

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
НЕВИННОМЫССКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ
(ФИЛИАЛ) СКФУ

Методические указания к лабораторным работам
(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)
Моделирование объектов и систем управления
Часть 2

Направление подготовки 15.03.04 Автоматизация технологических
процессов и производств

Направленность (профиль) "Информационно-управляющие системы"

Невинномысск, 2021

УДК 519.87
ББК 22.18

Печатается по решению УМС
Северо-Кавказского федерального
университета

Рецензенты: профессор кафедры высшей алгебры и геометрии,
доктор технических наук, *Н.И. Червяков*; доцент кафедры
информационных систем, электропривода и автоматики, кандидат
технических наук *А.А. Евдокимов*

Э.Е. Тихонов

**Моделирование объектов и систем управления. Методические указания
к лабораторным работам / – Невинномысск: Изд-во СКФУ, 2021.-92с.**

Пособие подготовлено в соответствии с программой дисциплины
«**Моделирование объектов и систем управления**», разработанной в
соответствии с требованиями ФГОС ВО направления подготовки
бакалавров.

УДК 519.87
ББК 22.18

© Э.Е. Тихонов, 2021
© НТИ, 2021

СОДЕРЖАНИЕ

Лабораторная работа №1 Инструмент моделирования систем Matlab/Simulink, построение простейших моделей.....	4
Лабораторная работа №2 Построение непрерывно-детерминированных моделей статических объектов и процессов.....	15
Лабораторная работа №3 Построение непрерывно-детерминированных моделей динамических объектов и процессов.....	27
Лабораторная работа №4 Построение дискретно-детерминированных моделей динамических объектов и процессов.....	39
Лабораторная работа №5 Вероятностное моделирование.....	60
Лабораторная работа №6 моделирование одноканальных систем массового обслуживания.....	72
Лабораторная работа №7 моделирование многоканальных систем массового обслуживания.....	83

Лабораторная работа №1 Инструмент моделирования систем Matlab/Simulink, построение простейших моделей

ЦЕЛЬ РАБОТЫ

- Знакомство с программной средой моделирования систем MATLAB/SIMULINK.
- Изучение основных блоков для построения моделей.
- Построение простейших моделей, проведение модельных экспериментов над моделями.

ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Перед выполнением лабораторной работы необходимо ознакомиться с темами:

- Основные приемы подготовки и редактирования модели [1, п.6].
- Установка параметров и проведение расчетов в модели [1, п.7,8].
- Библиотека блоков SIMULINK [1, п.9].

Ответить на вопросы:

1. Как создать новую модель в среде SIMULINK, как поместить блок в модель, как соединить блоки между собой?
2. Как задать параметры расчета времени перед запуском модели на выполнение?
3. Каково назначение и описание параметров блоков, используемых при выполнении лабораторной работы?

Описание программной среды SIMULINK

Программа SIMULINK является приложением к пакету MATLAB, так как работает только при наличии ядра MATLAB и использует функции, входящие в его состав.

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

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.

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.

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.

Основные блоки моделирования систем

Рассмотрим часто встречающиеся блоки SIMULINK. 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.

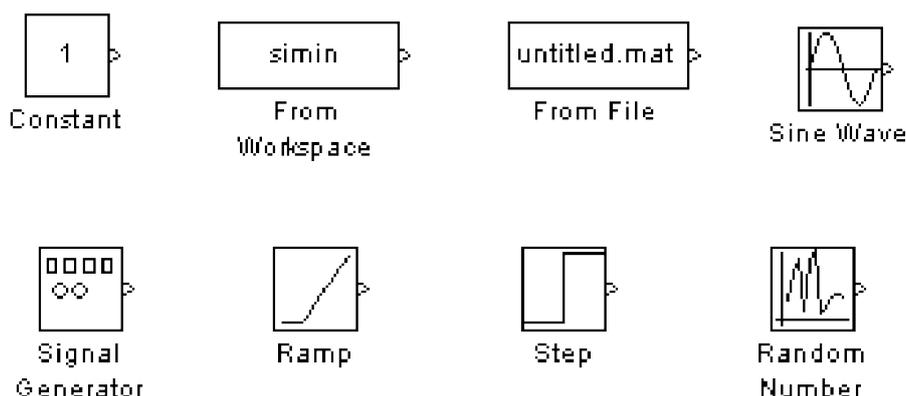


Рис. 1.1 Источники сигналов

Constant – блок задает постоянный по уровню сигнал;

From Workspace – блок считывания данных из рабочей области;

From File – блок считывания данных из файла;

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

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

Ramp – блок формирует линейный сигнал;

Step – блок формирует ступенчатый сигнал;

Random Number – блок формирует случайный сигнал с нормальным распределением уровня сигнала.

Приемники сигналов (Sinks). На рис. 1.2 представлены блоки-приемники сигналов.

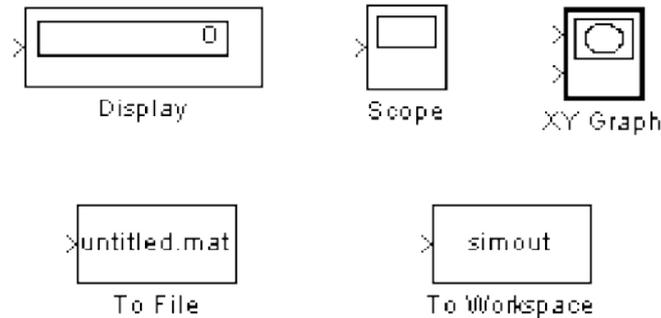


Рис. 1.2 Приемники сигналов

Display – цифровой дисплей, отображает значение полученного сигнала в виде числа;

Scope – виртуальный осциллограф, строит график функции полученного сигнала, зависящего от времени;

XY Graph – виртуальный графопостроитель, строит график одного сигнала в зависимости от другого (график вида $Y(X)$);

To File – блок записывает данные, поступающие на его вход, в файл;

To Workspace – блок записывает данные, поступающие на его вход, в рабочую область MATLAB.

Блоки математических операций. На рис. 1.3 представлены блоки математических операций.

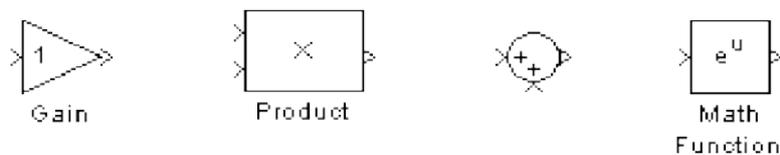


Рис. 1.3 Блоки математических операций

Gain – блок умножает входной сигнал на заданный коэффициент;

Product – блок перемножает два (или более) сигналов;

Sum – блок суммирует два (или более) сигналов;

Math Function – блок 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).

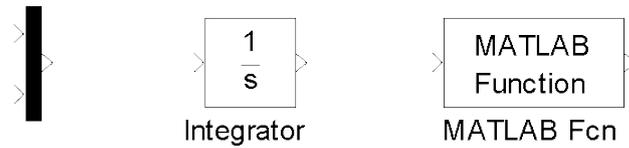


Рис. 1.4 Примеры часто встречающихся функций

Mux – мультиплексор, объединяет сигналы, пиктограмма которого показана первой слева на рис. 1.4;

Integrator – блок интегрирует сигнал;

MATLAB Fcn – блок формирует функцию, заданную пользователем на языке MatLab.

ПРАКТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Задание 1.1

Построить модель, формирующую сигнал вида $x(t) = 0.5 \sin \pi t + t^2$ на интервале $t \in [0, 2]$, и отобразить полученный сигнал на виртуальном осциллографе.

На рис. 1.5 приведен один из вариантов построения модели.

Входной сигнал $0.5 \sin \pi t$ задан в параметрах блока Sine Wave (рис. 1.6). Для сигнала t^2 использовались блоки: блок линейного сигнала и блок математической функции, где выбрана функция возведения в квадрат.

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

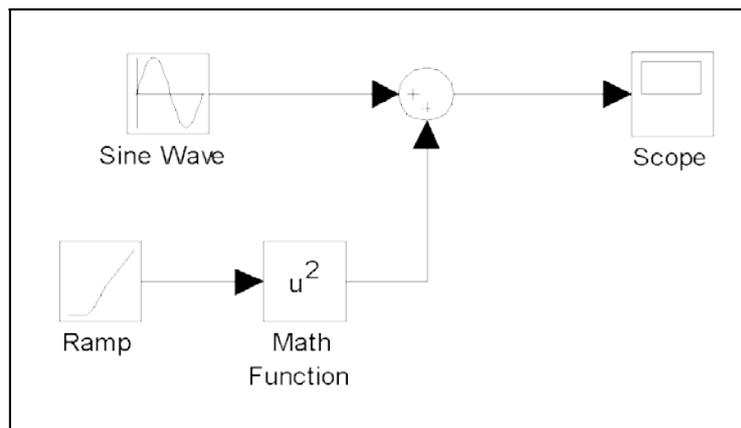


Рис. 1.5 Модель получения сигнала $x(t) = 0.5 \sin \pi t + t^2$

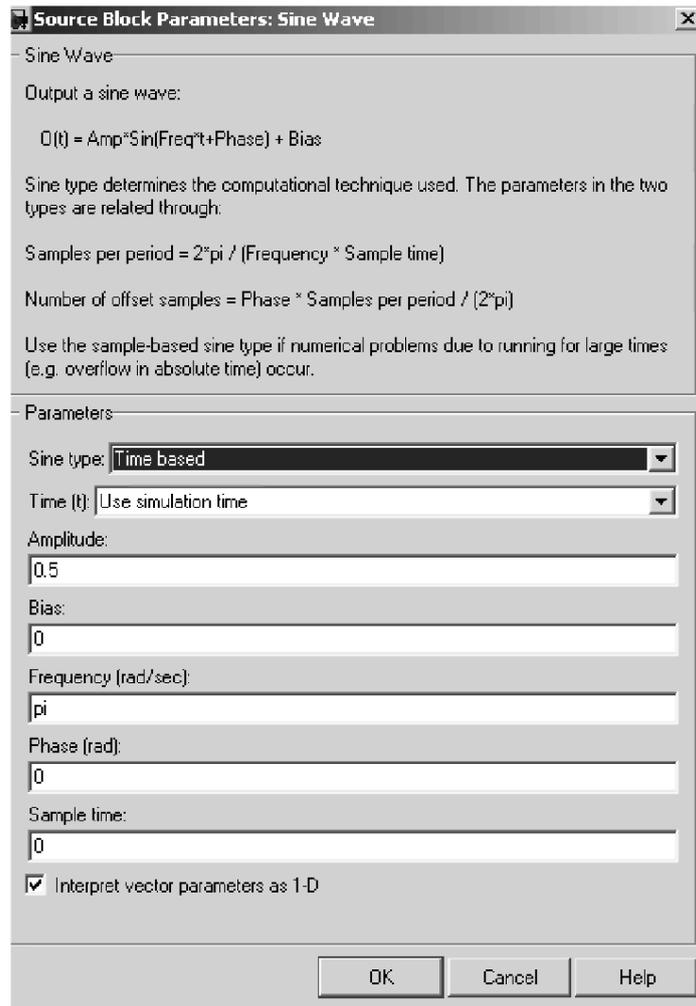


Рис. 1.6 Параметры блока Sine Wave

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

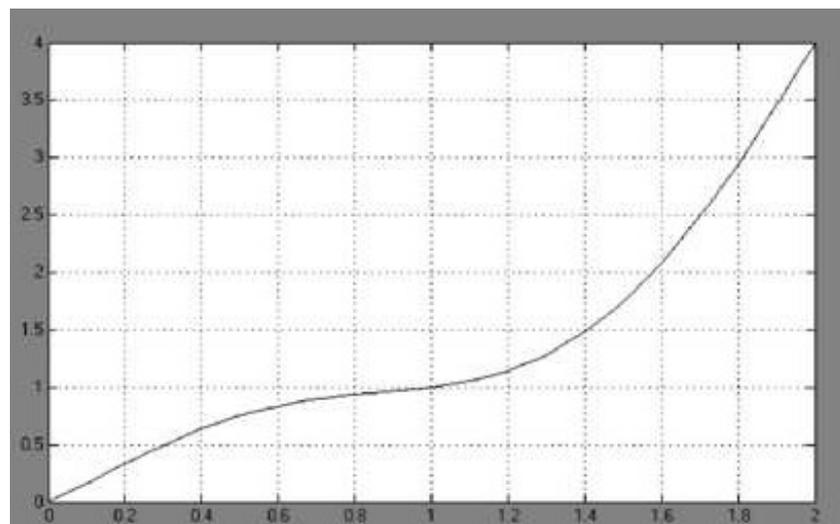


Рис. 1.7 Сигнал $x(t) = 0.5 \sin \pi t + t^2$ на экране виртуального осциллографа

Варианты для выполнения Задания 1.1

Построить модель, формирующую сигнал заданного вида на интервале $t \in [0, T]$, и отобразить полученный сигнал на виртуальном осциллографе.

При построении модели можно использовать следующие блоки: Constant, Integrator, Gain, Trigonometric Function, Math Function, Sum, Product, MinMax, Abs, Scope.

№	T	$x = f(t)$
1	2	$x(t) = (1 + 2 \sin(2t))^2 + 2t$
2	3	$x(t) = t + \sqrt{1 + 0.5 \sin(2t)}$
3	4	$x(t) = t^2 \cdot \cos(1.5 \cdot t)$
4	5	$x(t) = -3.5 + \sqrt{e^{-2t} + t^2}$
5	6	$x(t) = \min(5t, 100 - 2t^2)$
6	7	$x(t) = \sqrt{ -100 + 20t } + 2t$
7	1	$x(t) = \sqrt{t + \operatorname{tg}(0.5 \cdot t)} + 2.5$
8	2	$x(t) = \ln(1+t) + \sqrt{2t + 1.5}$
9	3	$x(t) = -2 + \sqrt{10^{-t} + t^2}$
10	4	$x(t) = e^{-2t} + \sqrt{1 + \sin(3t-1)}$
11	5	$x(t) = \cos(0.5 \cdot t) + t-2 $
12	6	$x(t) = \sqrt{e^{0.1 \cdot t} + t + 2t^2}$
13	7	$x(t) = \min(3t, 25 - t^2) + 3$
14	1	$x(t) = 5 \cdot \operatorname{tg} t^2 - 2 + t$
15	2	$x(t) = \sqrt{10^t + \sin(\pi t + 1)}$
16	3	$x(t) = (1 + 2 \cos t)^2 - 3t$
17	4	$x(t) = 2t - \sqrt{1 + 0.3 \cos(3t)}$
18	5	$x(t) = 1.5 + \sqrt{2^{-2t} + 3t^2}$
19	6	$x(t) = \ln(1 + 0.5t) + \sqrt{t + 0.5}$
20	7	$x(t) = \sin(0.2 \cdot t) + t-3 $
21	1	$x(t) = \sqrt{10^{2t} + \cos(\pi t - 1)}$
22	2	$x(t) = \min(5t, 10 - 2t^2)$

№	T	$x = f(t)$
23	3	$x(t) = \sqrt{-10 + 5t} - 3t^2$
24	4	$x(t) = 3^{-2t} + \sqrt{1 - \sin(3t+1)}$
25	5	$x(t) = t-3 + \cos(0.3-t)$

Задание 1.2

Построить структурные схемы виртуальных генераторов систем базисных функций.

Пример 1. На рис. 1.8, рис. 1.9 изображены две альтернативные схемы виртуальных генераторов системы степенных функций вида $S(t) = \{s_k(t) = t^k, k = \overline{0,4}\}$ при $t \in [0, 1]$.

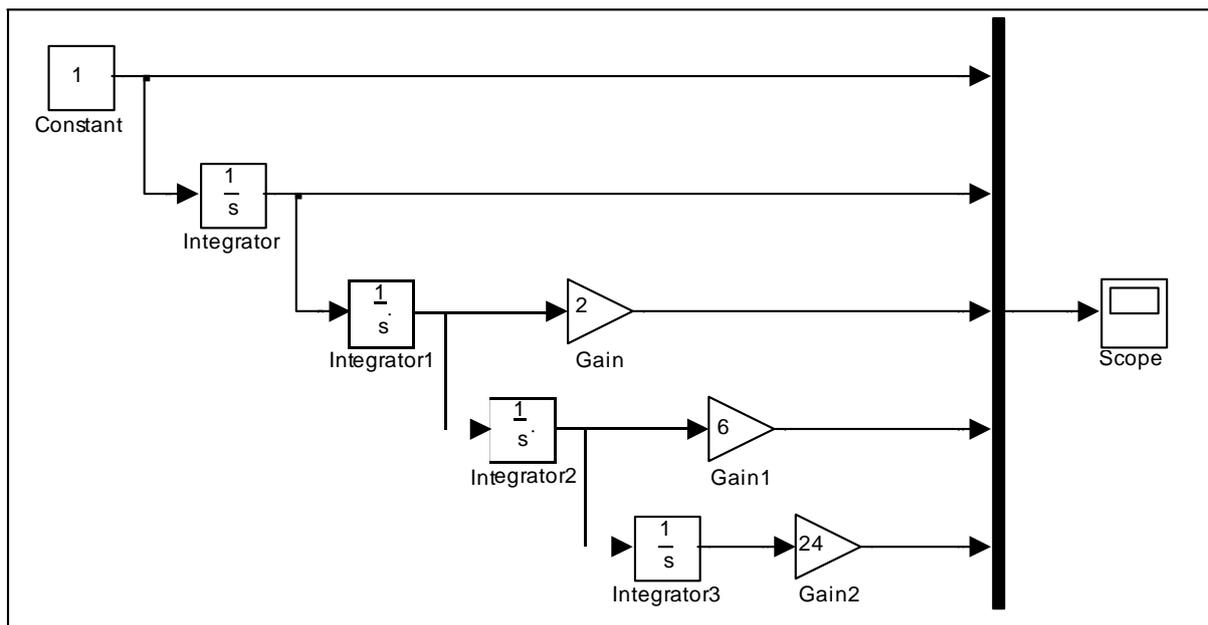


Рис. 1.8 Первый вариант структурной схемы виртуального генератора системы степенных базисных функций

Первая схема использует каскадное соединение интеграторов. На вход первого интегратора подается сигнал константы 1, на выходе получается $\int 1 \cdot dt = t$. Этот сигнал подается на вход второго интегратора, на выходе получается $\int t \cdot dt = \frac{t^2}{2}$, после умножения которого на 2 получается искомый сигнал. На вход третьего интегратора подается $\frac{t^2}{2}$, на

выходе получается $\frac{1}{2} \cdot \int t^2 dt = \frac{1}{2} \cdot \frac{t^3}{3}$. Для получения искомого сигнала результат следует умножить на 6 и т.д. Полученные выходные сигналы подаются на входы смесителя Mix для отображения на экране виртуального осциллографа Scope.

Во второй схеме используются система умножителей, последовательно формирующая сигналы степенных функций.

На рис 1.10 показана система степенных базисных функций.

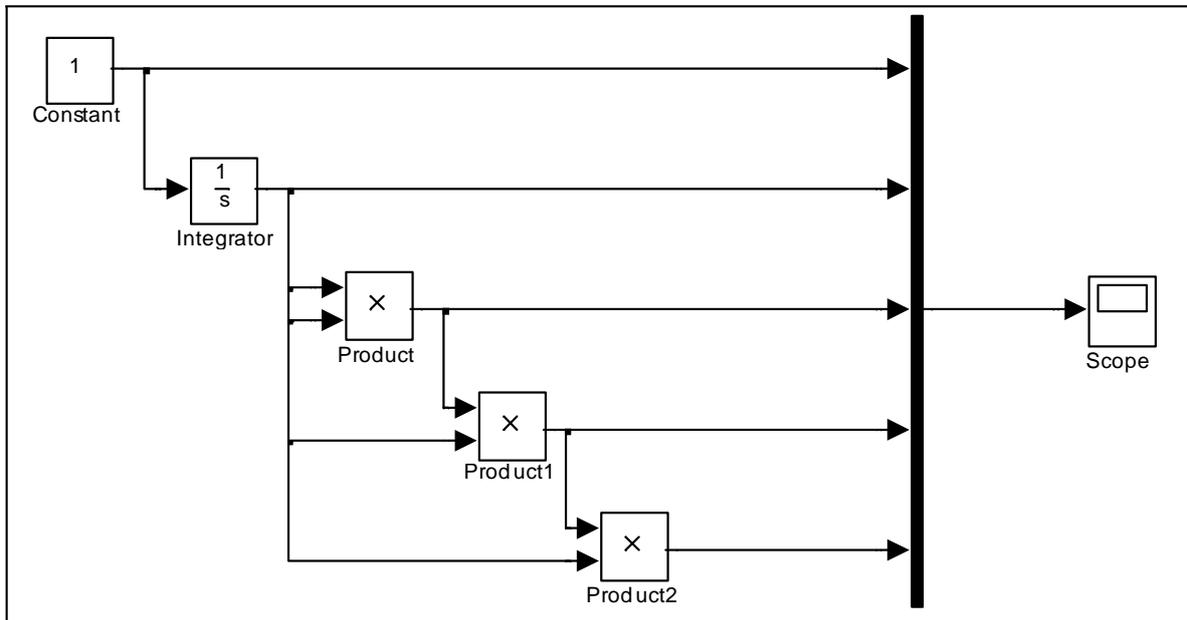


Рис. 1.9 Второй вариант структурной схемы виртуального генератора системы степенных базисных функций

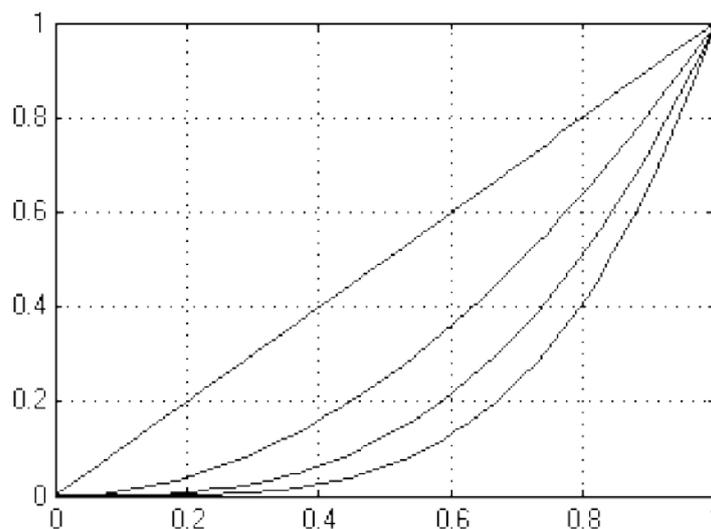


Рис. 1.10 Изображение системы степенных функций на экране виртуального осциллографа

Пример 2. На рис. 1.11 приведена структурная схема виртуального генератора системы экспоненциальных функций вида:
 $S(t) = \{s_k(t) = e^{kt}, k = \overline{0,4}\}$ при $t \in [0, 1]$.

В этой схеме используются 4 одинаковых, 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 $1 = e^0$.

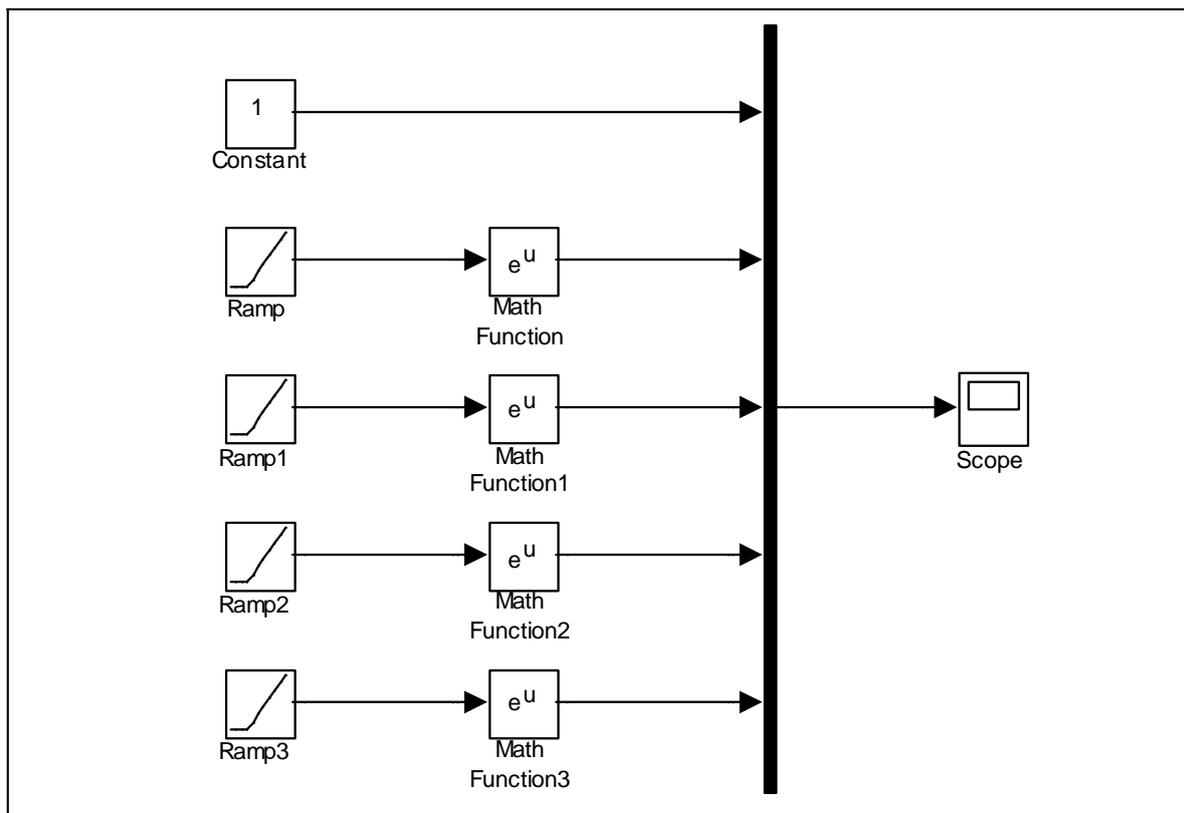


Рис. 1.11 Структурная схема виртуального генератора системы экспоненциальных функций

На рис. 1.12 изображен вид системы экспоненциальных функций на экране виртуального осциллографа.

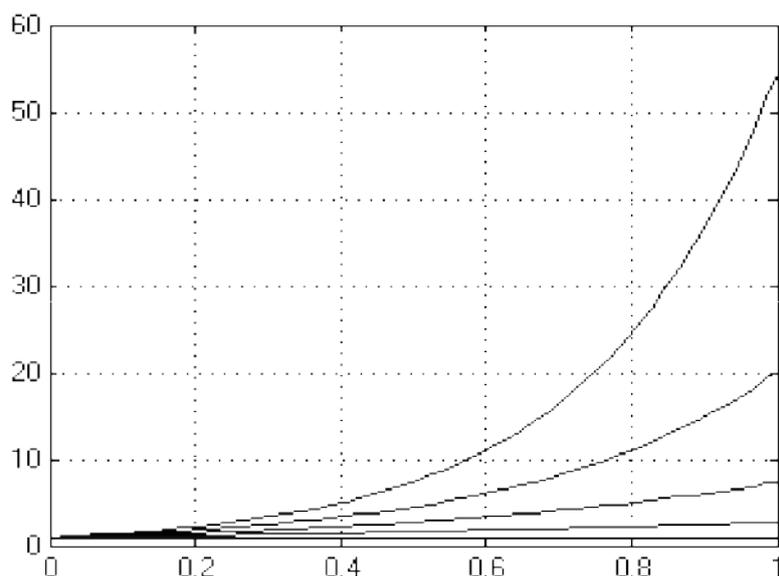


Рис. 1.12 Изображение системы экспоненциальных функций на виртуальном осциллографе

Варианты для выполнения Задания 1.2

Построить структурные схемы виртуальных генераторов систем базисных функций при $t \in [0, 1]$ и отобразить полученные сигналы на виртуальном осциллографе.

При построении модели использовать следующие блоки: Constant, Integrator, Gain, Trigonometric Function, Math Function, Mux, Scope.

№	$S(t) = \{s_k(t)\}$
1	$S(t) = \{1, \sin t, \cos t, \sin 2t, \cos 2t\}$
2	$S(t) = \{1, e^t, t \cdot e^t, t^2 \cdot e^t, t^3 \cdot e^t\}$
3	$S(t) = \{1, t, \sin 2t, \cos 2t\}$
4	$S(t) = \{1, te^t, te^{2t}, te^{3t}, te^{4t}\}$
5	$S(t) = \{1, \sin t, \cos t, t \cdot \sin t, t \cdot \cos t\}$
6	$S(t) = \{1, t \cdot e^t, t^2 \cdot e^{2t}, t^3 \cdot e^{3t}\}$
7	$S(t) = \{1, \sin 2t, \cos 2t, t \cdot \sin t, t \cdot \cos t\}$
8	$S(t) = \{1, e^t, t^2 \cdot e^t, t^4 \cdot e^t, t^6 \cdot e^t\}$
9	$S(t) = \{1, \sin t, \cos t, t \cdot \sin 2t, t \cdot \cos 2t\}$
10	$S(t) = \{1, t, te^{2t}, te^{4t}, te^{6t}\}$

№	$S(t) = \{s_k(t)\}$
11	$S(t) = \{\sin t, \cos t, \sin 2t, \cos 2t, \sin 3t, \cos 3t\}$
12	$S(t) = \{te^{-t}, te^{-2t}, te^{-3t}, te^{-4t}\}$
13	$S(t) = \{t, \sin 2t, \cos 2t, \sin 3t, \cos 3t\}$
14	$S(t) = \{e^{-t}, e^{-2t}, e^{-3t}, e^{-4t}\}$
15	$S(t) = \{e^t, \sin t, \cos t, e^{2t}, \sin 2t, \cos 2t\}$
16	$S(t) = \{1, \sin t, \sin 2t, \sin 3t, \sin 4t\}$
17	$S(t) = \{te^{-t}, t^2e^{-t}, t^3e^{-t}, t^4e^{-t}\}$
18	$S(t) = \{1, \cos t, \cos 2t, \cos 3t, \cos 4t\}$
19	$S(t) = \{te^t, t^2e^t, t^3e^t, t^4e^t\}$
20	$S(t) = \{1, t \sin t, t \sin 2t, t \sin 3t, t \sin 4t\}$
21	$S(t) = \{t \sin t, t \cos t, t^2 \sin t, t^2 \cos t\}$
22	$S(t) = \{1, t \cos t, t \cos 2t, t \cos 3t, t \cos 4t\}$
23	$S(t) = \{e^{-t} \cos t, e^{-t} \cos 2t, e^{-t} \cos 3t\}$
24	$S(t) = \{\sin t, t \sin t, t^2 \sin t, t^3 \sin t, t^4 \sin t\}$
25	$S(t) = \{\cos t, t \cos t, t^2 \cos t, t^3 \cos t, t^4 \cos t\}$

Структурную схему модели верхнего уровня (рис. 2.1) можно представить в виде блока, входными сигналами которого являются внешние воздействия, заданные величинами b_1, b_2, \dots, b_n , а выходными сигналами являются искомые значения переменных x_1, x_2, \dots, x_n .

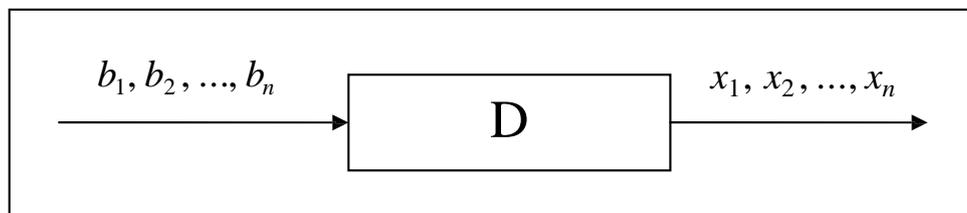


Рис. 2.1 Структурная схема модели верхнего уровня

В аналоговой вычислительной технике существует несколько способов моделирования объектов, описываемых системами алгебраических уравнений. Один из них сводится к решению системы обыкновенных дифференциальных уравнений, установившееся решение которой дает решение алгебраической задачи. Еще один метод основан на применении специального блока, который выполняет поиск корней алгебраических уравнений.

Для построения модели данного объекта применим *метод сведения к эквивалентной системе дифференциальных уравнений*. Введем в рассмотрение следующую систему дифференциальных уравнений:

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

Если при решении полученной системы все производные оказываются затухающими и в итоге принимают значения, равные нулю $\frac{dx_i}{dt} = 0$, то на выходе получается решение системы алгебраических уравнений $\{x_1, x_2, \dots, x_n\}$.

Для эквивалентности этих двух систем уравнений должно существовать установившееся решение системы дифференциальных уравнений $\frac{dx_i}{dt} = 0$, $i = \overline{1, n}$. Достаточным условием, обеспечивающим $x_i = \text{const}$

ПРАКТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Задание 2.1

Пример 1. Найти решение системы алгебраических уравнений:

$$\begin{cases} 4x_1 + 2x_2 = 14, \\ 2x_1 + 5x_2 = -5. \end{cases}$$

Система является системой с преобладающими диагональными элементами $a_{11} = 4 > 2$, $a_{22} = 5 > 2$, поэтому для ее решения применим метод сведения ее к эквивалентной системе дифференциальных уравнений

$$\begin{cases} \frac{dx_1}{dt} = -4x_1 - 2x_2 + 14, \\ \frac{dx_2}{dt} = -2x_1 - 5x_2 - 5. \end{cases}$$

Структурная схема модели данной системы приведена на рис. 2.2.

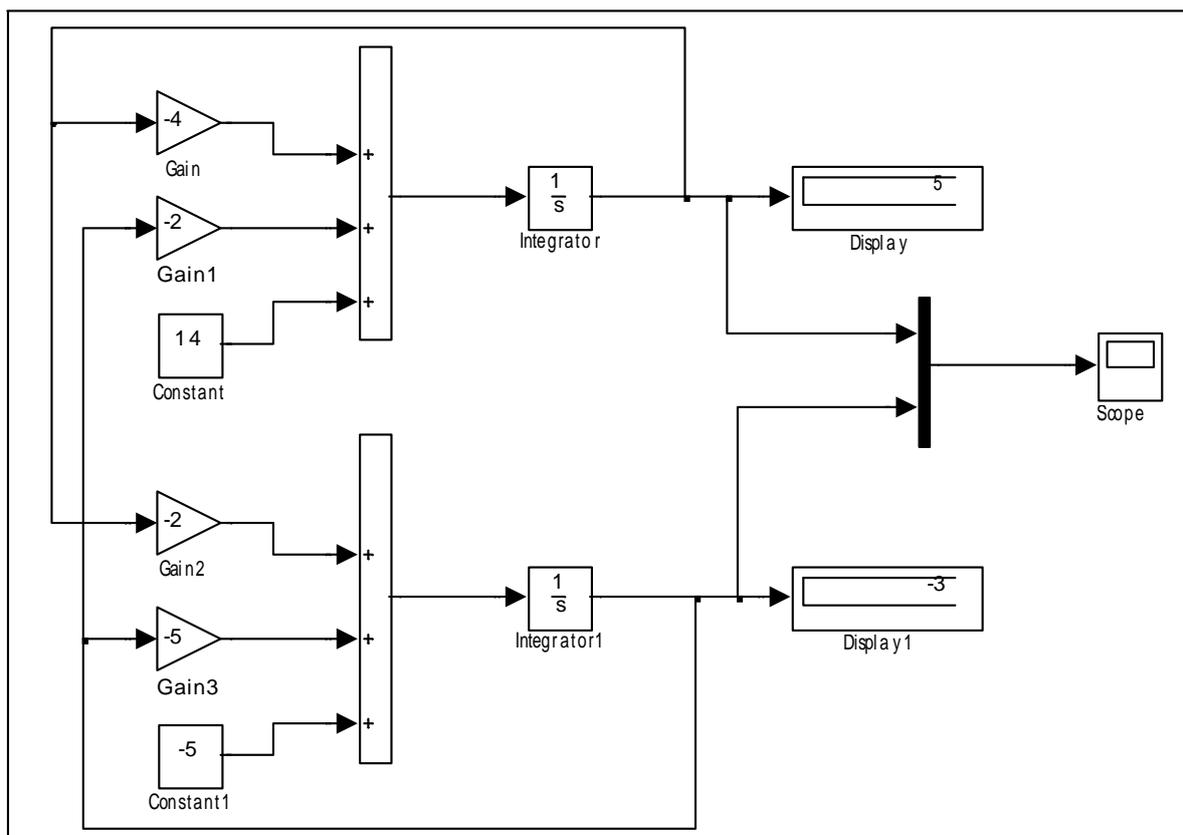


Рис.2.2 Структурная схема модели системы дифференциальных уравнений

Схема построена с использованием классических методов аналоговой вычислительной техники. Входными сигналами схемы являются значения 14 и -5 , при этом на входы интеграторов Integrator и Integrator1 подаются сформированные правые части первого и второго уравнения, а на выходах получаются значения искоемых переменных x_1, x_2 . Если наступит момент, когда эти значения перестанут изменяться, то они и являются решением системы.

Переходный процесс получения установившегося решения изображен на экране виртуального осциллографа (рис. 2.3).

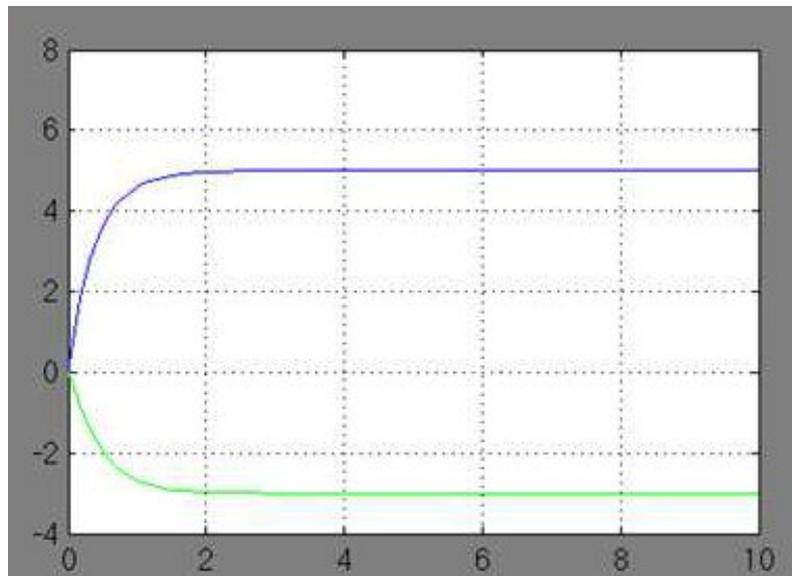


Рис. 2.3 Переходный процесс получения установившегося решения системы алгебраических уравнений

На рис. 2.3 видно, что после $t=2$ на выходах виртуальных интеграторов устанавливаются сигналы, соответствующие решению системы линейных алгебраических уравнений, $x_1 = 5$, $x_2 = -3$. Если выход на установившееся решение не происходит, то это означает, что решение системы не найдено, а саму систему следует преобразовать к виду системы с преобладающими диагональными элементами.

Пример 2. Найти решение системы алгебраических уравнений:

$$\begin{cases} \sin(2x - y) - 1.2x = 0.4, \\ 0.8x^2 + 1.5y^2 = 1. \end{cases}$$

Для решения системы будем использовать блок Algebraic Constraint, который выполняет поиск корней алгебраических уравнений. Систему представим в виде

$$\begin{cases} \sin(2x - y) - 1.2x - 0.4 = 0, \\ 0.8x^2 + 1.5y^2 - 1 = 0. \end{cases}$$

Входными сигналами схемы являются значения -0.4 и -1 . Сгенерированные выражения для вычисления левых частей уравнений направим на входы блоков Algebraic Constraint и Algebraic Constraint1, в которых зададим начальные значения выходных сигналов Initial guess, равными, например, $x = 0.4$, $y = 0$ соответственно.

Структурная схема модели данной системы приведена на рис. 2.4.

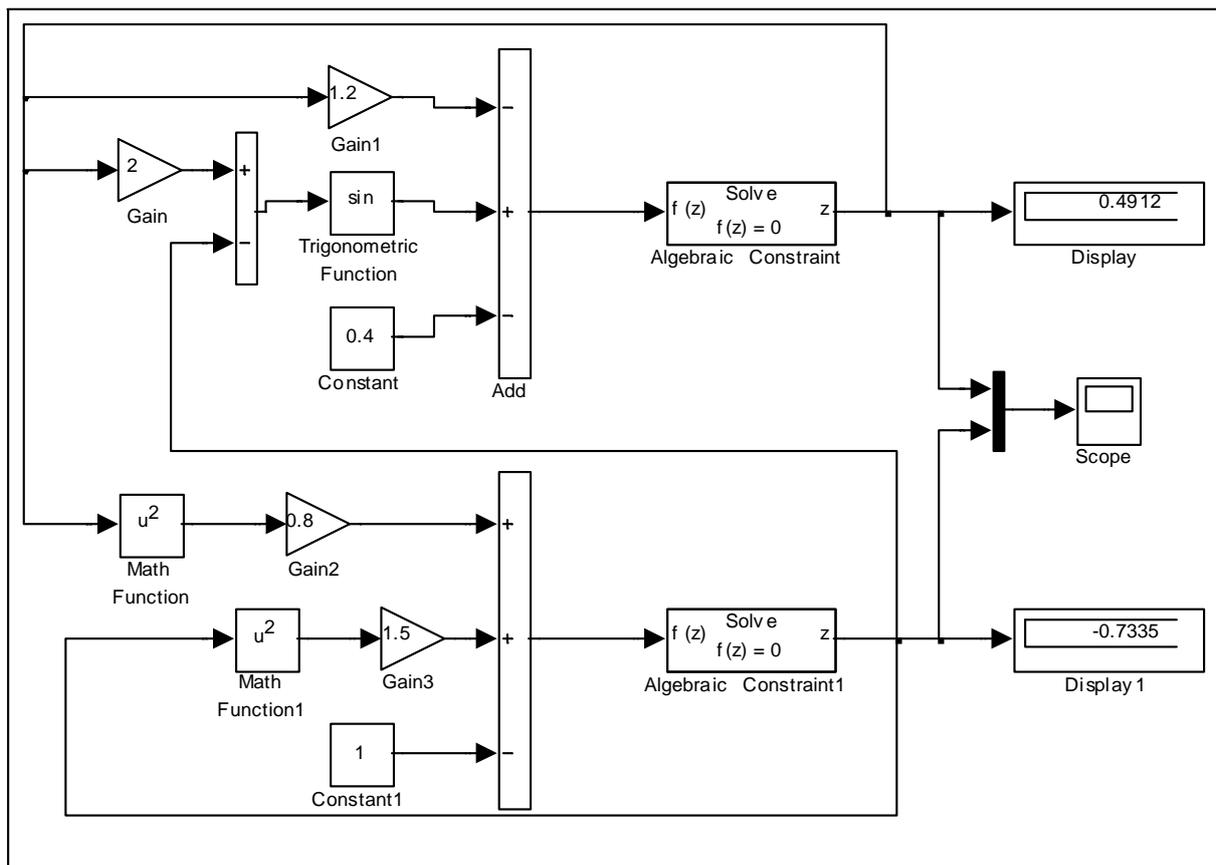


Рис. 2.4 Структурная схема модели системы алгебраических уравнений

Результатом работы модели является установившееся решение системы дифференциальных уравнений $x = 0.4912$, $y = -0.7335$, которое отображено в окнах блоков Display, Display1 (рис. 2.4), Scope (рис. 2.5), и которое является искомым решением системы алгебраических уравнений.

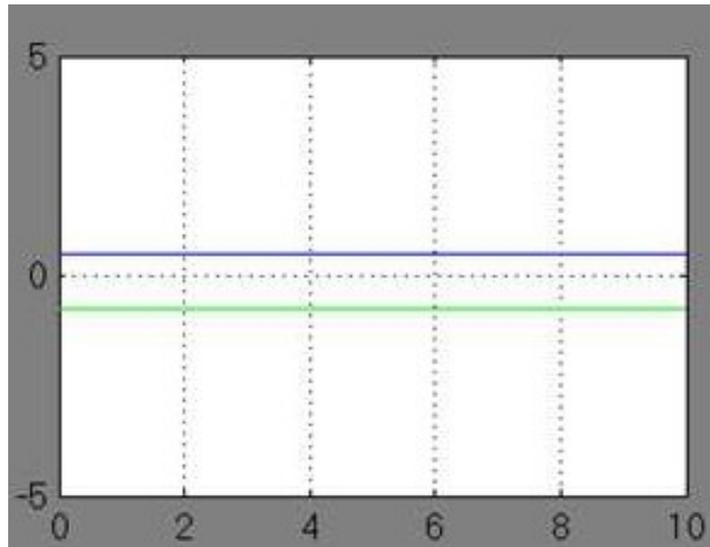


Рис. 2.5 Процесс получения решения системы алгебраических уравнений

Если изменить начальные значения выходных сигналов Initial guess блоков Algebraic Constraint и Algebraic Constraint1 на $x = -0.4$, $y = 0$, то получим второе решение системы алгебраических уравнений $x = -0.4391$, $y = -0.7509$.

Варианты для выполнения Задания 2.1

Синтезировать структурную схему моделей статических систем и получить соответствующее решение.

При построении модели можно использовать следующие блоки: Constant, Integrator, Gain, Trigonometric Function, Math Function, Sum, Product, Mux, Display, Scope, Algebraic Constraint.

№	Система 1	Система 2
1	$\begin{cases} \sin(x+1) - y = 1.2, \\ 2x + \cos y = 2. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.4) = x^2, \\ 0.6x^2 + 2y^2 = 1, x > 0, y > 0. \end{cases}$
2	$\begin{cases} \cos(x-1) + y = 0.5, \\ x - \cos y = 3. \end{cases}$	$\begin{cases} \sin(x+y) - 1.6x = 0, \\ x + y = 1, x > 0, y > 0. \end{cases}$
3	$\begin{cases} \sin x + 2y = 2, \\ x + \cos(y-1) = 0.7. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.1) = x^2, \\ x^2 + 2y^2 = 1. \end{cases}$
4	$\begin{cases} \cos x + y = 1.5, \\ 2x - \sin(y - 0.5) = 1. \end{cases}$	$\begin{cases} \sin(x+y) - 1.2x = 0.2, \\ x^2 + y^2 = 1. \end{cases}$

№	Система 1	Система 2
5	$\begin{cases} \sin(x+0.5) - y = 1, \\ x + \cos(y-2) = 0. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.3) = x^2, \\ 0.9x^2 + 2y^2 = 1. \end{cases}$
6	$\begin{cases} \cos(x+0.5) + y = 0.8, \\ \sin y - 2x = 1.6. \end{cases}$	$\begin{cases} \sin\left(\frac{x+y}{2}\right) - 1.3x = 0, \\ x + y = 1. \end{cases}$
7	$\begin{cases} \sin(x-1) = 1.3 - y, \\ x - \sin(y+1) = 0.8. \end{cases}$	$\begin{cases} \operatorname{tg}(xy) = x^2, \\ 0.8x^2 + 2y^2 = 1. \end{cases}$
8	$\begin{cases} 2y - \cos(x+1) = 0, \\ x + \sin y = -0.4. \end{cases}$	$\begin{cases} \sin\left(\frac{x+y}{2}\right) - 1.5x = 0.1, \\ x + y = 1. \end{cases}$
9	$\begin{cases} \cos(x+0.5) - y = 2, \\ \sin y - 2x = 1. \end{cases}$	$\begin{cases} \operatorname{tg}(xy) = x^2, \\ 0.7x^2 + 2y^2 = 1. \end{cases}$
10	$\begin{cases} \sin(x+2) - y = 1.5, \\ x + \cos(y-2) = 0.5. \end{cases}$	$\begin{cases} \sin(x+y) - 1.2x = 0.1, \\ x^2 + y^2 = 1. \end{cases}$
11	$\begin{cases} \sin(y+1) - x = 1.2, \\ 2y + \cos x = 2. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.2) = x^2, \\ 0.6x^2 + 2y^2 = 1. \end{cases}$
12	$\begin{cases} \cos(y-1) + x = 0.5, \\ y - \cos x = 3. \end{cases}$	$\begin{cases} \sin\left(\frac{x+y}{2}\right) = 1.5x - 0.1, \\ x + y = 1. \end{cases}$
13	$\begin{cases} \sin y + 2x = 2, \\ y + \cos(x-1) = 0.7. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.4) = x^2, \\ 0.8x^2 + 2y^2 = 1. \end{cases}$
14	$\begin{cases} \cos y + x = 1.5, \\ 2y - \sin(x-0.5) = 1. \end{cases}$	$\begin{cases} \sin(x+y) = 1.2x - 0.1, \\ x^2 + y^2 = 1. \end{cases}$
15	$\begin{cases} \sin(y+0.5) - x = 1, \\ y + \cos(x-2) = 0. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.1) = x^2, \\ 0.9x^2 + 2y^2 = 1. \end{cases}$
16	$\begin{cases} \cos(y+0.5) + x = 0.8, \\ \sin x - 2y = 1.6. \end{cases}$	$\begin{cases} \sin\left(\frac{x+y}{2}\right) - 1.4x = 0, \\ x + y = 1. \end{cases}$
17	$\begin{cases} \sin(y-1) + x = 1.3, \\ y - \sin(x+1) = 0.8. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.1) = x^2, \\ 0.5x^2 + 2y^2 = 1. \end{cases}$
18	$\begin{cases} 2x - \cos(y+1) = 0, \\ y + \sin x = -0.4. \end{cases}$	$\begin{cases} \sin\left(\frac{x+y}{2}\right) = 1.1x - 0.1, \\ x + y = 1. \end{cases}$
19	$\begin{cases} \cos(y+0.5) - x = 2, \\ \sin x - 2y = 1. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.2) = x^2, \\ x^2 + 2y^2 = 1. \end{cases}$

№	Система 1	Система 2
20	$\begin{cases} \sin(y+2) - x = 1.5, \\ y + \cos(x-2) = 0.5. \end{cases}$	$\begin{cases} \sin(x+y) - 1.5x = 0, \\ x^2 + y^2 = 1. \end{cases}$
21	$\begin{cases} \sin(x+1) - y = 1, \\ 2x + \cos y = 2. \end{cases}$	$\begin{cases} \operatorname{tg}(xy) = x^2, \\ 0.5x^2 + 2y^2 = 1. \end{cases}$
22	$\begin{cases} \cos(x-1) + y = 0.8, \\ x - \cos y = 2. \end{cases}$	$\begin{cases} \sin(x+y) = 1.2x - 0.2, \\ x + y = 1. \end{cases}$
23	$\begin{cases} \sin x + 2y = 1.6, \\ x + \cos(y-1) = 1. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.1) = x^2, \\ 0.7x^2 + 2y^2 = 1. \end{cases}$
24	$\begin{cases} \cos x + y = 1.2, \\ 2x - \sin(y-0.5) = 2. \end{cases}$	$\begin{cases} \sin(x+y) - 1.5x = 0.2, \\ x^2 + y^2 = 1. \end{cases}$
25	$\begin{cases} \sin(x+0.5) - y = 1.2, \\ x + \cos(y-2) = 0. \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0.3) = x^2, \\ 0.5x^2 + 2y^2 = 1. \end{cases}$

Задание 2.2

Найти решение системы линейных алгебраических уравнений:

$$\begin{cases} 5x_1 + 2x_2 + x_3 = 2.5, \\ 3x_1 + 7x_2 + 2x_3 = -1.5, \\ 4x_1 + 0.5x_2 + 5x_3 = 11.5 \end{cases}$$

Модель решения системы уравнений представлена на рис. 2.6.

Решение системы уравнений, отображенное на цифровых регистраторах (Display1 – Display3), равно $x_1 = 0.5$, $x_2 = -1$, $x_3 = 2$.

Переходный процесс получения установившегося решения изображен на экране виртуального осциллографа (рис. 2.7).

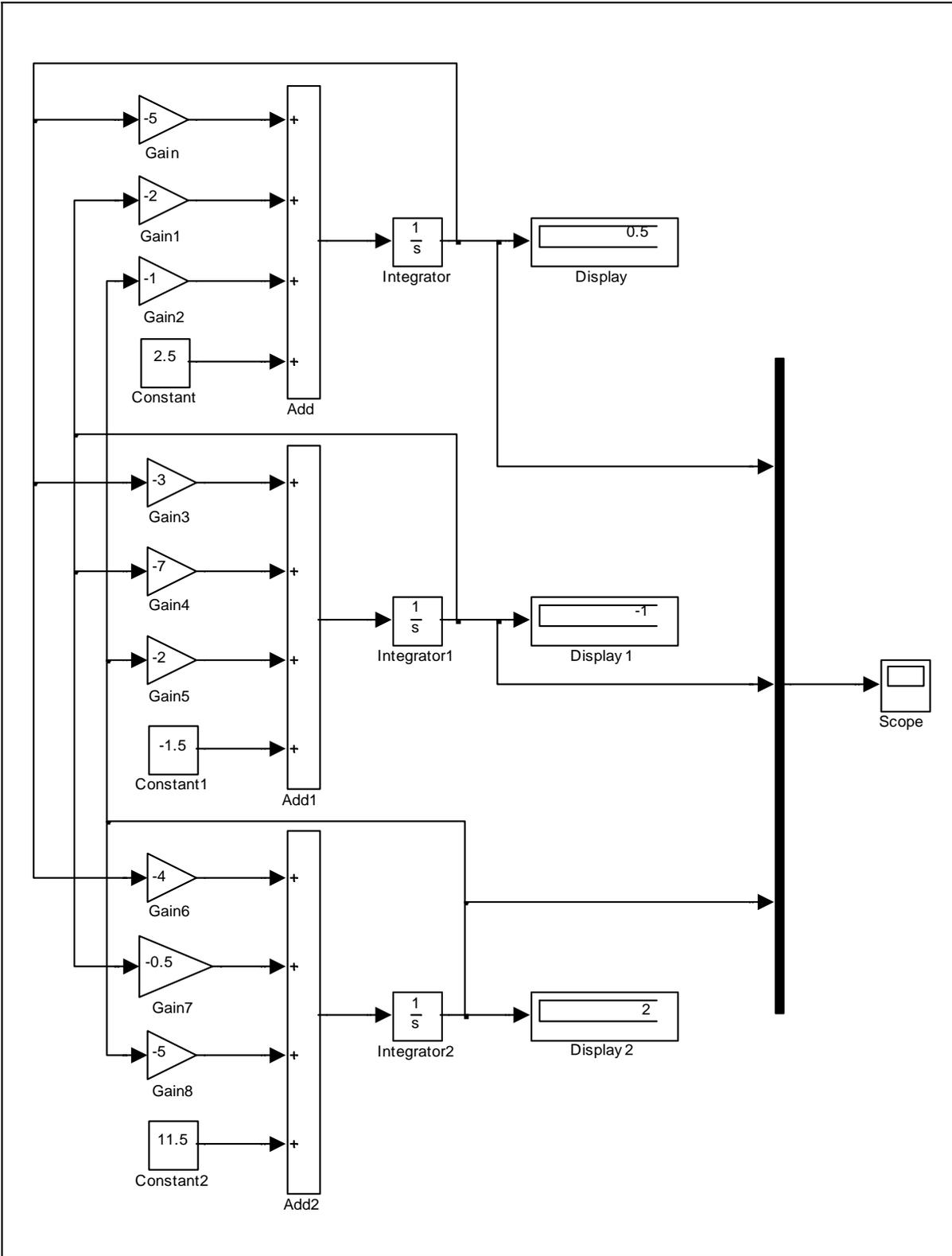


Рис. 2.6 Модель системы линейных алгебраических уравнений 3 порядка

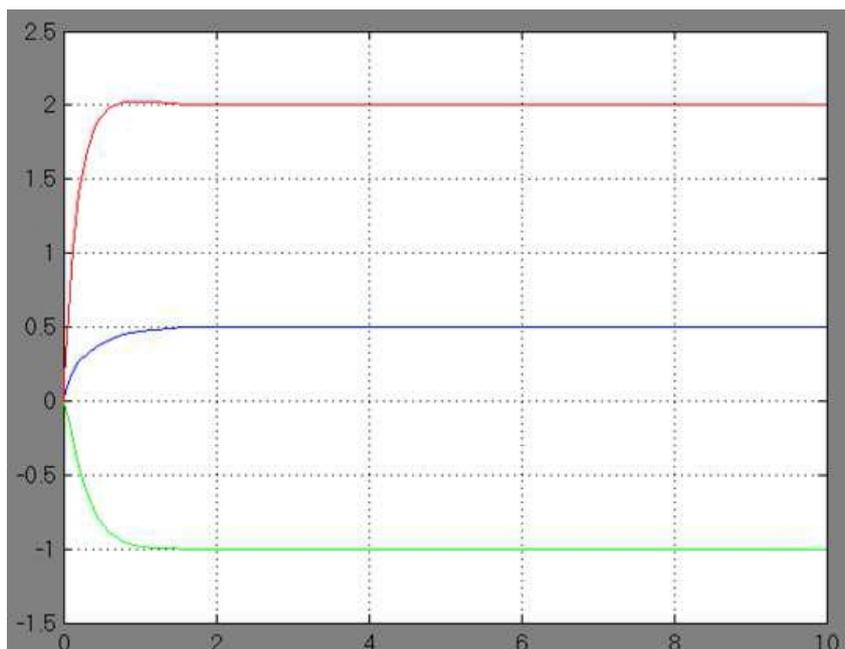


Рис. 2.7 Переходный процесс установления решения системы уравнений

Варианты для выполнения Задания 2.2

Синтезировать структурную схему модели статической системы и получить соответствующее решение.

При построении модели можно использовать следующие блоки: Constant, Integrator, Gain, Mux, Sum, Product, Scope, Display, Algebraic Constraint.

№	Система	№	Система
1	$\begin{cases} 5x_1 + 8x_2 - x_3 = -7, \\ x_1 + 2x_2 + 3x_3 = 1, \\ 2x_1 - 3x_2 + 2x_3 = 9. \end{cases}$	14	$\begin{cases} x_1 - 2x_2 + 3x_3 = 6, \\ 2x_1 + 3x_2 - 4x_3 = 16, \\ 3x_1 - 2x_2 - 5x_3 = 12. \end{cases}$
2	$\begin{cases} x_1 + 2x_2 + x_3 = 4, \\ 3x_1 - 5x_2 + 3x_3 = 1, \\ 2x_1 + 7x_2 - x_3 = 8. \end{cases}$	15	$\begin{cases} 3x_1 + 4x_2 + 2x_3 = 8, \\ 2x_1 - x_2 - 3x_3 = -1, \\ x_1 + 5x_2 + x_3 = 0. \end{cases}$
3	$\begin{cases} 3x_1 + 2x_2 + x_3 = 5, \\ 2x_1 + 3x_2 + x_3 = 1, \\ 2x_1 + x_2 + 3x_3 = 11. \end{cases}$	16	$\begin{cases} 2x_1 - x_2 + 3x_3 = 7, \\ x_1 + 3x_2 - 2x_3 = 0, \\ 2x_2 - x_3 = 2. \end{cases}$

№	Система	№	Система
4	$\begin{cases} x_1 + 2x_2 + 4x_3 = 31, \\ 5x_1 + x_2 + 2x_3 = 29, \\ 3x_1 - x_2 + x_3 = 10. \end{cases}$	17	$\begin{cases} 2x_1 + x_2 + 4x_3 = 20, \\ 2x_1 - x_2 - 3x_3 = 3, \\ 3x_1 + 4x_2 - 5x_3 = -8. \end{cases}$
5	$\begin{cases} 4x_1 - 3x_2 + 2x_3 = 9, \\ 2x_1 + 5x_2 - 3x_3 = 4, \\ 5x_1 + 6x_2 - 2x_3 = 18. \end{cases}$	18	$\begin{cases} x_1 - x_2 = 4, \\ 2x_1 + 3x_2 + x_3 = 1, \\ 2x_1 + x_2 + 3x_3 = 11. \end{cases}$
6	$\begin{cases} 2x_1 - x_2 - x_3 = 4, \\ 3x_1 + 4x_2 - 2x_3 = 11, \\ 3x_1 - 2x_2 + 4x_3 = 11. \end{cases}$	19	$\begin{cases} x_1 + 5x_2 - x_3 = 7, \\ 2x_1 - x_2 - x_3 = 4, \\ 3x_1 - 2x_2 + 4x_3 = 11. \end{cases}$
7	$\begin{cases} x_1 + x_2 + 2x_3 = -1, \\ 2x_1 - x_2 + 2x_3 = -4, \\ 4x_1 + x_2 + 4x_3 = -2. \end{cases}$	20	$\begin{cases} 11x_1 + 3x_2 - x_3 = 2, \\ 2x_1 + 5x_2 - 5x_3 = 0, \\ x_1 + x_2 + x_3 = 2. \end{cases}$
8	$\begin{cases} 3x_1 - x_2 = 5, \\ -2x_1 + x_2 + x_3 = 0, \\ 2x_1 - x_2 + 4x_3 = 15. \end{cases}$	21	$\begin{cases} 7x_1 + 5x_2 + 2x_3 = 18, \\ x_1 - x_2 - x_3 = 3, \\ x_1 + x_2 + 2x_3 = -2. \end{cases}$
9	$\begin{cases} 3x_1 - x_2 + x_3 = 4, \\ 2x_1 - 5x_2 - 3x_3 = -17, \\ x_1 + x_2 - x_3 = 0. \end{cases}$	22	$\begin{cases} 2x_1 + 3x_2 + x_3 = 1, \\ x_1 + x_3 = 0, \\ x_1 - x_2 - x_3 = 2. \end{cases}$
10	$\begin{cases} x_1 + x_2 + x_3 = 2, \\ 2x_1 - x_2 - 6x_3 = -1, \\ 3x_1 - 2x_2 = 8. \end{cases}$	23	$\begin{cases} x_1 - 2x_2 - 2x_3 = 3, \\ x_1 + x_2 - 2x_3 = 0, \\ x_1 - x_2 - x_3 = 1. \end{cases}$
11	$\begin{cases} 2x_1 + x_2 - x_3 = 1, \\ x_1 + x_2 + x_3 = 6, \\ 3x_1 - x_2 + x_3 = 4. \end{cases}$	24	$\begin{cases} 3x_1 + x_2 - 5x_3 = -7, \\ 2x_1 - 3x_2 + 4x_3 = -1, \\ 5x_1 - x_2 + 3x_3 = 0. \end{cases}$
12	$\begin{cases} 2x_1 - x_2 - 3x_3 = 3, \\ 3x_1 + 4x_2 - 5x_3 = 8, \\ 2x_2 + 7x_3 = 17. \end{cases}$	25	$\begin{cases} x_1 - 2x_2 + x_3 = 15, \\ 2x_1 + x_2 + 3x_3 = 9, \\ 2x_1 + 3x_2 + 2x_3 = -2. \end{cases}$
13	$\begin{cases} x_1 + 5x_2 + x_3 = -7, \\ 2x_1 - x_2 - x_3 = 0, \\ x_1 - 2x_2 - x_3 = 2. \end{cases}$		

Лабораторная работа №3 Построение непрерывно-детерминированных моделей динамических объектов и процессов

ЦЕЛЬ РАБОТЫ

- Определение динамических объектов и процессов.
- Моделирование динамических объектов и процессов.

ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Перед выполнением лабораторной работы необходимо ознакомиться с темами:

- Понятия о динамических объектах и процессах и способах их моделирования.
- Библиотека блоков SIMULINK [1, п.9].

Ответить на вопросы:

1. Какие объекты и процессы называются динамическими?
2. Какие математические модели используются для описания динамических объектов и процессов?
3. Каково назначение и описание параметров блоков, используемых при выполнении лабораторной работы?

Моделирование динамических объектов и процессов

Исследуемые объекты и процессы называются *динамическими*, если состояния, характеризующие эти объекты и процессы, изменяются с течением времени, то есть являются функциями, зависящие от времени.

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

Реализацию таких математических моделей в системе Simulink рассмотрим на ряде примеров.

ПРАКТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Задание 3.1

Построить модель физического маятника, находящегося под воздействием экспоненциально-затухающего косинусоидального возмущения.

Уравнение движения такого маятника имеет вид:

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

$$y(0) = y_0, \quad \left. \frac{dy(t)}{dt} \right|_{t=0} = y'_0$$

Выбирая числовые значения параметров, например: $a_1 = a_2 = 0.1$, $a_3 = -5$, $a_4 = 1$, $a_5 = 0.1$, получаем следующее уравнение:

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

$$y(0) = -1.5, \quad y'(0) = 2.$$

Проанализируем полученную математическую модель. Входным сигналом модели является внешнее воздействие на модель в виде функции $-5e^{-t} \cdot \cos(0.1t)$ из правой части уравнения. Выходным сигналом модели является искомая функция $y(t)$, которая получается двукратным интегрированием функции $y'(t)$, которую выражаем из уравнения в виде

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

Структурная схема модели верхнего уровня представлена на рис. 3.1.

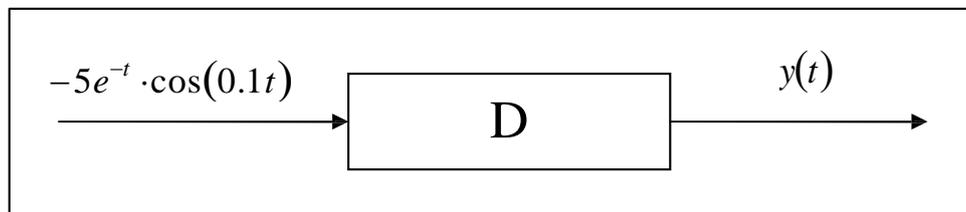


Рис. 3.1 Структурная схема модели верхнего уровня

Для построения модели сначала реализуем вычисление значения функции $-5e^{-t} \cdot \cos(0.1t)$, которое сформируем с помощью блоков Ramp (генератор аргумента t), блока Math Function, настроенного на реализацию экспоненциальной функции и блока Trigonometric Function, настроенного на вычисления функции $\cos u$.

На вход первого интегратора Integrator подается сгенерированное при помощи сумматора Sum выражение для функции $y'(t)$. Выходным сигналом из блока является функция $y'(t)$, которая направляется на вход

второго интегратора Integrator1, результатом работы которого окажется искомая функция $y(t)$. Коэффициенты уравнения при $y(t)$ и $y'(t)$ устанавливаются в окнах параметров масштабных блоков (Gain3 – Gain4). Начальные условия для функций задаются в окнах параметров соответствующих интеграторов (Integrator – Integrator1).

Структурная схема модели принимает вид, показанный на рис. 3.2.

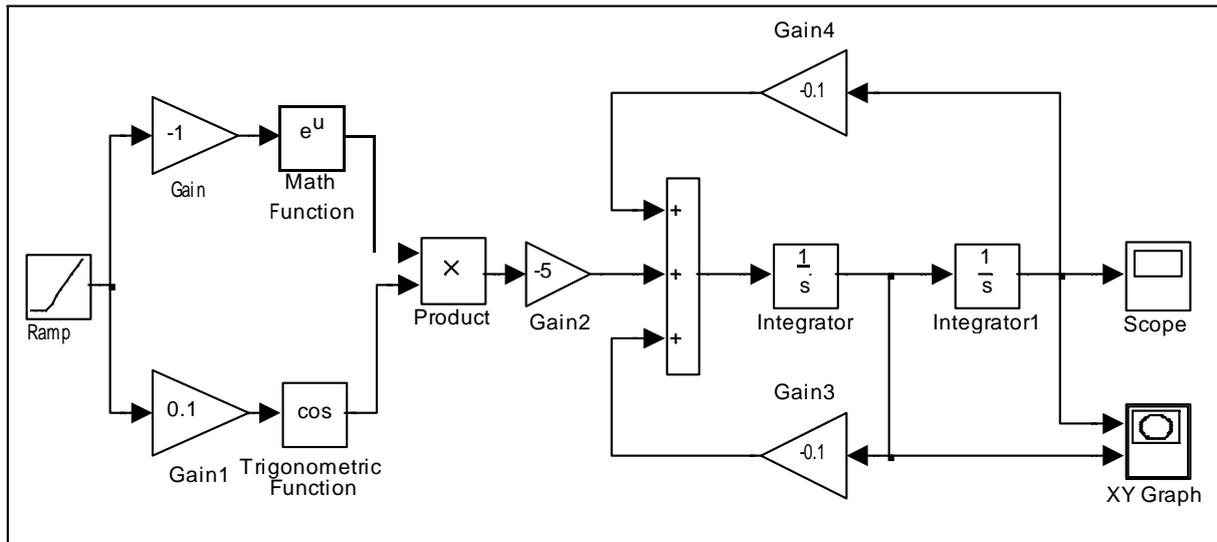


Рис. 3.2 Модель физического маятника

Результат работы модели показан на экране виртуального осциллографа Scope (рис. 3.3).

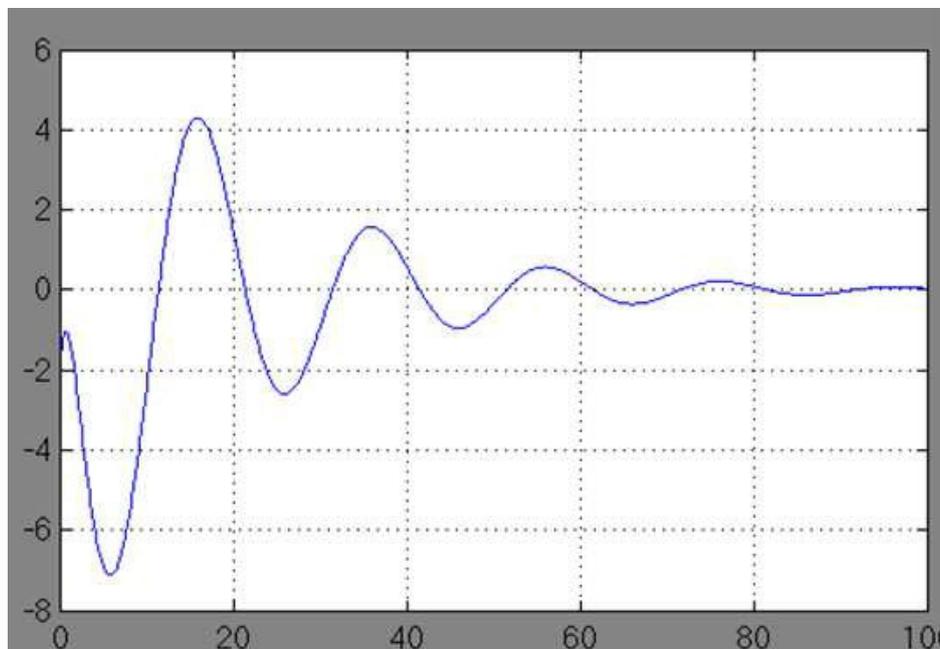


Рис. 3.3 График движения маятника во времени

Параметрический график зависимости производной сигнала от самого сигнала (фазовый портрет маятника) отображен при помощи блока XY Graph на рис. 3.4.

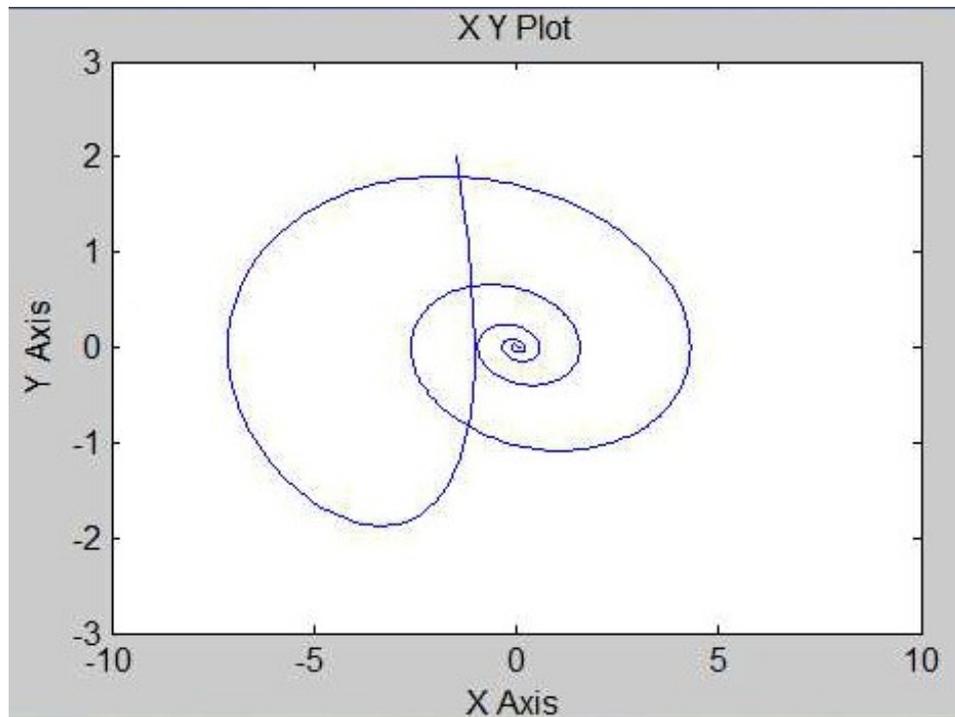


Рис. 3.4 Фазовый портрет маятника

Варианты для выполнения Задания 3.1

Построить модель динамической системы, рассчитать траекторию движения системы и представить ее в виде графика и фазового портрета.

- №1. Проанализировать поведение маятника с вязким трением $y' + 0.5y' + 2 \sin y = e^{-2t}$, $y(0) = 1.5$, $y'(0) = -2$.
- №2. Проанализировать траекторию движения качелей, подчиненного закону $y' + 0.3y' + (2 + \cos t)y = e^{-3t}$, $y(0) = 2$, $y'(0) = -1.5$.
- №3. Проанализировать колебания материальной точки, подчиненные закону $y' + 0.25 \cos t \cdot y = e^{-2t} \sin(0.5t)$, $y(0) = -2$, $y'(0) = 1.5$.
- №4. Проанализировать траекторию перемещения груза, подчиненного закону $y' + 0.2y' + 0.3 \sin y + 0.4y = e^{-3t}$, $y(0) = -2$, $y'(0) = -1.5$.
- №5. Проанализировать поведение решения уравнения Ван-дер-Поля $y' + 0.1(y^2 - 1)y' + 0.2y = e^{-2t}$, $y(0) = 1$, $y'(0) = -3$.

№6. Проанализировать поведение электрического колебательного контура
 $y' + 0.12y = e^{-t} \cos 2t, \quad y(0) = 3, \quad y'(0) = 2.$

№7. Проанализировать поведение модели, отображающей динамику боя,
 $y' - 10y = e^{-2t} \sin 3t, \quad y(0) = 2, \quad y'(0) = 2.$

№8. Проанализировать модель успокоения маятника
 $y' + \pi^2 y = \pi e^{-2t} \cos t, \quad y(0) = 3, \quad y'(0) = 1.$

№9. Проанализировать модель движения автомобиля при торможении
 $y' + 4y = e^{-3t} \sin 2t, \quad y(0) = -2, \quad y'(0) = -1.$

№10. Проанализировать модель генетического контроля бактерий
 $y' + 3.5y - \left(\frac{2.2}{y} - 0.1 \right) = \sin 0.2t, \quad y(0) = 5, \quad y'(0) = -2.$

№11. Проанализировать модель работы электродвигателя
 $y' = e^{-3t+2} \cos \left(\frac{\pi}{6} + 2t \right), \quad y(0) = 3, \quad y'(0) = -1.$

№12. Проанализировать модель работы ядерного реактора
 $y' + 0.25 \cos^2 t \cdot y = e^{-2t}, \quad y(0) = 25, \quad y'(0) = 0.$

№13. Проанализировать поведение маятника с вязким трением
 $y' + 0.35y' + 3 \sin y = e^{-3t}, \quad y(0) = 1.3, \quad y'(0) = -2.3.$

№14. Проанализировать траекторию движения качелей, подчиненного
закону $y' + 0.2y' + (5 + \cos 2t)y = e^{-2t}, \quad y(0) = 2.2, \quad y'(0) = -1.2.$

№15. Проанализировать колебания материальной точки, подчиненные
закону $y' + 0.25 \sin 2t \cdot y = e^{-2t} \cos(0.3t), \quad y(0) = -2.5, \quad y'(0) = 1.1.$

№16. Проанализировать траекторию перемещения груза, подчиненного
закону $y' + 0.3y' + 0.2 \cos y + 0.5y = e^{-2t}, \quad y(0) = -2.7, \quad y'(0) = 1.2.$

№17. Проанализировать поведение решения уравнения Ван-дер-Поля
 $y' + 0.2(y^2 + 1)y' + 0.3y = e^{-3t}, \quad y(0) = 1.7, \quad y'(0) = -3.1.$

№18. Проанализировать поведение электрического колебательного
контура $y' + 0.24y = e^{-3t} \sin 3t, \quad y(0) = 1, \quad y'(0) = 1.3.$

№19. Проанализировать поведение модели, отображающей динамику боя,
 $y' - 15y = e^{-3t} \cos 4t, \quad y(0) = 3, \quad y'(0) = 0.$

№20. Проанализировать модель успокоения маятника
 $y' + \pi^2 y = \pi e^{-3t} \sin 2t, \quad y(0) = 0.3, \quad y'(0) = 1.5.$

№21. Проанализировать модель генетического контроля бактерий
 $y' + 1.5y - \left(\frac{1.2}{y} - 0.5 \right) = \sin 0.3t, \quad y(0) = 3, \quad y'(0) = -1.$

№22. Проанализировать модель работы электродвигателя

$$y' = e^{-2t-2} \cos\left(\frac{\pi}{3} + t\right), \quad y(0) = 2.5, \quad y'(0) = -1.3.$$

№23. Проанализировать модель работы ядерного реактора

$$y' + 0.36 \sin^2 t \cdot y = e^{-3t}, \quad y(0) = 36, \quad y'(0) = 0.$$

№24. Проанализировать модель движения автомобиля при торможении

$$y' + 6y = e^{-2t} \cos 3t, \quad y(0) = -1, \quad y'(0) = -2.$$

№25. Проанализировать поведение модели, отображающей динамику боя,

$$y' - 9y = e^{-4t} \cos 3t, \quad y(0) = 2, \quad y'(0) = 0.$$

Задание 3.2

Построить модель динамической системы, описываемой дифференциальным уравнением 3-го порядка.

Пусть задано дифференциальное уравнение:

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

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

Структурная схема модели динамической системы, построенная по правилам аналоговой вычислительной техники, приведена на рис. 3.5.

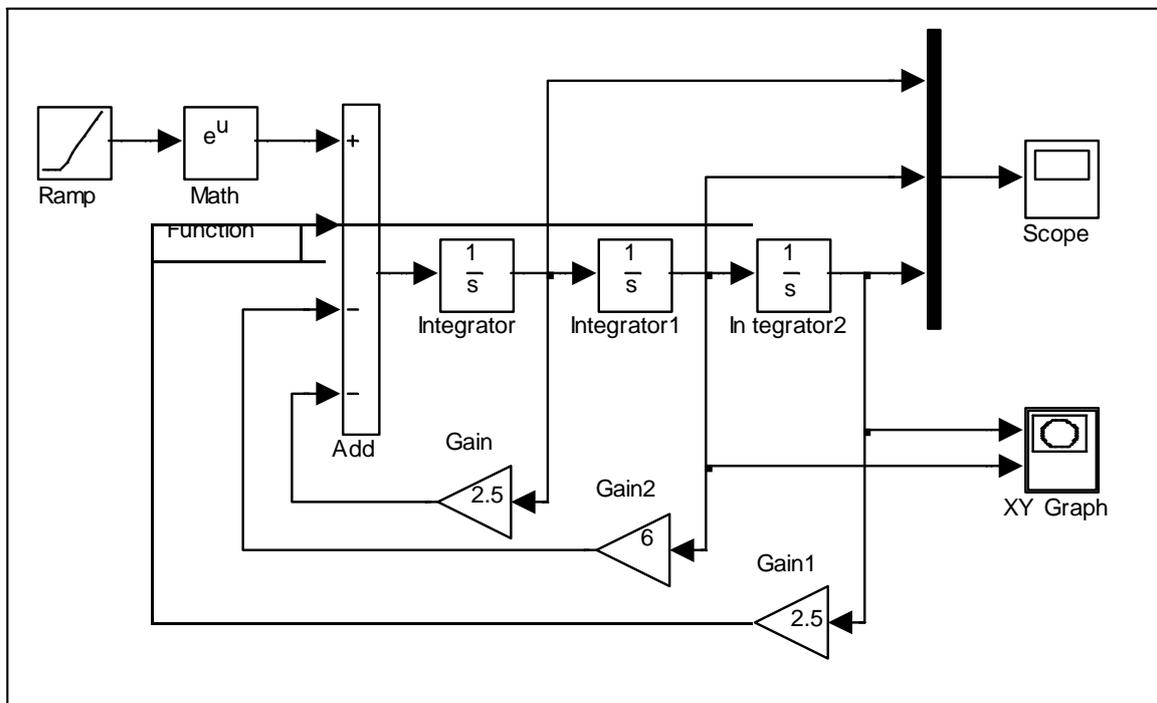


Рис. 3.5 Структурная схема модели динамической системы

При построении модели иногда требуется изменить направление входа и выхода у какого-либо блока. Для этого в контекстном меню блока нужно выбрать команду `Format \ Rotate Block \ Clockwise` и выполнить ее столько раз, пока вход и выход не примут нужное направление, например блоки `Gain`, `Gain1`, `Gain2`.

Визуализация переходного процесса показана на экране виртуального осциллографа (рис. 3.6). Фазовый портрет системы на экране виртуального регистратора (`XY Graph`) приведен на рис. 3.7.

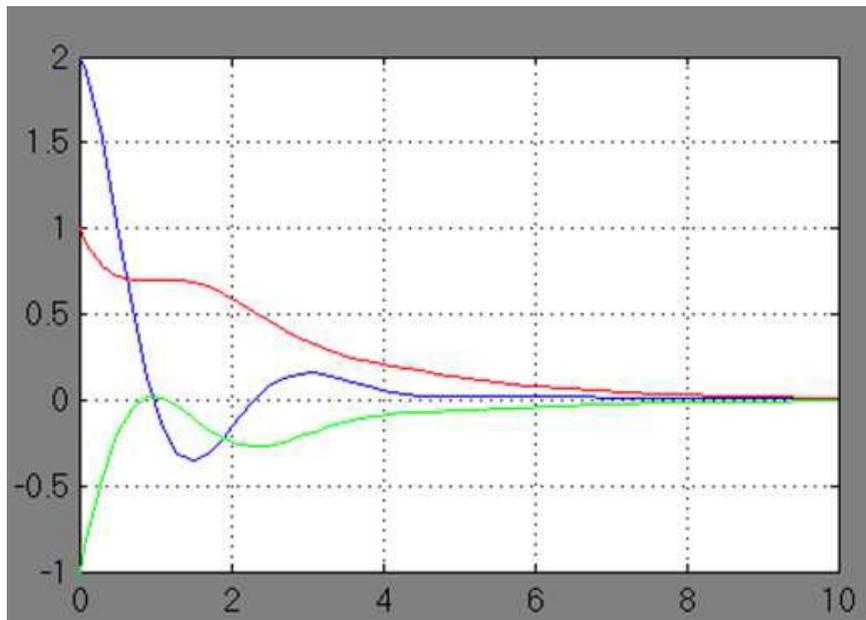


Рис. 3.6 Переходный процесс системы

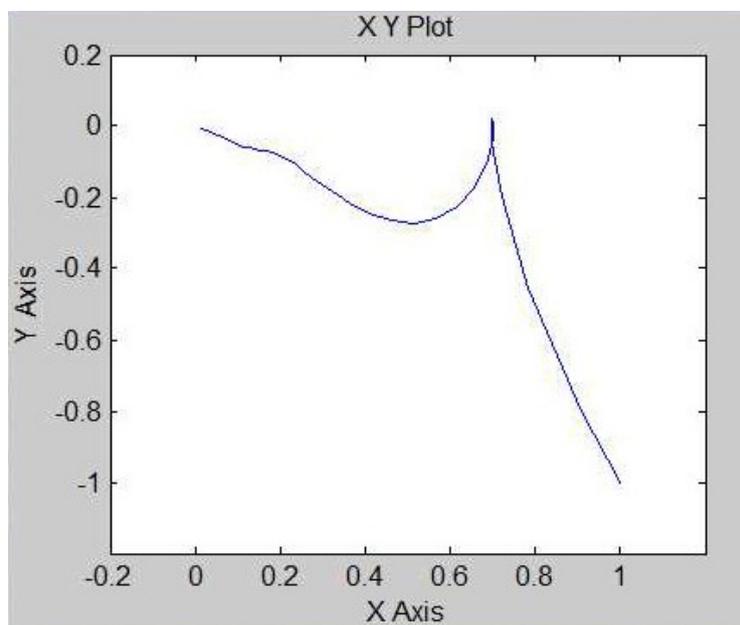


Рис. 3.7 Фазовый портрет системы

Варианты для выполнения Задания 3.2

Построить модель динамической системы, описываемой дифференциальным уравнением 3-го порядка.

Визуализировать функцию, первую и вторую производные решения. Построить фазовый портрет системы по функции и первой производной.

№1. $y''' + 0.5y'' + 0.9y' + 2y = \cos t, \quad y(0)=2, \quad y'(0)=-1, \quad y''(0)=0.$

№2. $y''' - 1.5y'' - 6y' - 2.5y = \sin t, \quad y(0)=-2, \quad y'(0)=1, \quad y''(0)=0.$

№3. $y'' + 3y' - 24y - 10y = e^{-t}, \quad y(0)=2, \quad y'(0)=1, \quad y''(0)=0.$

№4. $y''' - 1.5y'' - 6y' + 6y = \cos t, \quad y(0)=-2, \quad y'(0)=-1, \quad y''(0)=0.$

№5. $y''' + 3y'' - 24y' - 3y = \sin t, \quad y(0)=3, \quad y'(0)=2, \quad y''(0)=0.$

№6. $y'' - 1.5y' - 6y' + 5y = e^{-t}, \quad y(0)=-3, \quad y'(0)=-2, \quad y''(0)=0.$

№7. $y''' - 3y'' - 24y' - 8y = \cos t, \quad y(0)=-3, \quad y'(0)=2, \quad y''(0)=0.$

№8. $y''' - 1.5y'' - 6y' + 0.5y = \sin t, \quad y(0)=3, \quad y'(0)=-2, \quad y''(0)=0.$

№9. $y'' + 3y' - 24y' + y = e^{-t}, \quad y(0)=1, \quad y'(0)=-3, \quad y''(0)=0.$

№10. $y''' + 3y'' - 24y' + 10y = \cos t, \quad y(0)=-1, \quad y'(0)=-3, \quad y''(0)=0.$

№11. $y''' - 1.5y'' - 6y' + 4y = \sin t, \quad y(0)=-1, \quad y'(0)=3, \quad y''(0)=0.$

№12. $y'' - 3y' - 24y' - 5y = e^{-t}, \quad y(0)=1, \quad y'(0)=3, \quad y''(0)=0.$

№13. $y''' - 3y'' - 24y' - 3y = \cos t, \quad y(0)=4, \quad y'(0)=-1, \quad y''(0)=0.$

№14. $y''' - 3y'' - 24y' + 10y = \sin t, \quad y(0)=-4, \quad y'(0)=-1, \quad y''(0)=0.$

№15. $y'' - 2y' - 4y' + 7y = e^{-t}, \quad y(0)=4, \quad y'(0)=1, \quad y''(0)=0.$

№16. $y''' + 4.5y'' - 10.5y = \cos t, \quad y(0)=-4, \quad y'(0)=1, \quad y''(0)=0.$

№17. $y''' + 3y'' - 2y = \sin t, \quad y(0)=-1, \quad y'(0)=-4, \quad y''(0)=0.$

№18. $y'' - 12y' - 5y = e^{-t}, \quad y(0)=-1, \quad y'(0)=4, \quad y''(0)=0.$

№19. $y''' - 3y'' + 1.5y = \cos t, \quad y(0)=1, \quad y'(0)=-4, \quad y''(0)=0.$

№20. $y''' + 4.5y'' - 2y = \sin t, \quad y(0)=1, \quad y'(0)=4, \quad y''(0)=0.$

№21. $y'' - 12y' + 6y = e^{-t}, \quad y(0)=-3, \quad y'(0)=4, \quad y''(0)=0.$

№22. $y''' - 3y'' + 2.5y = \cos t, \quad y(0)=-3, \quad y'(0)=-4, \quad y''(0)=0.$

№23. $y''' + 3y'' - 3.5y = \sin t, \quad y(0)=3, \quad y'(0)=4, \quad y''(0)=0.$

№24. $y'' - 12y' + 10y = e^{-t}, \quad y(0)=3, \quad y'(0)=-4, \quad y''(0)=0.$

№25. $y''' - 4y'' + 2y = \cos t, \quad y(0)=-4, \quad y'(0)=-2, \quad y''(0)=0.$

Задание 3.3

Построить 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 имеют вид:

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

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

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

Входными сигналами модели являются функции времени, заданные в правых частях равенств, выходными сигналами – значения $x(t)$ и $y(t)$.

Структурная схема модели траектории камня приведена на рис. 3.8.

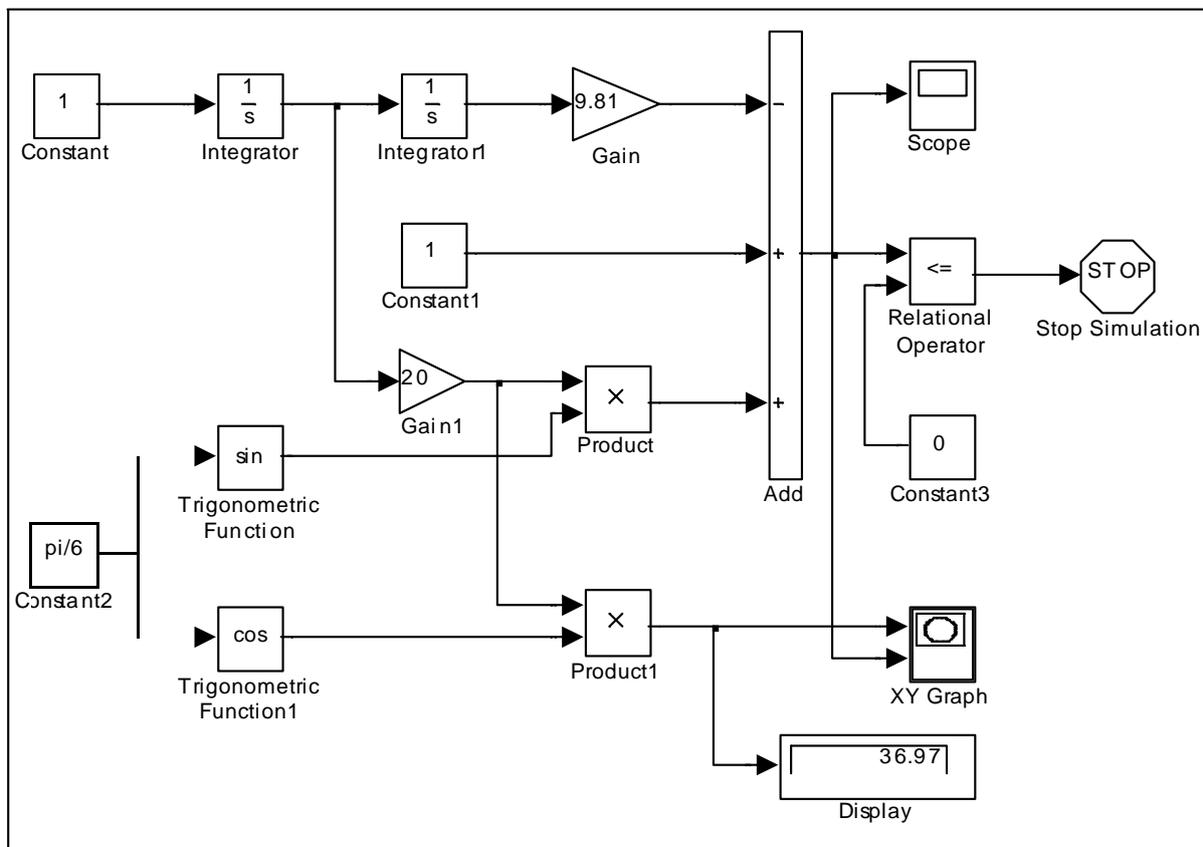


Рис. 3.8 Структурная схема модели траектории камня

Начальная высота полета камня задается блоком Constant1, начальная скорость полета задается блоком Gain1, угол наклона полета задается блоком Constant2, ускорение земного тяготения – блоком Gain.

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 станет равной нулю.

Показание цифрового регистратора Display соответствует длине пути по горизонтали, пройденной камнем до момента падения на землю. На рис. 3.9 показана траектория камня на экране виртуального регистратора. На рис. 3.10 показан график изменения во времени высоты полета камня.

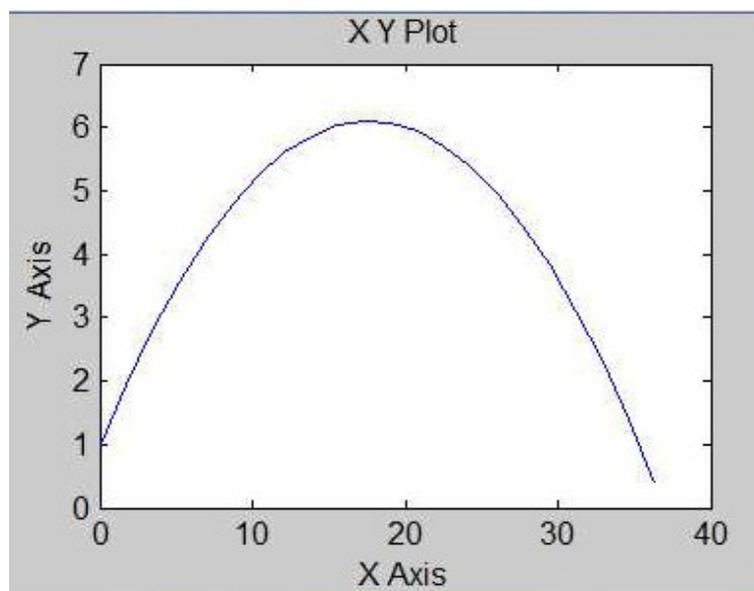


Рис. 3.9. Траектория полета камня

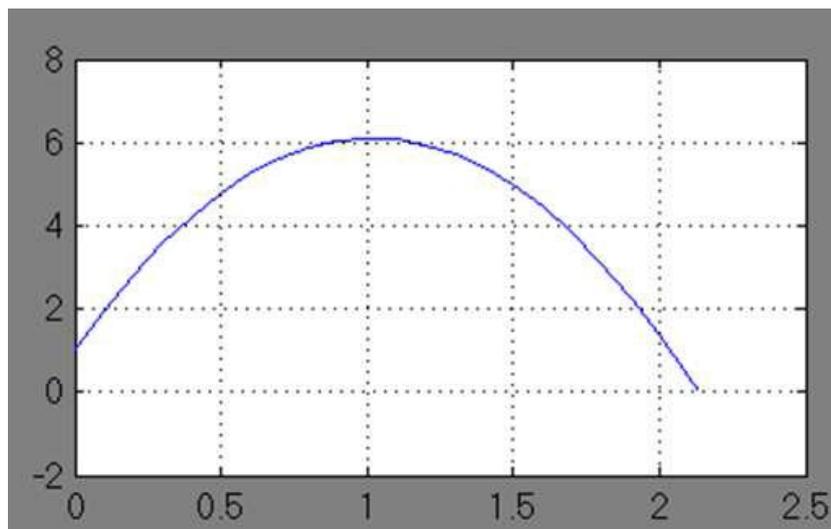


Рис. 3.10. График изменения во времени высоты полета камня

Варианты для выполнения Задания 3.3

- Вариант №1. $x_0 = 1.0, y_0 = 0.0, A = 5, B = 2, \omega = \pi/5.$
Вариант №6. $x_0 = 0.0, y_0 = 1.0, A = 2, B = 5, \omega = \pi/3.$
Вариант №11. $x_0 = -1.0, y_0 = 0.0, A = 3, B = 4, \omega = \pi/4.$
Вариант №16. $x_0 = 0.0, y_0 = -1.0, A = 4, B = 3, \omega = \pi/2.$
Вариант №21. $x_0 = 1.0, y_0 = 1.0, A = 1, B = 6, \omega = \pi/6.$

Варианты исходных данных применить для решения задачи.

Материальная точка движется на плоскости по закону

$$x(t) = x_0 + A \sin(\omega t), \quad y(t) = y_0 + B \cos(\omega t).$$

Реализовать модель траектории движения материальной точки в виде зависимости координат от времени и на фазовой плоскости.

- Вариант №2. $a = 10.$
Вариант №7. $a = 25.$
Вариант №12. $a = 30.$
Вариант №17. $a = 35.$
Вариант №22. $a = 40.$

Варианты исходных данных применить для решения задачи.

Вещество массой a разлагается на два составляющих его вещества. Скорость образования каждого из них пропорциональна массе неразложившегося вещества в соответствии с законом

$$x(t) = \frac{a(1-2^{-t})}{4}, \quad y(t) = \frac{3a(1-2^{-t})}{4},$$

Реализовать модель изменения масс веществ в зависимости от времени. Сколько времени понадобится для полного разложения вещества на составляющие?

- Вариант №3. $a = 50.0, v_0 = 0.1, m = 10, k = 0.02.$
Вариант №8. $a = 40.0, v_0 = 0.2, m = 15, k = 0.02.$
Вариант №13. $a = 30.0, v_0 = 0.3, m = 20, k = 0.02.$
Вариант №18. $a = 20.0, v_0 = 0.4, m = 25, k = 0.02.$
Вариант №23. $a = 10.0, v_0 = 0.5, m = 30, k = 0.02.$

Варианты исходных данных применить для решения задачи.

Материальная точка массы m притягивается центром с силой, пропорциональной расстоянию от точки до центра.

Движение начинается из некоторой точки на расстоянии a от центра с начальной скоростью v_0 .

$$x(t) = a \cos \frac{k}{\sqrt{m}} t, \quad y(t) = \frac{v_0 \sqrt{m}}{k} \sin \frac{k}{\sqrt{m}} t,$$

Реализовать модель траектории движения материальной точки в виде зависимости координат от времени и на фазовой плоскости.

Вариант №4. $x_0 = 1.0, \quad y_0 = 0.0, \quad a = 10, \quad \omega = 0.4\pi.$

Вариант №9. $x_0 = 0.0, \quad y_0 = 1.0, \quad a = 20, \quad \omega = 0.5\pi.$

Вариант №14. $x_0 = -1.0, \quad y_0 = 0.0, \quad a = 30, \quad \omega = 0.3\pi.$

Вариант №19. $x_0 = 0.0, \quad y_0 = -1.0, \quad a = 40, \quad \omega = 0.2\pi.$

Вариант №24. $x_0 = 1.0, \quad y_0 = 1.0, \quad a = 10, \quad \omega = 0.6\pi.$

Варианты исходных данных применить для решения задачи.

Материальная точка движется на плоскости по закону

$$x(t) = x_0 + (a - t) \sin(\omega t), \quad y(t) = y_0 + (a - t) \cos(\omega t).$$

Реализовать модель траектории движения материальной точки в виде зависимости координат от времени и на фазовой плоскости.

Вариант №5. $v_0 = 10.0, \quad y_0 = 1.5, \quad \alpha = 20^\circ, \quad g = 9.81.$

Вариант №10. $v_0 = 15.0, \quad y_0 = 2.0, \quad \alpha = 25^\circ, \quad g = 9.81.$

Вариант №15. $v_0 = 20.0, \quad y_0 = 2.5, \quad \alpha = 40^\circ, \quad g = 9.81.$

Вариант №20. $v_0 = 25.0, \quad y_0 = 3.0, \quad \alpha = 45^\circ, \quad g = 9.81.$

Вариант №25. $v_0 = 30.0, \quad y_0 = 1.8, \quad \alpha = 35^\circ, \quad g = 9.81.$

Варианты исходных данных применить для решения задачи.

Траектория полета снаряда, вылетевшего с начальной скоростью v_0 под углом α к горизонту из точки, находящейся на высоте y_0 над уровнем земли, определяется в соответствии с законом

$$y(t) = y_0 + v_0 \cdot \sin(\alpha) \cdot t - g \cdot \frac{t^2}{2}, \quad x(t) = v_0 \cdot \cos(\alpha) \cdot t,$$

Реализовать модель траектории полета снаряда под действием силы тяжести и определить расстояние от стартовой точки до точки падения снаряда.

Лабораторная работа №4 Построение дискретно-детерминированных моделей динамических объектов и процессов

ЦЕЛЬ РАБОТЫ

- Аппроксимация непрерывных и дискретных сигналов.

ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Перед выполнением лабораторной работы необходимо ознакомиться с темами:

- Задача аппроксимации непрерывных и дискретных сигналов.
- Метод наименьших квадратов.
- Модифицированный метод равных площадей.
- Блочно-импульсная аппроксимация.
- Библиотека блоков SIMULINK [1, п.9].

Ответить на вопросы:

1. Что называется задачей аппроксимацией функций?
2. Что называется аппроксимирующим полиномиальным спектром?
3. В чем заключается метод наименьших квадратов? Что он позволяет находить?
4. В чем заключается модифицированный метод равных площадей? Что он позволяет находить?
5. В чем состоит блочно-импульсная аппроксимация? Какие функции выбираются в качестве базисных?
6. Каково назначение и описание параметров блоков, используемых при выполнении лабораторной работы?
7. Как фрагмент модели превратить в подсистему? Для чего это делается?

Аппроксимация непрерывных и дискретных сигналов

При решении многих прикладных задач используются методы приближения довольно сложных для исследования функций более простыми, хорошо изученными функциями (например, многочленами). При этом исследуемая функция может быть задана как в аналитическом, так и дискретном виде (в виде экспериментальной таблицы большой размерности). Процедура замены заданной функции другой, более простой, функцией называется *аппроксимацией*. На практике такие задачи возникают, например, при преобразовании аналогового сигнала в цифровой и, наоборот, из цифрового сигнала в аналоговый.

Сформулируем задачу аппроксимации. Пусть непрерывная (дискретная) функция $x(t)$ задана на некотором интервале изменения аргумента $t \in [0, T]$. Требуется заменить эту функцию некоторой простой, но хорошо исследуемой функцией, сохранив поведение функции на указанном промежутке.

Для решения поставленной задачи выбирается некоторая система базисных функций $S(t) = \{s_k(t), k = \overline{0, n}\}$, заданных на том же интервале изменения аргумента. При этом заданная функция представляется некоторым обобщенным многочленом в виде разложения по базисным функциям

$$P_n(t) = \sum_{k=0}^n A_k s_k(t), \quad (4.1)$$

где коэффициенты A_k подлежат определению, при этом систему коэффициентов $A_k, k = \overline{0, n}$ обычно называют *аппроксимирующим полиномиальным спектром* (АПС) сигнала $x(t)$.

Решение задачи можно представить в виде структурной схемы модели верхнего уровня в виде последовательности двух блоков (рис. 4.1). Входными сигналами первого блока являются заданная функция $x(t)$ и выбранная система базисных функций $S(t)$. Аппроксимирующий спектр $A_k, k = \overline{0, n}$, являющийся выходом из первого блока, подается на вход второго блока для построения аппроксимирующей функции $P_n(t)$.

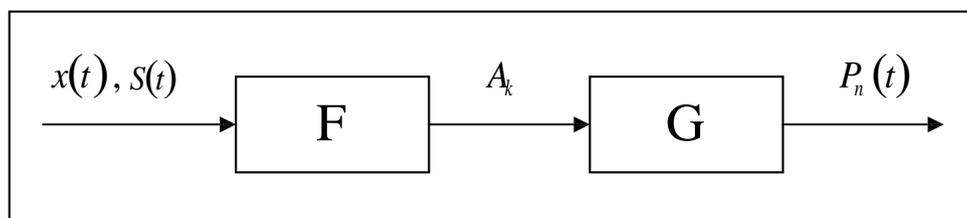


Рис. 4.1 Структурная схема аппроксимации функций

Для нахождения аппроксимирующего спектра $A_k, k = \overline{0, n}$, а следовательно и самой функции $P_n(t)$, рассмотрим два метода: метод наименьших квадратов и модифицированный метод равных площадей. В частности, для анализа дискретных динамических систем удобнее использовать метод равных площадей.

Метод наименьших квадратов

Задача определения аппроксимирующего спектра методом наименьших квадратов формулируется следующим образом: определить систему коэффициентов $A_k \quad k = \overline{0, n}$, которая минимизирует квадратичное

отклонение заданной функции от аппроксимирующего многочлена

$$\frac{1}{T} \int_0^T [P(t) - x(t)]^2 dt \rightarrow \min \quad \text{или} \quad \frac{1}{T} \int_0^T \left[\sum_{k=0}^n A_k s_k(t) - x(t) \right]^2 dt \rightarrow \min.$$

Необходимые условия нахождения минимума данной функции можно получить, если частные производные функции по всем A_k приравнять к нулю, что приводит к следующей системе алгебраических уравнений

$$\begin{cases} \int_0^T \left[\sum_{k=0}^n A_k s_k(t) - x(t) \right] \cdot s_0(t) dt = 0, \\ \int_0^T \left[\sum_{k=0}^n A_k s_k(t) - x(t) \right] \cdot s_1(t) dt = 0, \\ \dots \\ \int_0^T \left[\sum_{k=0}^n A_k s_k(t) - x(t) \right] \cdot s_n(t) dt = 0. \end{cases}$$

Систему уравнений для определения вектора аппроксимирующего спектра $A = \{A_k, k = \overline{0, n}\}$ можно записать в матричном виде

$$W \cdot A = Q, \quad \text{или} \quad A = W^{-1} \cdot Q, \quad (4.2)$$

где элементы матриц W и вектора Q вычисляются следующим образом:

$$w_{ij} = \int_0^T s_{i-1}(t) s_{j-1}(t) dt, \quad q_i = \int_0^T s_{i-1}(t) x(t) dt, \quad (4.3)$$

$$i = \overline{1, n+1}, \quad j = \overline{1, n+1}.$$

Сама же аппроксимирующая функция получается по формуле (4.1).

Модифицированный метод равных площадей

Метод равных площадей является более простым методом аппроксимации, основанный на определении коэффициентов аппроксимирующего многочлена (4.1), исходя из условия равенства нулю интегралов функций ошибки аппроксимации на некоторой системе интервалов изменения аргумента.

Суть модифицированного метода равных площадей заключается в следующем. Интервал изменения аргумента t разбивается на $(n + 1)$ часть (систему подинтервалов). Уравнение аппроксимирующей кривой подбирается так, чтобы площади, ограничиваемые заданной и аппроксимирующей функциями, были равны между собой для каждого участка в отдельности. Это приводит к системе линейных алгебраических уравнений, решение которой определяет аппроксимирующий спектр.

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

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

Например, при выборе системы подинтервалов вида $\left[\frac{iT}{n+1}, \frac{(i+1)T}{n+1} \right]$, $i = \overline{1, n+1}$, матрица W и вектор Q вычисляются по формулам:

$$w_{ij} = \int_0^{\frac{iT}{n+1}} s_{j-1}(t) dt, \quad q_i = \int_0^{\frac{iT}{n+1}} x(t) dt, \quad (4.4)$$

$$i = \overline{1, n+1}, \quad j = \overline{1, n+1}.$$

Вектор аппроксимирующего спектра $A = \{A_k, k = \overline{0, n}\}$ определяется по формуле (4.2), сама же аппроксимирующая функция по формуле (4.1).

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.

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

$$v_i(t) = \sigma(t - (i-1)h) - \sigma(t - ih). \quad (4.5)$$

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

$$\sigma(t) = \begin{cases} 0, & \text{если } t < 0, \\ 1, & \text{если } t \geq 0. \end{cases} \quad (4.6)$$

Аппроксимация сигнала по системе блочно-импульсных функций определяется по формулам (4.3) – (4.2) как

$$P_0(t) = \sum_{i=1}^m X_0(i) v_i(t), \quad \text{где} \quad X_0(i) = \frac{1}{h} \int_{(i-1)h}^{ih} x(t) dt. \quad (4.7)$$

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

$$w_i(t) = \begin{cases} \frac{2t}{h} + 1 - 2i \\ \frac{2t}{h} - 1 + 2i \end{cases} v_i(t), \quad (4.8)$$

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

$$P_1(t) = \sum_{i=1}^m (X_0(i) v_i(t) + X_1(i) w_i(t)). \quad (4.9)$$

Структурная схема модели верхнего уровня при аппроксимации функций с помощью блочно-импульсных базисных функций приведена на рис. 4.2.

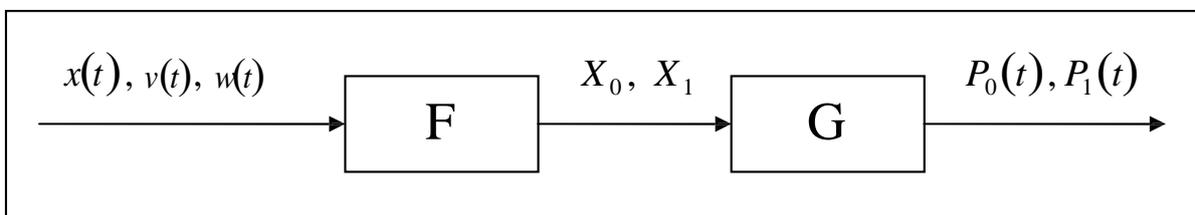


Рис. 4.2 Структурная схема блочно-импульсной аппроксимации

ПРАКТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Задание 4.1

Методом наименьших квадратов аппроксимировать сигнал $x(t) = \cos \pi t^2$ на интервале $[0,1]$. В качестве базисных функций взять степенные функции до 4-го порядка включительно.

Выполнение поставленной задачи разобьем на несколько этапов.

1. Сгенерируем непрерывный сигнал $x(t) = \cos \pi t^2$ на интервале $[0,1]$.
2. Сгенерируем систему степенных функций $S(t) = \{s_k(t) = t^k, k = \overline{0,4}\}$.
3. Определим вектор аппроксимирующего спектра $A = \{A_k, k = \overline{0,n}\}$ по формулам (4.3), (4.2).
4. Сгенерируем аппроксимирующий многочлен $P_n(t) = \sum_{k=0}^n A_k s_k(t)$.

Первый этап. Модель получения непрерывного сигнала $x(t) = \cos \pi t^2$ на интервале $[0,1]$ формируется известным по лабораторной работе №1 способом, результат представлен на рис. 4.3.

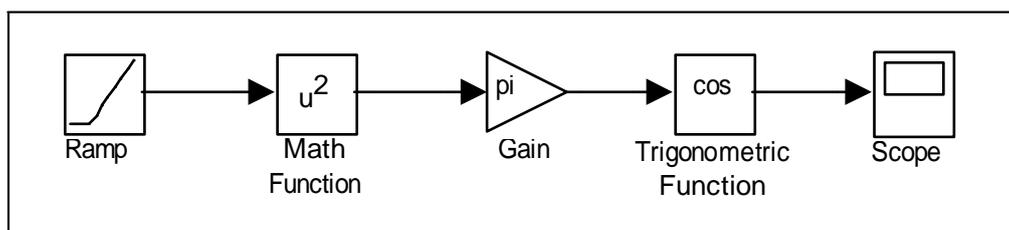


Рис. 4.3 Фрагмент модели построения сигнала $x(t) = \cos \pi t^2$

Однако, для того чтобы итоговая модель не была слишком громоздкой, отдельные этапы построения модели будем оформлять в виде подсистем, из которых и будет окончательно состоять модель.

Построенный фрагмент модели оформим как подсистему, для этого выделим все блоки, генерирующие сигнал $x(t) = \cos \pi t^2$, кроме блока Scope, и выполним команду Create Subsystem из меню Edit окна модели. Выделенный фрагмент будет помещен в подсистему, а вместо него в основном окне модели появится блок Subsystem.

Двойной щелчок по блоку Subsystem откроет дополнительное окно подсистемы, в котором отобразится спрятанный фрагмент, при этом выход подсистемы будет снабжен соответствующим выходным портом out1, который можно переименовать в $x(t)$.

Результат преобразований показан на рис. 4.4.

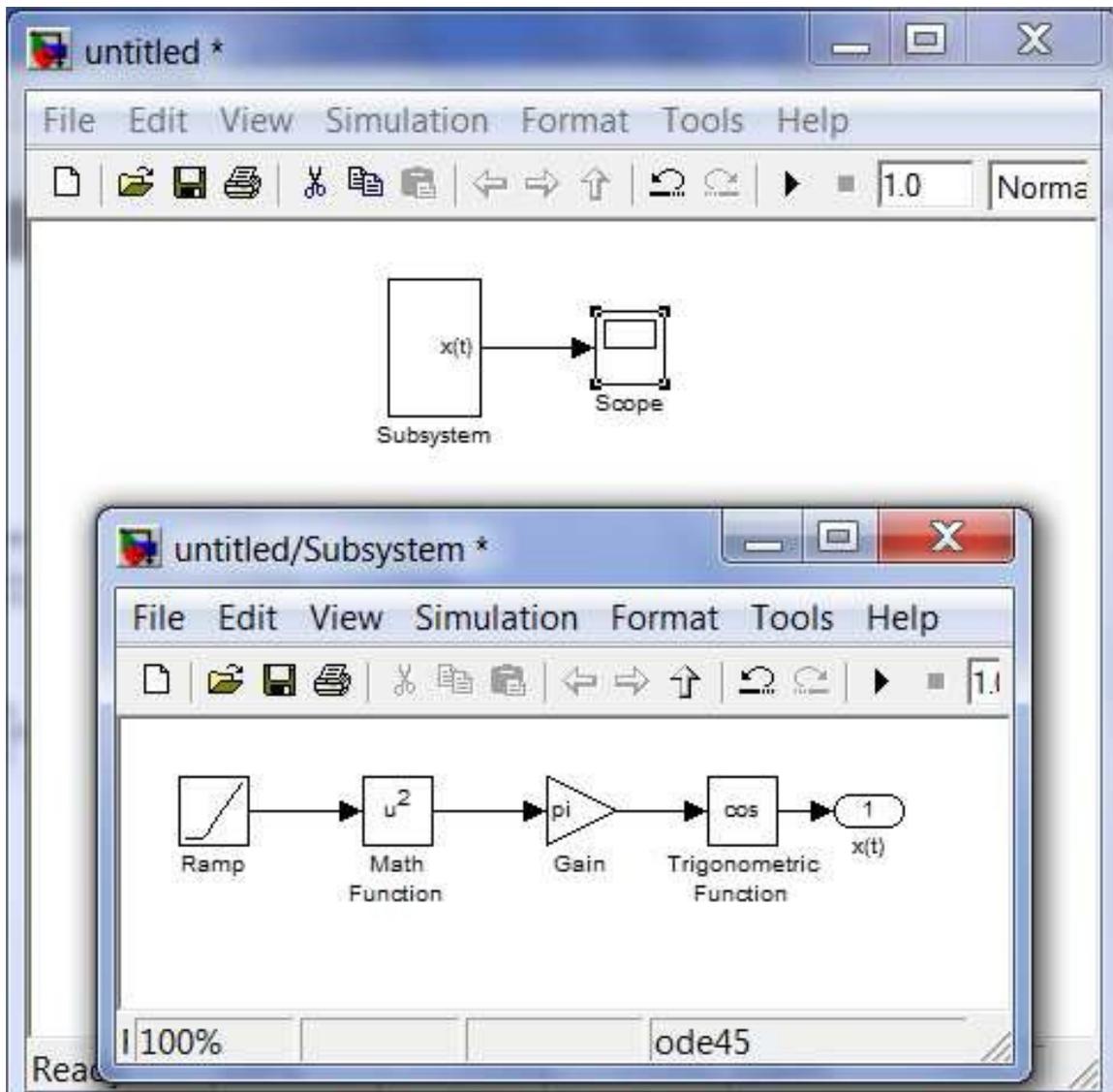


Рис. 4.4 Результат преобразования фрагмента системы в подсистему

Второй этап. Аналогичным образом поступим при построении системы базисных степенных функций $S(t) = \{s_k(t) = t^k, k = \overline{0,4}\}$. Сначала в основном окне построим структурную схему вычисления функций $\{1, t, t^2, t^3, t^4\}$ и преобразуем их в вектор при помощи блока Мух (рис. 4.5), затем выделим соответствующий фрагмент схемы и преобразуем его в подсистему описанным выше способом.

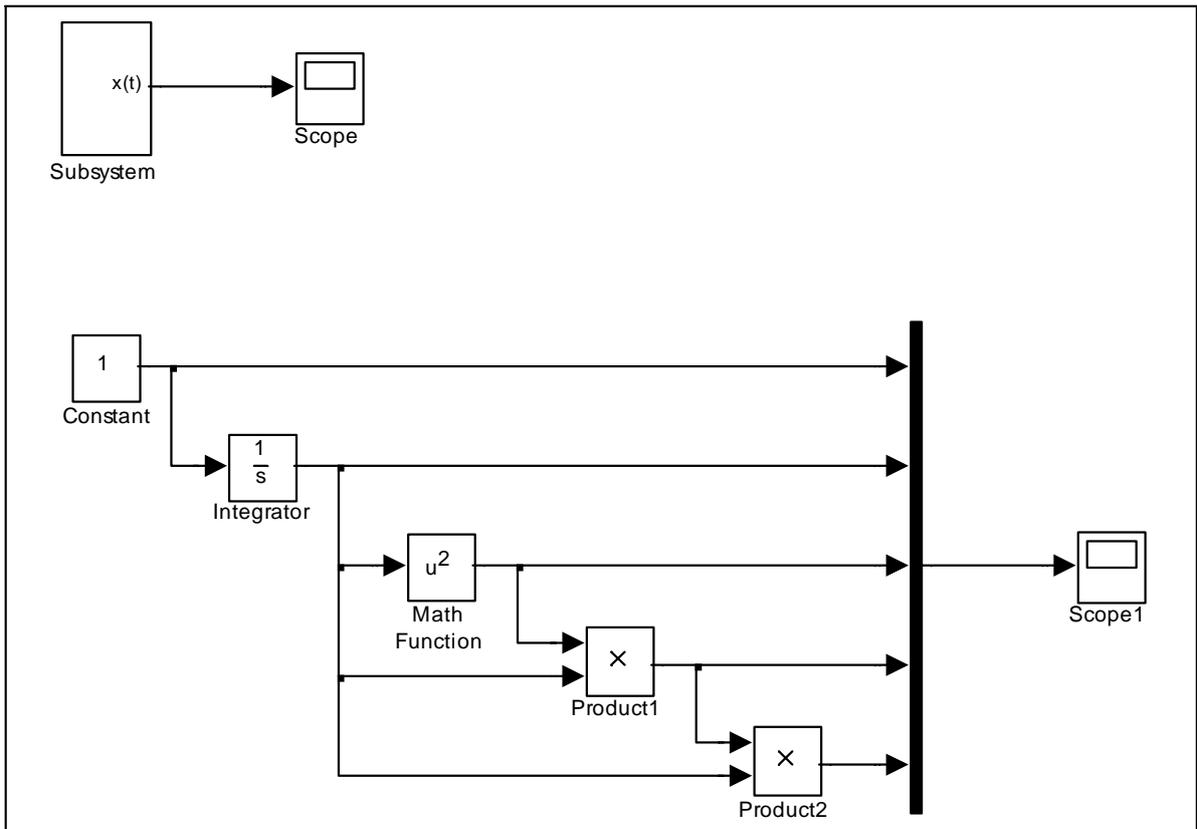


Рис. 4.5 Построение системы базисных функций

Модель в основном окне примет вид, показанный на рис. 4.6.

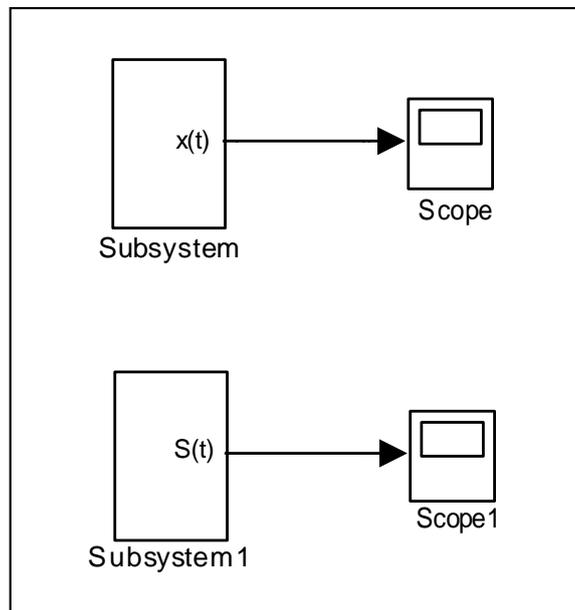


Рис. 4.6 Фрагмент модели, состоящий из подсистем

Третий этап. Определим вектор аппроксимирующего спектра $A = \{A_k, k = \overline{0, n}\}$ по формулам (4.3), (4.2). Вычисления коэффициентов можно сформировать при помощи схемы, показанной на рис. 4.7.

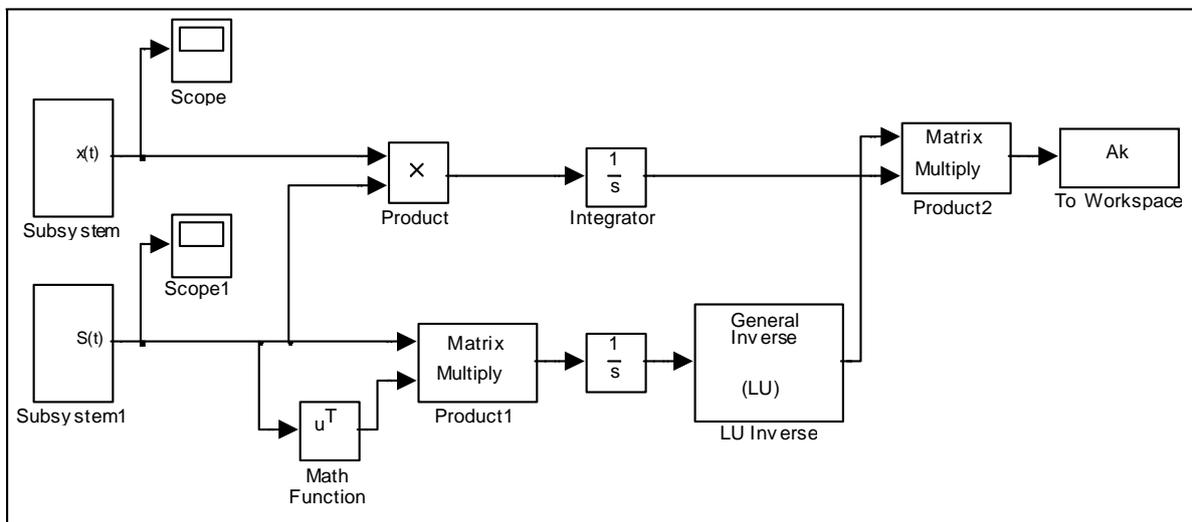


Рис. 4.7 Структурная схема определения коэффициентов аппроксимирующего многочлена

Здесь элементы q_i вектора Q получаются при умножении компонент вектора $S(t)$ на сигнал $x(t)$ с последующим интегрированием. Элементы w_{ij} матрицы W получаются при умножении вектора-строки $S^T(t)$ на вектор-столбец $S(t)$ с последующим интегрированием. Для получения обратной матрицы W^{-1} используется блок LU Inverse. Аппроксимирующий спектр $A_k, k = \overline{0, n}$ получается, в соответствии с формулой (4.2), умножением матриц W^{-1} и Q . Описанные вычисления преобразуем в подсистему, изображенную на рис. 4.8.

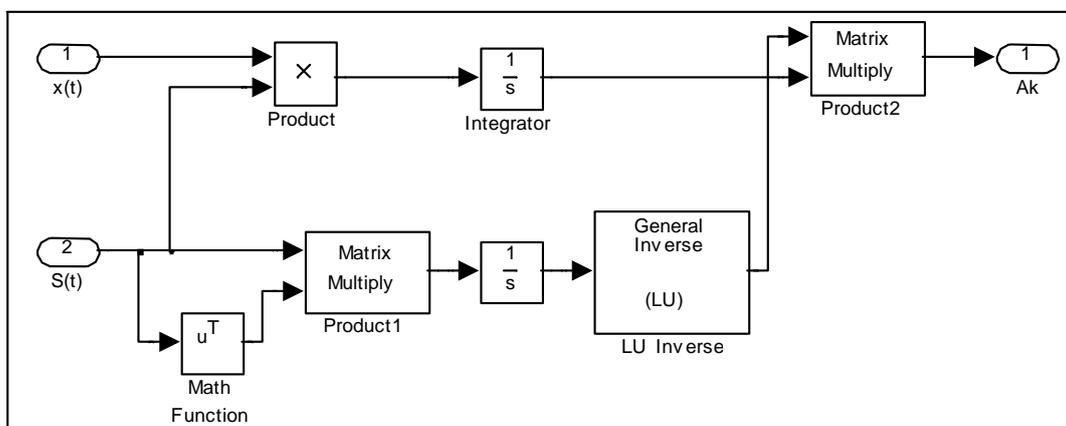


Рис. 4.8 Подсистема определения коэффициентов спектра

Модель в основном окне примет вид, показанный на рис. 4.9.

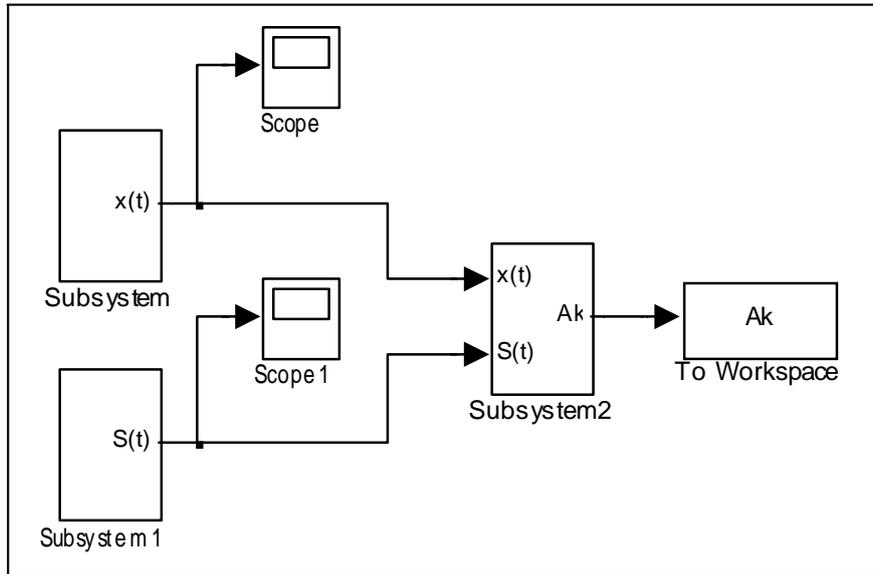


Рис. 4.9 Основная модель с тремя подсистемами

Осталось доработать блок To Workspace, используемый для записи результатов в рабочую область программы Matlab. На промежутке $[0,1]$ аппроксимирующий спектр A_k $k = 0,4$ генерируется в каждый момент времени $t \in [0,1]$ с некоторым варьируемым шагом. Для этого работой блока To Workspace необходимо управлять с помощью блока Step, генерирующего ступенчатый сигнал.

Преобразуем блок To Workspace в подсистему (рис. 4.10), в которую добавим блок Enable, при помощи которого можно управлять записью результатов в рабочую область. А самому блоку To Workspace зададим свойству Save format значение Structure With Time.

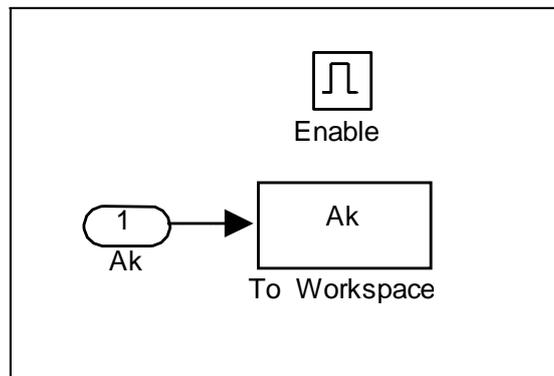


Рис. 4.10 Подсистема сохранения результатов расчета аппроксимирующего спектра в рабочую область

Окончательный вид основной модели с результатами моделирования на последнем шаге показан на рис. 4.11.

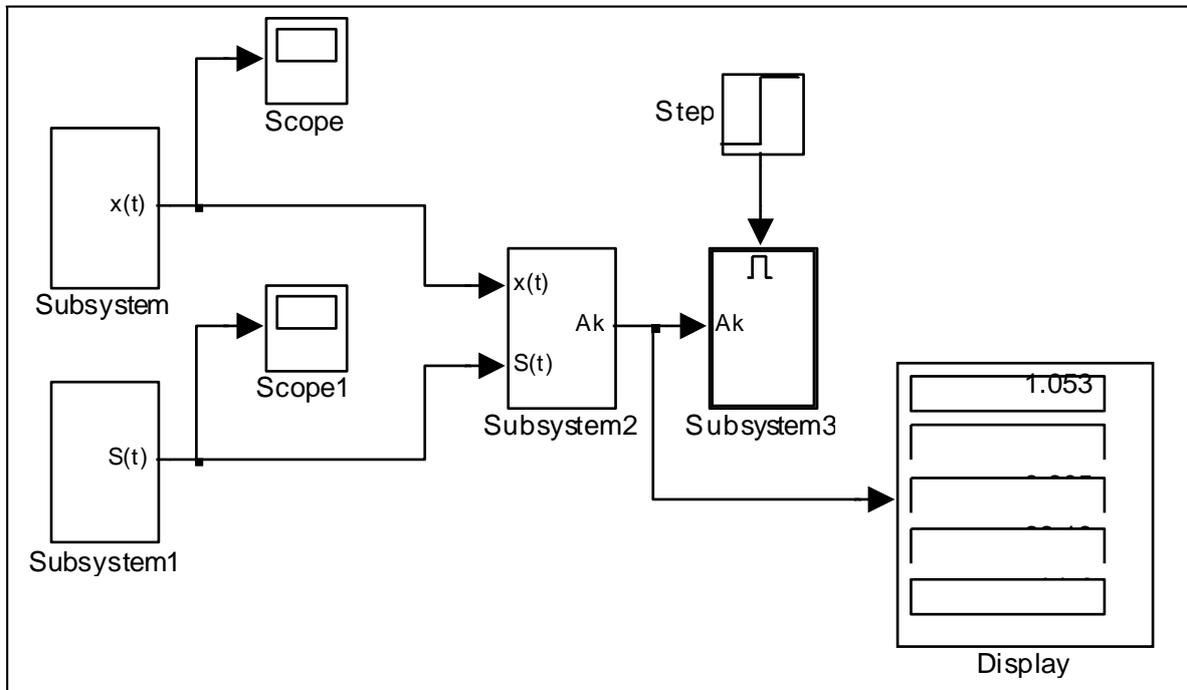


Рис. 4.11 Окончательный вид модели расчета коэффициентов по методу наименьших квадратов

Четвертый этап. После расчета коэффициентов аппроксимирующего спектра искомый аппроксимирующий сигнал восстанавливается по схеме, приведенной на рис. 4.12.

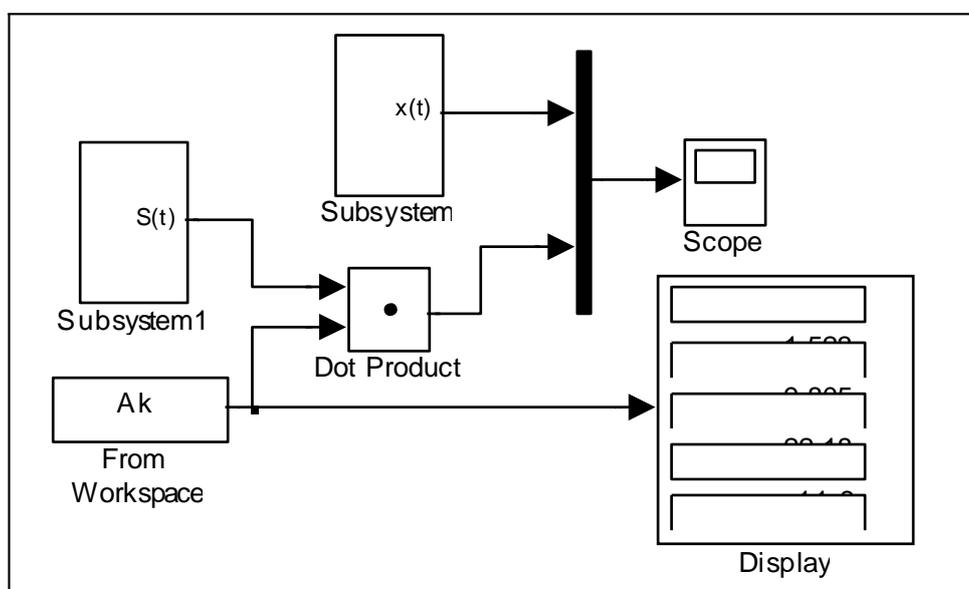


Рис. 4.12 Восстановление сигнала по найденному спектру

Результаты работы модели выводятся на экран виртуального осциллографа, при этом на рис. 4.13 слева показан исходный сигнал, а справа изображены исходный и наложившийся на него аппроксимирующий сигнал.

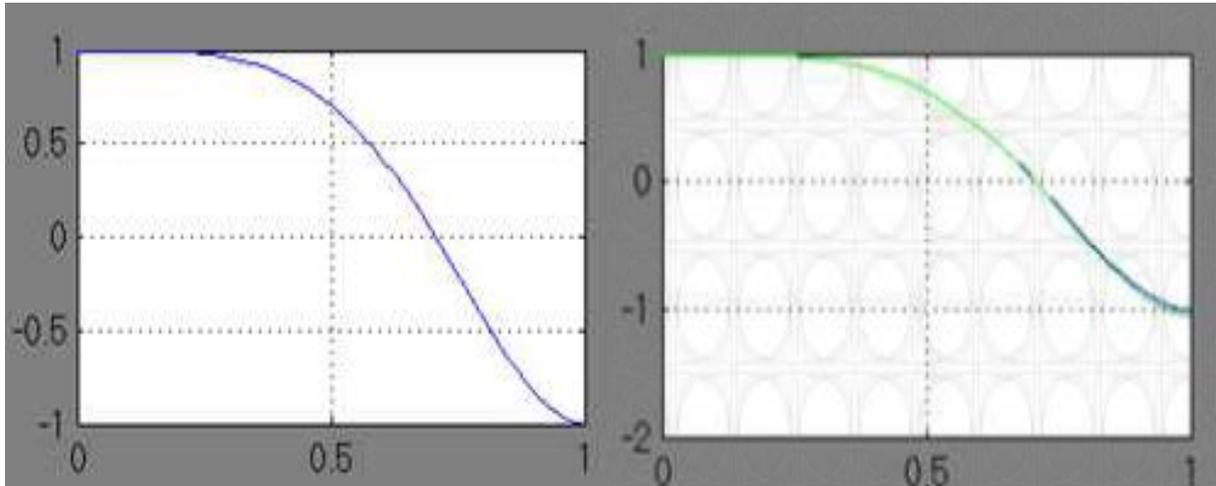


Рис. 4.13 Исходный сигнал $x(t) = \cos \pi t^2$ и его аппроксимация по методу наименьших квадратов

Варианты для выполнения Задания 4.1

Методом наименьших квадратов на интервале $[0,1]$ аппроксимировать сигнал из лабораторной работы №1 в соответствии со своим вариантом. В качестве базисных функций взять степенные функции вида $s(t) = \{s_k(t) = t^k, k = \overline{0,4}\}$, то есть систему $\{1, t, t^2, t^3, t^4\}$.

Задание 4.2

Модифицированным методом равных площадей аппроксимировать тот же сигнал $x(t) = \cos \pi t^2$ на интервале $[0,1]$. В качестве базисных функций взять те же степенные функции до 4-го порядка включительно.

Модель аппроксимации сигнала модифицированным методом равных площадей можно строить в соответствии с теми же этапами, что и модель в задании 4.1, а окончательную модель оформить как совокупность выполняемых подсистем.

Для упрощения построений можно воспользоваться результатами выполнения задания 4.1 и лишь преобразовать необходимые подсистемы.

Эти преобразования связаны с тем, что в соответствии с методом равных площадей интервал изменения аргумента разбивается, например, на пять подинтервалов $[0; 0,2]$, $[0,2; 0,4]$, $[0,4; 0,6]$, $[0,6; 0,8]$, $[0,8; 1,0]$ и на каждом из них строится система коэффициентов аппроксимирующего спектра по формулам (4.4), (4.2).

Первый этап. Подсистема построения сигнала $x(t) = \cos \pi t^2$ на интервале $[0, 1]$ может быть взята из задания 4.1 (рис. 4.4).

Второй этап. Подсистема построения базисных функций $\{1, t, t^2, t^3, t^4\}$ соответствует заданию 4.1 (рис. 4.14).

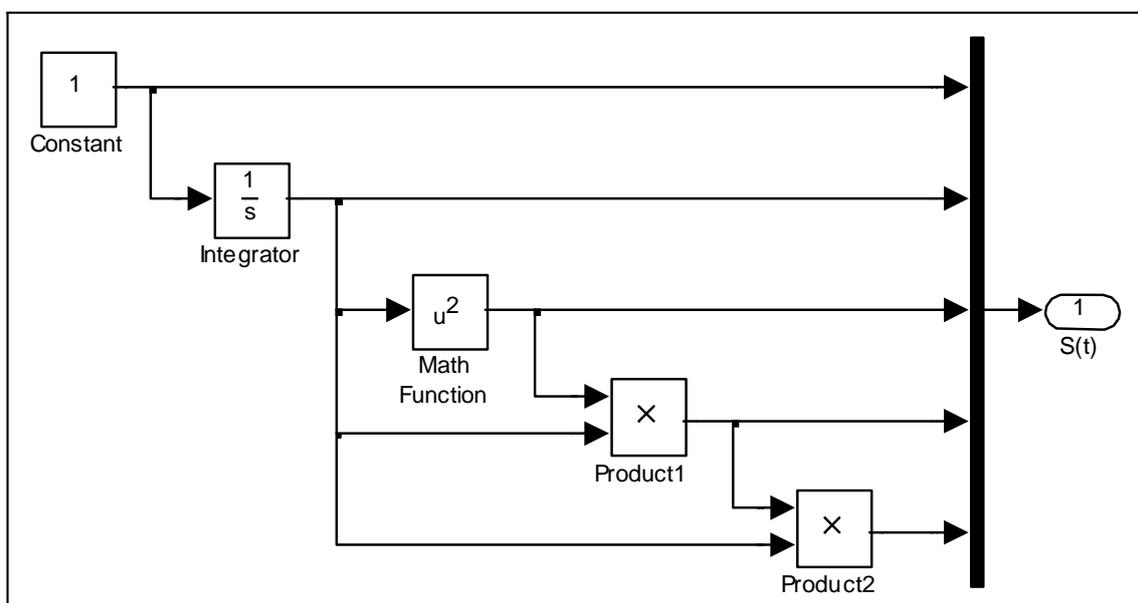


Рис. 4.14 Подсистема построения базисных функций

Третий этап. А вот подсистема построения вектора аппроксимирующего спектра Subsystem2 будет подвергнута значительным изменениям.

Исходя из формул (4.4), можно заметить, что как при расчете вектора Q , так и при расчете векторов-столбцов матрицы W общим является интегрирование функции на одних и тех же интервалах. Таким образом, для получения вектора Q необходимо проинтегрировать на заданной системе подинтервалов сигнал $x(t)$, а для получения столбцов матрицы W необходимо проинтегрировать 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. 4.15.

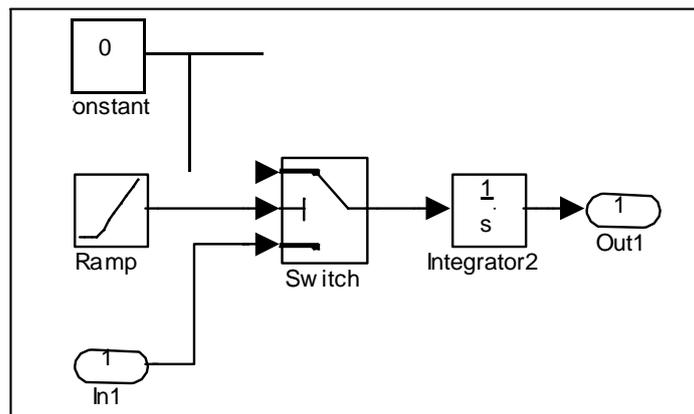


Рис. 4.15 Блок интегрирования функции на системе подинтервалов

На вход этого блока поступает сигнал – интегрируемая функция. Два других сигнала – нулевой и генератор линейно возрастающей функции времени t являются вспомогательными. Пока время 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 больше не влияет на значение интеграла.

Так как MatLab ориентирован в основном 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. 4.16.

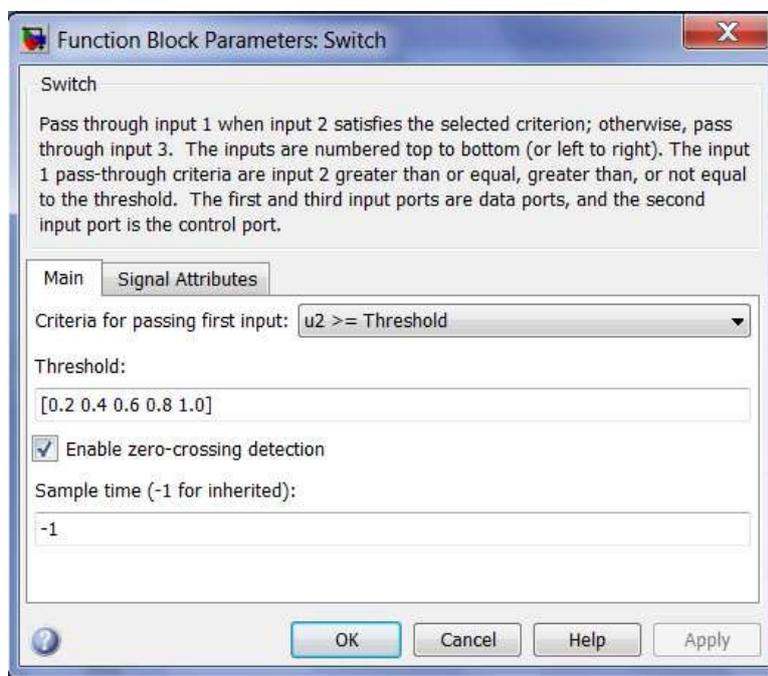


Рис. 4.16 Окно настройки параметров блока Switch

Теперь рассмотрим структурную схему блока Subsystem2, представленную на рис. 4.17.

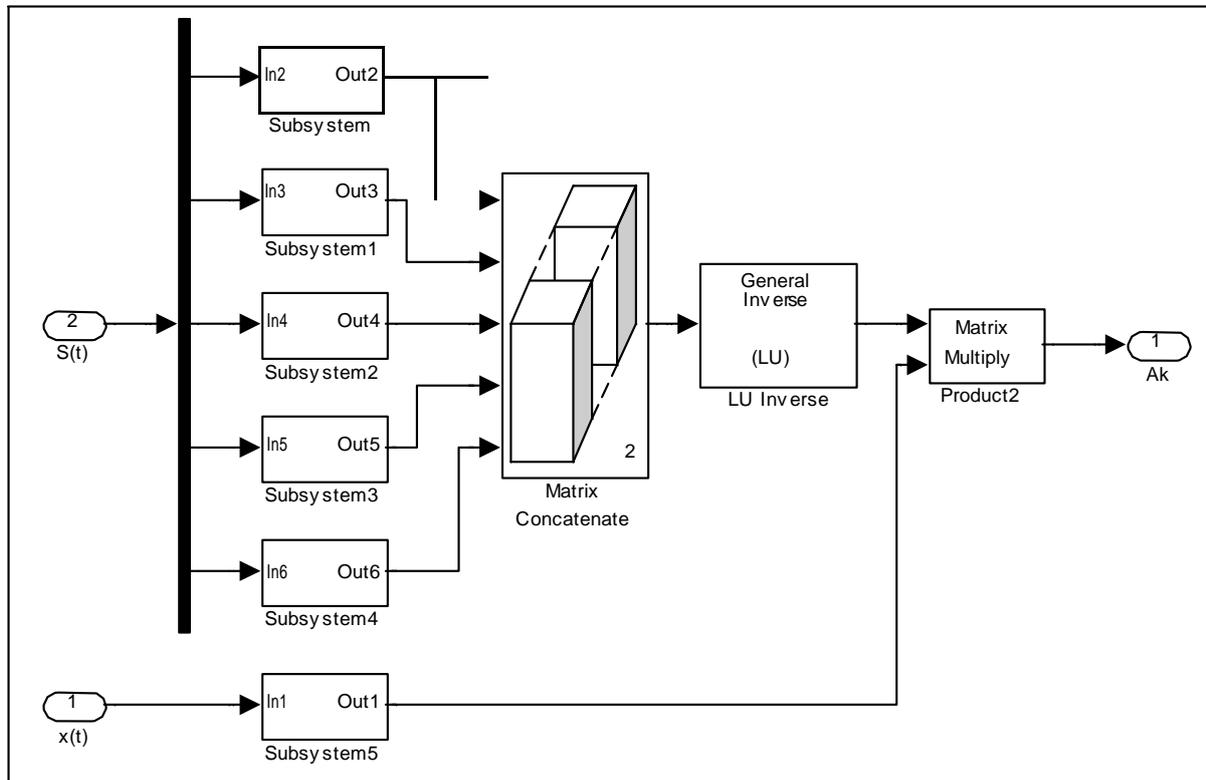


Рис. 4.17 Подсистема построения аппроксимирующего спектра

На вход блока подается сигнал $x(t)$ и система базисных функций $S(t)$. Подсистема Subsystem5 интегрирует сигнал и подает на выход вектор Q , а подсистемы Subsystem – Subsystem4 интегрируют векторы-столбцы матрицы W .

Блок Matrix Concatenate объединяет столбцы в матрицу. Далее размещены блоки расчета обратной матрицы и умножения матриц. На выходе получаем сигнал аппроксимирующего спектра, который появляется на экране дисплея модели верхнего уровня (рис. 4.18).

Четвертый этап. Рассмотрим задачу построения 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.

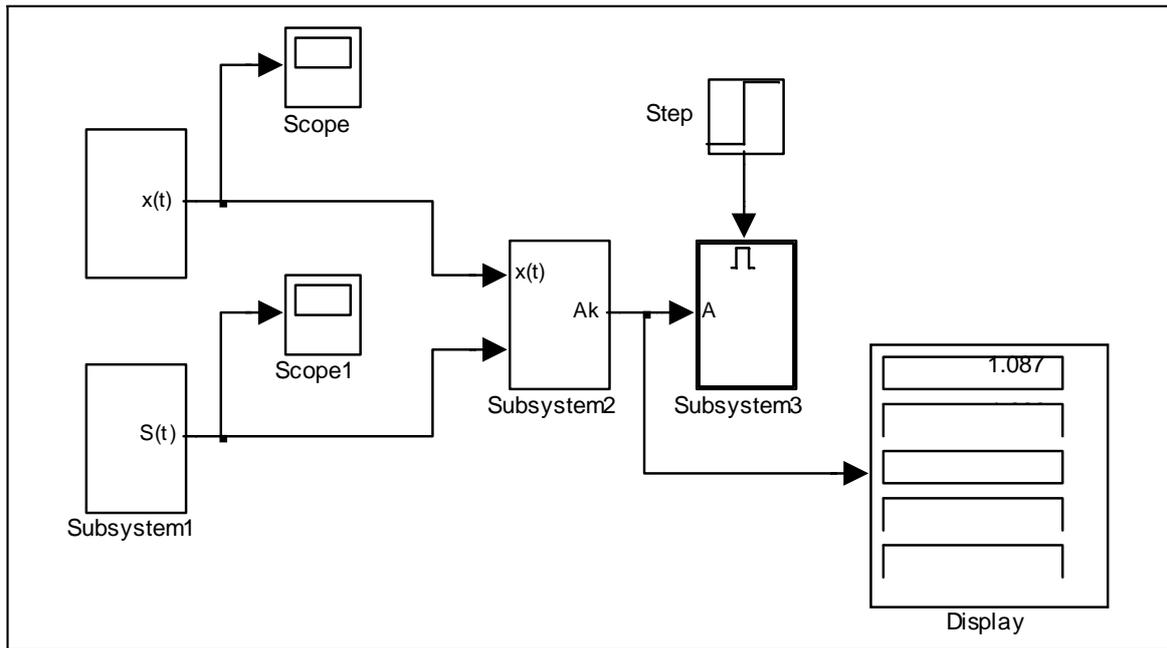


Рис. 4.18 Модель построения аппроксимирующего спектра с результатами

Структурная схема получения сигнала, как и в задании 4.1, будет иметь вид, показанный на рис. 4.19. 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 первоначальный сигнал. Графики сигнала и его аппроксимации, представленные на рис. 4.20, практически совпадают.

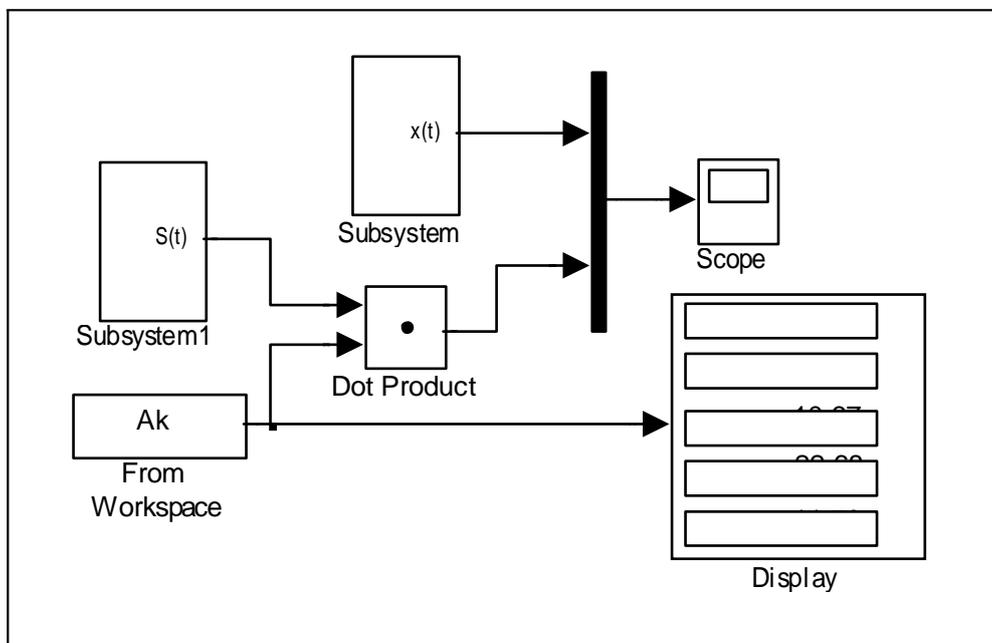


Рис. 4.19 Структурная схема получения аппроксимирующего сигнала

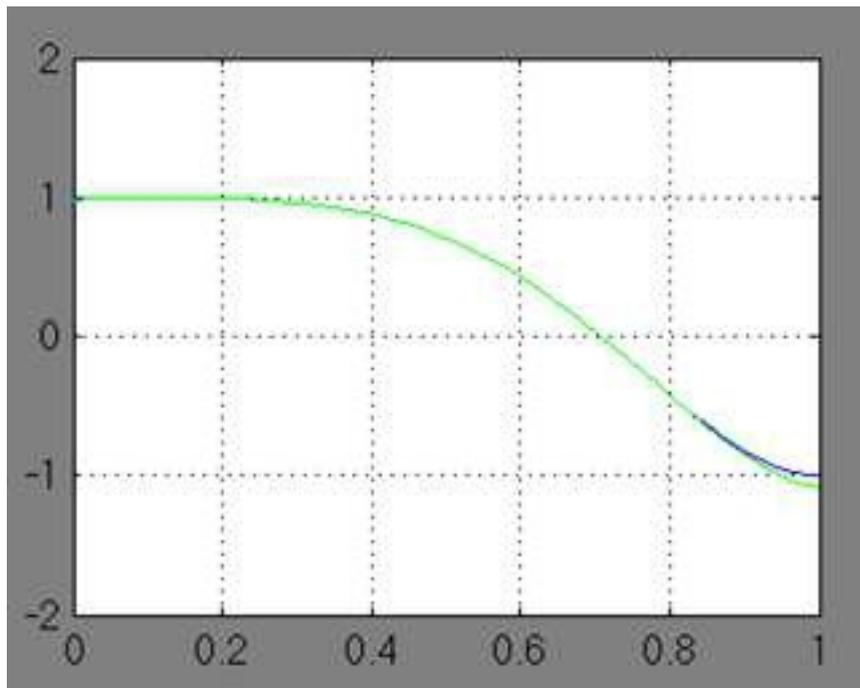


Рис. 4.20 График заданного сигнала и его аппроксимация

Рассмотренные модели применяются для решения большого количества задач, связанных с цифровой обработкой сигналов, в том числе с обработкой звуковых сигналов и изображений.

Варианты для выполнения Задания 4.2

Модифицированным методом равных площадей на интервале $[0, 1]$ аппроксимировать сигнал из лабораторной работы №1 в соответствии со своим вариантом. В качестве базисных функций взять степенные функции вида $S(t) = \{s_k(t) = t^k, k = \overline{0, 4}\}$, то есть систему $\{1, t, t^2, t^3, t^4\}$. Систему подинтервалов оставить как в примере.

Задание 4.3

Методом аппроксимации блочно-импульсными функциями нулевого порядка и локально-импульсными функциями первого порядка выполнить аппроксимацию сигнала $x(t) = \cos 2\pi t^2$ на интервале $[0, 1]$, $m = 10$.

Сгенерируем блочно-импульсные базисные функции нулевого порядка в соответствии с формулами (4.5), (4.6). Отрезок $[0, 1]$ разобьем на $m = 10$ частей длиной $h = 0.1$ и для каждого $i = \overline{1, m}$ вычислим

$v_i(t) = \sigma(t - (i-1)h) - \sigma(t - ih)$. Для этого оформим функцию

единичного

$$\sigma(t - (i-1)h) \begin{cases} 0, & \text{если } t < (i-1)h, \\ 1, & \text{если } t \geq (i-1)h. \end{cases} \quad \text{и} \quad \sigma(t - ih) \begin{cases} 0, & \text{если } t < ih, \\ 1, & \text{если } t \geq ih. \end{cases}$$

Для проверки условия на каждом частичном промежутке оформим блок Constant, содержащий значения $(i-1)h$ при $i = \overline{1, m}$, то есть постоянную $C = [0.0 \ 0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.6 \ 0.7 \ 0.8 \ 0.9]$ зададим в виде вектора с соответствующими компонентами или, что то же самое $C = [0.0:0.1:0.9]$ (рис. 4.21), а блок Constant1, содержащий значения ih при $i = \overline{1, m}$, зададим в виде диапазона значений $C = [0.1:0.1:1.0]$.

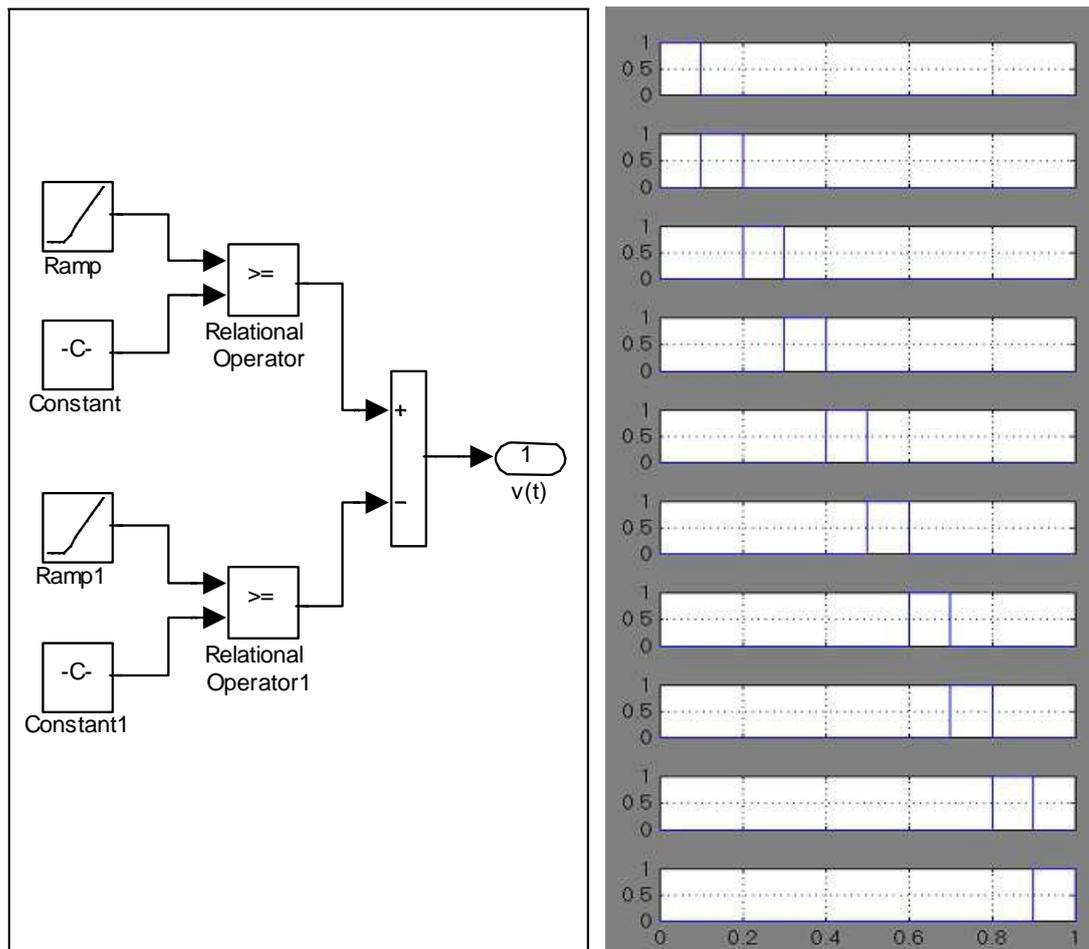


Рис. 4.21 Структурная схема, генерирующая блочно-импульсные функции нулевого порядка и графики блочно-импульсных функций

Рядом со схемой на рис. 4.21 показан вид блочно-импульсных функций нулевого порядка, образующих базис, при этом каждая из функций имеет столбик единичной высоты на соответствующем частичном промежутке.

В соответствии с формулами (4.8) построим систему локально-импульсных базисных функций первого порядка. При этом в блоке Constant1 (рис. 4.22) задаются значения $i = \overline{1, m}$ в виде вектора $C = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10]$ или, что то же самое $C = [1 : 10]$.

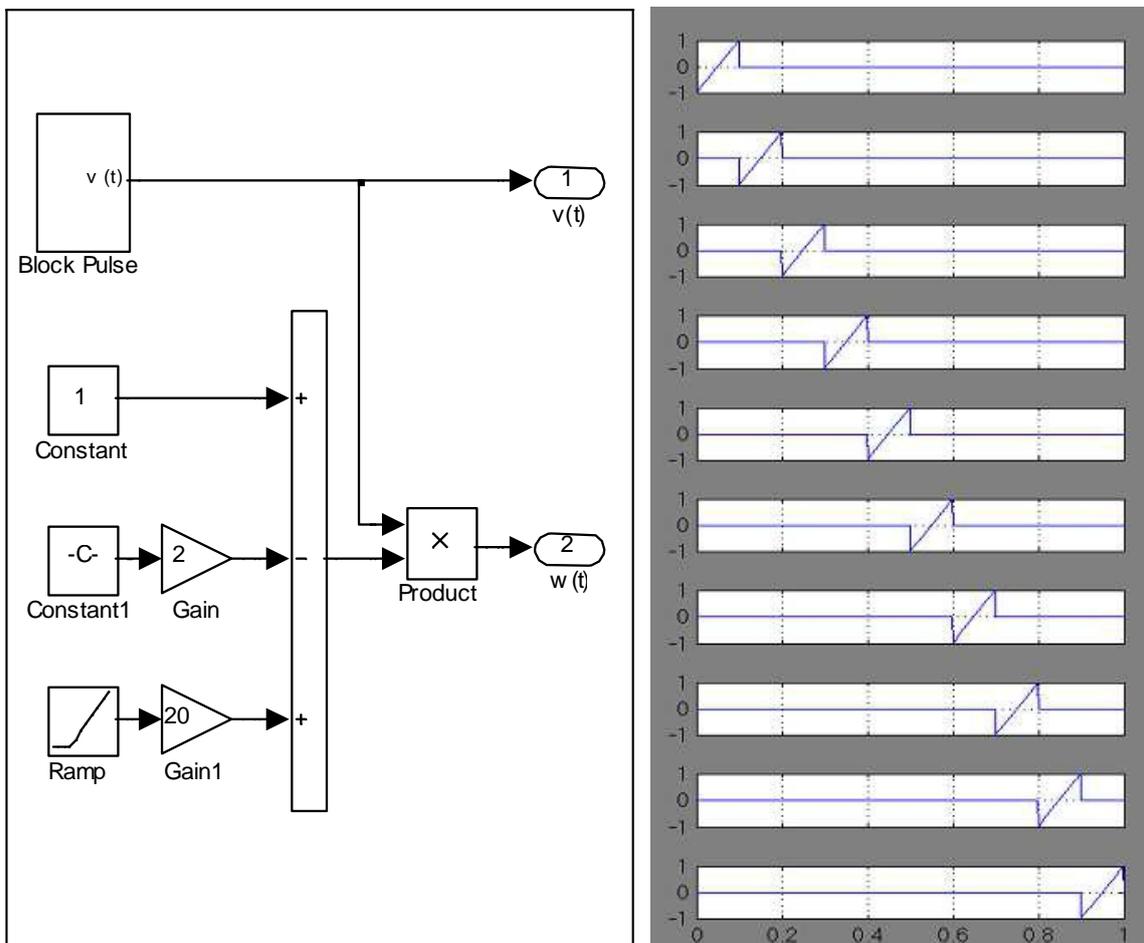


Рис. 4.22 Структурная схема, генерирующая локально-импульсные функции первого порядка и графики кусочно-линейных функций

Структурная схема анализа сигнала рассматриваемым методом представлена на рис. 4.23. Назначение блоков следующее:

- BF генерирует систему базисных функций;
- $X(t)$ генерирует анализируемый сигнал;
- X_0, X_1 рассчитывают элементы спектра по формулам метода наименьших квадратов;
- $Step$ определяет момент записи спектра в рабочую область.

Полученный спектр, кроме записи в рабочую область, можно выдать на дисплей.

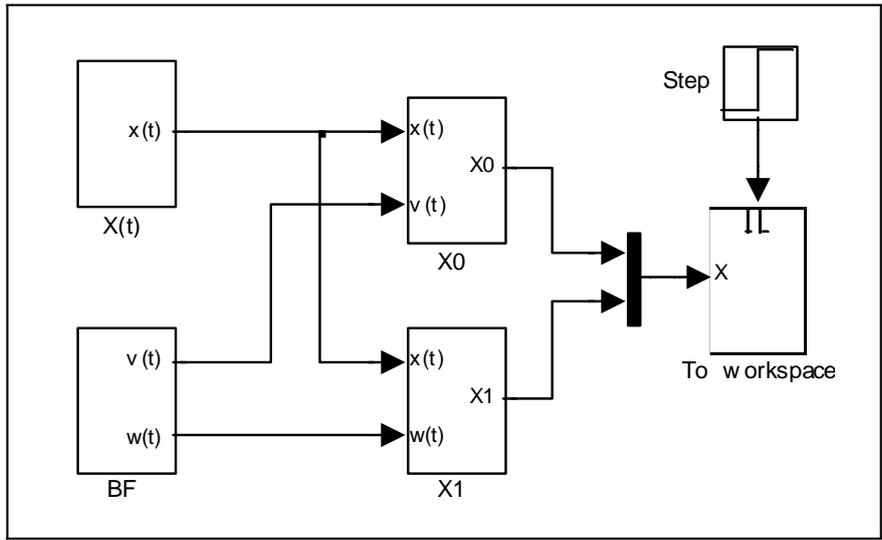


Рис. 4.23 Структурная схема анализа сигнала блочно-импульсным методом

Структурная схема, представленная на рис. 4.24, восстанавливает сигнал. На вход схемы подаются базисные функции и спектр анализируемого сигнала, на выход – аппроксимация сигнала по соответствующему методу и исходный сигнал.

На виртуальный осциллограф подаются исходный и восстановленный сигналы. Они представлены на рис. 4.25 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.

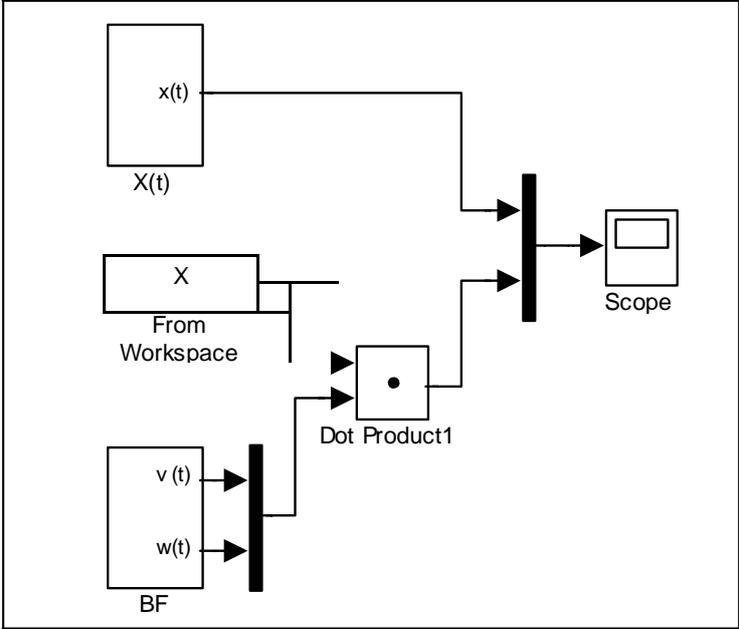


Рис. 4.24 Структурная схема синтеза сигнала

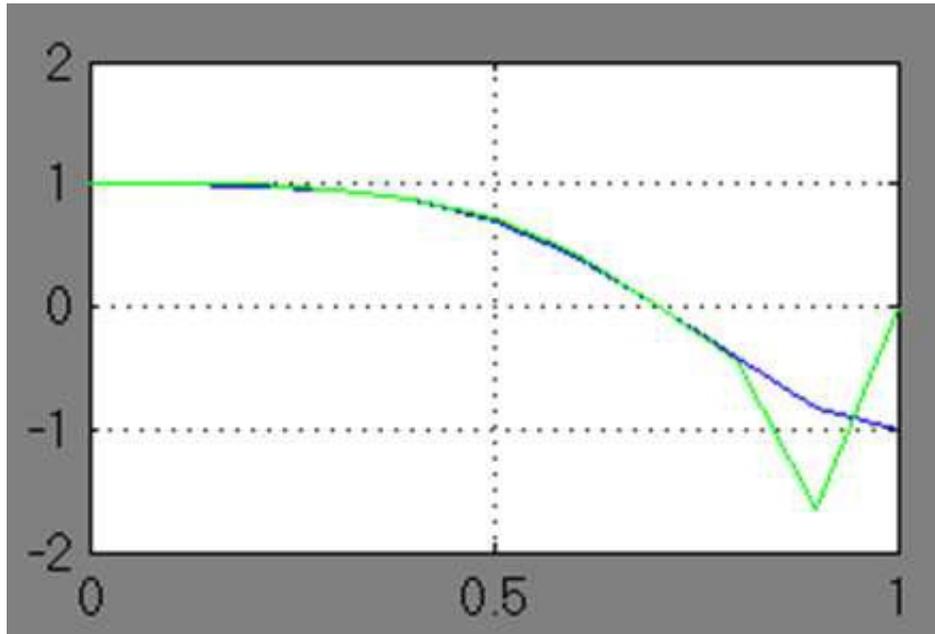


Рис. 4.25 Результаты аппроксимации сигнала

Варианты для выполнения Задания 4.3

Методом аппроксимации блочно-импульсными функциями нулевого и первого порядков выполнить аппроксимацию сигнала на интервале $[0, 1]$, $m = 10$ из лабораторной работы №1 в соответствие со своим вариантом.

Лабораторная работа №5 Вероятностное моделирование

ЦЕЛЬ РАБОТЫ

Изучить методы вероятностного моделирования с целью программной реализации алгоритмов формирования случайных величин с заданным законом распределения. Моделирование осуществляется в программных средах MATLAB:

- Построить модель простого случайного события, с заданной вероятностью.
- Построить модель простого случайного события, с заданной вероятностью сформировать с помощью генератора случайных чисел (ГСЧ) случайное событие на заданном интервале
- Построить модель с использованием генератора случайного числа, содержащую случайную величину, распределенную по нормальному закону распределения.

ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Перед выполнением лабораторной работы необходимо ознакомиться с темами:

- Вероятностное моделирование [1, п.2, 5, 7].
- Метод статических испытаний [1, п.2, 5, 7].
- Генераторы случайных чисел [1, п.2, 5, 7].
- Моделирование случайных событий [1, п.2, 5, 7].
- Моделирование случайных величин [1, п.2, 5, 7].

Ответить на вопросы:

1. Способы генерации случайных чисел.
2. Моделирование случайных событий.
3. Моделирование простого события.
4. Моделирование полной группы несовместных событий.
5. Моделирование случайных величин.
6. Моделирование непрерывных случайных величин.
7. Каково назначение и описание параметров блоков, используемых при выполнении лабораторной работы?

Моделирование случайных чисел

Для генерации случайных чисел можно воспользоваться следующими блоками: *Random Number* и *Uniform Random Number*.

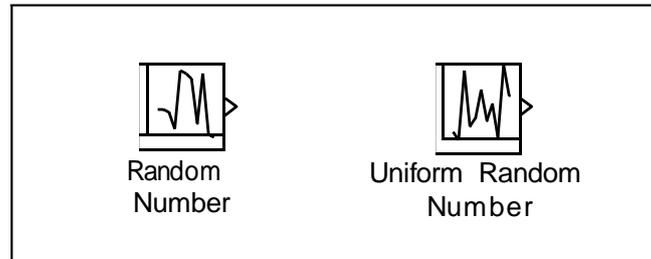


Рис. 5.1 Блоки, генерирующие случайные числа

Блок *Random Number* обеспечивает формирование дискретного сигнала, амплитуда которого является случайной величиной, распределенной по нормальному закону с заданными параметрами.

Блок *Uniform Random Number* обеспечивает формирование дискретного сигнала, амплитуда которого является случайной величиной, равномерно распределенной в заданном интервале.

Особенностью указанных блоков является то, что они в каждом сеансе моделирования генерирует одну и ту же последовательность случайных чисел. Для изменения генерируемой последовательности необходимо вручную изменить значение параметра *Initial seed*. При проведении большого числа повторных экспериментов с целью накопления статистических данных это не очень удобно. Поэтому для моделирования случайных событий можно воспользоваться генераторами случайных чисел, входящими в состав компоненты MATLAB.

Генератор непрерывной случайной величины, равномерно распределенной в заданном интервале, называется *unifrnd*. Вызов данной функции имеет вид:

$$\text{unifrnd}(A, B, M, N),$$

A, B – границы диапазона распределения;

M и N задают размер генерируемой матрицы случайных чисел.

При моделировании случайной величины эта функция может быть указана в качестве параметра настройки следующих блоков: MATLAB Fcn (раздел Nonlinear), Fcn (раздел Nonlinear), Constant (раздел Sources).

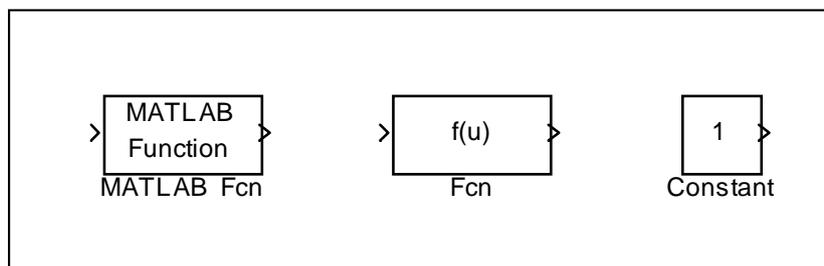


Рис. 5.2 Блоки, используемые для вызова функций, генерирующих случайные числа

Блок MATLAB Fcn - формирует функцию, заданную пользователем на языке MatLab

Блок Discrete-Time Integrator – дискретный сумматор времени. Выполняет суммирование интервалов времени между поступлениями входного и выходного сигнала.

Блок Relay – работает по аналогии с обычным реле, если входной сигнал превышает некоторое пороговое значение, то на выходе блока формируется «разрешающий» сигнал.

Блок Hit Crossing – позволяет идентифицировать момент времени, когда входной сигнал «пересекает» некоторое значение: при появлении такой ситуации на входе блока формируется единичный сигнал.

Моделирование непрерывных случайных величин

Использование случайных величин является наиболее универсальным и наиболее распространенным способом учета в модели случайных факторов, присущих реальным событиям и процессам. Примерами случайных величин могут служить: интервал времени до появления очередного клиента, длительность проведения технического обслуживания автомобиля, объем данных, считываемых с оперативной памяти ЭВМ.

Если известен закон распределения случайной величины, то мы можем ее представить в имитационной модели. При создании модели воспользуемся генераторами случайных чисел, входящими в состав компоненты MATLAB.

Генераторы случайных чисел можно найти используя справочную систему MATLAB:

- Открыть раздел *Statistics Toolbox\Probability Distributions\Random Number Generation*.
- В списке *Random Number Generation* выбрать функцию генератор, соответствующему требуемому закону распределения.

- Двойным щелчком ЛКМ на выбранной строке открыть страницу справочника, содержащую описание данного генератора; при этом в верхнем левом поле окна будет выведено название генератора; выделите его с помощью мыши и скопируйте в буфер обмена.
- В блок-диаграмме выбрать блок, в котором будет использоваться генератор, и открыть окно его настроек.
- Вставить из буфера обмена название генератора.

Имя требуемого генератора можно вводить вручную, в этом случае существует возможность ошибиться хотя бы в одном символе, что приведет к ошибке модели к необходимости ее коррекции и перезапуске модели.

ПРАКТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Задание 5.1

Построить модель простого случайного события, вероятность появления которого равна 0,3.

Пример 5.1. Случайное событие считается наступившим, если сформированное датчиком случайное число попало в диапазон $[0, p]$, где p – вероятность появления данного события.

Следовательно, модель должна содержать компоненты, обеспечивающие выполнение трех действий: генерацию случайного числа, сравнение его с p , идентификацию момента возникновения события.

Указанные действия можно реализовать, используя совместно блоки Uniform Random Number (раздел Sources) и Relational Operator (раздел Nonlinear), как показано на рис. 5.3.

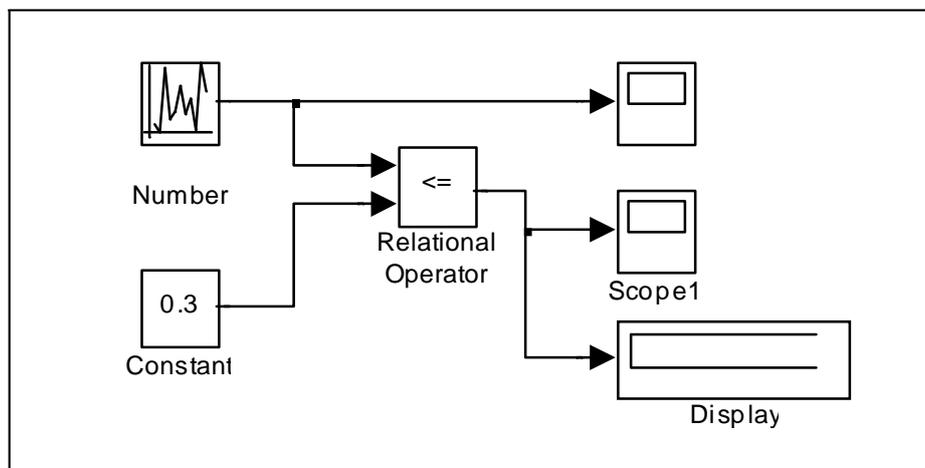


Рис. 5.3 Моделирование случайного события с заданной вероятностью наступления

Чтобы первый из блоков генерировал случайные числа, равномерно распределенные на интервале $[0;1]$, необходимо произвести соответствующую установку его параметров:

Minimum (нижняя граница диапазона): 0;

Maximum (верхняя граница): 1;

Sample time – шаг по времени, с которым генерируются случайные числа, позволяет определить количество случайных чисел, которые будут сформированы блоком в течение интервала моделирования: при Sample time, равном 0, новое случайное число генерируется через каждые 0.02 единицы модельного времени; при Sample time, равном интервалу моделирования, генерируется только одно случайное число.

Блок Scope обеспечивает графическое представление генерируемой последовательности случайных чисел. Блок Scope1 отображает значение 1 на некотором промежутке времени в том случае, когда событие произошло, и значение 0, когда событие не произошло.

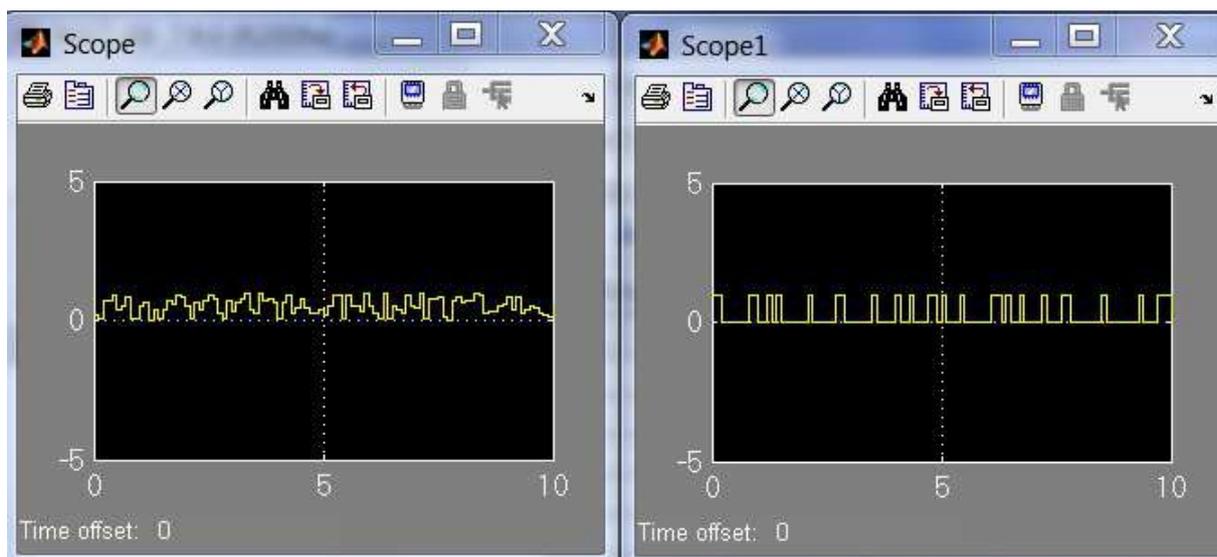


Рис. 5.4 Результат моделирования случайного события с заданной вероятностью наступления

Задание 5.2

Построить модель простого случайного события, вероятность появления которого равна 0,3, сформировать с помощью генератора случайных чисел (ГСЧ) случайное событие на интервале (0,1).

Пример 5.2. На рис. 5.5 приведен пример, обеспечивающей моделирование случайного события с помощью блока *Fcn*, в настройках которого следует указать обращение к генератору *unifrnd*.

В каждом сеансе моделирования генератор формирует новое случайное число. В этом можно убедиться с помощью дополнительного блока *Display1*, включенного в блок-диаграмму.

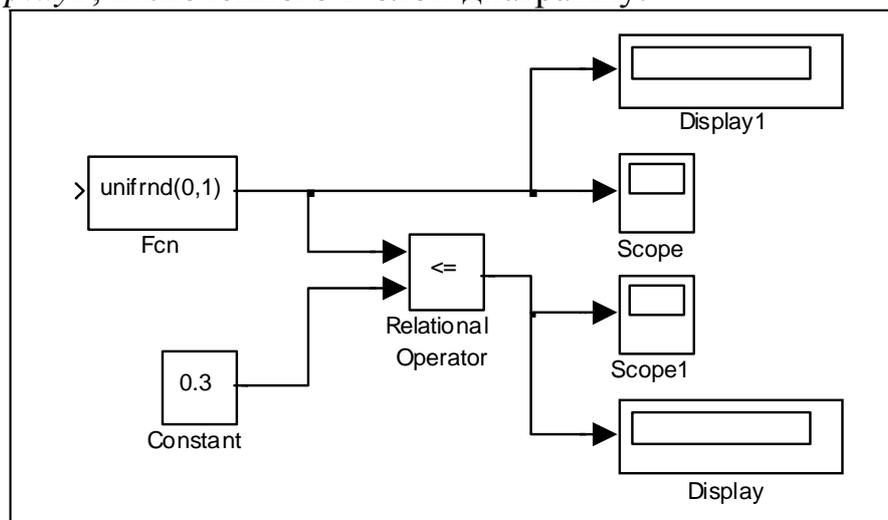


Рис. 5.5 Моделирование случайного события, основанное на неповторяющейся последовательности случайных чисел

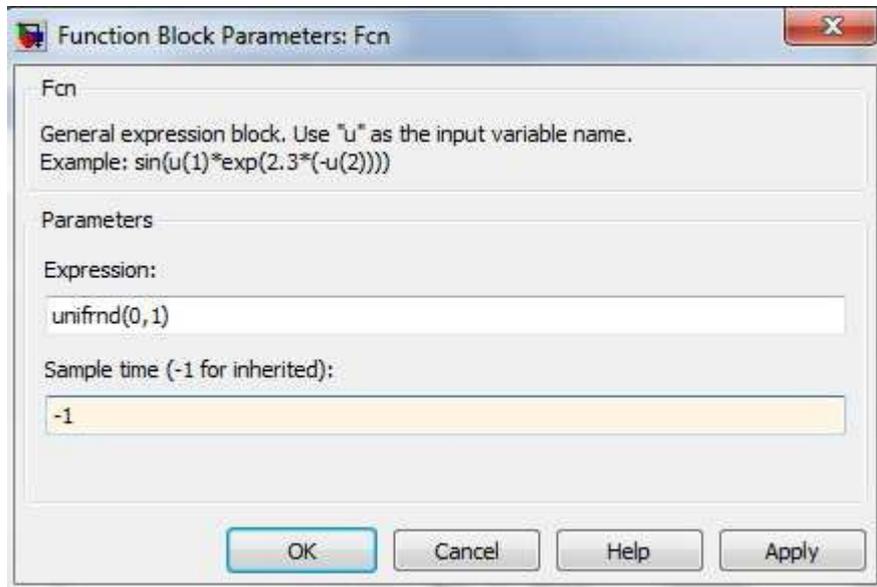


Рис. 5.6 Окно настройки блока Fcn

Пример 5.3. Появление случайного события можно интерпретировать еще одним способом: событие произошло, если некоторое связанное с ним число (признак) попало в заданный интервал или пересекло определенную границу.

Первый вариант может быть реализован с помощью блока *Relay*, второй – с помощью блока *Hit crossing* (оба блока из раздела *Nonlinear*).

На рис. 5.7 приведена блок-диаграмма, которая позволяет подсчитать число попаданий некоторого признака в заданный интервал. Работа блок-диаграммы основана на использовании блока *Relay*.

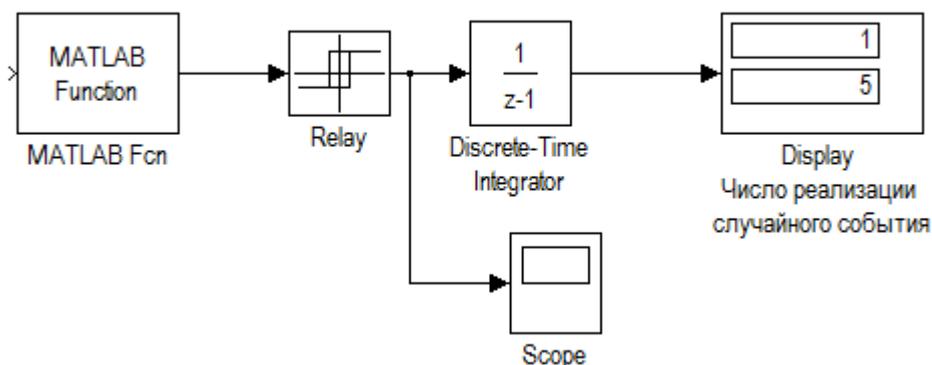


Рис. 5.7 Моделирования случайных событий с помощью блока Relay

Остальные блоки выполняют следующие функции:

MATLAB Fcn — генерирует последовательность СЧ, равномерно распределенных на интервале [0; 7];

Discrete-Time Integrator — подсчитывает число попаданий СЧ в диапазон [0; 5];

Display — выводит результат подсчета;

Scope — позволяет наблюдать за работой блока Relay.

Для блока Relay установлены следующие значения параметров:

Switch on point: 5 – точка включения (задает пороговое значение, при превышении которого происходит включение реле)

Switch off point: 1 – точка выключения (задает уровень сигнала при котором реле выключается)

Output when on: 0 – выход при включенном состоянии (определяет значение амплитуды разрешающего сигнала)

Output when off: 1 – выход при выключенном состоянии (уровень сигнала на реле, когда он находится в состоянии выключено).

Если исследователя интересует возможность хотя бы одной реализации случайного события на интервале моделирования, то удобнее воспользоваться блоком Hit Crossing.

На рис. 5.8 показан вариант блок-диаграммы, которая обеспечивает регистрацию момента превышения случайным числом заданного значения.

При этом параметры блока Hit crossing имеют значения:

Hit crossing offset: 4; (определяет значение, «пересечение» которого необходимо идентифицировать)

Hit crossing detection: rising - возрастание. (позволяет указать направление, которое интересует разработчиков модели)

При одновременном выполнении условий, задаваемых параметрами Hit crossing offset, Hit crossing detection, на выходе блока формируется единичный импульс. Его длительность определяется значениями параметрами Discrete-Time Integrator.

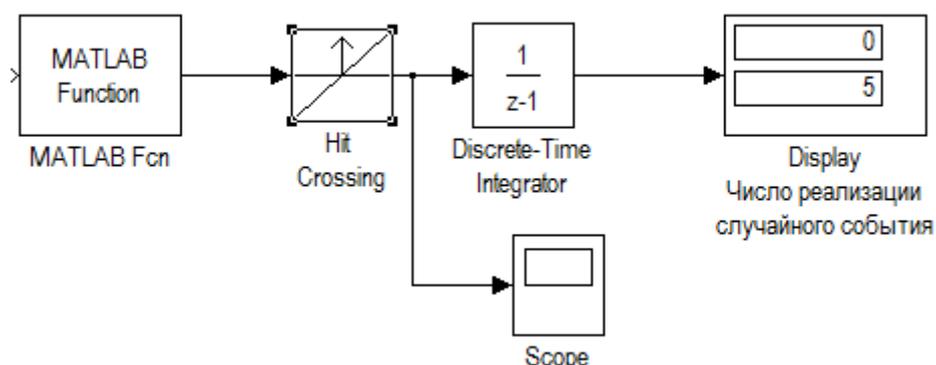


Рис. 5.8 Моделирования случайных событий с помощью блока Hit crossing

Остальные блоки используются так же, как в предыдущей блок-диаграмме.

Задание 5.4

Построить модель с использованием генератора случайного числа, содержащую случайную величину, распределенную по нормальному закону распределения.

Пример 5.4. Пусть имеется вычислительная система, содержащая 2 дисковых накопителя различной емкости: 600Мб (назовем его HD1) и 800 Мб (HD2). Данные поступают на каждый из накопителей от своего источника. Объем очередной порции информации является случайной величиной, распределенной по нормальному закону распределения.

Для первого источника закон распределения СВ имеет параметры $m_1=10$, $v_1=3$; для второго источника $m_2=20$, $v_2=7$. Требуется сравнить эффективность использования накопителей. При этом показателем эффективности будет служить коэффициент использования дискового пространства – K_u . Эта величина может быть рассчитана как отношение объема памяти, использованного на интервале моделирования, к полной емкости накопителя.

На Рис.5.9 показана модель, позволяющая решить поставленную задачу.

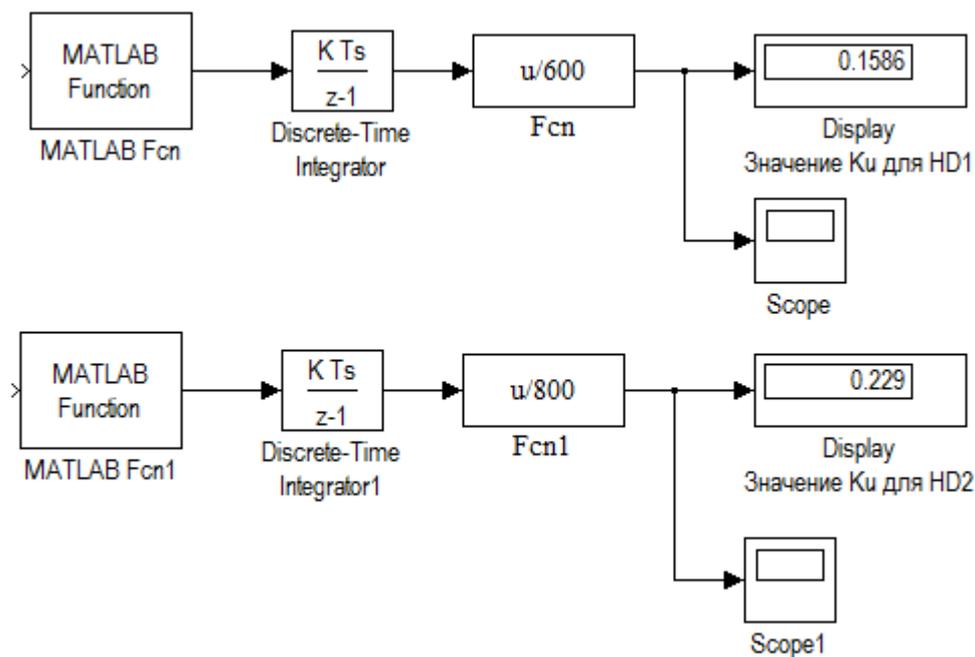


Рис. 5.9 Моделирование работы накопителей

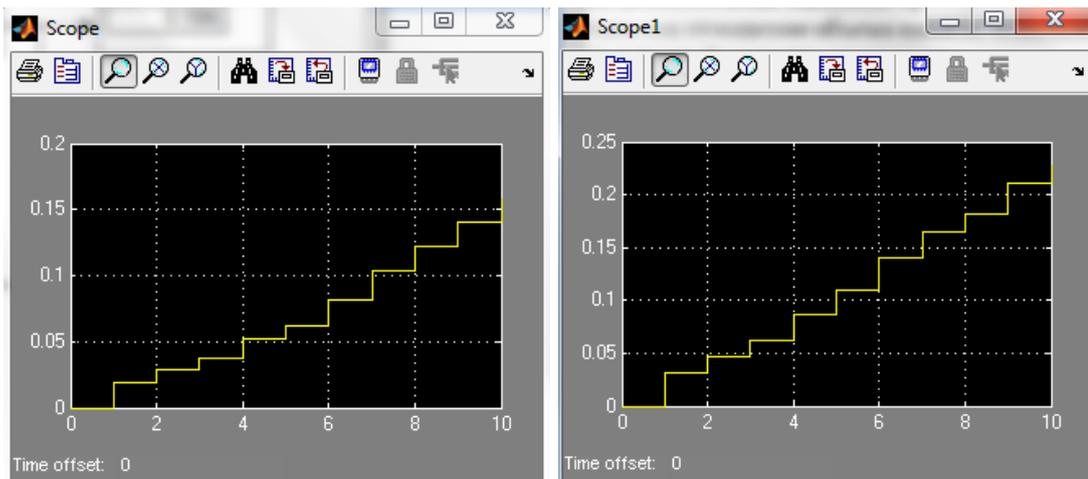


Рис. 5.10 Результат моделирования работы накопителей

Модель состоит из двух самостоятельных частей, каждая из которых моделирует работу одного из накопителей.

Блок MATLAB Fcn – является источником данных, в качестве параметра настройки блока указано имя генератора нормального распределения с соответствующим аргументами: $normrnd(10,3)$.

Блок Discrete-Time Integrator выполняет суммирование объема данных, поступающих от источника в накопитель, все параметры имеют значения, установленные по умолчанию.

Блок Fcn обеспечивает расчет значения K_u для текущего значения модельного времени ($u/600$).

Блок Display выводит на экран значение K_u . По истечении интервала моделирования оно представляет собой оценку данного показателя для накопителя HD1.

Перед запуском модели необходимо установить способ изменения модельного времени *Fixed-Step*.

Задание 5.5

Построить модель двух топливозаправочных станций и сравнить их эффективности использования на протяжении 11 дней.

Пример 5.5. Пусть имеется две топливозаправочные станции (ТЗС1 и ТЗС2). Показателем эффективности будет служить коэффициент оборудования топливозаправочных станций на протяжении 11 дней - K_u . Эта величина рассчитывается как отношение количества заправленных машин к потенциально возможной пропускной способности станций.

Первая топливозаправочная станция имеет 8 топливозаправочных колонок (ТЗК). Вторая топливозаправочная станция имеет 4 ТЗК. Среднее

время заправки одного автомобиля на любой из заправок составляет 5 мин. Работа ТЗС круглосуточная (24 часа). Количество автомобилей, заправленных в течение суток, – величина случайная и подчиняется нормальному закону распределения. Для первой ТЗС закон распределения СВ имеет параметры $m_1=1000$ авто, $v_1=250$ авто; для второй ТЗС: $m_2=850$ авто, $v_1=70$ авто.

На Рис.5.11 показана модель, позволяющая решить поставленную задачу.

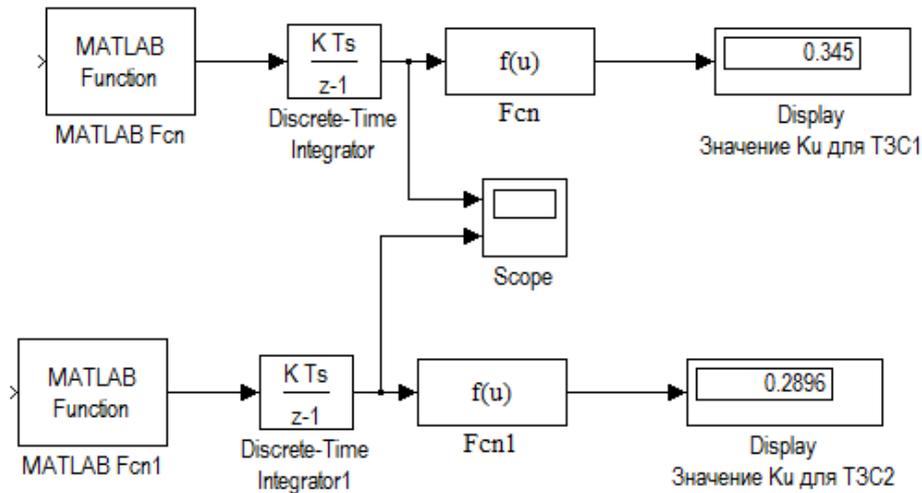


Рис. 5.11 Моделирования работы топливозаправочных станций ТЗС1 и ТЗС2

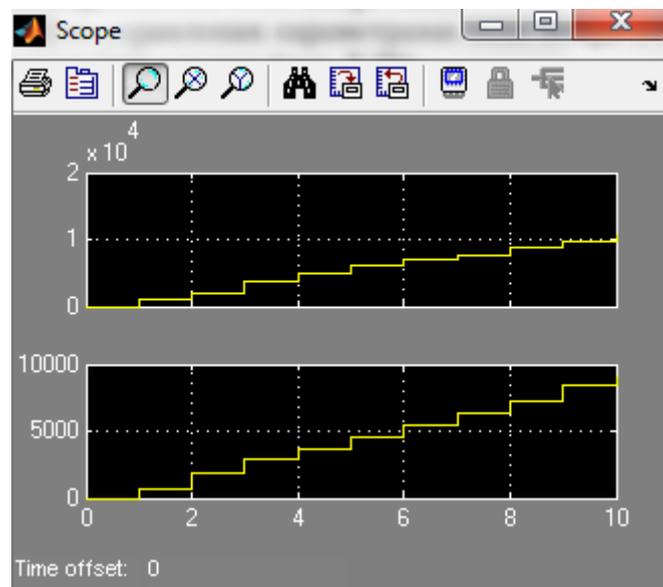


Рис. 5.12 Результат моделирования работы топливозаправочных станций ТЗС1 и ТЗС2

Модель состоит из двух самостоятельных частей, каждая из которых моделирует работу одной из топливозаправочных станций.

Блок MATLAB Fcn – в качестве параметра настройки блока указано имя генератора нормального распределения с соответствующим аргументами: для ТЗС1 $normrnd(1000, 250)$, ТЗС2 (850,70).

Блок Discrete-Time Integrator выполняет суммирование количества автомобилей, заправляющихся на ТЗС, все параметры имеют значения, установленные по умолчанию.

Блок Fcn обеспечивает расчет значения K_u (Рис.5.11).

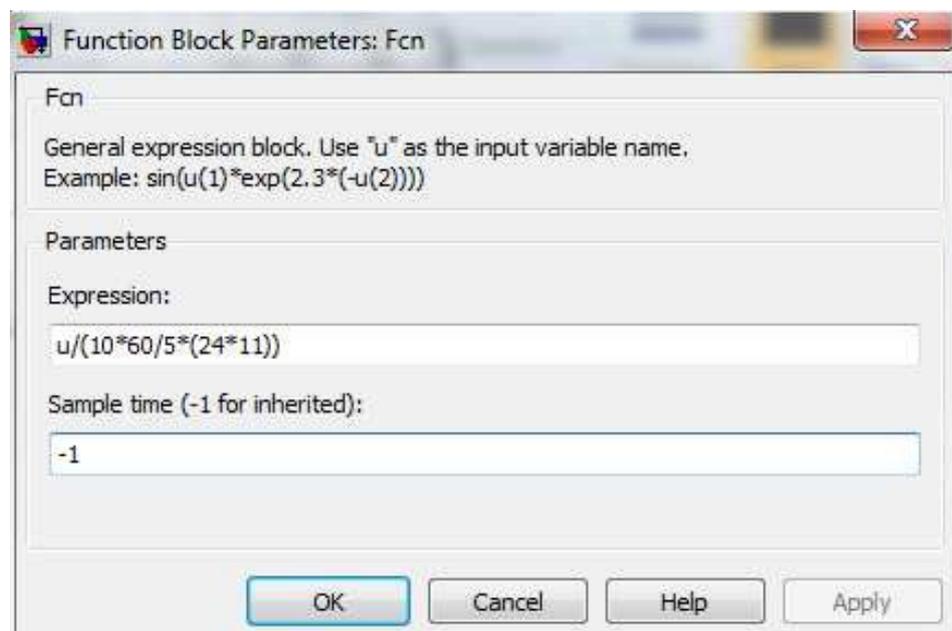


Рис. 5.12 окно настройки параметра Fcn

Лабораторная работа №6 моделирование одноканальных систем массового обслуживания

ЦЕЛЬ РАБОТЫ

- Моделирование одноканальных систем массового обслуживания с отказами;
- Моделирование одноканальных систем массового обслуживания с ожиданием при ограниченной очереди;
- Моделирование одноканальных систем массового обслуживания с ожиданием при неограниченной очереди.

ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Перед выполнением лабораторной работы необходимо ознакомиться с темами:

- Системы массового обслуживания и их характеристики [1, п.3, 5, 7];
- Одноканальные системы массового обслуживания [1, п.3, 5, 7];
- Виды представления времени в модели [1, п.3, 5, 7].

Ответить на вопросы:

1. Какие системы называются системами массового обслуживания?
2. Как классифицируются системы массового обслуживания по признаку их организации ?
3. Какие системы массового обслуживания называются системами ожиданием при ограниченной очереди?
4. Какие системы массового обслуживания называются системами ожиданием при неограниченной очереди?
5. Какие используются показатели эффективности одноканальных систем массового обслуживания ?
6. Какие исходные данные необходимы для моделирования одноканальных систем массового обслуживания?

7. Чем характеризуются потоки заявок на обслуживание?
8. Чем характеризуются механизмы обслуживания ?
9. По какому общему правилу вычисляется вероятность любого состояния системы массового обслуживания?
10. Что представляет собой моделирование с постоянным шагом?
11. Что представляет собой моделирование по особым состояниям?
12. Назначения блоков: MATLAB Fcn, Discrete-Time Integrator, Hit Crossing, Stop Simulation

Моделирования одноканальные системы массового обслуживания

Одноканальной называется система, у которой одна очередь, а так же канал для обслуживания только один, следовательно подходит, только если для перевозки пассажиров используется одна единица транспорта.

При разработке любой имитационной модели и планирования проведения модельных экспериментов необходимо соотносить между собой три представления времени:

- Реальное время, в котором происходит функционирование интегрируемой системы;
- Модельное (системное) время, в масштабе которого организуется работа модели;
- Машинное время, отражающее затраты времени ЭВМ на проведение имитации.

С помощью механизма модельного времени решаются следующие задачи:

- 1)) Отображается переход моделируемой системы из одного состояния в другое;
- 2)) Производится синхронизация работы компонент модели;
- 3)) Изменяется масштаб времени «жизни» (функционирования) исследуемой системы;
- 4)) Производится управление ходом модельного эксперимента;
- 5)) Моделируется квазипараллельная реализация событий в модели. Существуют два метода реализации механизма модельного времени:

- моделирование с постоянным шагом;
- моделирование по особым состояниям.

Выбор метода реализации механизма модельного времени зависит от назначения модели, ее сложности, характера исследуемых процессов, требуемой точности результатов.

Моделирование с постоянным шагом

При использовании метода постоянного шага отсчет системного времени ведется через фиксированные, выбранные исследователем интервалы времени. События в модели считаются наступившими в момент окончания этого интервала. Погрешность в измерении временных характеристик системы в этом случае зависит от величины шага моделирования Δt .

Метод моделирования с постоянным шагом используется на практике, если:

- События появляются регулярно, их распределение во времени достаточно равномерно;
- Число событий велико и моменты их появления близки;
- Моменты появления событий заранее определить невозможно.

Данный метод управления модельным временем достаточно просто реализуется в том случае, когда условия появления событий всех типов в модели можно представить как функцию времени.

Моделирование по особым состояниям

При моделировании по особым состояниям системное время каждый раз изменяется на величину, строго соответствующую интервалу времени до момента наступления очередного события.

В этом случае события обрабатываются в порядке их наступления, а одновременно наступившими считаются только те, которые являются одновременными в действительности.

Для реализации моделирования по особым состояниям требуется разработка специальной процедуры планирования событий (так называемого календаря событий).

Если известен закон распределения интервалов между событиями, то такое прогнозирование труда не составляет: достаточно к текущему значению модельного времени добавить величину интервала, полученную с помощью соответствующего датчика.

Моделирование по особым состояниям целесообразно использовать, если:

- события распределяются во времени неравномерно или интервалы между ними велики;
- предъявляются повышенные требования к точности определения взаимного положения событий во времени;
- необходимо учитывать наличие одновременных событий.

Дополнительное достоинство метода заключается в том, что он позволяет экономить машинное время, особенно при моделировании систем периодического действия, в которых события длительное время могут не наступать.

Моделирование одноканальных систем массового обслуживания в MATLAB.

При выборе одного из этих методов важное значение имеет тип моделируемой системы:

- Для непрерывных систем (с непрерывным временем смены состояний) по умолчанию используется переменный шаг приращения времени;
- Для дискретных систем следует устанавливать постоянный (фиксированный) шаг.

Такой подход не всегда оправдан, поскольку при моделировании непрерывных систем бывает удобнее определять очередное состояние системы как функцию времени, изменяющегося с заданной дискретностью. И наоборот, при моделировании дискретных систем величина очередного приращения времени зачастую определяется прогнозируемым моментом изменения состояния системы; причем смена состояний происходит, как правило, нерегулярно.

Поэтому полезно знать, каким образом при разработке моделей дискретных систем можно заставить модельное время изменяться по особым состояниям.

Команда Simulink / Configuration Parameters открывает диалоговое окно настроек параметров моделирования (рис.6.1).

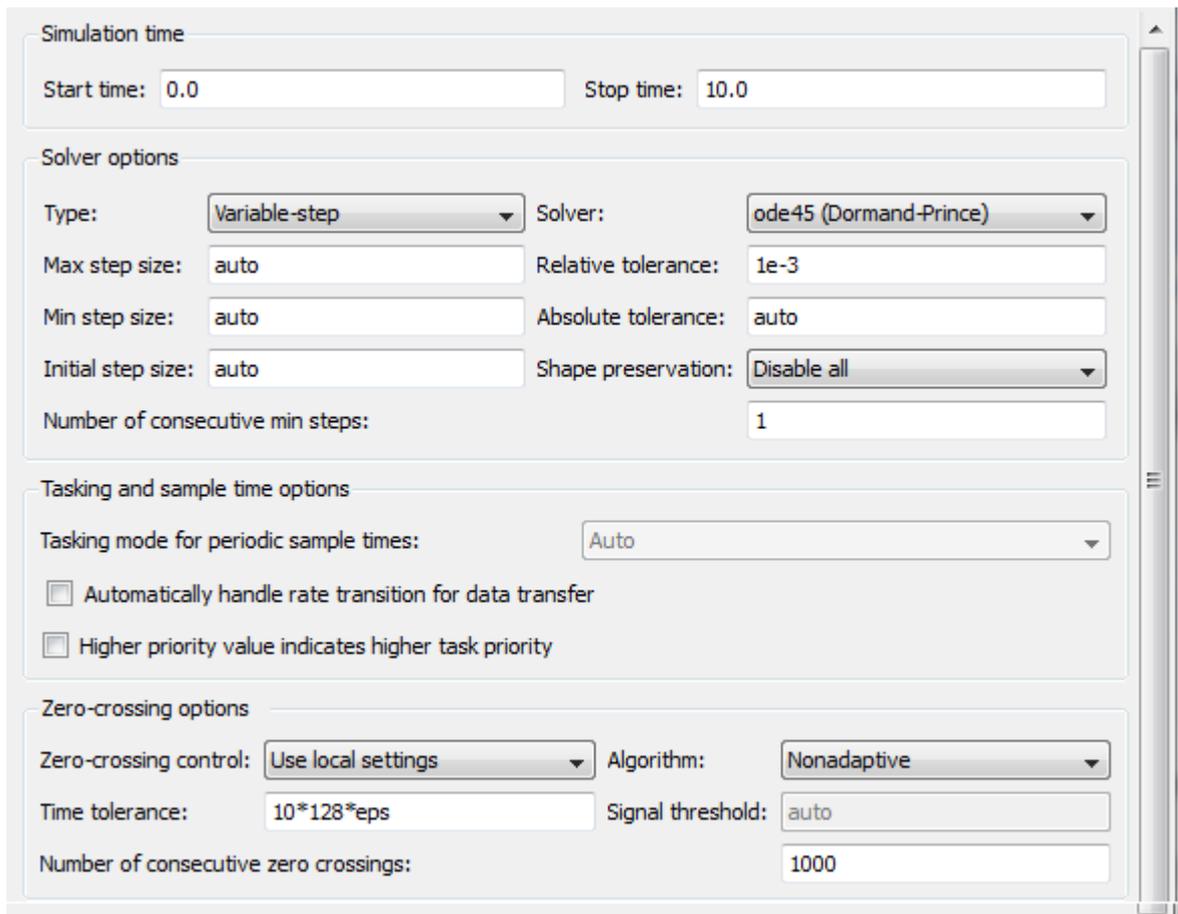


Рисунок 6.1 – Диалоговое окно настроек параметров моделирования

Simulation time (интервал моделирования) – величина интервала моделирования задается посредством указания начального (Start time) и конечного (Stop time) значений модельного времени;

Solver options (параметры расчета) – выбор метода реализации (расчета) модели;

Type (Тип) – позволяет выбрать способ изменения модельного времени; он содержит два пункта:

- Variable-step (моделирование с переменным шагом);
- Fixed-step (моделирование с фиксированным шагом).

Как правило, Variable-step используется для моделирования непрерывных систем, а Fixed-step – дискретных.

Solver (Расчет) – установка параметров расчета модели – разработчик может в ходе моделирования выбирать метод отображения динамики системы. С помощью закладки Solver система может быть отнесена к одному из следующих классов:

- С дискретным состоянием и дискретным временем перехода из одного состояния в другое;
- С дискретным состоянием и непрерывным временем переходов;
- С непрерывным состоянием и дискретным временем переходов;
- С непрерывным состоянием и непрерывным временем переходов.

Discrete – обеспечивает расчет дискретных состояний системы (и для непрерывного, и для дискретного времени переходов из состояния в состояние).

Solver /ODE – обеспечивают выбор метода расчета нового состояния для непрерывных систем. Эти методы различны для переменного (Variable-step) и фиксированного (Fixed-step) шага времени, но основаны на единой методике – решении обыкновенных дифференциальных уравнений (ODE).

Ниже двух раскрывающихся списков Type находится область, содержимое которой меняется в зависимости от выбранного способа изменения модельного времени.

При выборе Fixed-step в данной области появляется текстовое поле Fixed-step size (величина фиксированного шага) позволяющее указывать величину шага моделирования. Величина шага моделирования по умолчанию устанавливается системой автоматически (auto). Требуемая величина шага может быть введена вместо значения auto либо в форме числа, либо в виде вычисляемого выражения (то же самое относится и ко всем параметрам, устанавливаемым системой автоматически).

При выборе Variable-step в области появляются поля для установки трех параметров:

- Max step size – максимальный шаг расчета. По умолчанию он устанавливается автоматически (auto) и его значение в этом случае равно $(SforTime - StartTime)/50$. Довольно часто это значение оказывается слишком большим, и наблюдаемые графики представляют собой ломаные (а не плавные) линии. В этом случае величину максимального шага расчета необходимо задавать явным образом.
- Min step size – минимальный шаг расчета.
- Initial step size – начальное значение шага моделирования.

Для управления величиной шага моделирования при изменении модельного времени по особым состояниям используют два блока MATLAB Fcn и Discrete-Time Integrator (рис.6.2).

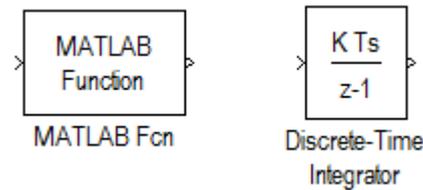


Рисунок 6.2 – Блоки, управления величиной шага моделирования

Блок MATLAB Fcn – обеспечивает формирование отрезков времени, длина которых распределена по заданному закону.

Блок Discrete-Time Integrator – выполняет суммирование этих отрезков, то есть «продвижении» модельного времени.

С помощью этих блоков можно реализовать механизм изменения модельного времени по особым состояниям для таких S-моделей, в которых условием наступления очередного события является истечение некоторого промежутка времени.

Управление окончанием моделирования

В MATLAB существует несколько способов окончания моделирования. Наиболее простой из них состоит в указании длительности интервала моделирования. При этом сеанс моделирования завершается при достижении модельным временем значения, заданного в поле Stop time при настройке параметров моделирования. Текущее состояние модели (или моделируемой системы) в данном случае не учитывается.

Также окончание моделирования мы можем задать с помощью блока Stop Simulation (Рис.6.3).

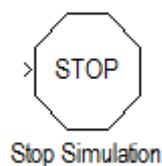


Рисунок 6.3 – Блок, окончания моделирования

Блок Stop Simulation – блок управления моделированием, который позволяет прервать моделирование при выполнении тех или иных условий.

С его помощью можно увязывать окончание сеанса моделирования с выполнением тех или иных условий:

- Переход моделируемой системы в определенное состояние;
- Завершения выполнения какой-либо операции (системой в целом или одной из ее подсистем);
- Достижения показателем эффективности заданного значения.

Условия окончания моделирования должны быть заданы в модели таким образом, чтобы при их наступлении на вход блока Stop Simulation поступал ненулевой сигнал (если сигнал векторный то хотя бы один элемент должен быть отличен от нуля).

ПРАКТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Задание 6.1 Построить одноканальную модель систем массового обслуживания потока заявок на обслуживание. Установить в параметрах моделирования фиксированный шаг моделирования.

Пример 6.1 Рассмотрим систему массового обслуживания, процессы которой мы хотим моделировать. Процесс функционирования такой системы можно рассматривать как последовательную смену ее состояний. Пусть, например, в одноканальной системе массового обслуживания происходит процесс обслуживания поступающих заявок. Предположим, необходимо моделировать поток посетителей супермаркета, подходящих к кассе для оплаты покупки. Наблюдения показали (или из других источников поступила информация), что в вечернее время с 17.00 до 18.00 в среднем через каждые 15 мин. приходит очередной посетитель. Среднее время между заявками на обслуживание $T_{cp}=1$ мин. Закон плотности распределения – экспоненциальный. Продвижение модельного времени задать с фиксированным шагом Fixed step.

Модель потока посетителей может выглядеть следующим образом (рис. 6.9).

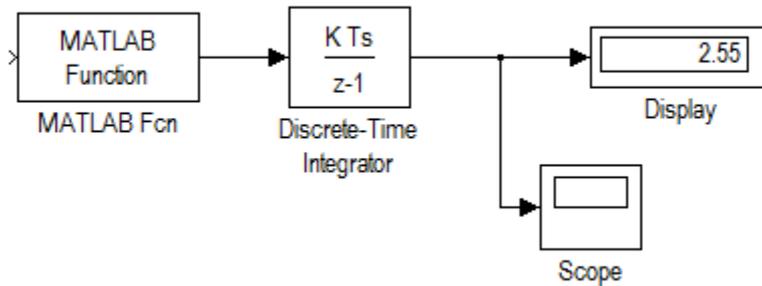


Рисунок 6.9 – Модель потока посетителей

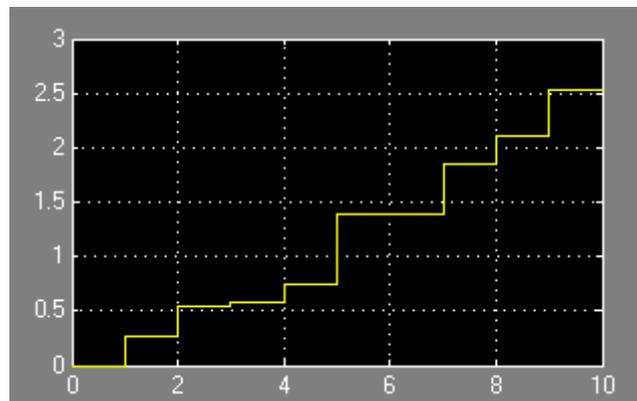


Рисунок 6.10 – Окно отображения общего время работы и количества посетителей

Блок MATLAB function имитирует случайные числа в соответствии с экспоненциальным законом распределения. С этой целью используется функция `expnrnd` с параметром 0.25, что соответствует среднему времени 15 мин. (15/60). Блок Discrete Time Integrator суммирует случайные числа, т.е. формирует время работы кассы нарастающим итогом. Блок Display показывает суммарное время работы кассы (в нашем случае оно будет случайное). Блок Scope1 отражает: по вертикали – общее время работы, которое складывается из случайных временных интервалов; по горизонтали – количество посетите-лей. С этой осью совпадает шаг моделирования.

В случае моделирования потока покупателей и стоимости их покупки управление модельным временем осуществляется по особым состояниям.

Задание 6.2

Построить модель потока заявок на обслуживание. Установить в параметрах моделирования переменный шаг моделирования.

Цель моделирования с помощью данной модели – прогнозирование доходов в супермаркете.

Пример 6.2 Выполнить модель потока заявок на обслуживание посетителей супермаркета) с обслуживанием заявок (под обслуживанием в данной задаче понимать прием денег в кассу).

Исходные данные для моделирования:

Закон плотности распределения интервалов между заявками – экспоненциальный.

Среднее время между заявками (интервал между покупателями) $T_{\text{ср}} = 1$ мин.

Закон плотности распределения суммы покупки- нормальный с параметрами $\text{МОЖ} = 200$ руб., $\text{СКО} = 50$ руб.

Время подсчета покупки принять равным «0».

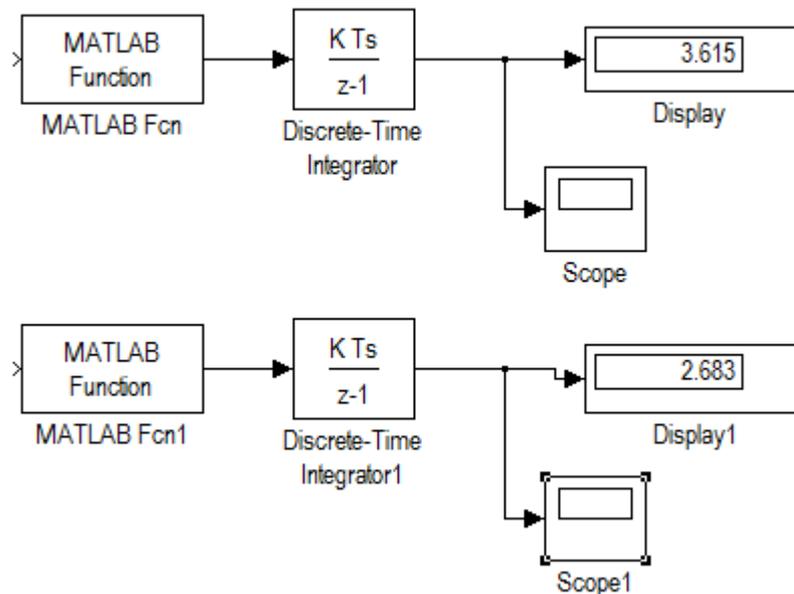


Рисунок 6.10 – Изменения модельного времени по особым состояниям

В блоке Scope, нижняя часть модели обеспечивает формирование отрезков времени, длина которых распределена по заданному закону (экспоненциальному), и продвижение модельного времени. Верхняя часть

модели имитирует случайную величину стоимости их покупки. Средняя стоимость покупки в модели принята за 0,3 тыс. руб. Из результатов одного прогона модели видно, что за 3 часа 60 минут в кассе супермаркета будет 2тыс. 600 рублей.

Лабораторная работа №7 моделирование многоканальных систем массового обслуживания

ЦЕЛЬ РАБОТЫ

Многоканальные системы массового обслуживания с ожиданием

- Моделирование одноканальных систем массового обслуживания с отказами;
- Моделирование одноканальных систем массового обслуживания с ожиданием при ограниченной очереди;
- Моделирование одноканальных систем массового обслуживания с ожиданием при неограниченной очереди.

ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Перед выполнением лабораторной работы необходимо ознакомиться с темами:

Моделирование многоканальных систем массового обслуживания [1, п.6,7,5].

Ответить на вопросы:

Что происходит с заявкой, поступившей в момент времени, когда все каналы обслуживания заняты в многоканальной системе массового обслуживания?

Что служит исходными данными для расчетов показателей эффективности многоканальных систем массового обслуживания?

Какие данные необходимы для моделирования многоканальных систем массового обслуживания?

Что представляет собой моделирование с переменным шагом?

Моделирование многоканальных системы массового обслуживания

Многоканальной называется система, у которой одна общая очередь, а каналов для обслуживания несколько, следовательно данный тип модели

наиболее подходящий для реализации, т.к. единиц техники обслуживающих пассажиров в большинстве случаев больше одной. Если будет использоваться только одна единица транспорта, то достаточно задать нужные параметры модели.

В качестве характеристик эффективности многоканальных систем массового обслуживания могут применяться следующие величины и функции:

- среднее количество заявок, которые может обслужить СМО в единицу времени;
- среднее количество заявок, получающих отказ и покидающих СМО;
- вероятность того, что поступившая заявка немедленно будет обслужена;
- среднее время ожидания в очереди;
- среднее количество заявок в очереди;
- средний доход СМО в единицу времени.

Системы с очередью делятся на:

Системы с неограниченным ожиданием, при этом поступившая в СМО задача становится в очередь и ждет обслуживания. Рано или поздно она будет обслужена;

Системы с ограниченным ожиданием, при этом на заявку в очереди накладываются ограничения, например ограниченное время пребывания в очереди, длина очереди, общее время пребывания в СМО. В зависимости от типа СМО для оценки эффективности могут быть применены разные показатели.

Синхронизация параллельных процессов

До этого мы рассматривали только асинхронные параллельные процессы, т.е. такие, которые не влияют друг на друга. «Привязку» таких процессов к единой оси модельного времени Simulink выполняет сам, освобождая от соответствующих проблем разработчика. Другое дело, когда имеют место синхронные параллельные процессы, состояние каждого из которых зависит от текущего состояния другого. При согласовании таких процессов должна учитываться специфика решаемой задачи, и без помощи разработчика Simulink здесь уже обойтись не может.

Для корректного управления модельным временем в моделях с синхронными параллельными процессами необходимо:

- Установить, какой из взаимодействующих процессов является подчиненным по отношению к другому.

- Определить, могут ли в подчиненном процессе происходить события, не связанные с изменением состояния управляющего процесса.
- Обеспечить приращение модельного времени на интервал времени до ближайшего события в управляющем процессе.
- Контролировать условия окончания сеанса моделирования.

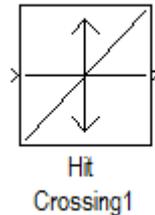


Рисунок 6.4 – Блок, идентификация моментов времени

Блок Hit Crossing (Обнаружено пересечение) – позволяет идентифицировать момент времени, когда входной сигнал «пересекает» некоторое значение: при появлении такой ситуации на входе блока формируется единичный сигнал.

Блок имеет три параметра настройки (Рис.6.5).

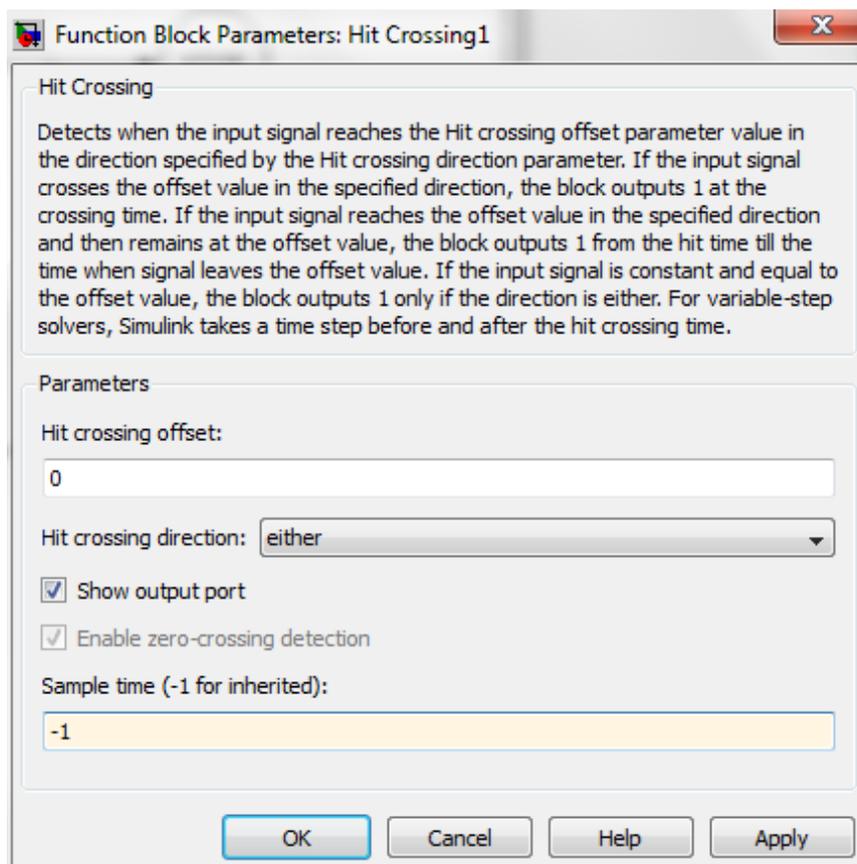


Рисунок 6.5 – Параметры настройки блока Hit Crossing

Hit Crossing offset – определяет значение, «пересечение» которого необходимо идентифицировать;

Hit Crossing direction – позволяет указать направление пересечения, которое интересует разработчика модели; значение этого параметра выбирается с помощью выпадающего меню, которое содержит три пункта:

- Rising (возрастание);
- Falling (убывание);
- Either (в обоих направлениях).

Show output port (показать выходной порт) – флажок, с помощью которого можно выбрать формат использования блока.

При одновременном выполнении условий, задаваемых параметрами Hit Crossing direction и Hit Crossing offset, на выходе блока формируется единичный импульс. Его длительность определяется значением параметра sample time блока, предшествующего в модели блоку Hit Crossing. Если этот параметр отсутствует, то единичный сигнал на выходе блока Hit Crossing существует до его следующего срабатывания.

ПРАКТИЧЕСКАЯ ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

Задание 7.1

Построить модель изменения модельного времени по особым состояниям, условием наступления очередного события является истечение некоторого промежутка времени.

Пример 7.1 Пусть имеется вычислительная система, содержащая 2 дисковых накопителя различной емкости: 600Мб (HD1) и 800 Мб (HD2). Длительность интервала моделирования равна 10 единицам модельного времени. Данные поступают на каждый из накопителей через интервалы времени, определяемые дискретностью изменения модельного времени $\Delta t = 0,2$.

Интервалы времени между поступлениями порций информации распределены по нормальному закону с параметрами $m_1=10$ $v_1=4$ ($\text{normrnd}(10,4)$), а для второго источника $m_2 =10$, $v_2 =7$.

Условием наступления событием является поступления на вход системы очередного сообщения, интервал времени между сообщениями является случайной величиной, распределенной по заданному закону. Запись информации на накопитель происходит мгновенно, и величина приращения модельного времени определяется только промежутком времени до появления очередной порции информации.

Модель работы будет выглядеть так, как показано на рис.6.6, для второго накопителя блок-диаграмма строится аналогичным образом.

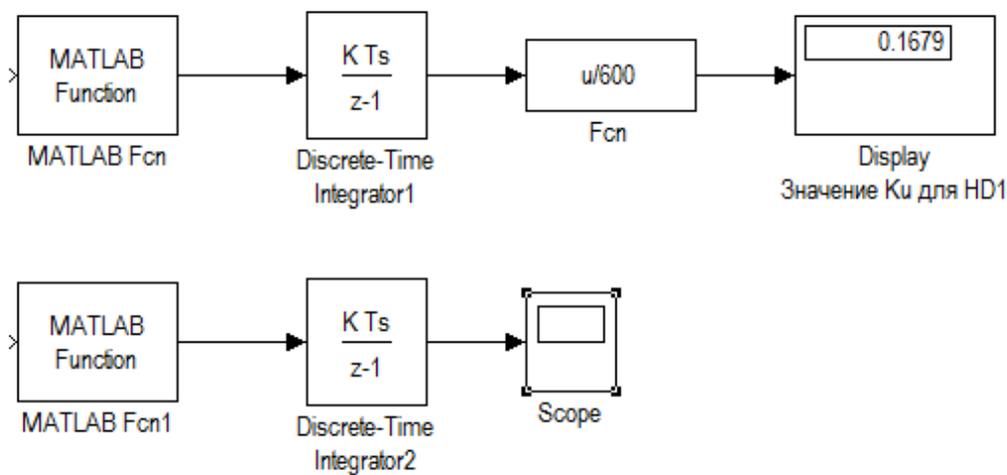


Рисунок 7.3 – Изменения модельного времени по особым стояниям

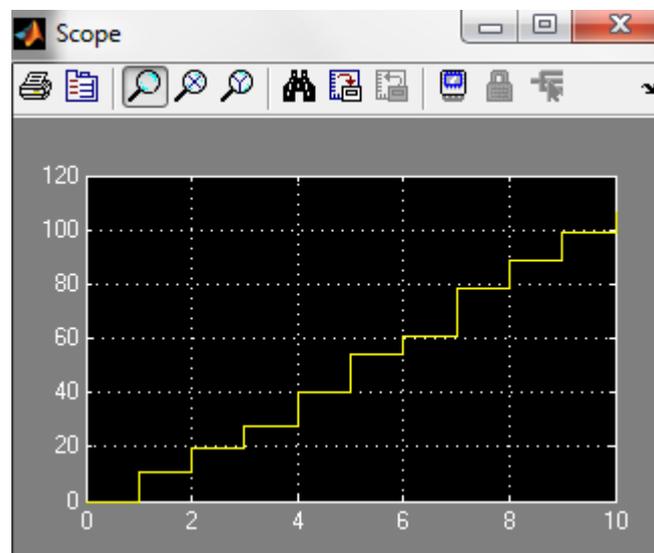


Рисунок 7.4 – Отображение модельного времени в блоке Scope

Для выполнения моделирования необходимо задать следующие параметры:

- 1) Время окончания моделирования $\text{Stop time}=10$;
- 2) Тип изменения модельного времени Variable-step ;
- 3) Интервалы времени между поступлениями порций информации распределены по нормальному закону с параметрами $m=10$ $v=4$ ($\text{normrnd}(10,4)$).

Изменения модельного времени отображается на блоке Scope . На его вход подается сигнал $\text{Discrete time Integrator}$, значения которого отображаются как функция от номера шага моделирования. Длительность последнего шага будет равна последнему значению сигнала на выходе блока $\text{Discrete time Integrator}$.

Тогда длительность интервала моделирования работы накопителей составит в среднем 100с.

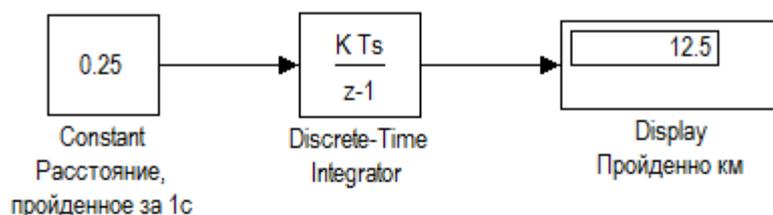
Задание 7.2

Моделирование с постоянным шагом

Пример 7.2 Некий самолет движется прямолинейно и с постоянной скоростью, а следующий за ним диспетчер должен ввести информацию о самолете до того, как он пересечет заданный рубеж.

Модель изображена на Рис 7.5.

Наблюдение за самолетом



Работа диспетчера

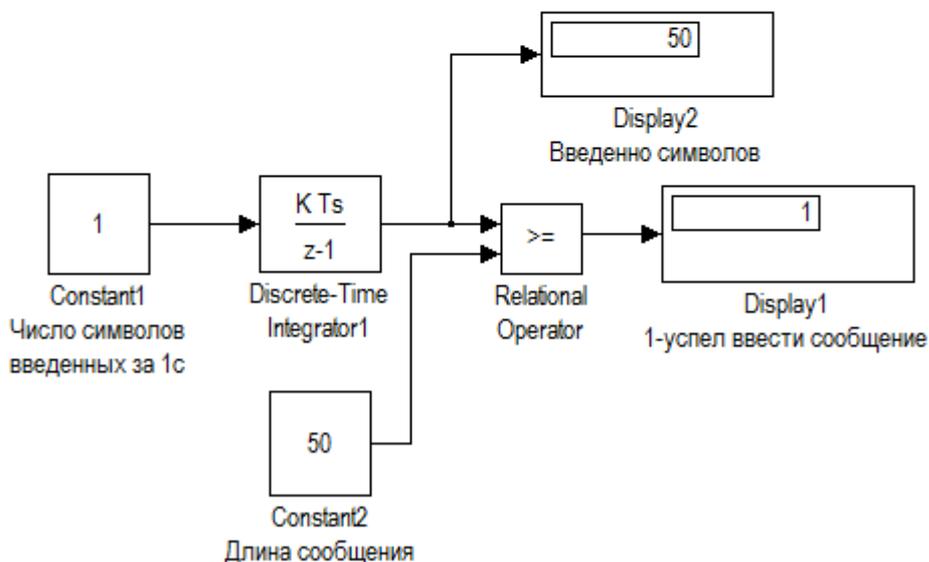


Рисунок 7.5 – Пример моделирования с постоянным шагом

Блок Discrete time integrator – вычисляет расстояние, пройденное самолетом, а блок Discrete time integrator1 – количество введенных диспетчером символов.

Для определенности будем считать, что фигурирующие в модели величины имеют следующие значения:

Скорость самолета: 900км/ч;

Расстояние до рубежа: 12,5 км;

Скорость ввода информации: 1 символ в секунду;

Длина текст: 50 символов.

Прежде чем установить параметры моделирования, мы должны:

- Привести скорость полета самолета и работы оператора к единому масштабу времени;
- Выбрать величину шага приращения модельного времени;
- Определить приемлемую длительность интервала моделирования.

Первый пункт выполнить проще всего: выражаем скорость самолета в км/с и получаем 0,25км/с.

Выполнение второго пункта требует сопоставление скоростей двух процессов; сравнив скорости, выбираем $\Delta t = 2c$.

Устанавливаем следующие параметры моделирования:

Top time: 50;

Type: Fixed step;

Fixed step size: 2.

Последний параметр требует небольшого пояснения: для всех блоков, имеющих параметр Sample time, он также должен быть равен 2.

Выполнив моделирование получаем ответ на поставленный ранее вопрос: диспетчер успеет (хотя и с трудом) ввести требуемую информацию.

Выбор шага моделирования $\Delta t = 2$ играет большую роль если мы изменим его на 3, тогда диспетчер не успеет ввести сообщение, сколько раз бы мы не повторяли эксперимент.

Задание 6.6

Собрать модель двух синхронных процессов на примере работы супермаркета. Оплата покупателя в кассу может начаться только при условии подхода его к кассе. Шаг моделирования вычисляется как сумма двух временных интервалов: промежутка до нового покупателя и длительности обслуживания его в кассе.

1-й процесс – процесс обслуживания заявки (процесс оплаты в кассе, при этом время обслуживания не равно «0», данный процесс подчиненный по отношению ко второму процессу).

2-й процесс – процесс потока заявок на обслуживание (поток покупателей, подходящих к кассе). Интервал между заявками (покупателями) соизмерим с временем обслуживания заявки (процессом оплаты).

Модель будет иметь вид, показанный на рис. 7.6.

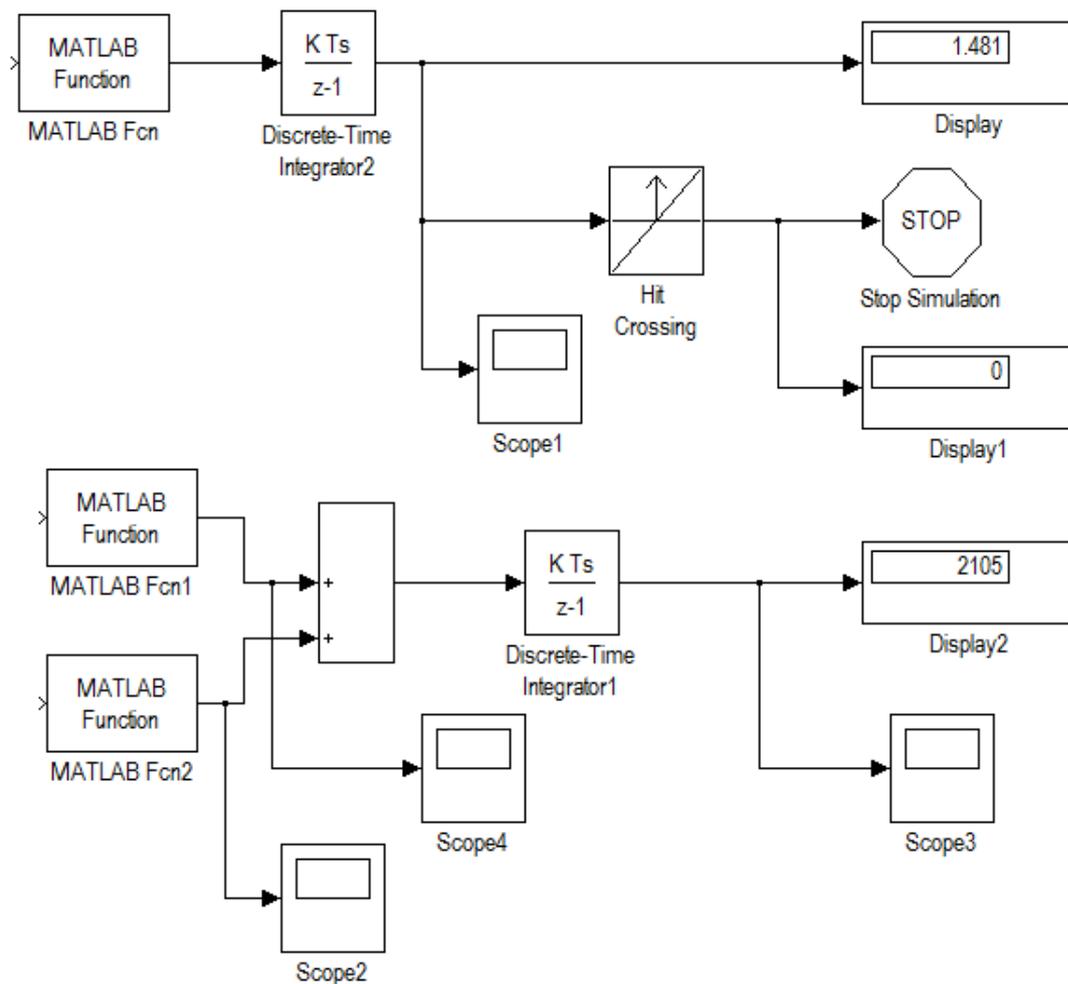


Рисунок 7.6 – Управление модельным временем при наличии двух синхронных процессов

Исходные данные для моделирования:

- Закон плотности распределения интервалов между заявками и времени обслуживания – экспоненциальный.
- Среднее время между заявками (интервал между покупателями) $T_{\text{ср}} = 1$ мин, среднее время обслуживания заявки (обслуживания в кассе) $T_{\text{ср.касс}} = 5$ мин.
- Закон плотности распределения суммы покупки – нормальный с параметрами $\text{МОЖ} = 200$ руб., $\text{СКО} = 50$ руб.;
- моделирование закончить по условию (сумма, поступившая в кассу, равна 2000 руб.).

Варианты для выполнения задания:

Задание 7.1

Проведите моделирование. Опишите результаты моделирования.

Доработать предыдущую имитационную модель для случая двухканальной системы обслуживания (например, две кассы). Окончание моделирования выполнить по условию просмотра 300 единиц модельного времени. Среднее время обслуживания в 1-й кассе (кассир более опытный и расторопный) $T_{\text{ср}} = 1$ мин., во 2-й кассе 5 мин. Интервал между покупателями $T_{\text{ср}} = 0,5$ мин. для обеих касс. Остальные условия такие же, как в предыдущей задаче. Часть модели, реализующая продвижение модельного времени по особым состояниям, представлена на рис.

Задание 7.2

Смоделировать работу двух накопителей HD1 и HD2 с использованием блока Stop Simulation. Условием окончания сеанса моделирования является исчерпания емкости хотя бы одного из накопителей (если накопитель заполнен то $K_u \geq 1$). Установить интервал моделирования Stop time =100.

Вычислительная система, содержит 2 дисковых накопителя различной емкости: 600Мб (назовем его HD1) и 800 Мб (HD2). Данные поступают на каждый из накопителей от своего источника. Объем очередной порции информации является случайной величиной, распределенной по нормальному закону распределения.

Для первого источника закон распределения СВ имеет параметры $m_1=10$, $v_1=3$; для второго источника $m_2=20$, $v_2=7$.