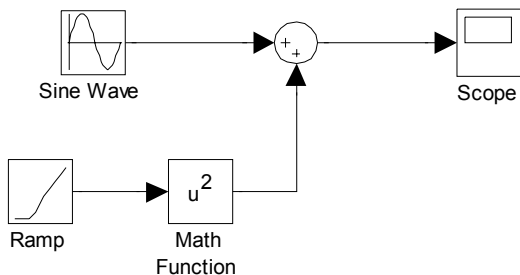


Рис. 1.5. Модель сигнала $x(t) = 0,5 * \sin \pi t + t^2$.

Сигнал $0,5 \sin \pi t$ задан в параметрах блока Sine Wave (рис. 1.6).

Для сигнала t^2 использовались два блока – блок линейного сигнала и блок математической функции, где была выбрана функция возведения в квадрат.

Результаты работы выведены на экран виртуального осциллографа (рис. 1.7).

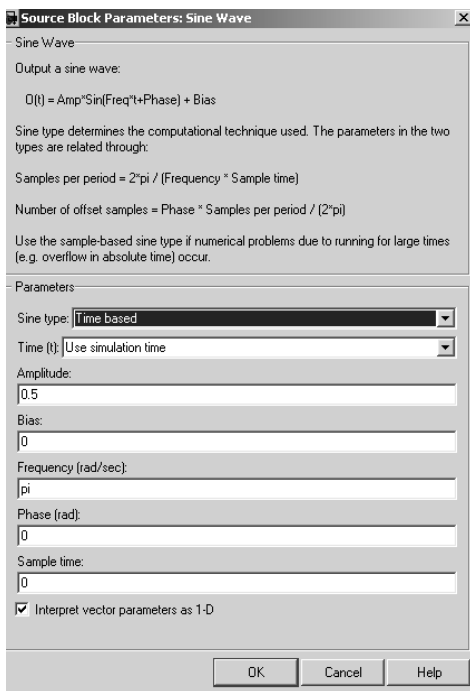


Рис. 1.6. Параметры блока Sine Wave.

Интервал моделирования задан в пределах от 0 до 2 в окне меню Simulation/ Configuration Parameters.

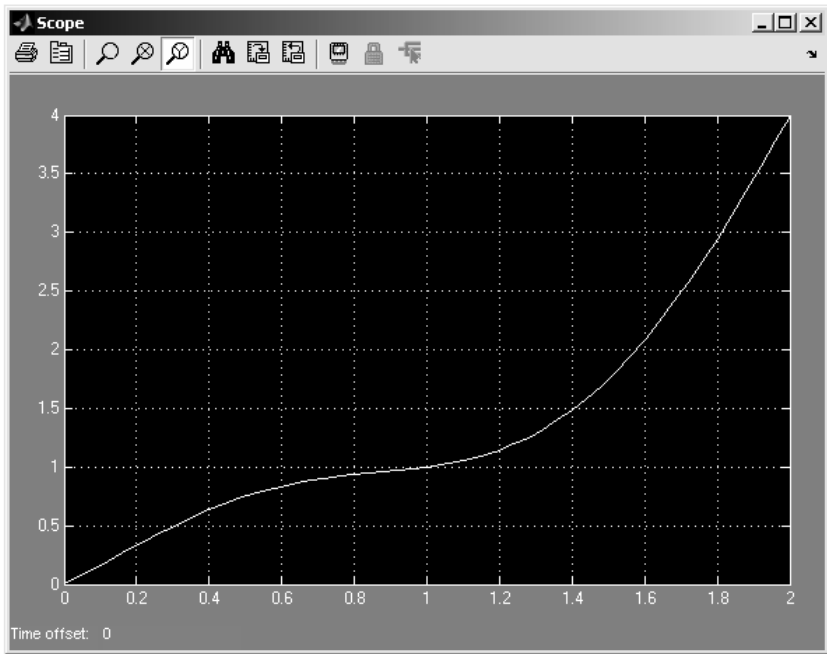


Рис. 1.7. Сигнал $x(t) = 0,5 * \sin \pi t + t^2$ на экране виртуального осциллографа.

Ниже приведены структурные схемы виртуальных генераторов систем базисных функций, которые будут использоваться в дальнейшем для полиномиальной аппроксимации методами наименьших квадратов и равных площадей. Для построения таких структурных схем могут быть использованы различные подходы. На рис. 1.8, 1.9 изображены две альтернативные схемы виртуальных генераторов системы степенных функций вида: $s_k(t) = \left\{ t^k \right\}_{k=0}^4$.

Первая из них использует каскадное соединение интеграторов, на вход первого из которых подается сигнал константы. Выходные сигналы интеграторов масштабируются с помощью масштабных звеньев и подаются вместе с сигналом константы (1) на входы смесителя (mux) для отображения на экране виртуального осциллографа (scope). Во второй схеме используются система умножителей, последовательно формирующая сигналы степенных функций. На рис 1.10 показан вид сформированной системы функций.

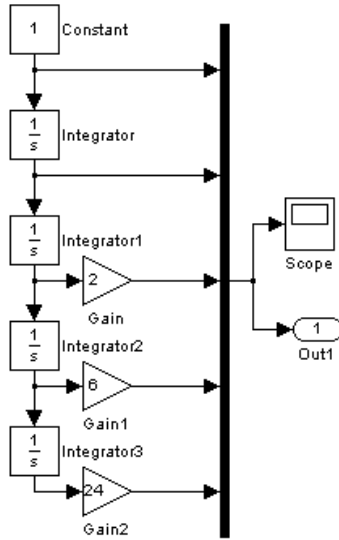


Рис. 1.8. Первый вариант структурной схемы виртуального генератора системы степенных базисных функций.

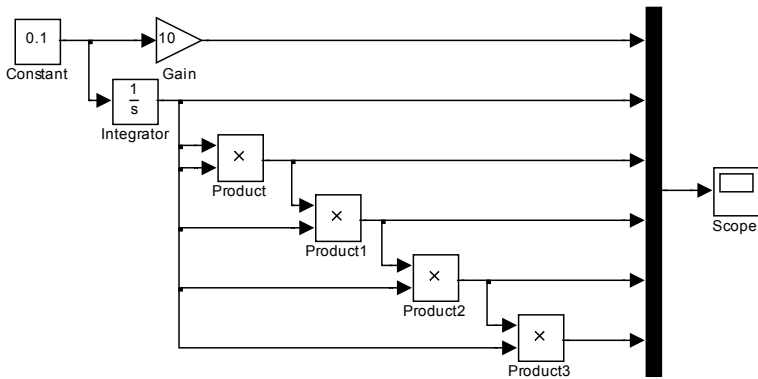


Рис. 1.9. Второй вариант структурной схемы виртуального генератора системы степенных базисных функций.

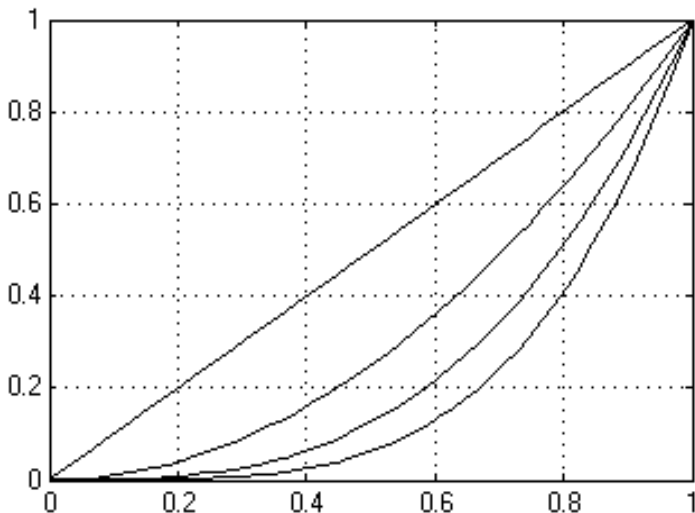


Рис. 1.10. Изображение системы степенных функций на экране виртуального осциллографа

На рис. 1.11 приведена структурная схема виртуального генератора системы экспоненциальных функций вида: $s_k(t) = \{e^{kt}\}_{k=0}^4$. В этой схеме используются 4 идентичных последовательно соединенных блока источника линейного сигнала (Ramp) и блока математической функции (Math Function). С помощью меню параметров решающих блоков задаются различные наклоны линейных сигналов (аргументов экспоненциальных функций) и выбирается тип функции блоков математической функции (в данном случае – exp). Выходные сигналы блоков математической функции через смеситель подаются на вход блока виртуального осциллографа. К одному из входов смесителя подключен сигнал константы (1), изображающий первую из экспоненциальных функций (e^0). Альтернативный вариант этой структурной схемы приведен на рис. 1.12. Здесь экспоненциальные функции кратного аргумента формируются с помощью умножителей. Параметры решающих элементов как и ранее задаются с помощью меню параметров. На рис.1.13 изображен вид системы экспоненциальных функций на экране виртуального осциллографа.

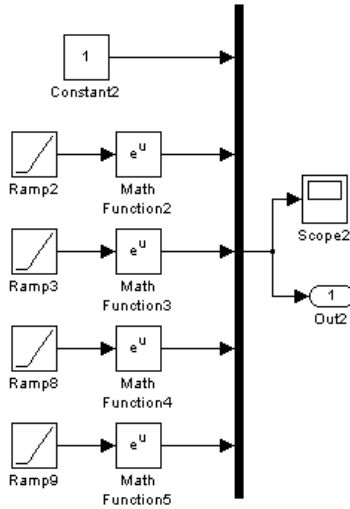


Рис. 1.11. Структурная схема виртуального генератора системы экспоненциальных функций (вариант 1)

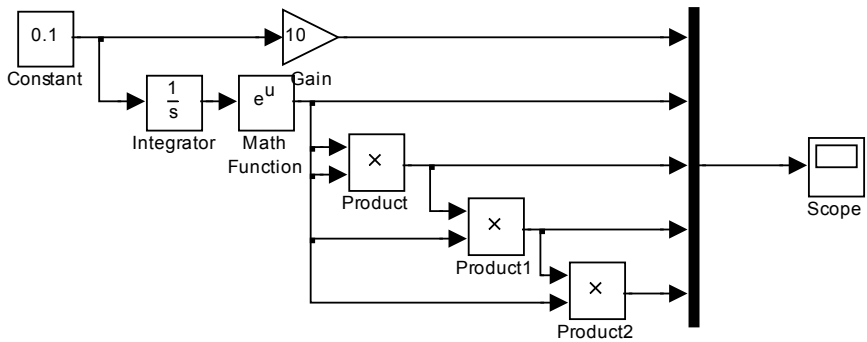


Рис. 1.12. Структурная схема виртуального генератора системы экспоненциальных функций (вариант 2)

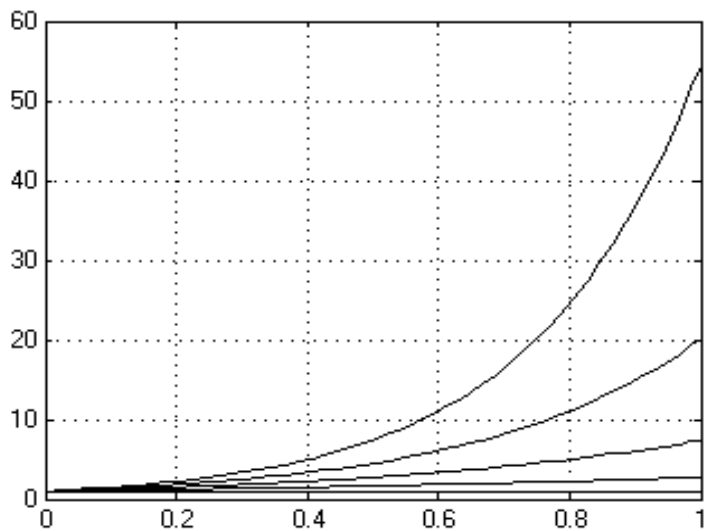


Рис. 1.13. Изображение системы экспоненциальных функций на виртуальном осциллографе.

2. МОДЕЛИ АЛГЕБРАИЧЕСКИХ ОБЪЕКТОВ

В аналоговой вычислительной технике существует несколько способов моделирования объектов, описываемых системами алгебраических уравнений [22 – 25]. Один из них, наиболее распространенный, сводится к решению системы обыкновенных дифференциальных уравнений, установившееся решение которой дает решение алгебраической задачи.

Пусть безинерционный объект описывается системой уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \dots & \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Для построения модели данного объекта применим метод сведения к эквивалентной системе дифференциальных уравнений.

Введем следующую систему дифференциальных уравнений:

$$\begin{aligned} \frac{dx_1}{dt} + a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - b_1 &= 0 \\ \frac{dx_2}{dt} + a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n - b_2 &= 0 \\ \dots & \\ \frac{dx_n}{dt} + a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n - b_n &= 0 \end{aligned}$$

Как только все производные затухнут ($\frac{dx_i}{dt} = 0$), получаем сигнал решения $\{x_1, x_2, \dots, x_n\}$.

Для эквивалентности этих двух систем уравнений должно обеспечиваться затухающее решение системы дифференциальных уравнений. Достаточным условием, обеспечивающим затухающее решение, является положительная определенность матрицы коэффициентов линейной системы уравнений. Это возможно, в частности, при условии, когда $a_{ii} \geq \sum_{j=1}^n a_{ij}$, $i \neq j$.

Пример 2.1.

Найти решение системы линейных алгебраических уравнений:

$$4x_1 + 2x_2 = 14$$

$$2x_1 + 5x_2 = -5$$

Перейдем к эквивалентной системе дифференциальных уравнений:

$$\frac{dx_1}{dt} = 14 - 4x_1 - 2x_2$$

$$\frac{dx_2}{dt} = -5 - 2x_1 - 5x_2$$

Структурная схема модели данной системы приведена на рис. 2.1. Она построена с использованием классических методов аналоговой вычислительной техники. Переходный процесс установления решения изображен на экране виртуального осциллографа (рис. 2.2).

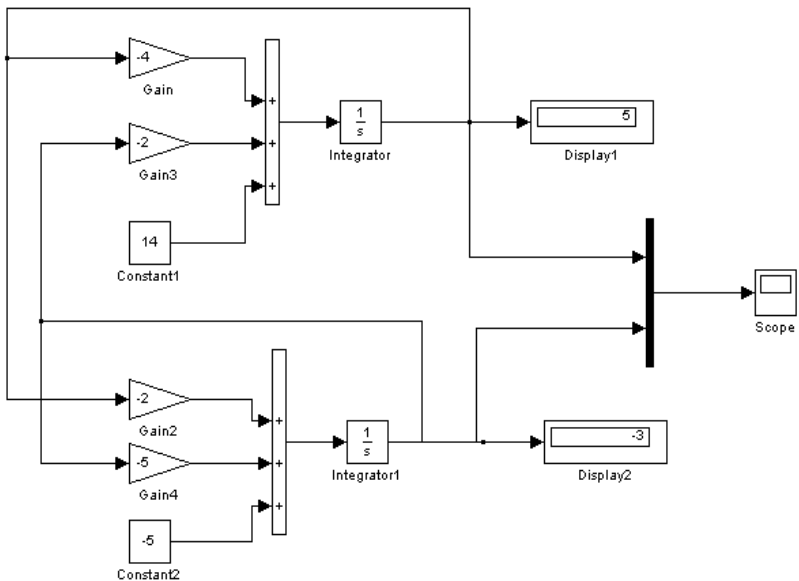


Рис.2.1. Структурная схема модели системы дифференциальных уравнений, эквивалентной системе линейных алгебраических уравнений второго порядка

На рис. 2.2 видно, что после $t = 2$ на выходах виртуальных интеграторов устанавливаются сигналы, соответствующие решению системы линейных алгебраических уравнений:

$$x_1 = 5,$$
$$x_2 = -3.$$

Расширение класса алгебраических объектов, матрица коэффициентов которых имеет произвольный вид, возможно путем использования методов квазианалогового моделирования [22, 23]. Примеры таких методов будут рассмотрены позднее.

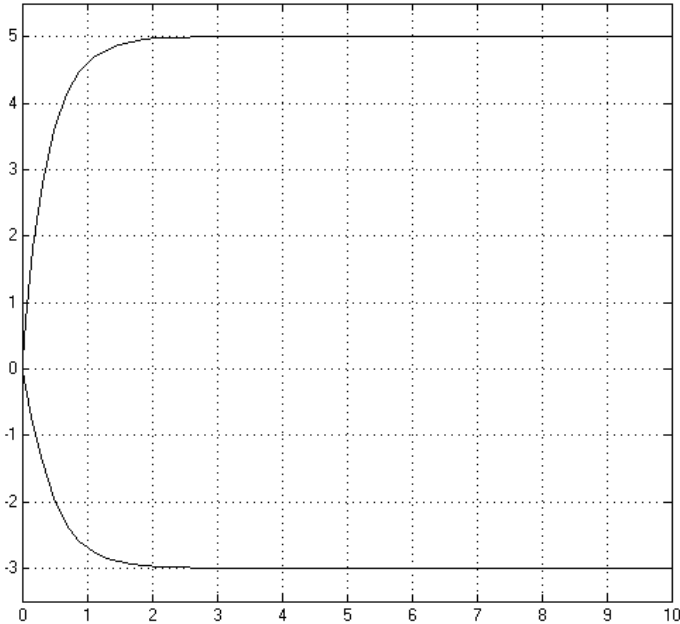


Рис. 2.2. Переходный процесс установления решения системы линейных алгебраических уравнений путем сведения к эквивалентной дифференциальной системе уравнений

Пример 3.2.

Найти решение системы:

$$5x_1 + 2x_2 + x_3 = 2,5$$
$$3x_1 + 7x_2 + 2x_3 = -1,5$$
$$4x_1 + 0,5x_2 + 5x_3 = 11,5$$

Модель системы и результаты ее работы представлены на рис. 2.3, 2.4 соответственно.

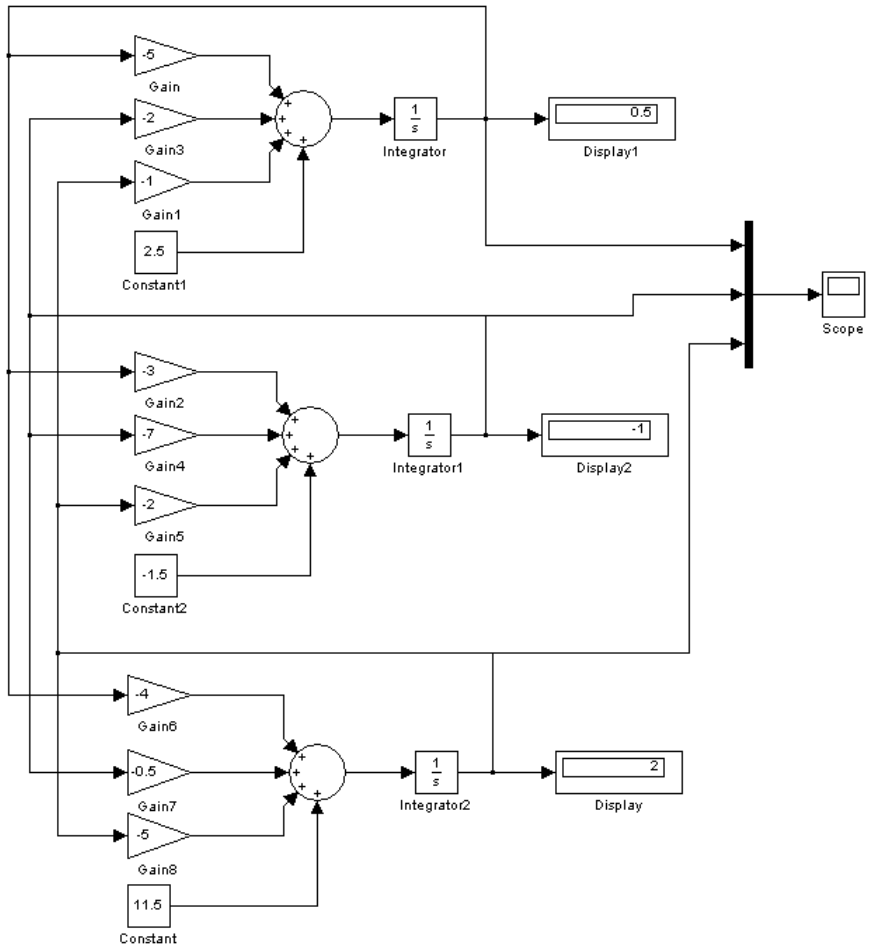


Рис. 2.3. Модель системы линейных алгебраических уравнений 3 порядка.

Решение системы уравнений, отображенное на цифровых регистраторах (Display1 – Display3) равно:

$$x_1 = 0,5$$

$$x_2 = -1.$$

$$x_3 = 2$$

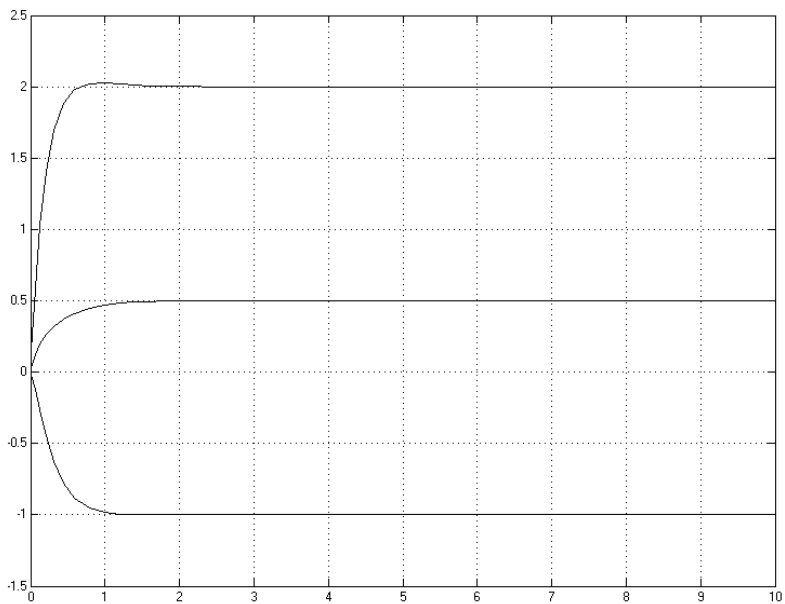


Рис. 2.4. Переходный процесс установления решения системы уравнений.

3. АППРОКСИМАЦИЯ СИГНАЛОВ

В этой разделе рассмотрим широко известный аппроксимационный метод наименьших квадратов для функции одной переменной, а также альтернативный метод, который позволяет снизить вычислительные затраты и упростить реализацию на ПЭВМ – модифицированный метод равных площадей. В частности, в Simulink-структурах анализа динамических систем удобнее использовать именно метод равных площадей благодаря его преимуществам.

Пусть непрерывный сигнал $x(t)$ задан на некотором интервале изменения аргумента $[0, T]$. Выбирается некоторая система функций $\bar{\mathbf{S}} = \{s_k(t)\}_{k=0}^n$, заданных на том же интервале изменения аргумента. Из теории аппроксимации известно [6, 18], что сигнал может быть представлен на определенном интервале изменения аргумента обобщенным полиномом

$$x_a(t) = \sum_{k=0}^n X(k)s_k(t). \quad (3.1)$$

Систему коэффициентов $X(k)$ обычно называют аппроксимирующим полиномиальным спектром (АПС) сигнала $x(t)$.

3.1. Метод наименьших квадратов

Задача определения аппроксимирующего спектра методом наименьших квадратов формулируется следующим образом [18, 26]: определить систему величин $X(k)$, минимизирующую норму $\|\beta(t)\|$ функции ошибки

$$\beta(t) = x_a(t) - x(t).$$

В качестве нормы ошибки будем использовать условие

$$\frac{1}{T} \int_0^T r(t)\beta^2(t)dt = \min,$$

где $r(t)$ - весовая функция.

Вектор АПС $\bar{\mathbf{X}} = \{X(k)\}_{k=0}^n$ сигнала находится по формуле:

$$\bar{\mathbf{X}} = \mathbf{W}^{-1} \cdot \bar{\mathbf{Q}}, \quad (3.2)$$

где элементы операционных матриц \mathbf{W} и и вектора $\bar{\mathbf{Q}}$ вычисляются следующим образом:

$$w_{ij} = \int_0^T r(t) s_{i-1}(t) s_{j-1}(t) dt,$$

$$q_i = \int_0^T r(t) s_{i-1}(t) x(t) dt, \quad (3.3)$$

$$i, j = 1, \dots, n + 1.$$

Рассмотрим использование метода наименьших квадратов на примере.

Пример 3.1.

Необходимо аппроксимировать методом наименьших квадратов сигнал $x(t) = \cos \pi t^2$ на интервале $[0, 1]$. В качестве базисных функций взять степенные функции 5-го порядка.

Структурные схемы определения коэффициентов по формулам 3.2-3.3 представлены на рис. 3.1 и 3.2.

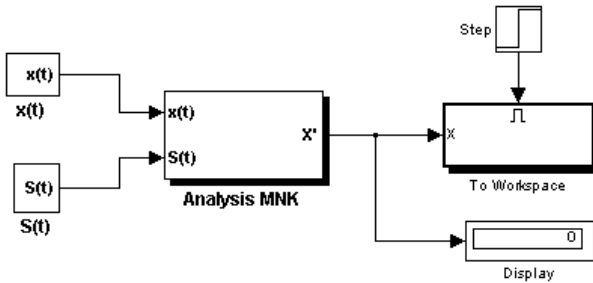


Рис. 3.1. Структурная схема определения коэффициентов аппроксимирующего полинома

Основу схемы составляет подсхема Analysis MNK, которая в развернутом виде приведена на рис. 3.1б. Виртуальные индикаторы Scope и Display, показанные на этой подсхеме, не являются обязательными для определения коэффициентов аппроксимирующего полинома, а предназначены для контроля правильности построения структурной схемы в процессе отладки виртуальной модели.

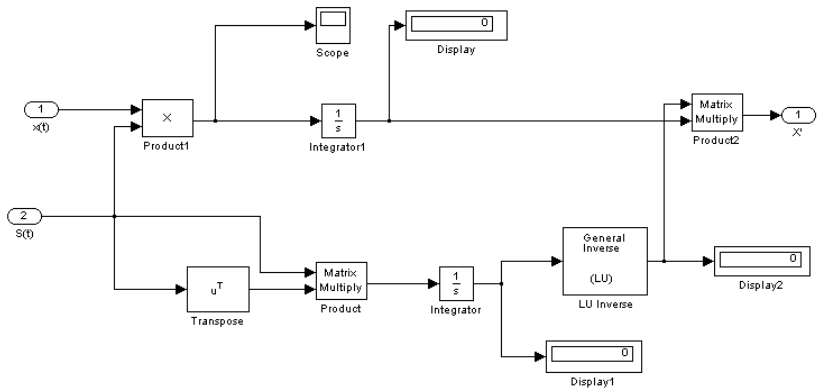


Рис. 3.2. Расчет коэффициентов по методу наименьших квадратов.

После расчета коэффициентов сигнал восстанавливается по схеме, приведенной на рис. 3.3.

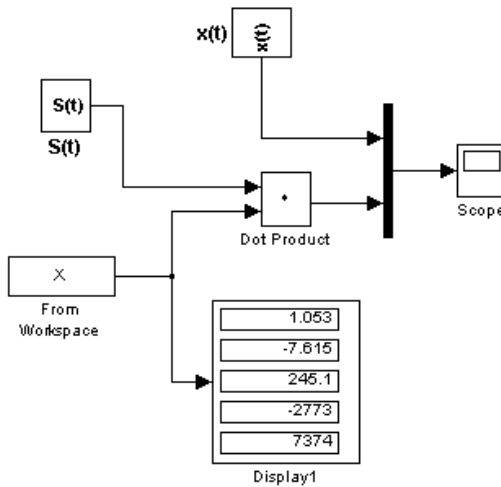


Рис. 3.3. Восстановление сигнала по найденному спектру

Результаты работы модели выводятся на экран виртуального осциллографа, и изображены на рис. 3.4.

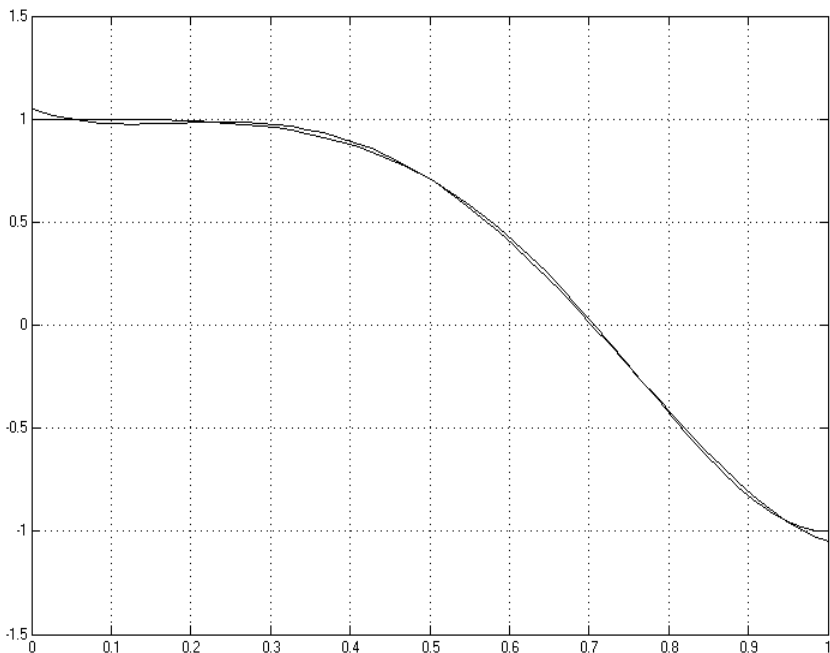


Рис. 3.4. Исходный сигнал $x(t) = \cos \pi t^2$ и его аппроксимация по методу наименьших квадратов.

3.2. Модифицированный метод равных площадей

Метод равных площадей является более простым методом аппроксимации, основанный на определении коэффициентов аппроксимирующего полинома, исходя из условия равенства нулю интегралов функций ошибки аппроксимации на некоторой системе интервалов изменения аргумента. Идея метода равных площадей, как метода ручных вычислений, приведена у Мелентьева П.В. [20], однако, как операционный метод он был развит в работе Симак Л.А. [21, 26].

Суть модифицированного метода равных площадей заключается в следующем. Интервал изменения аргумента t разбивается на $(n+1)$ не обязательно равных частей (подинтервалов). На рис. 3.5 представлены возможные типы систем подинтервалов. Уравнение аппроксимирующей кривой подбирается так, чтобы площади, ограничиваемые аппроксимируемой и аппроксимирующей функциями, были равны между собой для каждого участка в

отдельности. Это приводит к системе линейных алгебраических уравнений, решение которой определяет аппроксимирующий спектр.

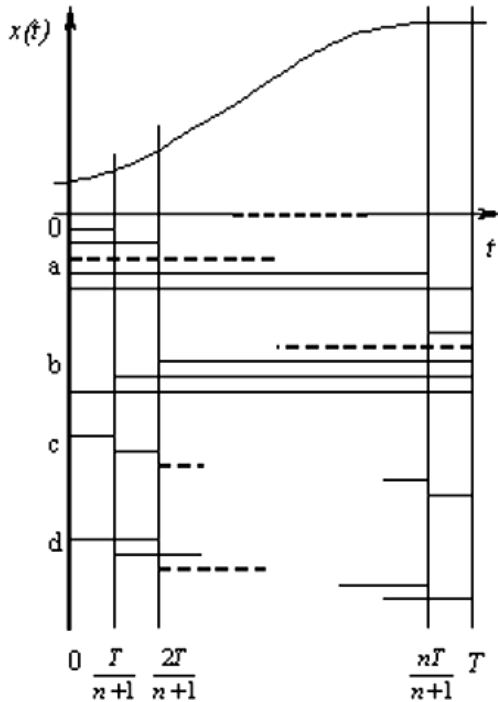


Рис. 3.5. Системы подинтервалов для метода равных площадей

Условия равенства указанных площадей (интегралов) для некоторой системы линейно-независимых подинтервалов имеет вид:

$$\int_{t_m}^{t_i} r(t) \left[\sum_{k=0}^n X(k) s_k(t) - x(t) \right] dt = 0. \quad (3.4)$$

Например, при выборе системы подинтервалов вида $[0, \frac{iT}{n+1}]$, $i = 1, \dots, n+1$, (рис. 3.5(a)) матрица \mathbf{W} и вектор $\bar{\mathbf{Q}}$ вычисляются по формулам:

$$w_{ij} = \int_0^{\frac{iT}{n+1}} r(t) s_{j-1}(t) dt,$$

$$q_i = \int_0^{\frac{iT}{n+1}} r(t) x(t) dt,$$

$$i, j = 1, \dots, n+1. \quad (3.5)$$

Рассмотрим реализацию модифицированного метода равных площадей в среде MatLab/Simulink. Модель, описывающая модифицированный метод равных площадей, представлена на рис. 3.6.

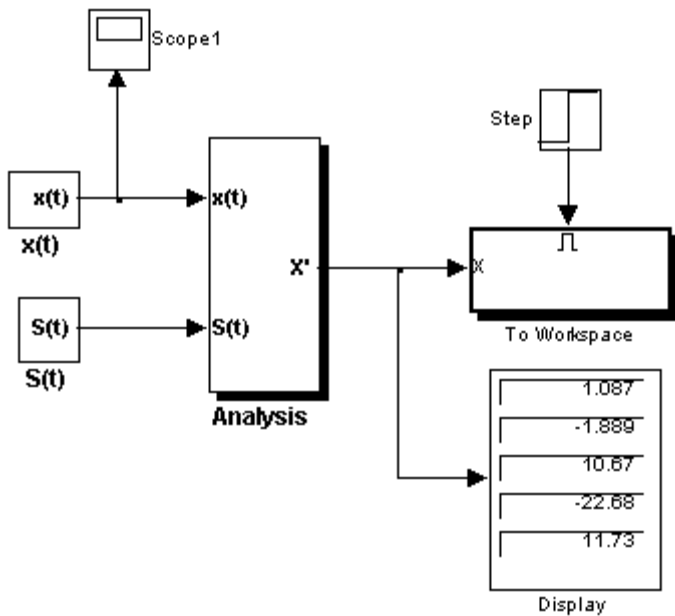


Рис. 3.6. Структурная схема реализации модифицированного метода равных площадей

Основной частью схемы является блок анализа (Analysis). Его входные параметры – непрерывный сигнал $x(t)$ и система базисных функций $\mathbf{S}(t)$. На выходе находится дисплей, на котором отображается полученный спектр, и блок To Workspace, записывающий спектр в рабочую область. Рассмотрим пример из предыдущего подраздела. Необходимо определить

аппроксимирующий спектр функции $x(t) = \text{Cos}(\pi t^2)$ на интервале $[0, 1]$. Интервал разбивается на пять подинтервалов $[0; 0,1]$, $[0; 0,2]$, $[0; 0,3]$, $[0; 0,4]$, $[0; 0,5]$. Базисные функции имеют вид $s_k(t) = t^k$, $k = \overline{0, n}$.

Рассмотрим подробнее блоки, задающие сигнал и базисные функции. На рис. 3.7 показан блок $X(t)$.

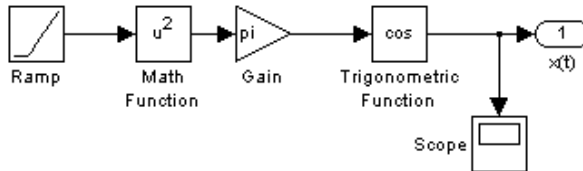


Рис. 3.7. Блок, генерирующий аппроксимируемый сигнал

Источником является генератор линейно возрастающего сигнала Ramp, который в данном случае ассоциируется с временем t . Далее, на рисунке видно, как с помощью блоков математических операций (возведение в степень, умножение на константу, преобразование входного сигнала с помощью тригонометрической функции), на выходе получаем заданный сигнал $x(t)$. Для наглядности перед выходом подключен «виртуальный осциллограф» (Scope), на котором в процессе моделирования отображается сигнал $x(t) = \text{Cos}(\pi t^2)$. Общий вид окна блока Scope с его панелью инструментов изображен на рис. 3.8.

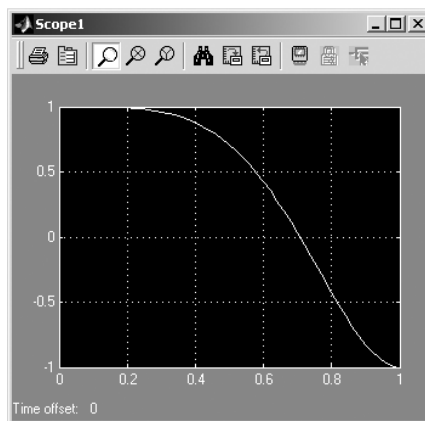


Рис. 3.8. Панель инструментов виртуального осциллографа и график сигнала, выведенный на осциллограф

Блок $\underline{S}(t)$ генерирует систему базисных функций $s_k(t) = t^k, k = 0, n$, его структурная схема показана на рис. 3.9.

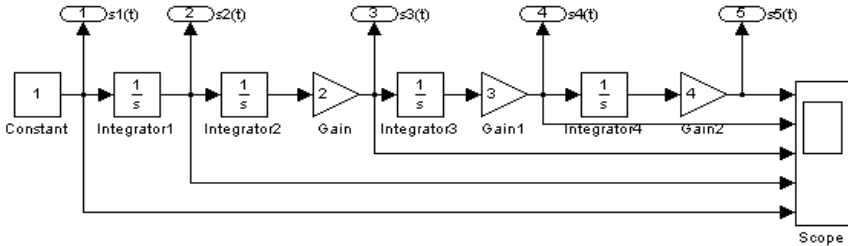


Рис. 3.9. Структурная схема виртуального генератора системы базисных функций.

Поскольку система базисных функций имеет удобный для моделирования вид, можно из первой функции $s(t) = 1$ с помощью интегрирования (блок Integrator) получить следующие. Блоки умножения на константу Gain, Gain1, Gain2 необходимы для нормирования степенных функций.

Полученные функции выводятся на выход этой подсистемы и на виртуальный осциллограф. Изображения базисных функций показаны на рис. 3.10.

Перед тем, как перейти к описанию центрального блока модели Analysis, необходимо отметить, что в подсистемах этого блока используется конструкция, которую можно использовать как при расчете вектора \bar{Q} , так и при расчете векторов-столбцов матрицы \mathbf{W} . Общим является интегрирование функции на одних и тех же интервалах. Таким образом для получения вектора \bar{Q} необходимо проинтегрировать на заданной системе подинтервалов сигнал $x(t)$, а для получения столбцов матрицы \mathbf{W} – базисные функции, на той же системе подинтервалов. Это позволяет для выполнения указанных операций использовать одну и ту же подсистему, структурная схема которой представлена на рис. 3.11.

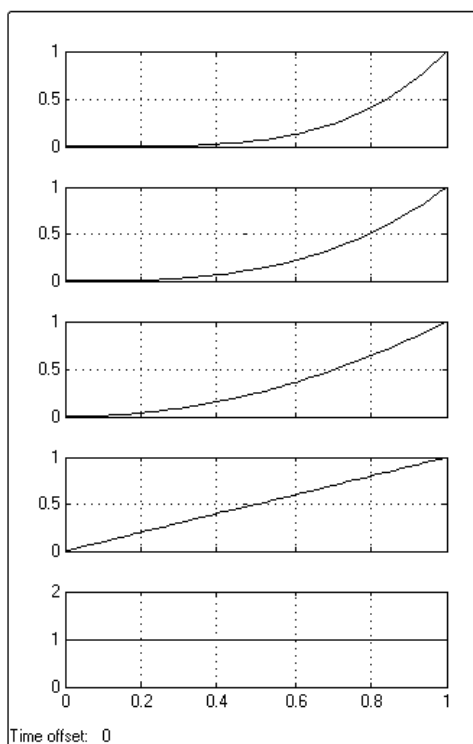


Рис. 3.10. Система базисных функций

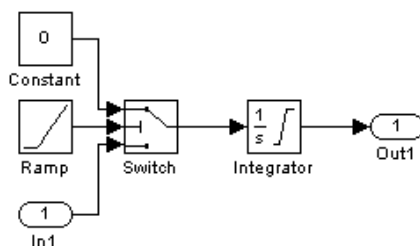


Рис. 3.11. Блок интегрирования функции на системе подинтервалов.

На вход этого блока поступает сигнал – интегрируемая функция. Два других – сигнал, нулевой сигнал и генератор линейно возрастающей функции (в данном случае он описывает время t) являются вспомогательными. Пока аргумент t находится в пределах заданного подинтервала, переключатель Switch подает на интегратор сигнал $x(t)$. Но как только время выходит за пределы подинтервала, сигнал переключается на нулевой и, таким образом, больше не влияет

на значение интеграла. Так как MatLab ориентирован в первую очередь на обработку массивов данных, то имеет смысл задавать в параметрах блоку Switch не один подинтервал, а вектор, состоящий из всех пяти подинтервалов, тогда на выходе также получим вектор значений интеграла. Определение системы подинтервалов с помощью параметров блока Switch показано на рис. 3.12.

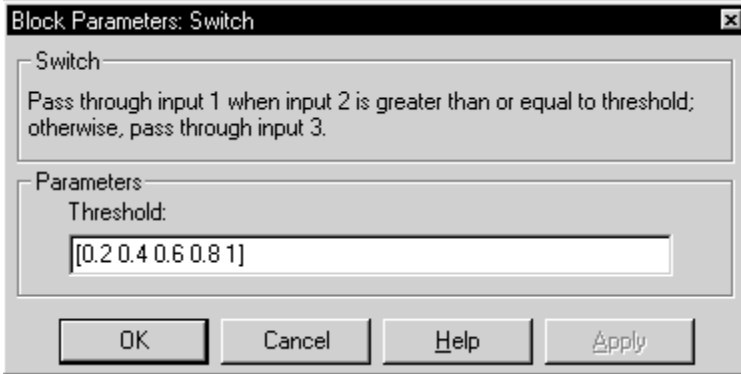


Рис. 3.12. Окно настройки параметров блока Switch.

Теперь рассмотрим структурную схему блока Analysis.

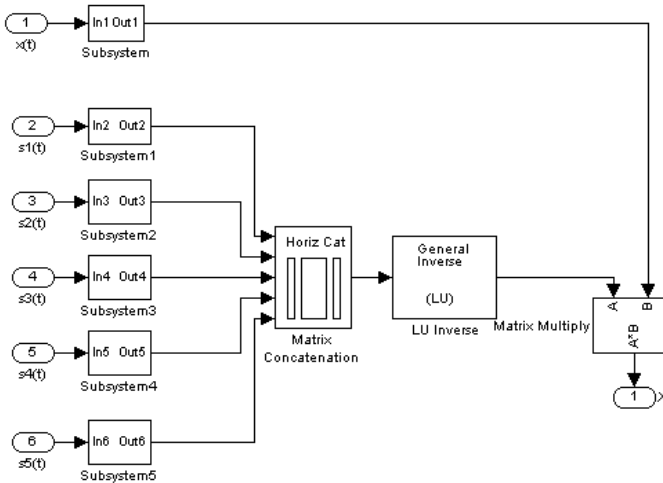


Рис. 3.13. Структурная схема блока Analysis

На вход блока подается сигнал $x(t)$ и система базисных функций $s(t)$. Подсистема Subsystem интегрирует сигнал и подает на выход

вектор \bar{Q} , а подсистемы Subsystem 1-5 – векторы-столбцы матрицы \mathbf{W} . Блок Matrix Concatenation объединяет столбцы в матрицу. Далее размещены блоки расчета обратной матрицы и умножения матриц. В итоге на выходе получаем сигнал аппроксимирующего спектра, который появляется на экране дисплея модели самого высокого уровня (рис. 3.6). На рис. 3.14 приведено значение спектра функции, о рассматриваемого в качестве примера.

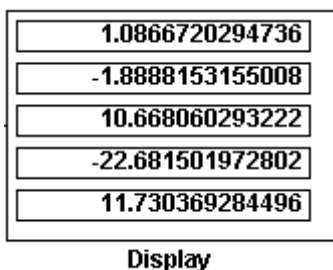


Рис. 3.14. Значение аппроксимирующего спектра.

Для детального рассмотрения метода равных площадей возьмем другую систему интервалов $[\frac{iT}{n}, T]$, $i = 0, \dots, n-1$. В этом случае модель метода будет отличаться параметрами блока Switch (вектор пороговых значений будет иметь вид: [0 0.2 0.4 0.6 0.8]) и в блоке интегрирования функции на системе подинтервалов (рис. 3.11) поменяются местами входной сигнал $x(t)$ и нулевой сигнал. Аппроксимирующий спектр для такой системы подинтервалов представлен на рис. 3.15.

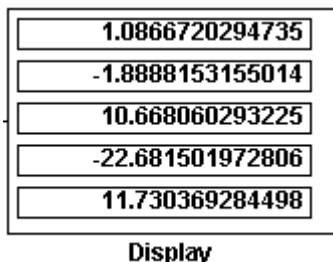


Рис. 3.15. Значения аппроксимирующего спектра для другой системы подинтервалов.

Можно рассмотреть обратную задачу – построение сигнала по заданному спектру в системе базисных функций. Для наглядности

используем спектр, полученный выше, на основе его построим аппроксимацию и сравним ее с реальным сигналом.

Структурная схема синтезирования сигнала будет иметь вид (рис. 3.16.). Блоки $X(t)$ и $S(t)$ были описаны ранее. Блок From Workspace извлекает из рабочей области спектр сигнала. Базисные функции умножаются на соответствующие коэффициенты спектра, суммируются и выводятся на виртуальный осциллограф. На этот же осциллограф подается заданный сигнал. Графики сигнала и его аппроксимации представлены на рис. 3.17.

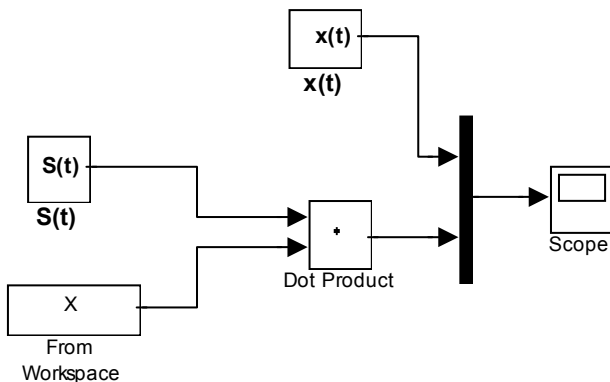


Рис. 3.16. Структурная схема синтеза сигнала.

В модели был рассмотрен случай, когда количество базисных функций равнялось количеству подынтервалов. Но это необязательно. Можно увеличить точность вычислений за счет большего числа подынтервалов. Однако, в этой модели необходимо блок расчета обратной матрицы заменить на блок вычисления псевдообратной матрицы. Этот блок можно найти в дополнительной библиотеке Simulink – DSP (Digital Signal Processing) Blockset.

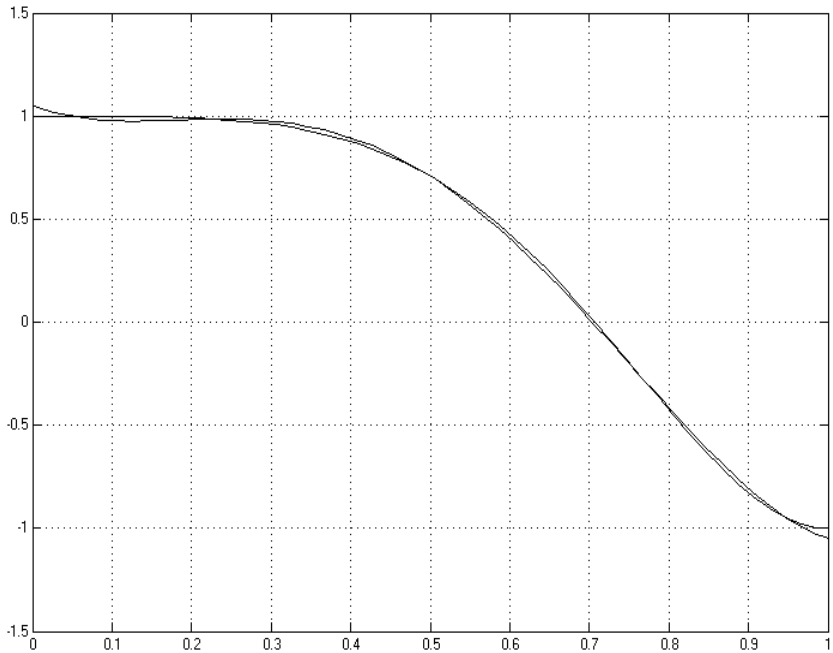


Рис. 3.17. График сигнала и его аппроксимация.

Этот набор блоков является одним из мощных инструментов системы Simulink. Библиотека обеспечивает решение практически всех типов задач, связанных с цифровой обработкой сигналов, в том числе с обработкой звуковых сигналов и изображений.

3.3. Блочно-импульсная аппроксимация

Одной из наиболее распространенных и удобных для применения в задачах обработки и моделирования сигналов динамических систем оказывается базисная система блочно-импульсных функций [21]. Рассмотрим блочно-импульсную аппроксимацию и некоторые необходимые операции над спектрами.

Пусть отрезок $[0, T]$ разбит на m равных частей, длиной $h = T/m$. На решетке аргумента $\{ih\}_{i=1}^m$ задается система базисных функций

$$v_i(t) = \sigma(t - (i - 1)h) - \sigma(t - ih). \quad (3.6)$$

Здесь $i = 1, \dots, m$; $v_i(t)$ – прямоугольные блочно-импульсные функции; $\sigma(t)$ – функция единичного скачка:

$$\sigma(t) = \begin{cases} 0, & \text{если } t < 0, \\ 1, & \text{если } t \geq 0. \end{cases} \quad (3.7)$$

Аппроксимация сигнала по системе блочно-импульсных функций определяется по формулам (3.2)–(3.3) как

$$\tilde{x}(t) = \sum_{i=1}^m X_0(i) v_i(t), \quad (3.8)$$

где

$$X_0(i) = \frac{1}{h} \int_{(i-1)h}^{ih} x(t) dt. \quad (3.9)$$

Если к блочно-импульсным базисным функциям добавить кусочно-линейные функции:

$$w_i(t) = \left(\frac{2t}{h} + 1 - 2i \right) v_i(t), \quad (3.10)$$

то аппроксимация сигнала по системе локально-импульсных базисных функций первого порядка существенно улучшится и будет определяться по формуле:

$$x_a(t) = \sum_{i=1}^m (X_0(i) v_i(t) + X_1(i) w_i(t)). \quad (3.11)$$

На рис. 3.18, 3.19 представлены структурные схемы моделей, генерирующих блочно-импульсные базисные функции нулевого и первого порядков.

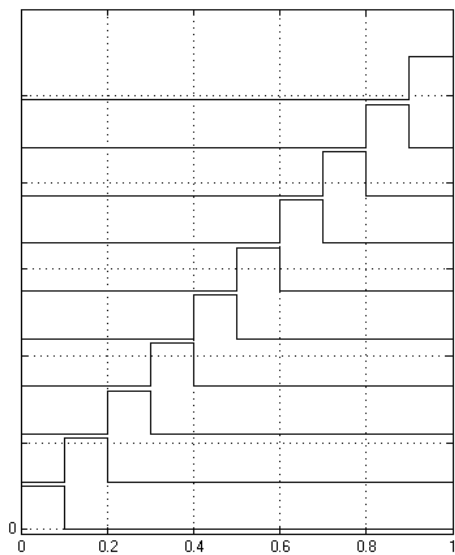
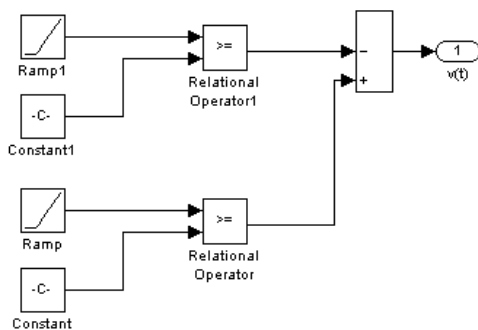


Рис. 3.18. Структурная схема, генерирующая блочно-импульсные функции нулевого порядка, и график блочно-импульсных функций.

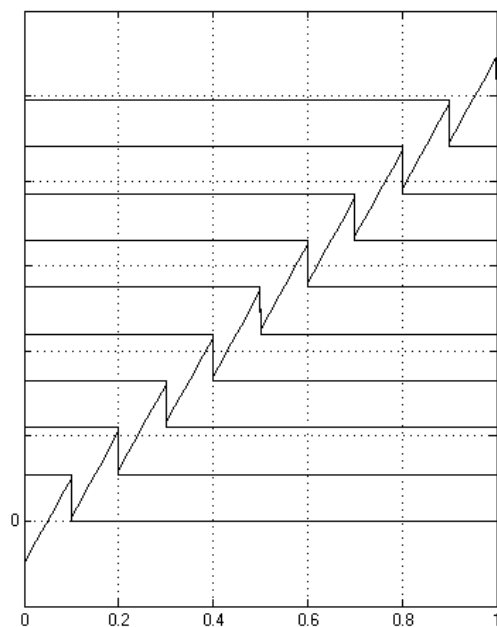
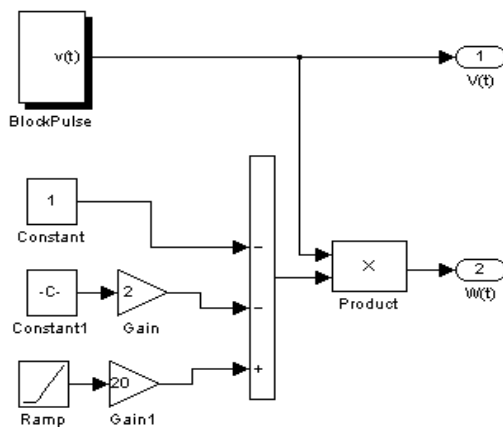


Рис. 3.19. Структурная схема, генерирующая блочно-импульсные функции первого порядка, и график кусочно-линейных функций.

Пример 3.2.

Рассмотрим реализацию метода аппроксимации блочно-импульсными функциями нулевого и первого порядков в среде MATLAB/Simulink. В качестве примера используем сигнал $x(t) = \cos 2\pi t^2$ на интервале $[0, 1]$, $m=10$. Структурная схема анализа сигнала рассматриваемым методом представлена на рис. 3.20. Назначение блоков следующее:

- BF генерирует систему базисных функций;
- $X(t)$ генерирует анализируемый сигнал;
- X_0, X_1 рассчитывают элементы спектра;
- Step определяет момент записи спектра в рабочую область. Полученный спектр, кроме записи, выдается на дисплей.

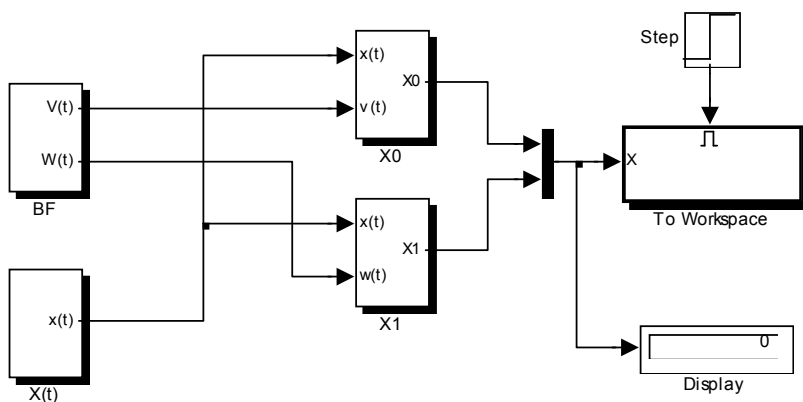


Рис. 3.20. Структурная-схема анализа сигнала блочно-импульсным методом.

Структурная схема, представленная на рис. 3.21, восстанавливает сигнал. На вход схемы подаются базисные функции и спектр анализируемого сигнала, на выход – аппроксимация сигнала по соответствующему методу и исходный сигнал.

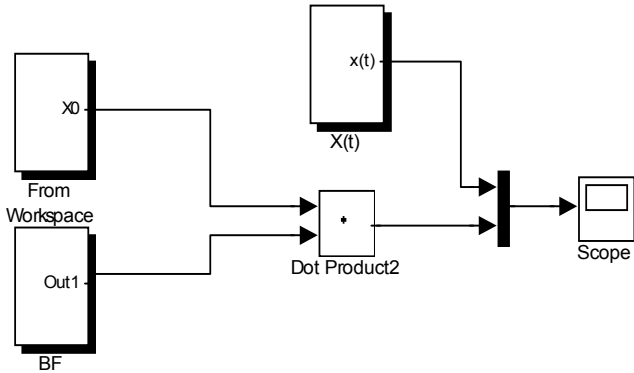


Рис. 3.21. Структурная схема синтеза сигнала по методу БИФ.

На виртуальный осциллограф подаются исходный и восстановленный сигналы. Они представлены на рис. 3.22 для обоих методов соответственно. Из графиков видно, что даже при небольшом числе подынтервалов (m) получается довольно точная аппроксимация сигнала. Точность можно повышать, разбивая интервал определения сигнала на большее количество подынтервалов.

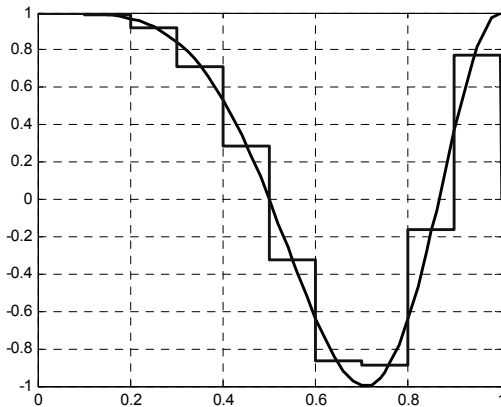


Рис. 3.22. Результаты аппроксимации сигнала.

4. МОДЕЛИ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

Распространенным способом описания поведения динамической системы является система дифференциальных или интегро-дифференциальных уравнений.

Реализацию таких математических моделей в Simulink рассмотрим на ряде примеров.

Пример 4.1. Модель физического маятника, находящегося под воздействием экспоненциально-затухающего косинусоидального возмущения.

Уравнение движения такого маятника имеет вид:

$$\frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_2 y(t) = a_3 e^{-a_4 t} \cdot \cos a_5 t,$$

$$y(t) \Big|_{t=0} = y_0,$$

$$\frac{dy(t)}{dt} \Big|_{t=0} = y'_0.$$

Выбрав числовые значения параметров, например: $a_1 = a_2 = 0,1$, $a_3 = -5$, $a_4 = 1$, $a_5 = 0,1$, получим следующее уравнение:

$$y'' + 0.1y' + 0.1y = -5e^{-t} \cdot \cos(0.1t),$$

$$y(0) = -1.5,$$

$$y'(0) = 2.$$

Структурная схема модели будет иметь вид, показанный на рис. 4.1.

Результаты работы модели показан на экране виртуального осциллографа (рис.4.2), а параметрический график зависимости производной сигнала от сигнала (фазовый портрет маятника) изображен на рис. 4.3.

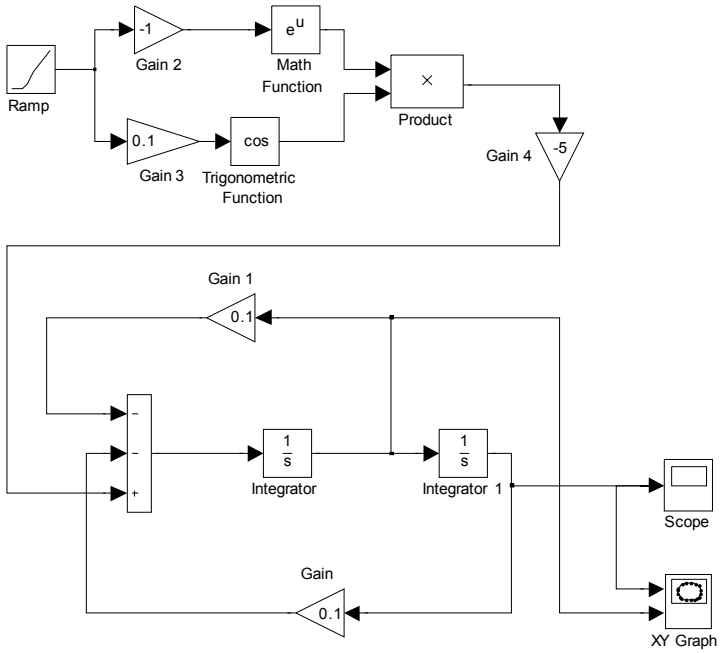


Рис. 4.1. Модель физического маятника

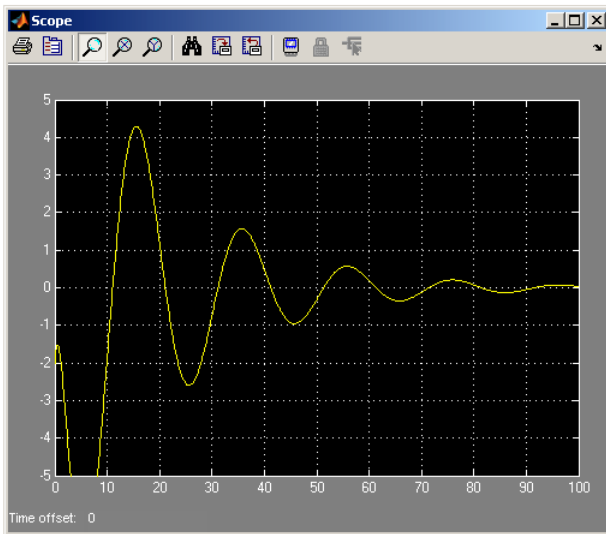


Рис. 4.2. Движение физического маятника

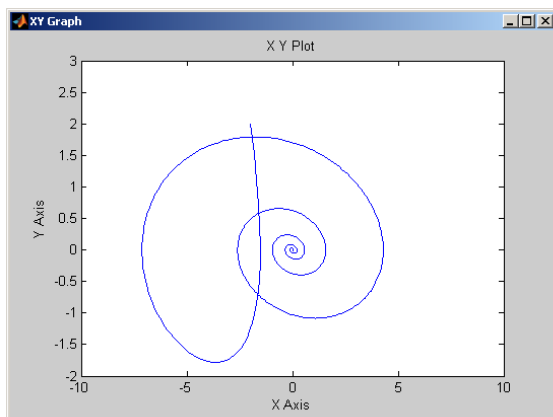


Рис. 4.3. Фазовый портрет физического маятника.

Пример 4.2. Модель динамической системы, описываемой дифференциальным уравнением 3-го порядка.

Задано дифференциальное уравнение:

$$\frac{d^3 y(t)}{dt^3} + 2.5 \frac{d^2 y(t)}{dt^2} + 6 \frac{dy(t)}{dt} + 2.5 y(t) = e^{-t},$$

$$y(0) = 1; y'(0) = -1; y''(0) = 2.$$

Структурная схема модели динамической системы, построенная по обычным правилам аналоговой вычислительной техники, приведена на рис.4.4.

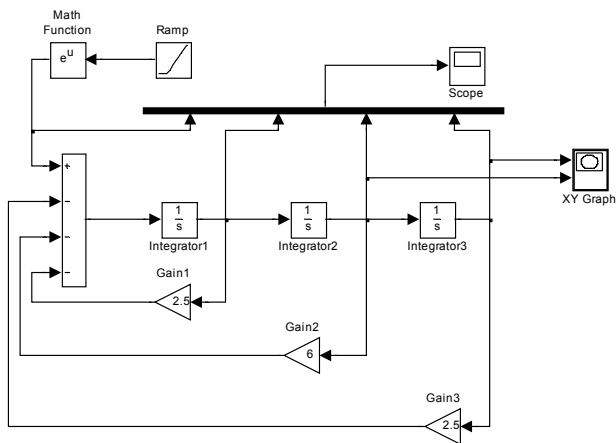


Рис. 4.4. Структурная схема модели динамической системы 3 порядка.

Коэффициенты уравнения устанавливаются в окнах параметров масштабных блоков (Gain1 – Gain3). Начальные условия для функций и производных в окнах параметров интеграторов (Integrator1 – Integrator 3). Правая часть дифференциального уравнения сформирована с помощью блоков Ramp (генератор аргумента $-t$) и блока MathFunction, настроенного на реализацию экспоненциальной функции. Визуализация переходного процесса показана на экране виртуального осциллографа (рис. 4.5). Фазовый портрет системы на экране виртуального двухкоординатного регистратора (XY Graph) приведен на рис. 4.6.

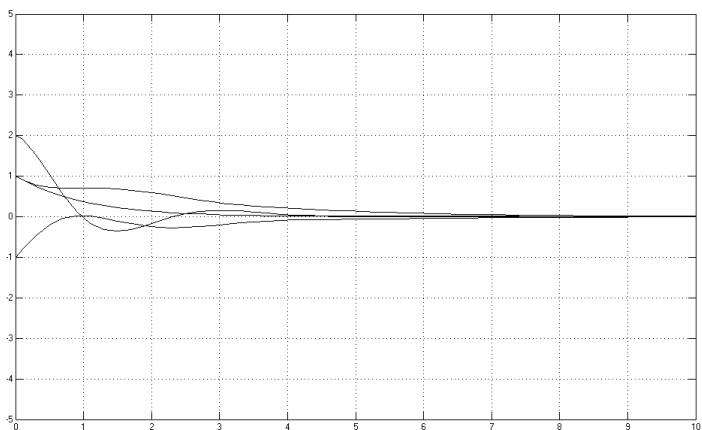


Рис. 4.5. Переходный процесс системы

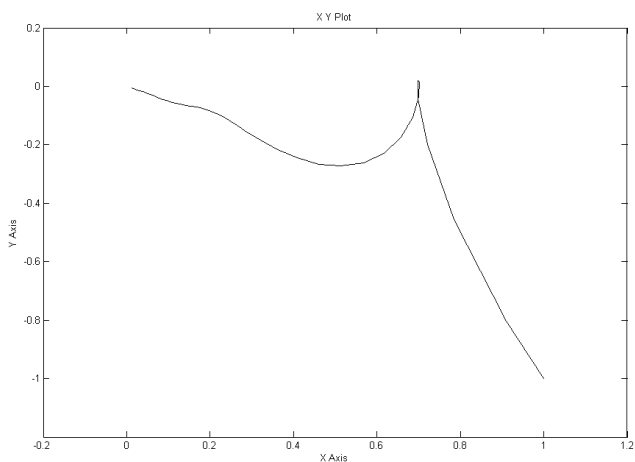


Рис. 4.6. Фазовый портрет системы

Пример 4.3 Модель траектории полета тела, брошенного с начальной скоростью под углом к горизонту. Предположим, что наблюдатель, находясь над уровнем земли на высоте 1 м, бросил камень под углом 30 градусов к горизонту с начальной скоростью 20 м/сек. Необходимо реализовать модель траектории полета камня под действием силы тяжести и определить расстояние от наблюдателя до точки падения камня. Влиянием атмосферы на полет камня пренебречь. Уравнения движения камня, как это следует из школьного курса физики, имеют вид:

$$y = y_0 + v \cdot \sin(\alpha) \cdot t - g \cdot \frac{t^2}{2},$$

$$x = v \cdot \cos(\alpha) \cdot t,$$

где: $y_0 = 1\text{ м}$, $v = 20\text{ м/сек}$, $\alpha = 30^\circ$, $g = 9.81\text{ м/сек}^2$.

Структурная схема модели траектории камня приведена на рис. 4.7.

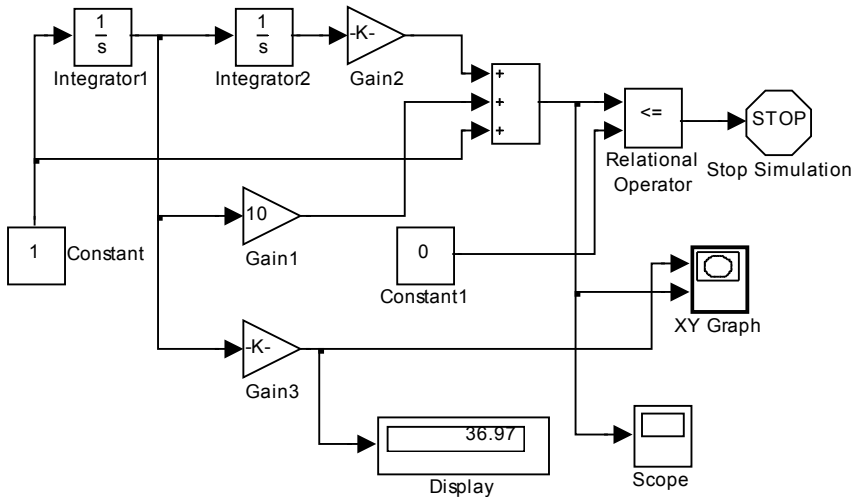


Рис. 4.7. Структурная схема модели траектории камня

Вертикальная составляющая начальной скорости камня задается блоком Gain1, горизонтальная составляющая - блоком Gain3, ускорение земного тяготения – блоком Gain2. Значение текущей высоты полета камня как функции времени формируется на выходе сумматора. Сигнал окончания моделирования формируется блоками Relational Operator и Stop Simulation в момент времени, когда

высота y сравнивается с нулем. Показания цифрового регистратора соответствует длине пути по горизонтали, пройденной камнем до момента соприкосновения с землей. На рис. 4.8 и 4.9 показаны соответственно траектория камня на экране виртуального двухкоординатного регистратора и график изменения во времени высоты полета камня.

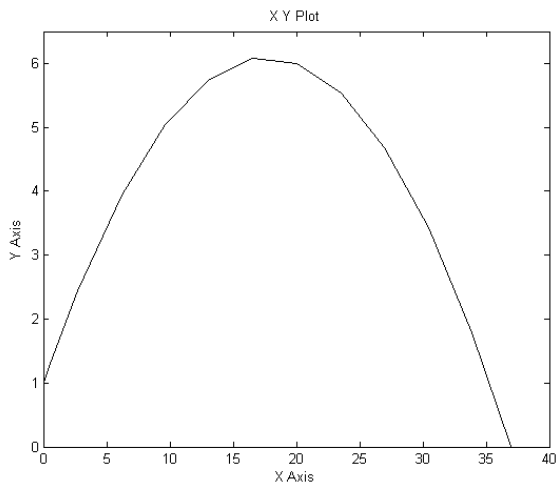


Рис.4.8. Траектория полета камня, брошенного под углом 30 градусов к горизонту с начальной высоты 1м, с начальной скоростью 20 м/сек.

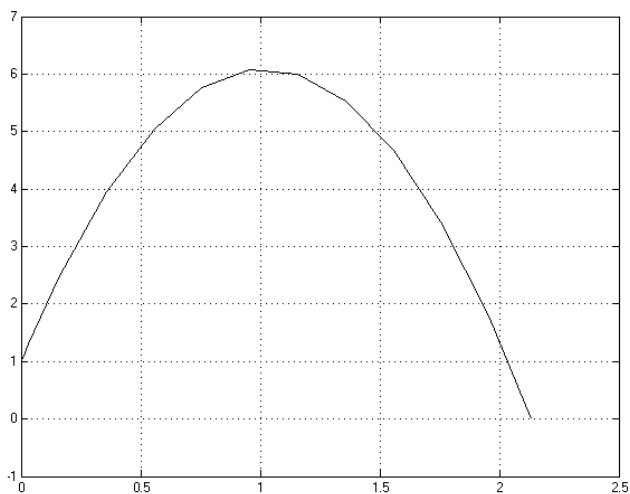


Рис.4.9. График изменения во времени высоты полета камня.

5. ОБРАТИМОСТЬ МОДЕЛЕЙ И ЗАДАЧИ ОПТИМИЗАЦИИ

Методы аналогового моделирования являются мощной общепризнанной технологией изучения широкого круга проблем естествознания и техники. Теория квазианалогового моделирования, предложенная и развитая в работах академика Г.Е. Пухова и его последователей в 70-х годах прошлого века, существенно расширила использование методов математического и компьютерного моделирования [2-5, 22, 23]. Однако, дальнейшее развитие и использование моделирования для решения задач анализа, синтеза, управления динамическими системами существенно сдерживалось рядом особенностей элементной базы аналоговой вычислительной техники и самим способом представления информации в аналоговых структурах. Среди этих ограничений следует упомянуть следующие:

- низкая точность задания параметров аналоговых решающих элементов;
- существенная зависимость параметров от различных внешних факторов (изменение температуры, влияние электромагнитных наводок и шумов и т.п.);
- низкая степень автоматизации ввода исходных данных, набора топологии моделирующей среды и вывода результатов моделирования.

Разработка аналого-цифровых вычислительных машин и комплексов частично ослабляла негативное влияние указанных факторов на совершенство аналогового подхода, однако, радикально не решала проблемы существенного совершенствования аналоговых структур.

В настоящее время благодаря развитию мощных программных комплексов моделирования и, в особенности, методов графического и визуального программирования [1, 16, 27], предоставляющих разработчику моделирующих структур виртуальный инструментарий и широкий набор виртуальных решающих блоков, появилась реальная возможность снова обратиться к технологии аналогового и квазианалогового моделирования.

Данный раздел построен следующим образом. Рассмотрены свойства обратимых и необратимых решающих элементов моделирующих структур, реализующих математические выражения в явной и неявной формах. Далее кратко описан ряд виртуальных решающих блоков графических приложений комплекса MATLAB/Simulink/SimPowerSystem и их назначение, разработаны структуры виртуальных моделирующих цепей для решения систем линейных

алгебраических уравнений, неравенств и задач линейного программирования. Рассмотрение проводится на простых иллюстративных примерах.

5.1. Обратимость решающих элементов

Функциональные блоки моделирующих систем подразделяются на обратимые и необратимые в зависимости от того, могут ли они преобразовывать информацию только в прямом направлении (от входа к выходу) или способны выполнять как прямые, так и обратные операции. Так, например, умножитель, имеющий два входа для операндов-сомножителей и один выход для произведения, является необратимым решающим блоком, так как невозможно задать произведение и один из сомножителей и определить значение второго сомножителя. Таким же необратимым решающим блоком является классический операционный усилитель. Напряжение на выходе усилителя определяется напряжениями на его входах и передаточными характеристиками по каждому из входов. Для необратимых решающих блоков характерна явная форма задания математической операции, выполняемой блоком:

$$z = f(x_1, x_2, \dots, x_m), \quad (5.1)$$

где: z – результат выполнения математической операции (выходной сигнал решающего блока, $x_i (i := 1, \dots, m)$ - операнды (входные сигналы решающего блока). Неявная форма задания математической операции характерна для обратимых решающих блоков:

$$\varphi(x_1, x_2, \dots, x_m) = 0. \quad (5.2)$$

В выражении (5.2) все сигналы равноправны, а само выражение является функциональным ограничением на значения входящих в него сигналов. Обратимые решающие блоки, реализующие выражения типа (5.2), фактически являются моделями функциональных ограничений и очень удобны для построения моделей сложных систем по участкам, т.е. с использованием методов и средств диакоптики. Приложение Simulink системы MATLAB в основном ориентировано на решение задач теории автоматического управления и содержит виртуальные аналоги необратимых решающих блоков. Использование другого приложения MATLAB – SimPowerSystem в сочетании с методами квазианалогового моделирования [24, 25] позволяет построить обратимые решающие блоки и обратимые моделирующие структуры на их основе.

5.2. Виртуальные решающие элементы системы MATLAB/Simulink/SimPowerSystem

Библиотека виртуальных решающих элементов графических приложений системы MATLAB включает в себя чрезвычайно большое количество элементов. Часть таких элементов описывалась в разделе 1. Для удобства рассмотрения приведем и прокомментируем свойства элементов, которые потребуются для решения поставленных в работе задач. На рис. 5.1 приведена совокупность используемых элементов. Необходимо отметить, что форматы задания сигналов в приложениях Simulink и SimPowerSystem отличаются друг от друга, что не позволяет соединять блоки различных приложений друг с другом без согласования. Пиктограммы содержат обозначения входных сигналов блоков Simulink в виде стрелки «>», тогда как полюса блоков приложения SimPowerSystem обозначены в виде квадратов «□». Для согласования сигналов этих приложений некоторые блоки приложения SimPowerSystem содержат полюса, характерные для каждого из приложений в отдельности: «>» и «□».

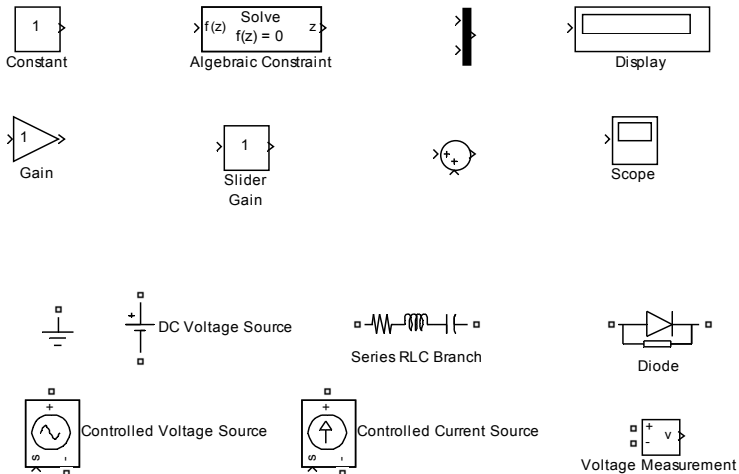


Рис. 5.1. Совокупность используемых элементов

Приложение Simulink

- Источник постоянного сигнала (Constant) - предназначен для введения сигнала, не меняющегося в процессе

моделирования. Величина сигнала задается с помощью меню элемента.

- Масштабный преобразователь (Gain) – изменяет масштаб сигналов, поступающих на его вход. Коэффициент преобразования задается с помощью меню элемента.
- Скользящий масштабный преобразователь (Slider Gain) – изменяет масштаб сигналов, поступающих на его вход. Аналог ползункового регулятора. Допускает изменение масштаба сигнала в процессе моделирования. Коэффициент преобразования задается с помощью меню элемента или вручную мышью.
- Сумматор (элемент без надписи, 3-й слева во 2-й строке рис. 5.1) – производит алгебраическое суммирование входных сигналов. Количество входных сигналов и их знаки (+, -) задаются с помощью меню элемента.
- Алгебраическое ограничение (Algebraic Constraint) – обеспечивает нулевой сигнал на входе путем изменения выходной величины z таким образом, чтобы $f(z)$ равнялось нулю. Для этого необходимо иметь канал обратной связи, чтобы выходной сигнал мог изменять входной с заданным функциональным ограничением. Фактически, этот элемент является виртуальным аналогом операционного усилителя с сигналами стандарта Simulink.
- Мультиплексор (Mux), (элемент без надписи, 3-й слева в 1-й строке рис. 5.1) – мультиплексирует (смешивает) сигналы. Количество входов задается с помощью меню элемента. Этот элемент, будучи подключенным к цифровому дисплею или виртуальному осциллографу, превращает эти индикаторы в многоканальные.
- Цифровой регистратор (Display) – отображает численные значения сигнала в случае их установления (в результате окончания процесса моделирования).
- Виртуальный осциллограф (Scope) – отображает форму сигналов, изменяющихся во времени. Параметры задаются с помощью меню элемента.

Приложение SimPowerSystem

- Виртуальный источник электродвижущей силы (ЭДС) (DC Voltage Source) – задает значение и полярность э.д.с. постоянного тока.
- Виртуальная ветвь электрической цепи в виде последовательного соединения резистора, конденсатора и катушки индуктивности. Задание параметров ветви и ее

топологии (частных случаев) обеспечивается с помощью меню (окна параметров).

- Виртуальный диод (Diode) – аналог диода. Падение напряжения в проводящем направлении и обратное сопротивление в запертом состоянии задаются с помощью меню элемента и могут изменяться в широких пределах. По умолчанию падение напряжения на диоде в проводящем состоянии равно 800 мВ (аналог кремниевого полупроводникового диода).
- Управляемый источник напряжения (Controlled Voltage Source) – Виртуальный источник электродвижущей силы, значение э.д.с. которого определяется сигналом формата Simulink. Подходит для обеспечения связи элементов приложений Simulink и SimPowerSystem.
- Управляемый источник тока (Controlled Current Source) – виртуальный источник тока, величина тока которого определяется сигналом формата Simulink. Подходит для обеспечения связи элементов приложений Simulink и SimPowerSystem.
- Измеритель напряжения (Voltage Measurement) – виртуальный вольтметр приложения SimPowerSystem. Кроме измерения сигналов в системе SimPowerSystem может быть использован для обеспечения связи элементов приложений Simulink и SimPowerSystem.

5.3. Необратимый линейный преобразователь в системе Simulink

Ограничимся линейным преобразованием с 4 сигналами: $a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = 0$. Обобщение на большее число сигналов является тривиальной задачей. Структурная схема преобразователя в среде Simulink приведена на рис. 5.2.

Схема состоит из 4 масштабных преобразователей, сумматора и элемента алгебраического ограничения. Параметры масштабных преобразователей соответствуют коэффициентам линейного преобразования. Для корректной работы преобразователя необходимо определить, какой из 4 сигналов является искомым. Пусть для определенности это будет x_3 . Следующим этапом необходимо соединить выход (Out1) со входом (In3). На остальные три входа должны быть поданы значения сигналов x_1, x_2, x_4 . В результате работы системы будет реализовано следующее линейное

преобразование: $x_4 = -\frac{1}{a_4}(a_1x_1 + a_2x_2 + a_3x_3)$. В этой структурной

схеме элемент алгебраического ограничения выполняет роль виртуального операционного усилителя, формирующего такой выходной сигнал, который обеспечивает на его входе нулевой сигнал. Если такую схему свернуть в подсхему, то из таких подсхем можно строить необратимые модели систем линейных алгебраических уравнений (по классификации квазианалоговых моделей – α -аналоги системы линейных алгебраических уравнений).

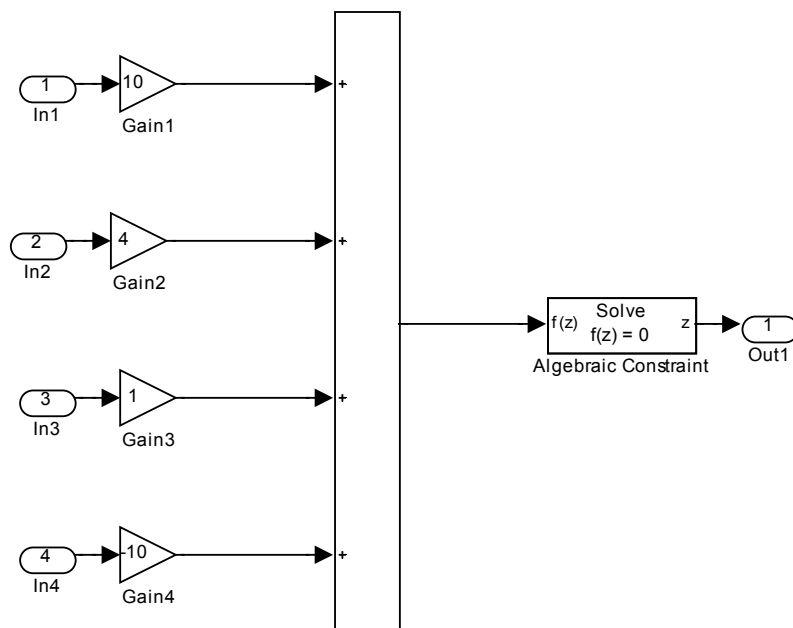


Рис. 5.2. Структурная схема модели линейного преобразователя

5.4. α -аналоговая модель системы линейных алгебраических уравнений

Пример 5.1. Из подсхем типа рис. 5.2 можно построить модель системы линейных алгебраических уравнений 3 порядка, поскольку один из входов будет использован для задания правой части соответствующего уравнения. Структурная схема модели системы

линейных алгебраических уравнений с 3 неизвестными приведена на рис. 5.3. В ее состав входят три подсистемы рис. 5.2, источник постоянного сигнала, мультиплексор и цифровой регистратор (дисплей). Одноименные полюсы подсистем объединены и подключены к выходам соответствующих подсистем, формирующих значения неизвестных x_1, x_2, x_3 , объединенные четвертые полюсы подключены к источнику постоянного сигнала. Описанная структурная схема позволяет определять решения системы линейных алгебраических уравнений лишь в случае, когда матрица коэффициентов системы является положительно определенной. В данном примере параметры схемы соответствовали следующей системе уравнений:

$$10x_1 + 4x_2 + x_3 = 10,$$

$$4x_1 + 10x_2 + x_3 = -29,$$

$$x_1 + 4x_2 + 10x_3 = -3.5.$$

На цифровом регистраторе рис. 5.3 показано решение системы уравнений ($x_1 = 2.5$, $x_2 = -4$, $x_3 = 1$).

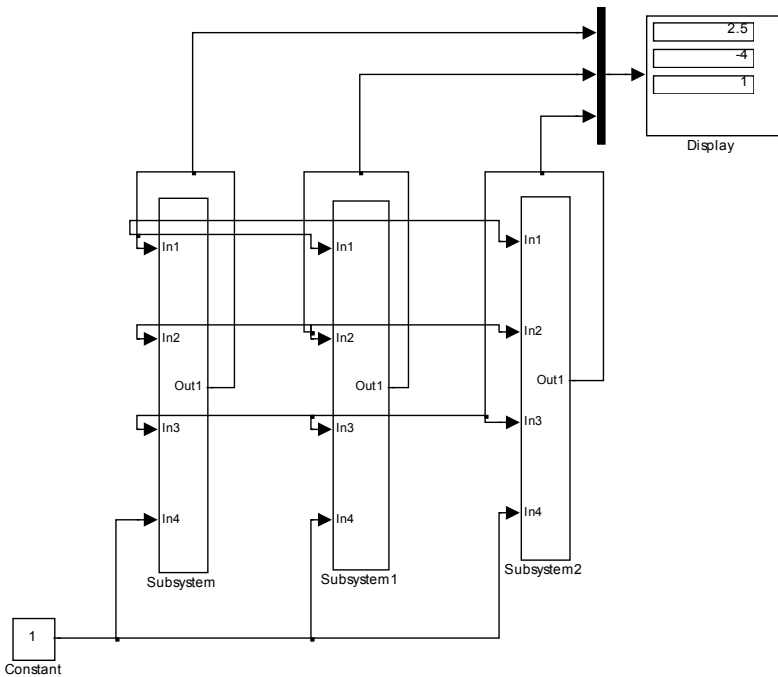


Рис. 5.3. Структурная схема α -аналоговой модели системы линейных алгебраических уравнений 3 порядка.

5.5. Виртуальный аналог операционного усилителя

В библиотеках приложений Simulink и SimPowerSystem, к сожалению, нет виртуального аналога операционного усилителя, без которого нельзя сконструировать обратимые квазианалоговые модели. Такой виртуальный аналог можно построить, используя элемент алгебраического ограничения приложения Simulink и виртуальные измеритель напряжения и управляемый источник напряжения приложения SimPowerSystem. Структурная схема такого виртуального операционного усилителя приведена на рис. 5.4. Измеритель напряжения и управляемый источник напряжения фактически являются согласующими элементами, обеспечивающими совместимость решающих блоков Simulink в среде SimPowerSystem. При включении структурной схемы рис. 5.4 с соответствующими цепями обратной связи на выходном полюсе Conn2 формируется такой сигнал, при котором сигнал на входном полюсе Conn1 равен нулю. Структурная схема рис. 5.4 будет свернута в подсхему, используемую в последующих структурах обратимых квазианалоговых моделей. Рассмотрим структуру обратимого квазианалогового преобразователя p – типа [23, 25].

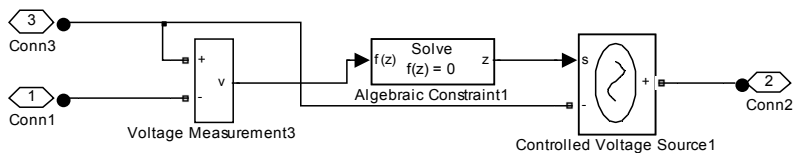


Рис. 5.4. Виртуальный аналог операционного усилителя.

5.6. Обратимый линейный преобразователь

Структура p -аналогового обратимого линейного преобразователя приведена на рис. 5.5. Подсистема 1 является виртуальным аналогом операционного усилителя, рассмотренного выше. Полюс Conn1 должен быть заземлен, полюса Conn2-Conn5 соответствуют сигналам, ограниченным соотношением:

$$a_2 x_2 + a_3 x_3 + a_5 x_5 = 0 .$$

Коэффициенты указанного соотношения устанавливаются заданием проводимостей верхнего ряда резисторов схемы рис. 5.5.

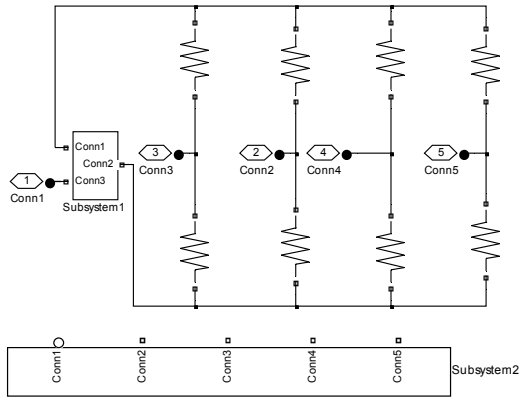


Рис. 5.5. Структурная схема обратимого линейного преобразователя и его представление в виде подсистемы.

5.7. Обратимая модель системы линейных алгебраических уравнений

Пример 5.2. Рассматривая структурную схему обратимого линейного преобразователя рис. 5.5 как модель линейного алгебраического уравнения, можно построить модель системы линейных алгебраических уравнений путем объединения нескольких обратимых линейных преобразователей. На рис. 5.6 представлена модель системы линейных алгебраических уравнений 3 порядка. Параметры обратимых линейных преобразователей (ОЛП), обозначенные на рис. 5.6, как Subsystem1, Subsystem2, Subsystem3, устанавливаются пропорциональными коэффициентам уравнений и их правым частям. Одноименные полюса обратимых линейных преобразователей объединяются, чтобы обеспечить совместность ограничений, накладываемых каждым ОЛП. Полюса Conn1 заземляются, а виртуальный источник э.д.с. подключается к объединенному полюсу Conn5. Виртуальные вольтметры, мультимплексор и цифровой регистратор обеспечивают измерение и вывод значений неизвестных системы уравнений. В данном примере решалась следующая система уравнений:

$$2x_1 + 5x_2 + x_3 = 11,$$

$$4x_1 + x_2 + 3x_3 = 7,$$

$$x_1 + x_2 + x_3 = 4.$$

Неизвестные системы уравнений, указанные на цифровом регистраторе, равны: $x_1 = -1$, $x_2 = 2$, $x_3 = 3$.

Continuous
pow ergui

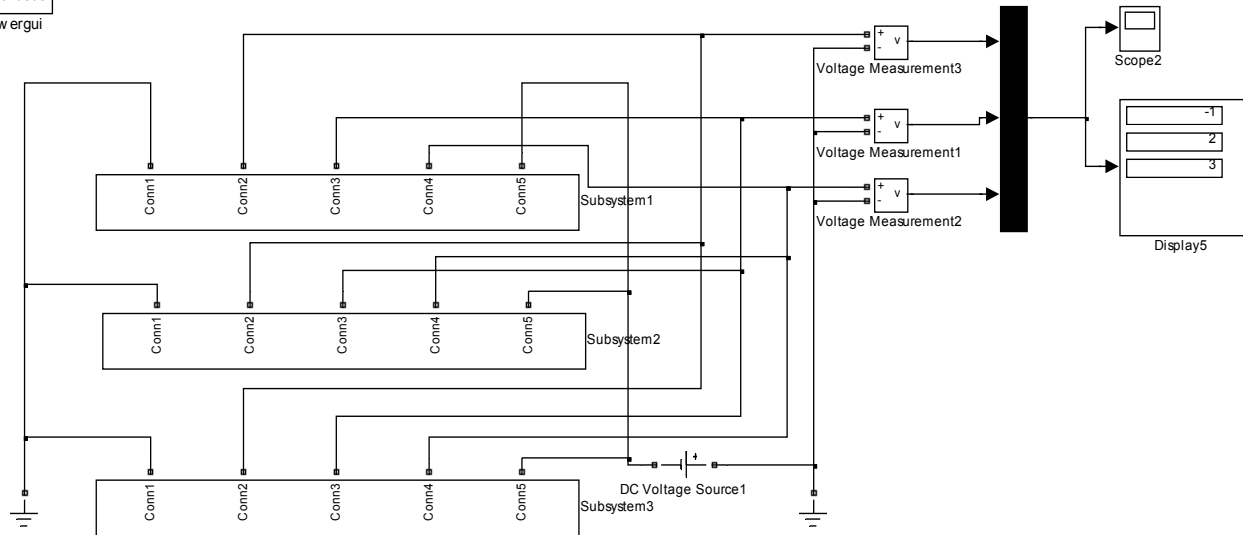


Рис.5.6. Структурная схема обратимой модели системы линейных алгебраических уравнений

5.8. Обратимая модель системы линейных неравенств

Рассмотренные выше структуры обратимых линейных преобразователей могут быть положены в основу построения обратимой модели системы линейных неравенств. Ограничимся случаем системы трех линейных неравенств, зависящих от двух переменных вида:

$$a_{11}x_1 + a_{12}x_2 \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 \leq b_2,$$

$$a_{31}x_1 + a_{32}x_2 \geq b_3.$$

Известно, что любая система линейных неравенств может быть сведена к эквивалентной системе линейных уравнений путем введения дополнительных переменных, на которые накладываются условия неотрицательности (неположительности). Преобразования приведенной системы неравенств приводят к следующим выражениям:

$$a_{11}x_1 + a_{12}x_2 + x_3 = b_1,$$

$$a_{21}x_1 + a_{22}x_2 + x_4 = b_2,$$

$$a_{31}x_1 + a_{32}x_2 + x_5 = b_3,$$

$$x_3 \geq 0, x_4 \geq 0, x_5 \leq 0.$$

Пример 5.3. Структурная схема модели обратимой модели системы линейных неравенств, приведенная на рис. 5.7, содержит три подсистемы обратимых линейных преобразователей (Subsystem, Subsystem1, Subsystem2), объединенных одноименными полюсами (Conn1, Conn2, Conn3, Conn5), которые реализуют модель системы трех линейных алгебраических уравнений с 5 переменными. Полюса подсистем Conn5 подключены к виртуальным диодам (Diode1, Diode2, Diode4), обеспечивающим неотрицательность дополнительных переменных x_3, x_4 и неположительность x_5 . Описанная методика может применяться и при построении обратимых моделей задач линейного программирования, к рассмотрению которых мы и переходим.

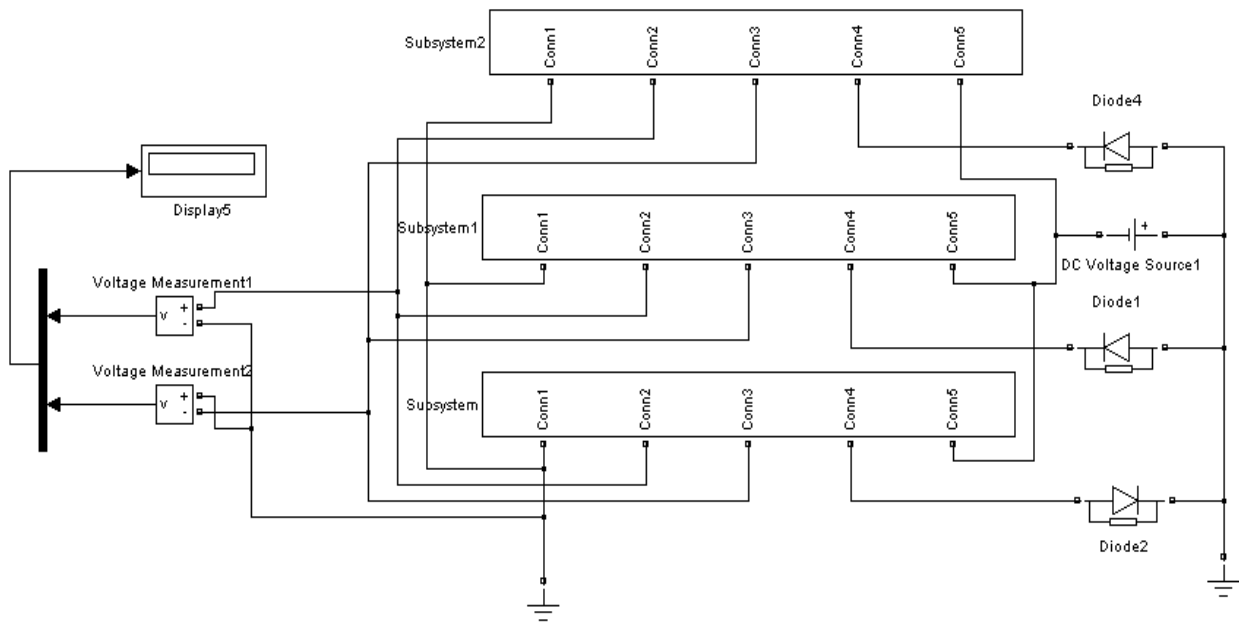


Рис. 5.7. Структурная схема обратимой модели системы линейных неравенств

5.9. Обратимая модель задачи линейного программирования

Пример 5.4. Рассмотрим следующую задачу линейного программирования: максимизировать целевую функцию $\mu = c_1x_1 + c_2x_2$, зависящую от переменных x_1, x_2 , удовлетворяющих следующей системе линейных неравенств:

$$a_{11}x_1 + a_{12}x_2 \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 \leq b_2,$$

$$x_1 \geq 0, x_2 \geq 0.$$

Структурная схема обратимой модели задачи линейного программирования изображена на рис. 5.8. Она содержит три обратимых линейных преобразователя, два из которых вместе с виртуальными диодами реализуют первые два неравенства задачи линейного программирования. Два последних неравенства (неотрицательности переменных) реализуются еще двумя виртуальными диодами (Diode2, Diode3), подключенными к полюсам ОЛП Conn2, Conn3. Максимизация целевой функции обеспечивается с помощью виртуального источника тока, управляемого сигналом константы.

Параметры структурной схемы, которые задавались в данном примере, соответствовали следующим значениям коэффициентов задачи линейного программирования: $a_{11} = 3$, $a_{12} = 2$, $a_{21} = 1$, $a_{22} = 4$, $b_1 = 6$, $b_2 = 4$, $c_1 = c_2 = 1$. Значения переменных, соответствующих оптимальному решению задачи линейного программирования ($x_1 = 1.6$, $x_2 = 0.6$, $-\mu = -2.2$), указаны на цифровом регистраторе. Для решения задачи минимизации целевой функции при тех же ограничениях достаточно изменить знак константы в параметрах источника постоянного сигнала (Constant), подключенного к управляющему входу виртуального источника тока (ControlledCurrentSource).

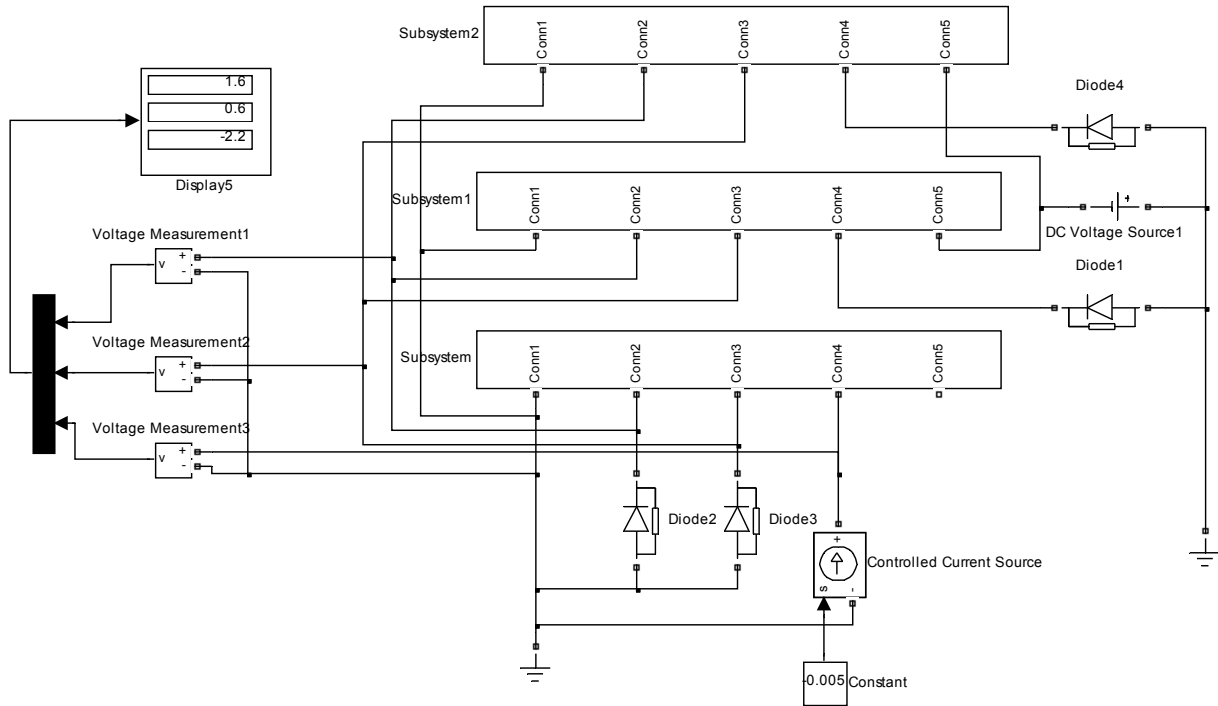


Рис. 5.8. Структурная схема обратимой модели задачи линейного программирования

Рассмотренные методы реализации обратимых решающих элементов в приложениях визуального программирования системы MATLAB позволяют использовать весь арсенал методов квазианалогового моделирования для реализации компьютерных моделей широкого класса задач. Это оказалось возможным благодаря конструированию виртуального аналога операционного усилителя из функциональных блоков приложений Simulink и SimPowerSystem. Результаты компьютерных экспериментов с различными обратимыми моделями задач линейной алгебры и математического программирования показали возможность реализации динамического диапазона машинных переменных виртуальных моделей порядка 60 Дцб и эквивалентных коэффициентов усиления виртуальных аналогов усилителей без обратной связи порядка 10^{15} . Для выяснения динамических характеристик обратимых моделей необходимо проведение дополнительных исследований.

5.10. Модель транспортной задачи линейного программирования

Частным, но весьма распространенным видом задач линейного программирования, является транспортная задача. Классическая транспортная задача линейного программирования предусматривает нахождение оптимального плана перевозки однородного продукта в транспортной сети между пунктами производства и потребления продукта, при котором минимизируется стоимость транспортных издержек на перевозку. Математическая модель транспортной задачи имеет следующий вид [8, 10]. Пусть имеются m пунктов производства однородного продукта с объемами производства $a_i, i := 1, \dots, m$ и n пунктов потребления этого продукта с объемами потребления $b_j, j := 1, \dots, n$. Пункты производства связаны с пунктами потребления транспортной сетью, образующей, в общем случае $m \times n$ маршрутов с заданными стоимостями перевозки единицы продукта c_{ij} по маршруту $i-j$. Если обозначить неизвестные объемы перевозок по маршруту $i-j$ как x_{ij} , математическая модель транспортной задачи будет иметь вид:

$$\sum_{j=1}^n x_{ij} = a_i, i := 1, \dots, m, \quad (5.3)$$

$$\sum_{i=1}^m x_{ij} = b_j, j := 1, \dots, n, \quad (5.4)$$

$$x_{ij} \geq 0, \quad (5.5)$$

$$\mu = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min . \quad (5.6)$$

Выражение (5.3) является условием баланса объемов производства и перевозок продукта. Аналогично, выражение (5.4) является условием баланса объемов перевозок и потребления. Условие (5.5) требует неотрицательности перевозок (нецелесообразность перевозок из пунктов потребления в пункты производства). Общие транспортные издержки (стоимость перевозки) определяется выражением (5.6) и должна минимизироваться выбором объемов перевозок по различным маршрутам.

Известна аналоговая электрическая модель транспортной задачи линейного программирования [10] в виде цепи постоянного тока, содержащей идеальные источники напряжения, тока и диоды. Топология цепи повторяет топологию транспортной сети, моделями маршрутов являются ветви цепи из последовательно соединенных диодов и источников напряжения, источники тока используются для моделирования пунктов производства и потребления продукта. Уравнения (5.3) и (5.4) воспроизводятся как уравнения первого закона Кирхгофа электрической цепи-модели, условия неотрицательности (5.5) реализуются односторонней проводимостью диодов. В цепи устанавливается такое распределение токов, моделирующих объемы перевозок, при котором минимальна мощность, рассеиваемая в модели транспортной сети [7]. Таким образом реализуется минимум выражения (5.6) и получается оптимальное решение транспортной задачи.

Для построения виртуальной модели транспортной задачи в среде визуального программирования необходимо сконструировать структурные схемы источников тока, моделирующих пункты производства и потребления продукта, а также структурную схему модели ветви. Далее, можно используя эти структурные схемы в качестве подсистем, синтезировать структурную схему виртуальной модели транспортной задачи в целом.

Структурная схема виртуальной модели пункта производства и его изображение в виде подсхемы показаны на рис. 5.9.

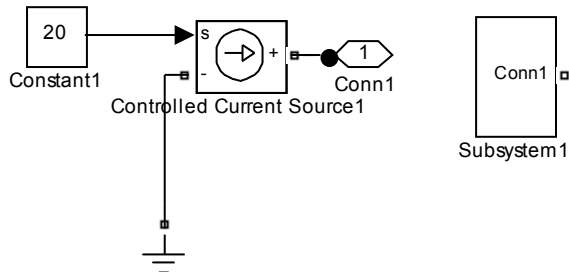


Рис. 5.9. Модель пункта производства.

Схема состоит из управляемого источника тока (Controlled Current Source1) приложения SimPowerSystems, значение тока которого задается блоком константы (Constant1) приложения Simulink. Отрицательный полюс источника тока заземлен, а положительный подключен к выходному полюсу (Conn1). Значение константы, устанавливаемое в окне параметров блока Constant1, определяется величиной объема производства пункта производства, моделью которого является Subsystem1.

Структурная схема виртуальной модели пункта потребления строится аналогично и приведена на рис.5.10.

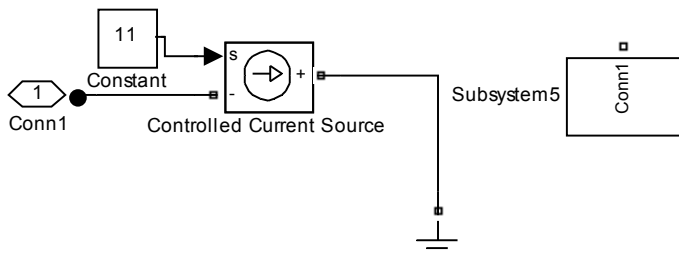


Рис. 5.10. Модель пункта потребления.

Структурная схема виртуальной модели ветви транспортной сети и ее изображение в виде подсхемы приведены на рис. 5.11.

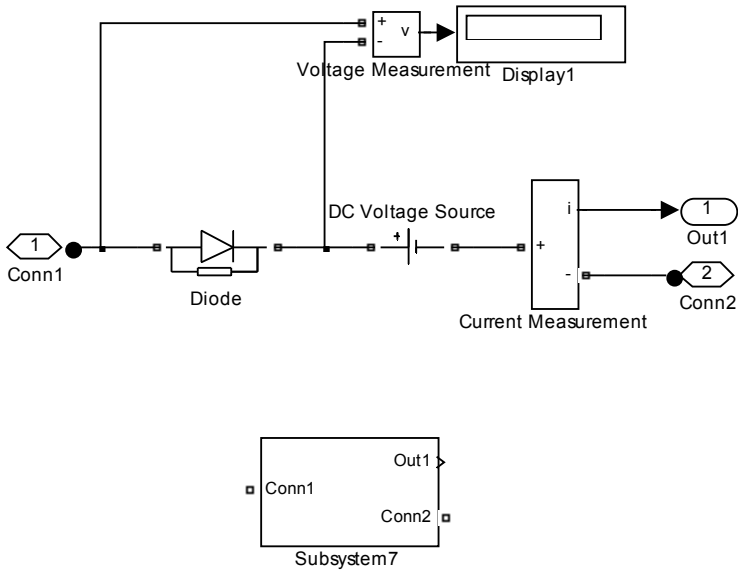


Рис. 5.11. Модель ветви транспортной сети.

Основу модели ветви составляют последовательное соединение виртуального диода (Diode) и виртуального источника электродвижущей силы постоянного тока (DC Voltage Source). Анод виртуального диода соединен с входным полюсом виртуальной модели ветви (Conn1), а отрицательный полюс виртуального источника э.д.с. через виртуальный измеритель тока (Current Measurement) соединен с выходным полюсом модели ветви (Conn2). Выходной полюс Out1 виртуального измерителя тока формирует сигнал стандарта Simulink и может подключаться к цифровым регистраторам этого приложения непосредственно или через мультиплексор Mux. Виртуальный вольтметр, образованный виртуальным измерителем напряжения Voltage Measurement и цифровым регистратором Display1, не являются обязательными элементами модели ветви и предназначены для оценивания состояния виртуального диода (открытого или закрытого). Входные полюса моделей ветвей транспортной сети (Conn1) подключаются к выходным полюсам (Conn1) моделей пунктов производства, а выходные полюса моделей ветвей транспортной сети (Conn2) подключаются к входным полюсам (Conn1) моделей пунктов потребления. Порядок построения структурной схемы виртуальной модели транспортной задачи рассмотрим на примере.

Пример 5.5. Дана следующая транспортная задача линейного программирования. Найти оптимальный вариант перевозки цемента,

производимого двумя цементными заводами на три стройки, расположенными в том же районе. Производительности цементных заводов составляют соответственно 200 и 100 тонн в сутки. Потребности строящихся объектов составляют 110, 140 и 50 тонн в сутки. Стоимости перевозки 1 тонны цемента для возможных 6 маршрутов в условных единицах и объемы цемента, подлежащего перевозке, приведены в таблице.

Таблица

	Об-кт № 1	Об-кт № 2	Об-кт № 3
З-д № 1	30	10	30
З-д № 2	30	50	60

На рис. 5.12 приведена структурная схема виртуальной модели транспортной задачи, построенная из элементов, рассмотренных выше (рис. 5. 9 – 5.11). Виртуальные модели заводов № 1 и № 2 представлены элементами Subsystem1 и Subsystem2. Виртуальные модели строительных объектов № 1, №3, представлены элементами Subsystem5 и Subsystem10. Так как задача является сбалансированной (суммарный объем производства цемента равен суммарной потребности строительных объектов), модели одного из объектов воспроизводить не требуется. В нашем случае выбран объект № 2. Модели различных маршрутов представлены на схеме следующими подсистемами:

Для завода № 1 – Subsystem7, Subsystem3 и Subsystem8,

Для завода № 2 – Subsystem4, Subsystem6 и Subsystem9.

Виртуальные токи, изображающие объемы перевозок в оптимальном варианте, оцениваются цифровым регистратором Display1, на который поступают сигналы от виртуальных моделей ветвей через мультиплексор.

Значения объемов производства и потребления продукта задавались в окнах параметров блоков Constant виртуальных моделей пунктов производства и потребления продукта (Subsystem1, 2, 5, 10) в масштабе 0.1. Величины стоимости перевозки единицы продукта по различным маршрутам задавались в окнах параметров блоков DC Voltage Source виртуальных моделей ветвей транспортной сети (Subsystem7, 3, 8, 4, 6, 9) также в масштабе 0.1. Результаты моделирования, приведенные на экране цифрового регистратора имеют вид:

$$x_{11} = 1 \quad x_{12} = 14 \quad x_{13} = 10$$

$$x_{21} \approx 0 \quad x_{22} = 5 \quad x_{23} \approx 0$$

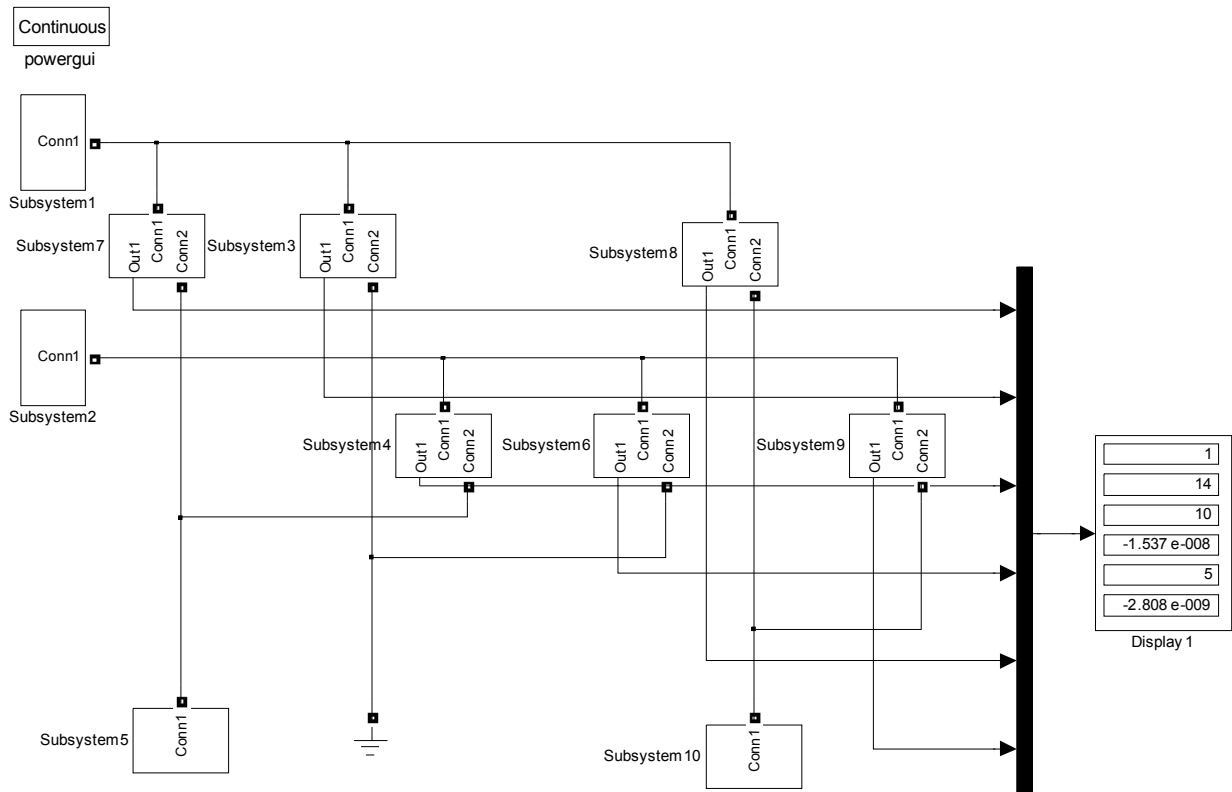


Рис.5.12. Структурная схема модели транспортной задачи.

6. УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

При самостоятельном выполнении упражнений, приведенных в данном разделе, могут оказаться полезными следующие рекомендации и порядок работы над задачами:

- После загрузки системы MATLAB открыть приложение Simulink.
- В открывшемся окне Simulink Library Browser мышью выбрать File->New->Model, либо символ нового файла, либо нажать клавиши Ctrl+N. В результате на экране появится изображение рабочей области, в которой необходимо конструировать структурную схему модели решаемой задачи. Эта рабочая область может быть сохранена как файл с именем, выбираемым пользователем.
- Перенести мышью в рабочую область необходимые решающие элементы, источники сигналов и виртуальные измерительные приборы из библиотеки Simulink в открытом окне Simulink Library Browser.
- Скопировать в необходимом количестве элементы, которые встречаются более одного раза, и разместить их в рабочей области так, чтобы их было удобно соединять друг с другом виртуальными проводниками. Оптимальное размещение элементов является достаточно непростой задачей и зависит от опыта и умения исследователя.
- В случае необходимости можно изменить пространственную ориентацию блоков. Для этого необходимо разместить курсор мыши над изображением блока, выделить его однократным нажатием левой клавиши мыши и выбрать мышью команду Format->Rotate Block, либо одновременным нажатием клавиш Ctrl+R. Одна команда поворачивает блок на 90 градусов.
- Соединить блоки между собой в соответствии со структурной схемой решаемой задачи (сформировать структурную схему модели) с помощью мыши.
- Установить необходимые параметры решающих элементов и других блоков, входящих в структурную схему модели с помощью окна параметров блока, вызов которого производится путем двукратного нажатия на левую кнопку мыши, когда курсор находится в пределах изображения блока.
- В окне Simulation строки меню рабочей области выбрать команду Start. Запуск процесса моделирования можно также осуществить одновременным нажатием клавиш CTRL+T.

- Следует помнить, что виртуальные решающие блоки и элементы системы Simulink обладают и другими возможностями, использовать которые можно с помощью электронной системы Help или других учебных пособий по системе MATLAB / Simulink.

6.1. Сформировать и визуализировать сигналы заданной формы.

Упражнение 1

- $y = (1 + 2 \sin(2t))^2$

Указание. Использовать блоки: Constant, Integrator, Gain, Trigonometric Function, Math Function, Sum, Scope,.

Упражнение 2

- $y = \sqrt{1 + 0.5 \sin(2t)}$

Указание. Использовать те же блоки, что и в упр. 1, перестроив блок Math Function.

Упражнение 3

- $y = t \cdot e^{-t^2} \cdot \cos(2\pi t)$

Указание. Использовать блоки: Constant, Integrator, Gain, Trigonometric Function, Math Function, Product, Scope.

Упражнение 4

- $y = 3.5 + 0.3t - 0.06t^2 - \sqrt{e^{-2t} + t^2}$

Указание. Использовать в необходимых количествах блоки: Constant, Integrator, Gain, Math Function, Sum, Scope.

Упражнение 5

- $y = \min(5t, 100 - 2t^2)$

Указание. Использовать блоки: Constant, Integrator, Gain, Math Function, Sum, MinMax, Scope.

Упражнение 6

- $y = \sqrt{|-100 + 20t|}$

Указание. Использовать блоки: Constant, Integrator, Gain, Abs, Math Function, Scope.

6.2. Сформировать структурные схемы.

Упражнение 7. Сформировать структурную схему системы, генерирующую фигуру эллипса на экране виртуального двухкоординатного регистратора XY Graph на основе параметрического задания уравнения эллипса:

$$\begin{aligned}x(t) &= A \cdot \sin(\omega t), \\y(t) &= B \cdot \cos(\omega t).\end{aligned}$$

при следующих значениях параметров: $A = 5$, $B = 2$, $\omega = \pi/5$.

Указание. Использовать блоки: Ramp, Gain, Trigonometric Function, XY Graph, установить необходимые масштабы по осям двухкоординатного регистратора.

Упражнение 8. Сформировать структурную схему системы, генерирующую фигуру спирали на экране виртуального двухкоординатного регистратора XY Graph на основе параметрического задания математической модели спирали:

$$\begin{aligned}x(t) &= (10 - t) \cdot \sin(0.4\pi t), \\y(t) &= (10 - t) \cdot \cos(0.4\pi t).\end{aligned}$$

Указание. Использовать блоки: Constant, Integrator, Gain, Trigonometric Function, Product, Scope, XY Graph, установить необходимые масштабы по осям двухкоординатного регистратора.

Упражнение 9. Построить структурную схему генератора системы базисных функций $s_k(t) = \{e^{-kt}\}_{k=0}^4$.

Указание. Использовать блоки: Constant, Integrator, Gain, Math Function, Mux, Scope.

Упражнение 10. Построить структурную схему генератора системы базисных функций $s_k(t) = \{\sin kt, \cos kt\}_{k=0}^4$.

Указание. Использовать блоки: Constant, Integrator, Gain, Trigonometric Function, Mux, Scope.

Упражнение 11. Построить структурную схему генератора системы базисных функций $s_k(t) = \{1, t, \sin t\}$.

Указание. Использовать блоки: Constant, Integrator, Gain, Trigonometric Function, Mux, Scope.

6.3. Построить виртуальные модели.

Упражнение 12. Синтезировать структурную схему модели системы линейных алгебраических уравнений:

$$\begin{cases} 2x_1 - x_2 - 6.5 = 0 \\ -x_1 + 2x_2 - x_3 - 1 = 0 \\ -x_2 + 2x_3 - 0.5 = 0 \end{cases}$$

Указание. Использовать блоки: Constant, Gain, Sum, Algebraic Constraint, Mux, Display.

Упражнение 13. Синтезировать структурную схему модели дифференциального уравнения 3 порядка:

$$\frac{d^3y}{dx^3} + 0.5 \frac{d^2y}{dx^2} + 0.9 \frac{dy}{dx} + 2y = 0.$$

с начальными условиями:

$$y(0) = 2, y'(0) = -1, y''(0) = 0.$$

Построить фазовый портрет системы.

Указание. Использовать блоки: Integrator, Gain, Sum, Mux, Display, XY Graph. Выбрать диапазон изменения переменных для XY Graph с учетом амплитуд решения и его производной.

Упражнение 14. Синтезировать структурную схему модели дифференциального уравнения 2 порядка:

$$\frac{d^2y}{dt^2} + 0.1 \frac{dy}{dt} + 0.5y = 1$$

с начальными условиями:

$$y(0) = 1, y'(0) = -1.$$

Построить фазовый портрет системы. Визуализировать функцию и первую производную решения.

Указание. Использовать блоки: Integrator, Gain, Constant, Sum, Mux, Display, XY Graph. Выбрать диапазон изменения переменных для XY Graph с учетом амплитуд решения и его производной.

В следующем разделе приведены структурные схемы рассмотренных в данном пособии иллюстративных примеров и результаты моделирования, представленные на экранах виртуальных измерительных приборов. При самостоятельном выполнении упражнений сначала рекомендуется составить и отладить структурную схему модели, и лишь потом обратиться к помощи.

7. ИЛЛЮСТРАЦИИ К УПРАЖНЕНИЯМ

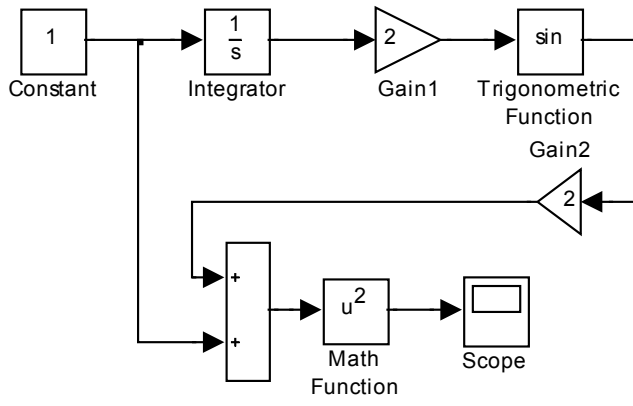


Рис. 7.1. Схема к упражнению 1.

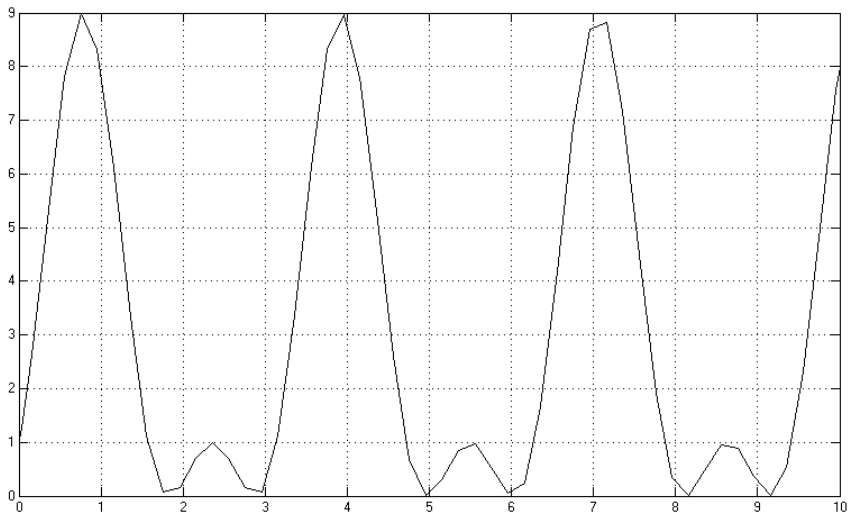


Рис. 7.2. Решение упражнения 1 на осциллографе.

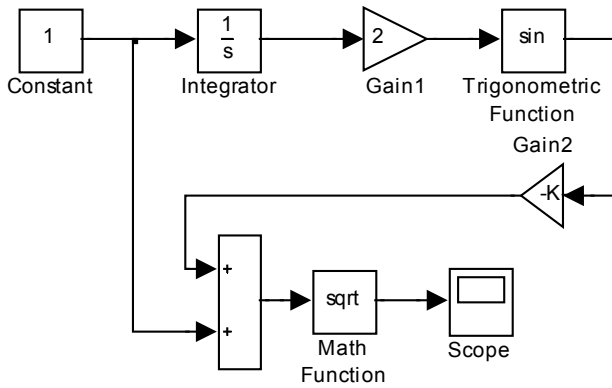


Рис. 7.3. Схема к упражнению 2.

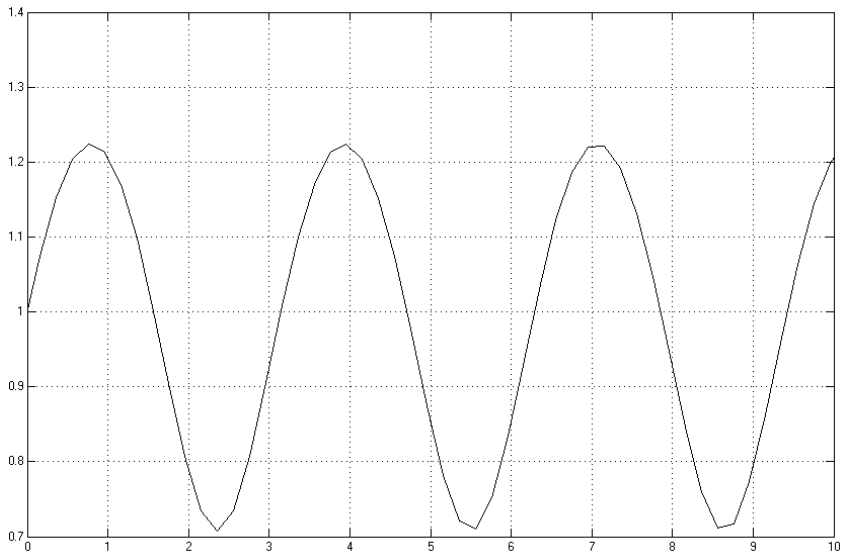


Рис. 7.4. Решение упражнения 2 на осциллографе.

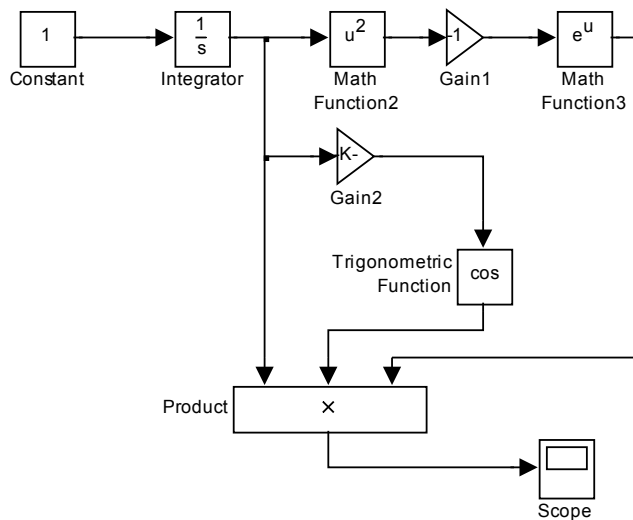


Рис. 7.5. Схема к упражнению 3.

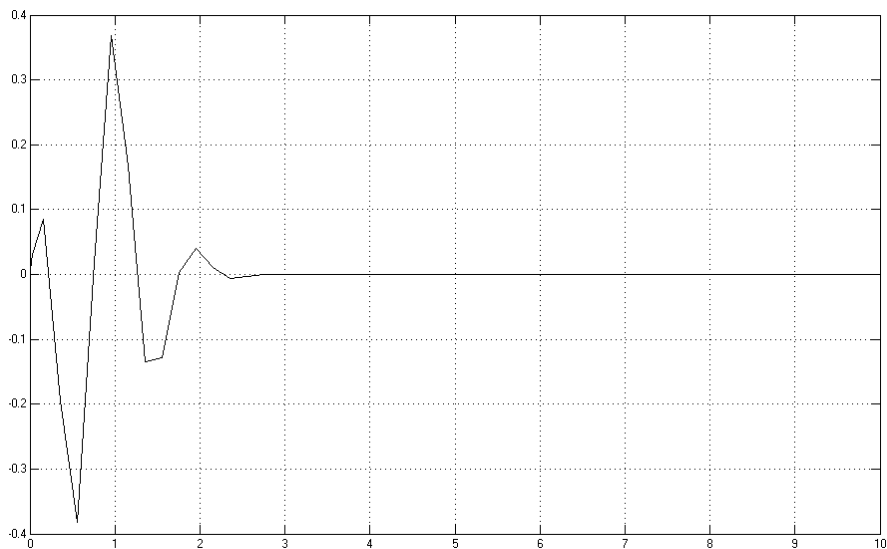


Рис. 7.6. Решение упражнения 3 на осциллографе.

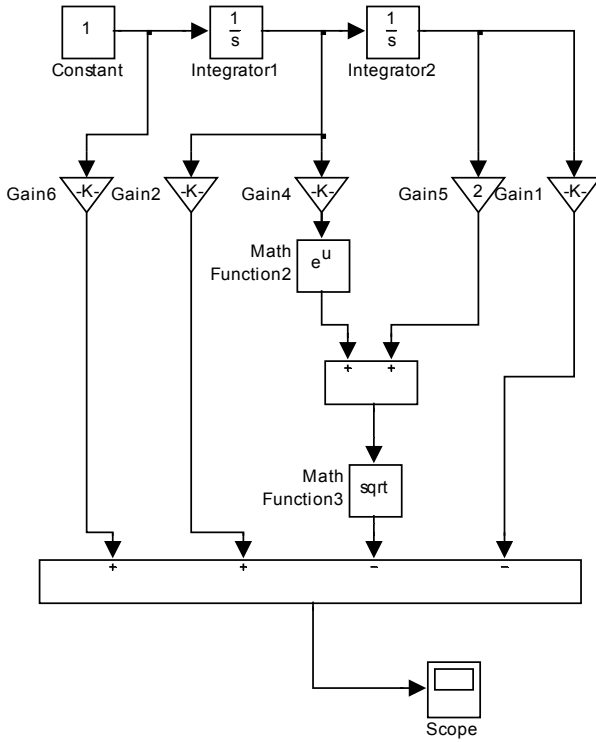


Рис. 7.7. Схема к упражнению 4.

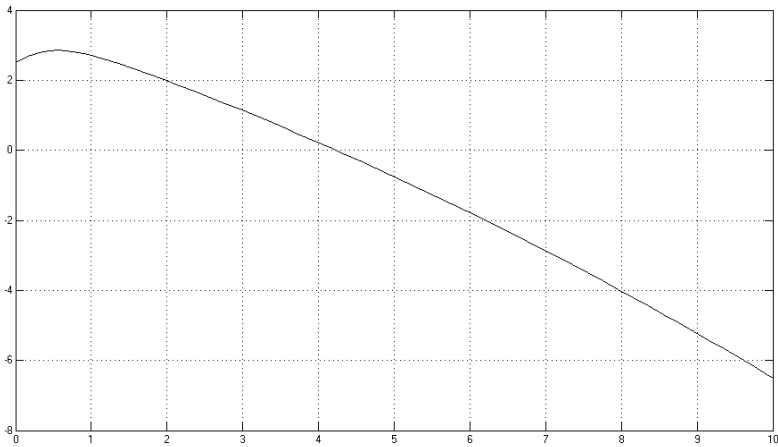


Рис. 7.8. Решение упражнения 4 на осциллографе.

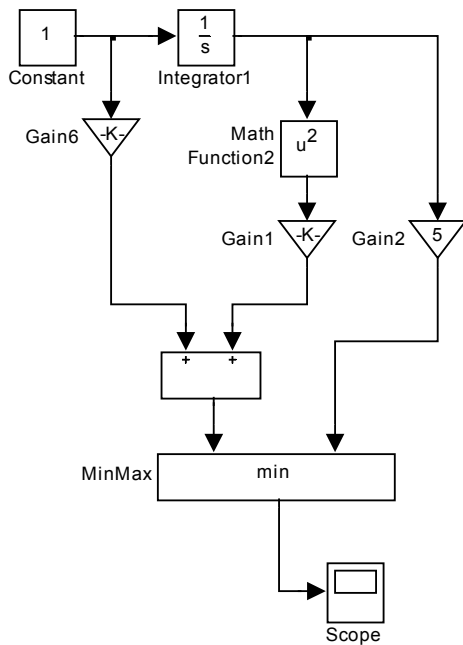


Рис. 7.9. Схема к упражнению 5.

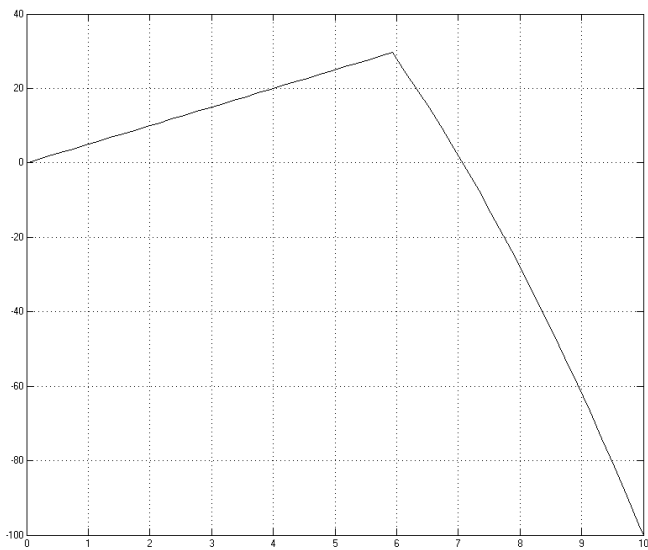


Рис. 7.10. Решение упражнения 5 на осциллографе.

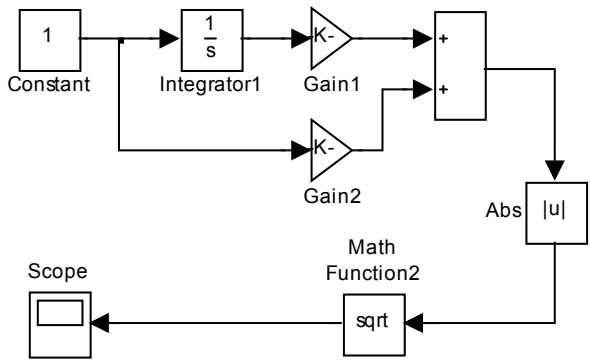


Рис. 7.11. Схема к упражнению 6.

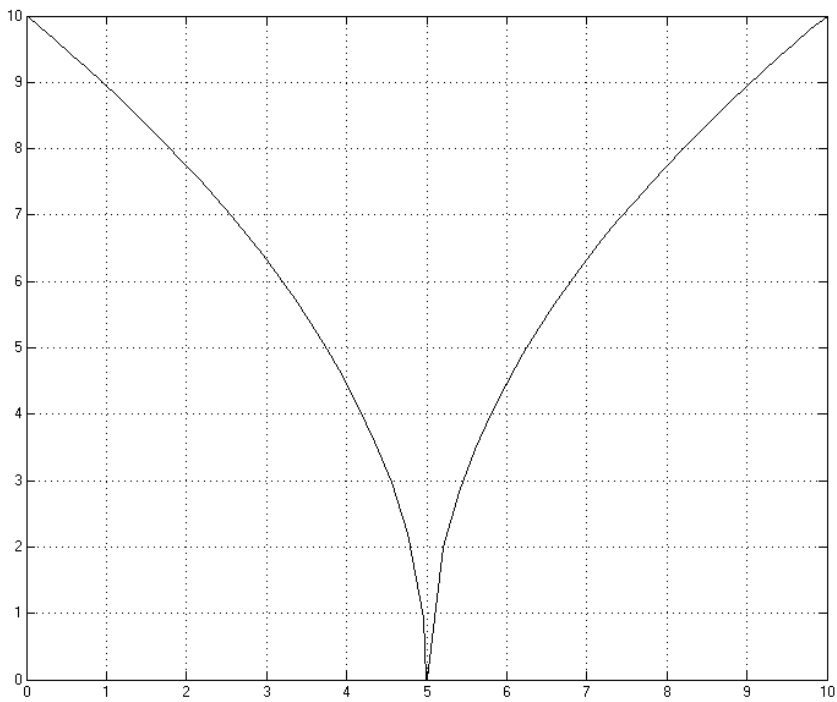


Рис. 7.12. Решение упражнения 6 на осциллографе.

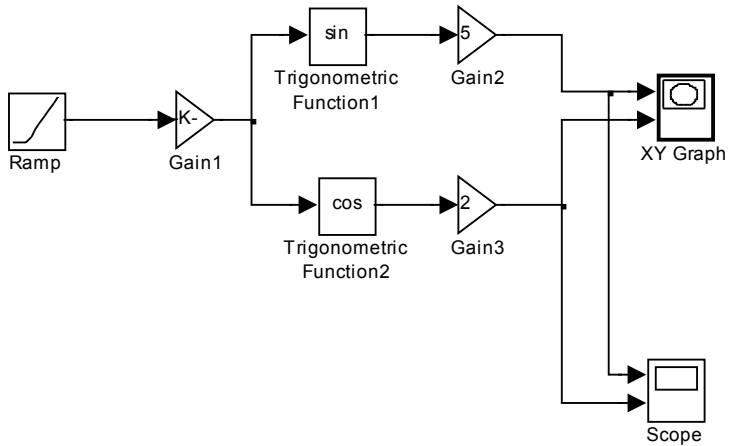


Рис. 7.13. Схема к упражнению 7.

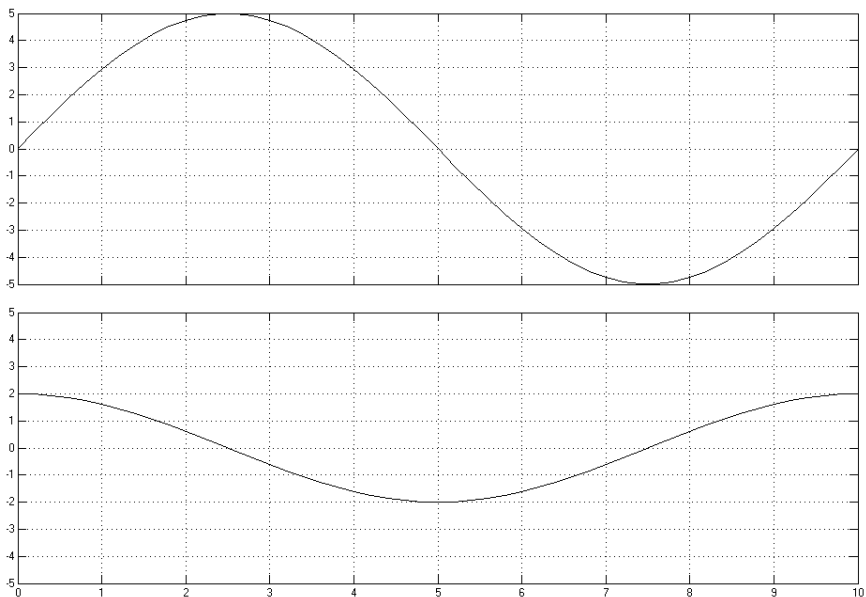


Рис. 7.14. Решение упражнения 7 на осциллографе.

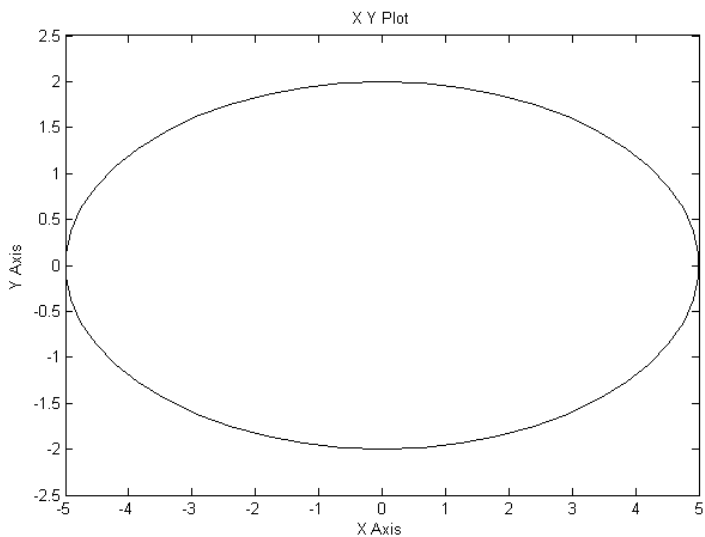


Рис. 7.15. Фазовый портрет системы (упражнение 7).

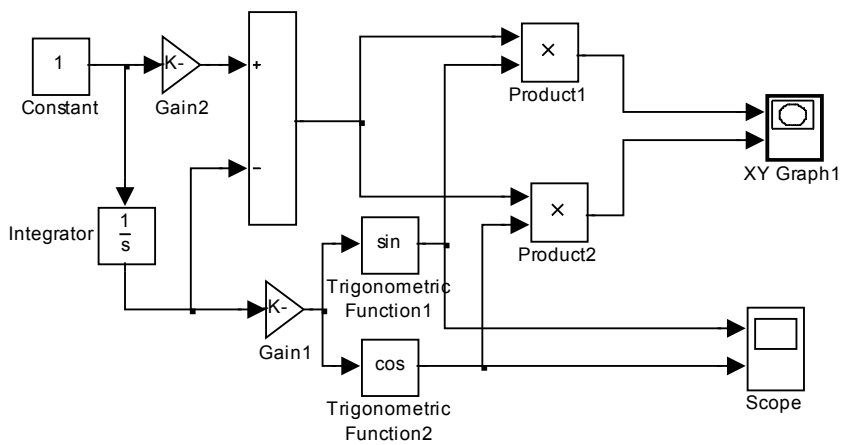


Рис. 7.16. Схема к упражнению 8.

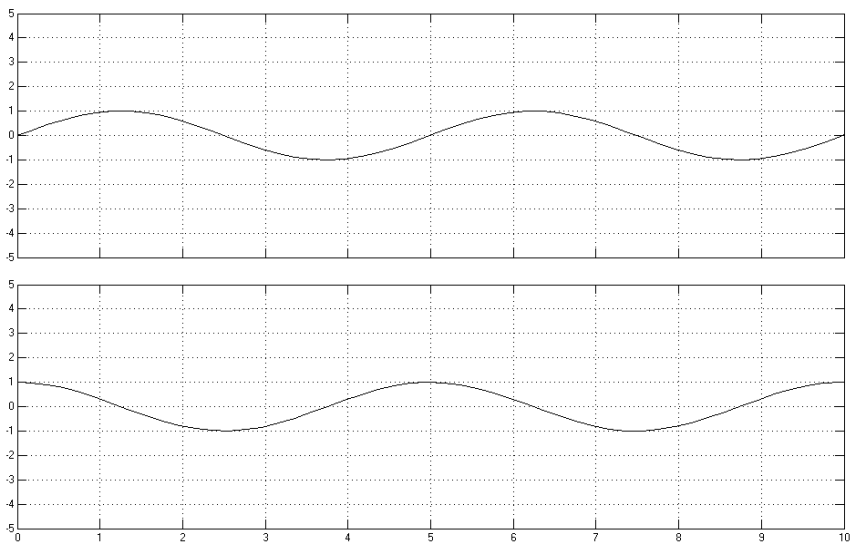


Рис. 7.17. Решение упражнения 8 на осциллографе.

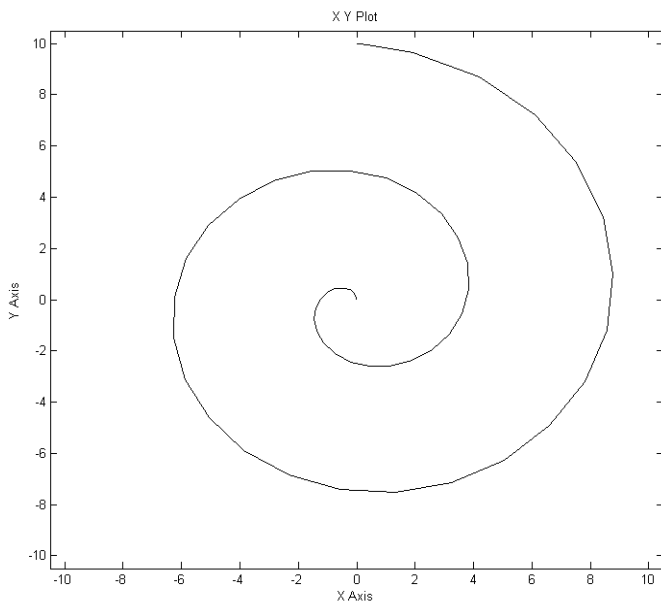


Рис. 7.18. Фазовый портрет системы (упражнение 8).

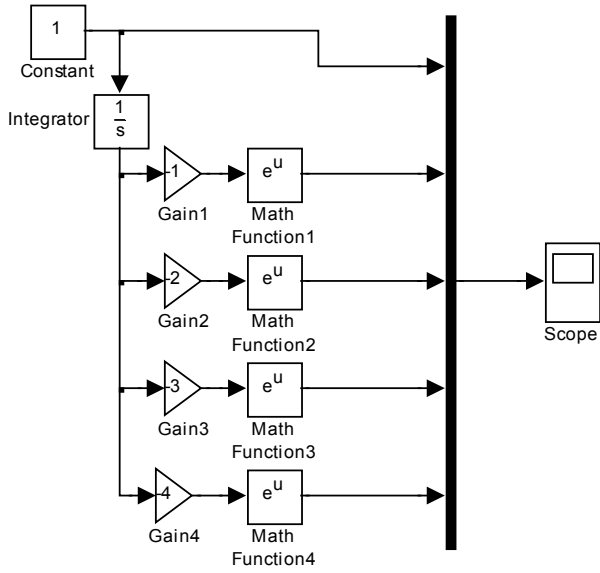


Рис. 7.19. Схема к упражнению 9.

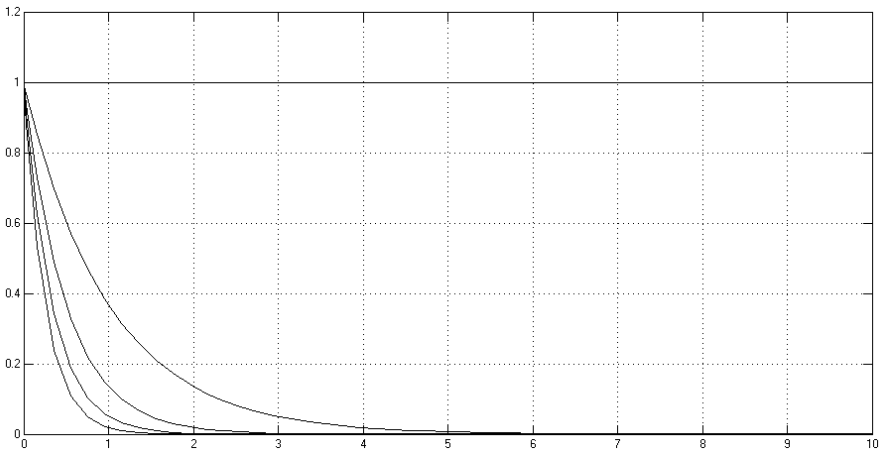


Рис. 7.20. Решение упражнения 9 на осциллографе.

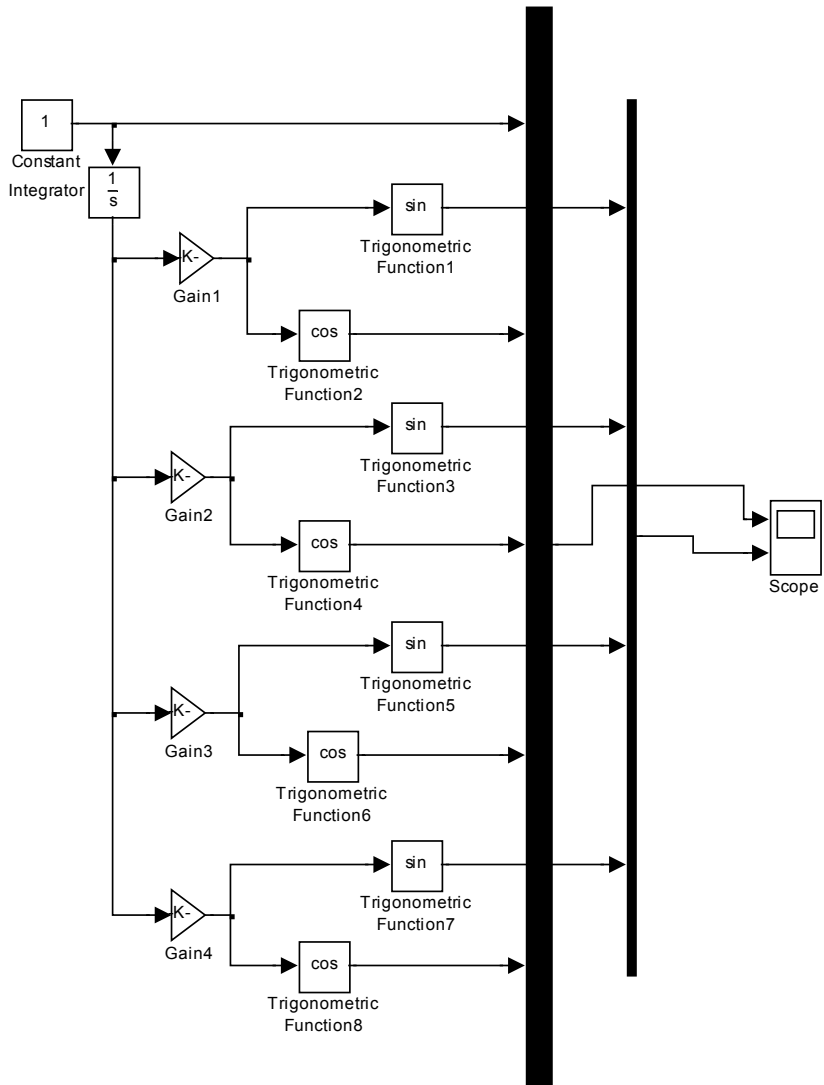


Рис. 7.21. Схема к упражнению 10.

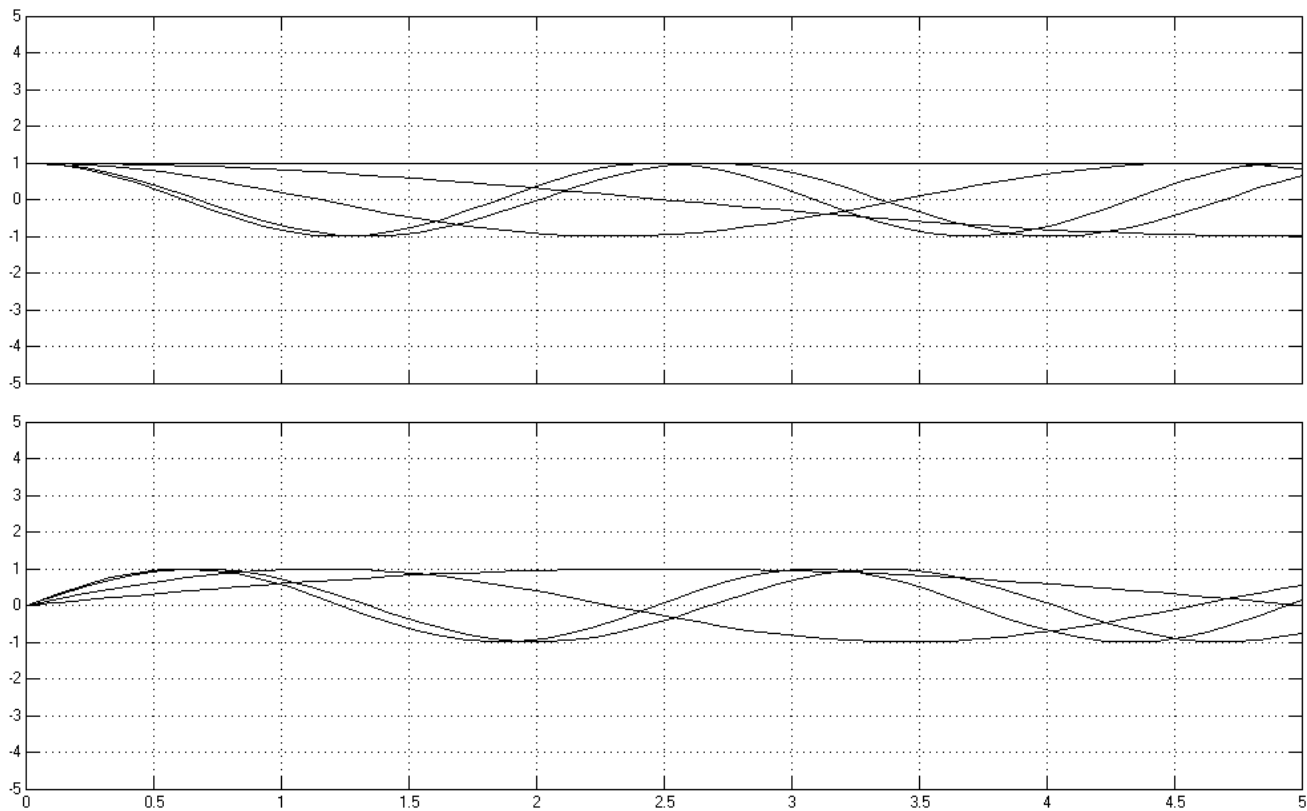


Рис. 7.22. Решение упражнения 10 на осциллографе.

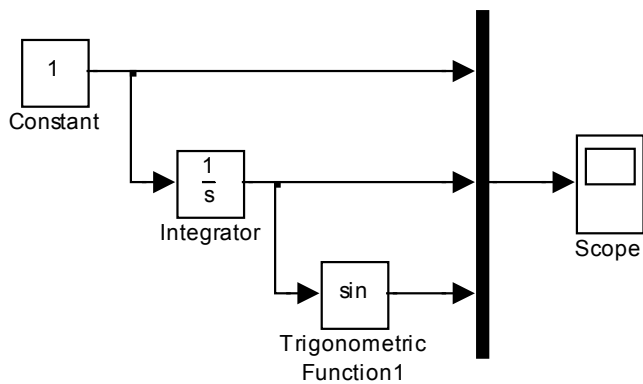


Рис. 7.23. Схема к упражнению 11.

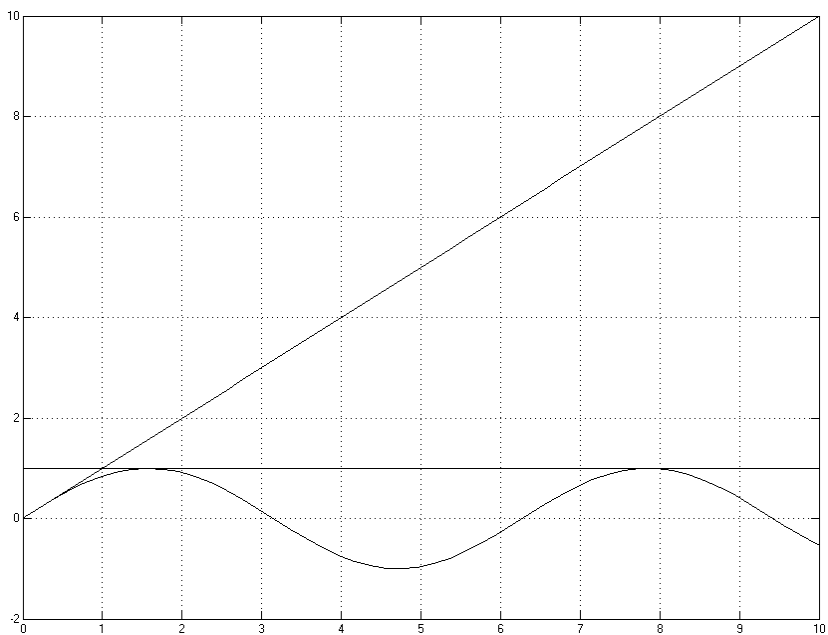


Рис. 7.24. Решение упражнения 11 на осциллографе.

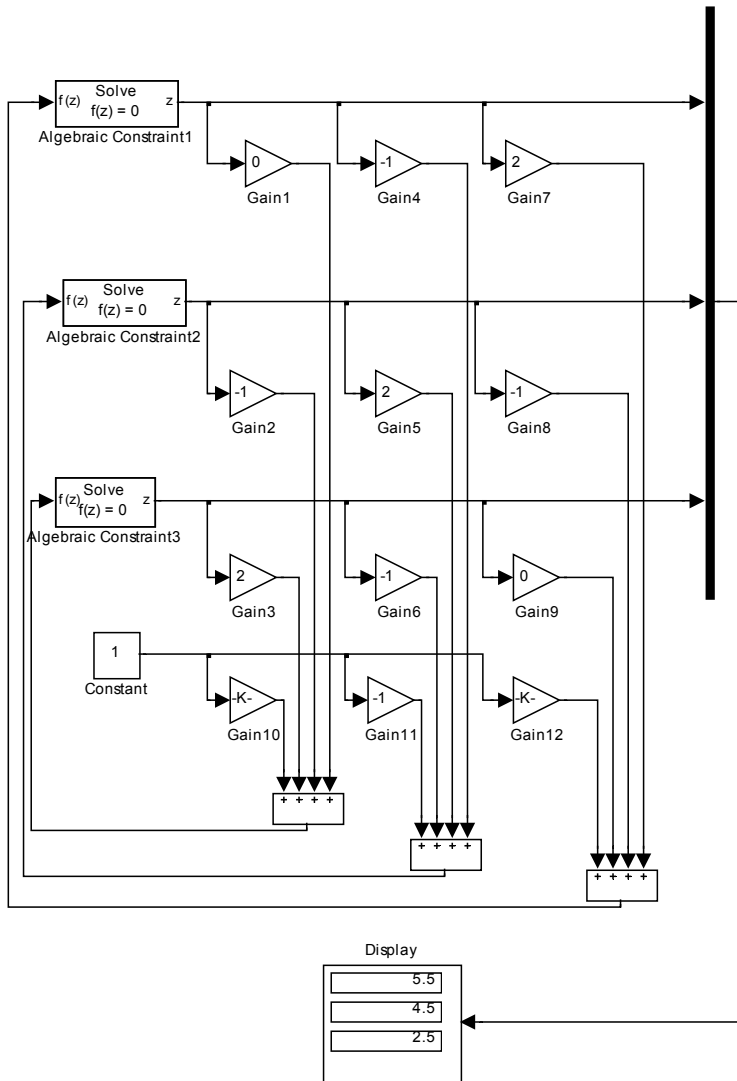


Рис. 7.25. Схема к упражнению 12.



Рис. 7.26. Решение упражнения 12 на виртуальном осциллографе (зависимость высоты полета от времени)

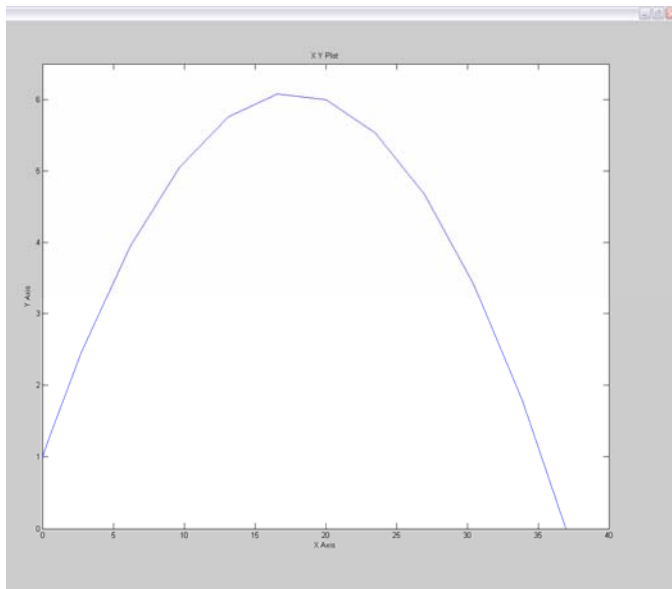


Рис. 7.27. Решение упражнения 12 (траектория полета камня на двухкоординатном регистраторе XY-Graph).

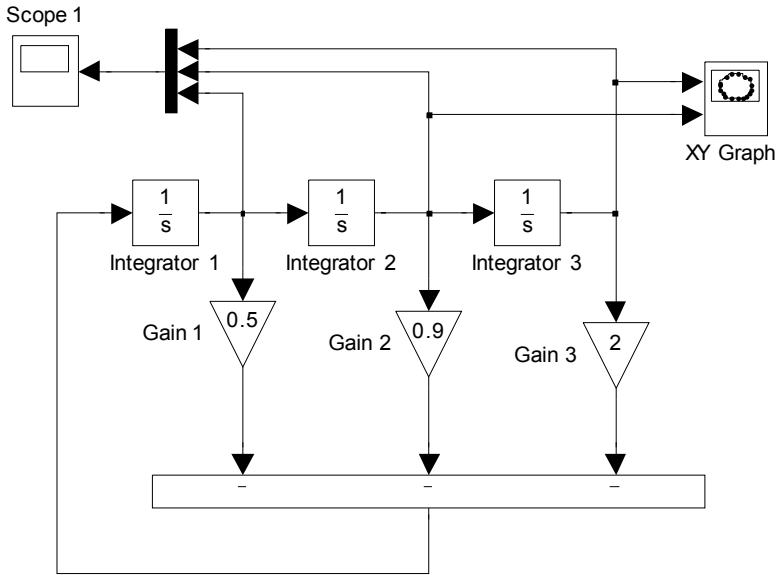


Рис. 7.28. Схема к упражнению 13.

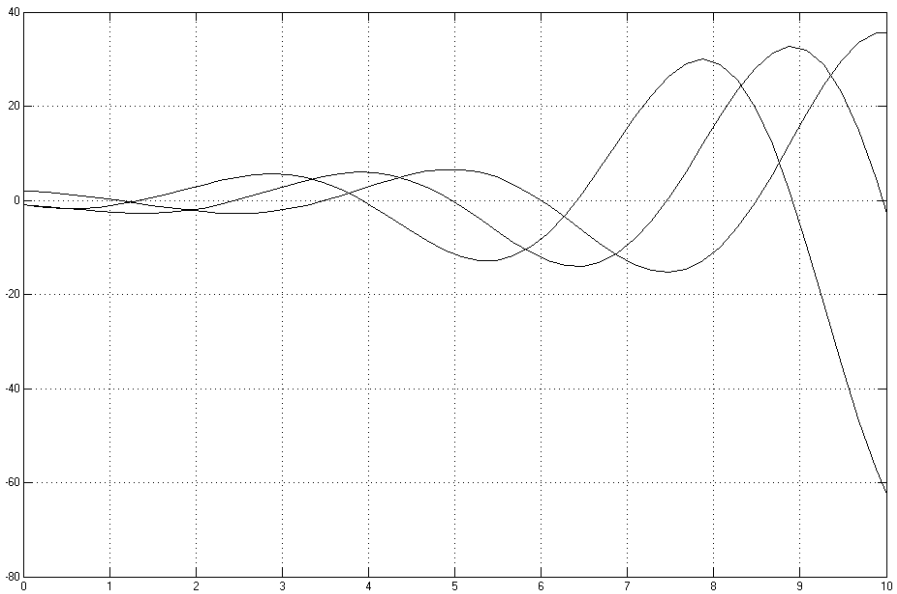


Рис. 7.29. Решение упражнения 13 на осциллографе.

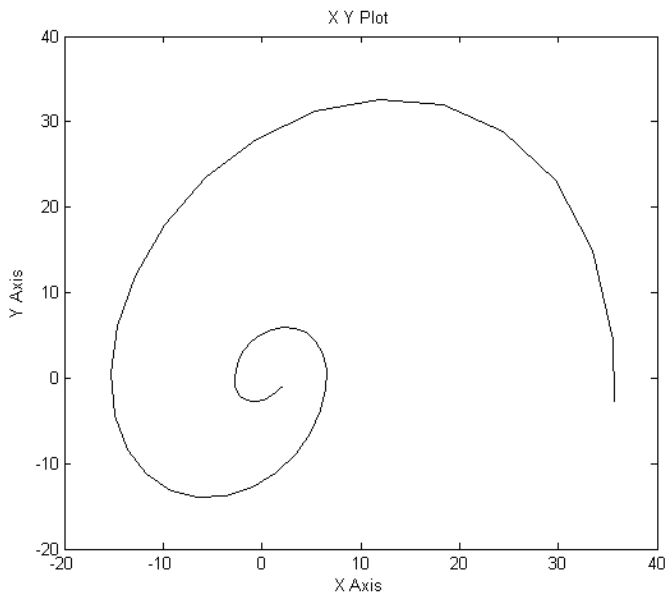


Рис. 7.30. Фазовый портрет системы (упражнение 13).

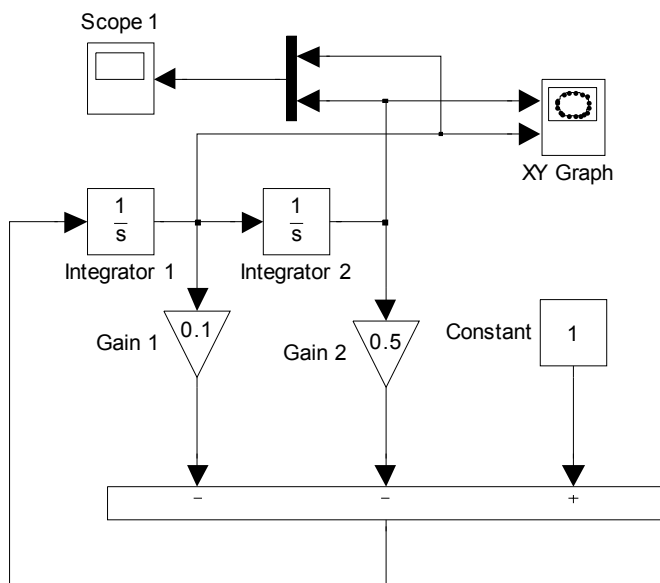


Рис. 7.31. Схема к упражнению 14.

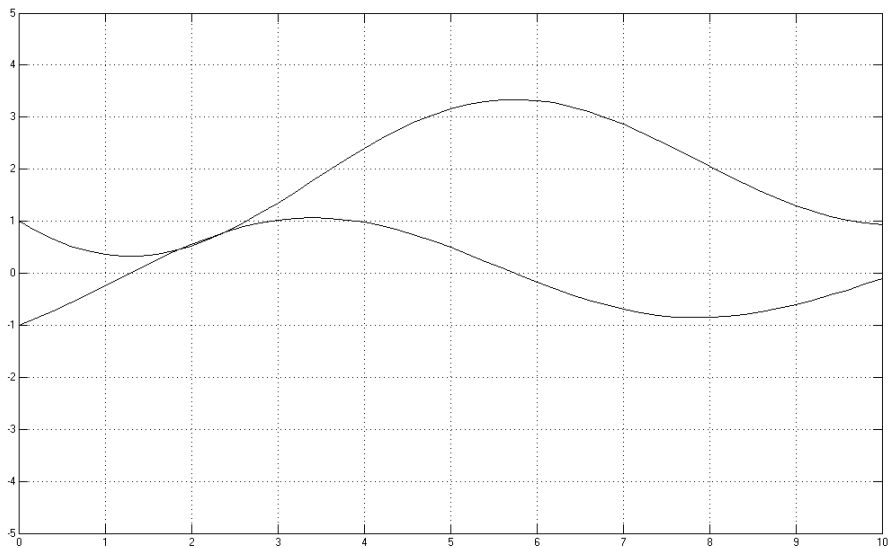


Рис. 7.32. Решение упражнения 14 на осциллографе.

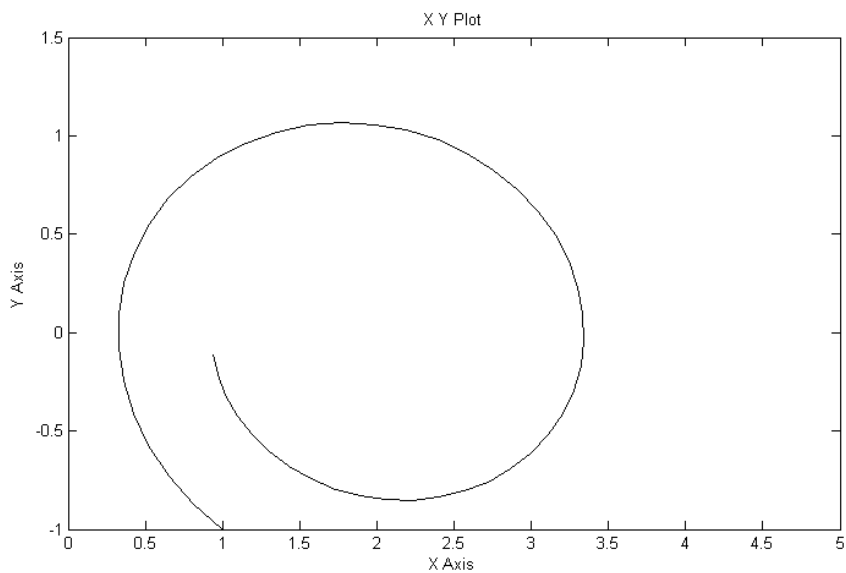


Рис. 7.33. Фазовый портрет системы (упражнение 14).

8. ЛИТЕРАТУРА

1. *Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н.* MATLAB 7. – СПб.: БХВ – Петербург, 2005. – 1104 с.
2. *А.с. 156703 СССР.* Моделирующее устройство для решения задач линейного программирования/ Г.Е. Пухов, Б.А.Борковский, В.В.Васильев, А.Е.Степанов, О.Н.Токарева. – Заявл.18.09.1962, № 795137/26-24; Оpubл. в Б.И., 1963, № 16; МКИ G 06d.
3. *А.с. 282761 СССР.* Квазианалоговое моделирующее устройство для решения задач линейного программирования/ Г.И. Грездов, Л.А.Симак, Ю.П. Космач. – Заявл.21.09.1969, № 1349448/18-24; Оpubл. в Б.И., 1970, № 30; МКИ G 06g 7/34.
4. *А.с. 304600 СССР.* Квазианалоговое моделирующее устройство / Г.И. Грездов, Л.А.Симак, Ю.П. Логвиненко. – Заявл.18.09.1969, № 1349833/18-24; Оpubл. в Б.И., 1971, № 17; МКИ G 06g 7/34.
5. *А.с. 409249 СССР.* Обратимый преобразователь /В.В.Васильев. – Заявл. 05.07.71, № 1677956/18-24; Оpubл. в Б.И., 1973, № 48; МКИ G 06o 3/00.
6. *Ахизер Н.И.* Лекции по теории аппроксимации. – М.-Л.: Гос. изд-во техн.-теор. лит., 1947. – 324 с.
7. *Васильев В.В.* О принципе наименьшего действия в электрических цепях // Сб. «Теоретическая электротехника».- Львов: Изд-во Львовского университета, 1972. – вып. 13.- С. 56 – 58.
8. *Васильев В.В., Ралдугтн Е.А.* Электронные модели задач на графах.- К.: Наукова Думка, 1987. – 152 с.
9. *Данилов А.И.* Компьютерный практикум по курсу «Теория управления». SIMULINK – моделирование в среде MATLAB. – М.: МГУИЭ, 2002, 128 с.
10. *Деннис Дж.* Математическое программирование и электрические цепи. – М.: Изд-во иностр. лит., 1961. – 212 с.
11. *Дьяконов В.П.* MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. Полное руководство пользователя. – М.: СОЛОН-Пресс. -2002. – 768 с.
12. *Дьяконов В.П., Круглов В.* Математические пакеты расширения MATLAB. Специальный справочник. – СПб.: Питер, 2001. – 480 с.
13. *Дьяконов В.П., Круглов В.* MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. – СПб.: Питер, 2002. – 448 с.
14. *Дэбни Дж., Хартман Т.* Simulink 4. Секреты мастерства. – М.: БИНОМ. Лаборатория знаний, 2003. – 403 с.
15. *Загидуллин Р.Ш.* LabView в исследованиях и разработках. – М.: Горячая линия – Телеком, 2005. – 352 с.

16. *Карлацук В.И.* Электронная лаборатория на IBM PC. Программа Electronics Workbench и ее применение. – М.: СОЛОН – Пресс, 2003. – 736 с.
17. *Кондрашов В.Е., Королев С.Б.* MATLAB как система программирования научно-технических расчетов. – М.: Мир, 2002. – 350 с.
18. *Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров. – М.: Наука, 1977. – 832 с.
19. *Лазарев Ю.Ф.* MatLAB 5.x. – К.: Издательская группа BHV, 2000. – 384 с.
20. *Мелентьев П.В.* Приближенные вычисления. – М.: Физматгиз, 1962. – 388 с.
21. *Моделирование динамических систем: Аспекты мониторинга и обработки сигналов / Васильев В.В., Грездов Г.И., Симак Л.А. и др.* – К.: НАН Укр., 2002. – 344 с.
22. *Пухов Г.Е.* Избранные вопросы теории математических машин. – К.: Изд-во АН УССР, 1964. – 263 с.
23. *Пухов Г.Е.* Методы анализа и синтеза квазианалоговых электронных цепей. – К.: Наукова думка, 1967. – 568 с.
24. *Пухов Г.Е., Кулик М.Н.* Гибридное моделирование в энергетике. – К.: Наук. думка, 1977. – 151 с.
25. *Пухов Г.Е., Васильев В.В.* Теория математического моделирования и математические машины непрерывного действия. – К.: КИИГА, 1971. – 110 с.
26. *Симак Л.А.* Аппроксимирующие полиномиальные спектры непрерывных сигналов и их применение. – Киев, 1987. – 46 с. – (Препр. / АН УССР. Ин-т проблем моделирования в энергетике; 112).
27. *Черных И.В.* SIMULINK: среда создания инженерных приложений. – М.: ДИАЛОГ-МИФИ, 2003. – 496 с.

**Национальная Академия наук Украины
Институт проблем моделирования в энергетике им. Г.Е.Пухова
Отделение гибридных моделирующих и управляющих систем в
энергетике**

**Министерство образования и наук Украины
Национальный авиационный университет
Кафедра электротехники и светотехники**

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ И СИСТЕМ В СРЕДЕ MATLAB/SIMULINK.

УЧЕБНОЕ ПОСОБИЕ

**Всеволод Викторович Васильев
Лилия Алексеевна Симак
Анна Михайловна Рыбникова**

Печатается по решению ученого совета Отделения гибридных
моделирующих и управляющих систем в энергетике ИПМЭ им.
Г.Е.Пухова НАН Украины

Подготовка к печати, компьютерный набор, верстка
Анна Михайловна Рыбникова
Виктория Владимировна Чёчь

Подписано к печати 12.03.08. Формат 60 x 84/16. Бумага офсетная

Усл. печ. л. 8,0. Уч.- изд. л. 7.4

Тираж 100 экз.

Заказ № _____

Отпечатано ООО «Академпресс»

Свидетельство о внесении в Государственный реестр издателей, изготовителей и
распространителей продукции № ДК 2634 от 04.10.06 г.

Адрес: 01133, г. Киев, ул. Кутузова, 18/7

Телефон: 286-42-38