

СЕКЦИЯ 5.
Моделирование в Simulink

Часть 1.

Председатель:
д. т. н., профессор И. В. Черных

Оглавление

Анимица О. В., Кувшинов В. М. ПРОГРАММА ПОДДЕРЖКИ БАЗ ДАННЫХ «DATABASE MANAGER» ДЛЯ КОМПЛЕКСА «FLIGHTSIM» В СРЕДЕ ПРОГРАММИРОВАНИЯ MATLAB/SIMULINK	1349
Блинов О. В. МОДЕЛИРОВАНИЕ СИСТЕМЫ КОНТРОЛЯ ПЛОТНОСТИ ТКАНИ ПО УТКУ	1368
Борисов П. А. ПРИМЕНЕНИЕ MATLAB/SIMULINK ДЛЯ ИЗМЕРЕНИЯ И ОЦЕНКИ КАЧЕСТВА ЭЛЕКТРОЭНЕРГИИ В ТРЕХФАЗНЫХ СИММЕТРИЧНЫХ СИСТЕМАХ С АКТИВНЫМИ ПРЕОБРАЗОВАТЕЛЯМИ	1372
Браславский И. Я., Ишматов З. Ш., Плотников Ю. В. ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ SIMULINK ДЛЯ ОЦЕНКИ ПОТРЕБЛЕНИЯ ЭЛЕКТРОЭНЕРГИИ АСИНХРОННЫМ ЭЛЕКТРОПРИВОДОМ	1387
Браславский И. Я., Костылев А. В., Мезеушева Д. В., Степанюк Д. П., Кириллов А. В. МОДЕЛИРОВАНИЕ ВЕНТИЛЬНЫХ АСИНХРОННЫХ ЭЛЕКТРОПРИВОДОВ	1395
Васильев В. В., Симак Л. А., Тодорова А. М. МОДЕЛИ ДИНАМИЧЕСКИХ СИСТЕМ НЕЦЕЛОГО ПОРЯДКА В СРЕДЕ MATLAB/SIMULINK	1404
Васильева И. Л. ИМИТАЦИОННАЯ МОДЕЛЬ ЛЕСНОЙ ОТРАСЛИ ПРОИЗВОДСТВЕННО–ТЕХНОЛОГИЧЕСКОГО СЕКТОРА ЭКОНОМИКИ	1413
Веселов М. В. РЕШЕНИЕ ЗАДАЧ ГИДРАВЛИЧЕСКОГО РАСЧЕТА СИСТЕМ ЗАПРАВКИ РАКЕТЫ КОСМИЧЕСКОГО НАЗНАЧЕНИЯ СРЕДСТВАМИ ПАКЕТА SIMULINK	1418
Выдолоб Г. М. ИНТЕРАКТИВНАЯ АППРОКСИМАЦИЯ ФУНКЦИЙ В СРЕДЕ MATLAB/SIMULINK	1422
Ганеев А. Р., Шаньгин Е. С. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ АДАПТИВНОГО УПРАВЛЕНИЯ ДОБЫЧЕЙ НЕФТИ В ПАКЕТЕ MATLAB/SIMULINK	1427
Гордеев А. С., Гурьянов Д. В., Рындюк К. Д. ИМИТАЦИОННАЯ МОДЕЛЬ ОПРЕДЕЛЕНИЯ ТОВАРНОГО СОРТА ПЛОДОВ	1443

Грачев А. Д. SIMULINK МОДЕЛЬ МИКРОМОЩНОГО ПРОГРАММИРУЕМОГО ОПЕРАЦИОННОГО УСИЛИТЕЛЯ.....	1453
Данилов А. И. ПОСТРОЕНИЕ НА ЭЛЕМЕНТАХ SIMULINK ИЗМЕРИТЕЛЕЙ ФАЗЫ, ЧАСТОТЫ И АМПЛИТУДЫ	1458
Жесткова Ю. Е., Чернецов В. И. МОДЕЛИРОВАНИЕ СРЕДСТВАМИ ПАКЕТА SIMULINK ИНТЕГРИРУЮЩИХ АНАЛОГО-ЦИФРОВЫХ ПРЕОБРАЗОВАТЕЛЕЙ.....	1466
Жуков К. Г. ПРИМЕНЕНИЕ ПАКЕТА FIXED-POINT BLOCKSET В РАЗРАБОТКЕ УСТРОЙСТВ РЕАЛЬНОГО ВРЕМЕНИ	1474
Ионов Ю. Г., Чекмарев С. В. ИМИТАЦИОННАЯ МОДЕЛЬ ИЗМЕРИТЕЛЬНОГО ПРЕОБРАЗОВАТЕЛЯ ПО ОПРЕДЕЛЕНИЮ ПАРАМЕТРОВ ПЛАЗМЕННОГО ПОТОКА.....	1483
Ишматов З. Ш., Волков М. А. ИСПОЛЬЗОВАНИЕ МАТЛАВ ДЛЯ СИНТЕЗА МИКРОПРОЦЕССОРНЫХ СИСТЕМ УПРАВЛЕНИЯ МЕТОДОМ ПОЛИНОМИАЛЬНЫХ УРАВНЕНИЙ	1491
Казаков В.А., Михотин В.Д., Тихонова Е.А. МОДЕЛИРОВАНИЕ СРЕДСТВАМИ SIMULINK/POWER SYSTEM BLOCKSET ПЕРЕМЕННЫХ RLC-ЭЛЕМЕНТОВ ЭЛЕКТРИЧЕСКИХ ЦЕПЕЙ	1501
Комаров М. В, Широков С. В, Соломин О. В, Данчин И. А. МОДЕЛИРОВАНИЕ И ОБРАБОТКА ВИБРАЦИОННЫХ СИГНАЛОВ СИСТЕМЫ «РОТОР-ПОДШИПНИКИ СКОЛЬЖЕНИЯ- ДЕМПФЕРЫ» В СИСТЕМЕ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ MATLAB/SIMULINK	1515
Коровкин М. В., Погожев С. В. ВОПРОСЫ ПОСТРОЕНИЯ СЕТЕВЫХ КОМПЛЕКСОВ В MATLAB/SIMULINK	1524
Кувшинов В. М., Анимица О. В. РАЗВИТИЕ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ МОДЕЛИРОВАНИЯ И АНАЛИЗА ДИНАМИКИ САМОЛЕТА С СИСТЕМОЙ УПРАВЛЕНИЯ В СРЕДЕ ПРОГРАММИРОВАНИЯ MATLAB/SIMULINK («FLIGHTSIM»).....	1533

УДК 629.735.33

ПРОГРАММА ПОДДЕРЖКИ БАЗ ДАННЫХ «DATABASE MANAGER» ДЛЯ КОМПЛЕКСА «FLIGHTSIM» В СРЕДЕ ПРОГРАММИРОВАНИЯ MATLAB/SIMULINK

Анимица О. В., Кувшинов В. М.

Центральный Аэрогидродинамический институт (ЦАГИ)

им. проф. Н.Е. Жуковского, Жуковский

e-mail: vmkouvsh@mail.progtech.ru

Введение

Во многих областях техники модели различных устройств являются достаточно сложными и содержат большое количество разнообразных нелинейных характеристик в виде одномерных, двумерных и трехмерных зависимостей.

В процессе разработки модели какого-нибудь устройства или при его модификации многие параметры и характеристики (нелинейные зависимости) могут уточняться, изменяться, дополняться и так далее.

При работе с достаточно сложными моделями технических устройств в MATLAB/Simulink является очень удобным хранить все необходимые численные данные параметрически, то есть в виде имен переменных и массивов, которые содержатся в рабочем поле MATLAB Workspace.

Для долговременной работы все эти данные (одномерные и многомерные массивы характеристик, а также параметры модели в виде чисел или векторов) удобно содержать в одном файле, который можно хранить на диске и переносить с одного компьютера на другой, а также редактировать. Для моделирования в Simulink данные из файла такого типа можно загружать в рабочее поле MATLAB Workspace.

Для выполнения таких операций в моделях среды MATLAB/Simulink разработана программа поддержки баз данных **Database Manager** (функция **dbm**), обладающая следующими свойствами:

1. возможность численного и графического редактирования содержимого базы данных (одномерных, двумерных, трехмерных числовых массивов);
2. возможность создания новых баз данных, редактирования и дополнения существующих, путем «захвата» числовых массивов из различных источников;
3. возможность автоматизированного формирования элементов моделей Simulink в виде одномерных, двумерных, трехмерных таблиц типа Look-Up Table.

Database Manager запрограммирована полностью на языке MATLAB (содержит в общей сложности около 12 000 строк кода) и является хорошо испытанным в работе приложением. Программа разрабатывалась, проверялась в работе и улучшалась с учетом замечаний и пожеланий пользователей на протяжении трех лет.

Данная программа имеет обычный для Windows оконно-диалоговый интерфейс и предназначена для работы в среде MATLAB версии 6.1 и выше. Она работает с базами данных в виде структур MATLAB, тип которых описан далее в разделе 1.

Эти структуры содержатся в mat-файлах. Имя mat-файла должно быть таким же, как и имя самой структуры, поэтому нельзя изменять имена mat-файлов вне программы **Database Manager**. Для mat-файлов, содержащих структуры баз данных, принято особое расширение *.dbm.

Так как программа написана полностью в виде файлов-функций MATLAB, при ее работе не происходит никакого взаимодействия с рабочим полем MATLAB Workspace, кроме тех случаев, где ставится задача загрузки содержимого базы данных в рабочее поле.

Все основные функции, диалоги и интерфейсные элементы программы описаны ниже в разделе 2.

Способ взаимодействия программы и моделей Simulink описан в разделе 3.

1. Структура базы данных

В качестве базы данных рассматривается набор численных массивов, описывающих характеристики какого-либо устройства. Часть этих массивов является аргументами, а другая часть — функциями от этих аргументов. Также в базу данных могут входить некие параметры в качестве отдельных чисел либо числовых массивов (векторов).

Например, при моделировании динамики самолета в качестве аргументов базы данных аэродинамических характеристик часто используются массивы точек угла атаки α , числа Маха M , углов отклонений органов управления, а в качестве функций от этих аргументов - аэродинамические коэффициенты C_y , m_z и другие. В качестве параметров в базу данных часто входят инерционно-массовые и геометрические параметры самолета (вес, моменты инерции, площадь крыла, хорда и т. п.).

Все вышеуказанные численные данные могут использоваться при построении динамической модели устройства в среде MATLAB/Simulink, где параметры обычно используются как значения констант, постоянных коэффициентов усиления и в других элементах, а аргументы и функции — в одномерных, двумерных, трехмерных таблицах интерполяции.

Кроме этого, в базе данных может присутствовать и некоторая текстовая часть — описание базы данных — комментарий.

Все эти численные и текстовые данные удобно хранить, считывать и записывать в виде отдельных файлов, а для работы с ними необходимо иметь специальный инструмент — программу поддержки базы данных.

При работе с базой данных необходимо иметь возможность обрабатывать наборы данных (массивов), различных по размеру (и даже по типу, если базу данных нужно снабдить текстовыми комментариями).

В среде MATLAB существуют механизмы для работы с подобными разнородными данными класса структур. Структура — объект, состоящий из независимых полей, которые содержат данные разного типа (числа или массивы чисел, текст или массивы текста) и которые сами могут являться структурами. Структуру в MATLAB можно создать и редактировать при помощи команд **fieldnames**, **setfield**, **getfield** и других. Командами **save** и **load** осуществляется запись и считывание структур из так называемых mat-файлов (файлов особого формата для хранения данных в MATLAB). Расширение mat-файлов по умолчанию *.mat, но пользователь при записи может задавать и другие расширения.

Таким образом, в среде MATLAB целесообразно формировать базы данных в виде структур. Имя структуры является также именем mat-файла, в котором она хранится. Как указывалось выше, в качестве расширения mat-файлов, с которыми работает программа обработки баз данных, принято *.dbm.

Так как база данных состоит из 4-х основных типов данных (аргументы, функции, параметры, комментарии), эту структуру (в дальнейшем - Struct) целесообразно составить из 4 основных полей:

Struct.Arg — набор массивов аргументов,

Struct.Fun(1:n) — набор массивов функций,

Struct.Prm — набор параметров,

Struct.Comments — текстовые комментарии.

Поле комментариев содержит общую текстовую часть базы данных, например:

Struct.Comments = «База данных для модели No. 1».

Поля аргументов и параметров представляют собой подструктуры. В этих подструктурах поля носят имена величин, описываемых ими, и содержат числа либо вектора. Например:

Struct.Prm.g = 9.81 — параметр «g» равен 9.81;

Struct.Arg.Alpha = [0:10:90] — в массиве аргумента «Alpha» содержится 10 точек от 0 до 90 с шагом 10.

Для каждой функции задается несколько свойств. Поэтому поля функций представляют собой пронумерованные по количеству функций подструктуры с 6 полями:

Struct.Fun(i).ArgNames — наименования аргументов функции (для графиков);

Struct.Fun(i).ArgArr — имена массивов аргументов функции;

Struct.Fun(i).Name — наименование функции (для графиков);
Struct.Fun(i).ArrName — имя массива функции, используемое в моделях;

Struct.Fun(i).Arr — численный массив значений табличной функции;

Struct.Fun(i).FunComments — текстовый комментарий к данной функции.

Поля 1 и 3 используются в основном при выводе функций на графики, где можно преобразовать обычный текст в греческие, подстрочные, надстрочные и другие особые символы, используя технологию TeX. Например, если в поле **Name** функции содержится строка «C_y(alpha)», то заголовок на графике будет выглядеть как C_y(α) (см. рис. 2, 8, 9, 10), что очень удобно при использовании графиков для отчетной документации.

Поля 2 и 4 используются для работы в MATLAB Workspace. В них содержатся имена массива функции и массивов ее аргументов (из списка аргументов в основной структуре). Поле 5 содержит сам массив функции (одномерный, двумерный или трехмерный). Поле 6 содержит любые текстовые пояснения к данной функции.

Общий формат структуры базы данных представлен на рис. 1.

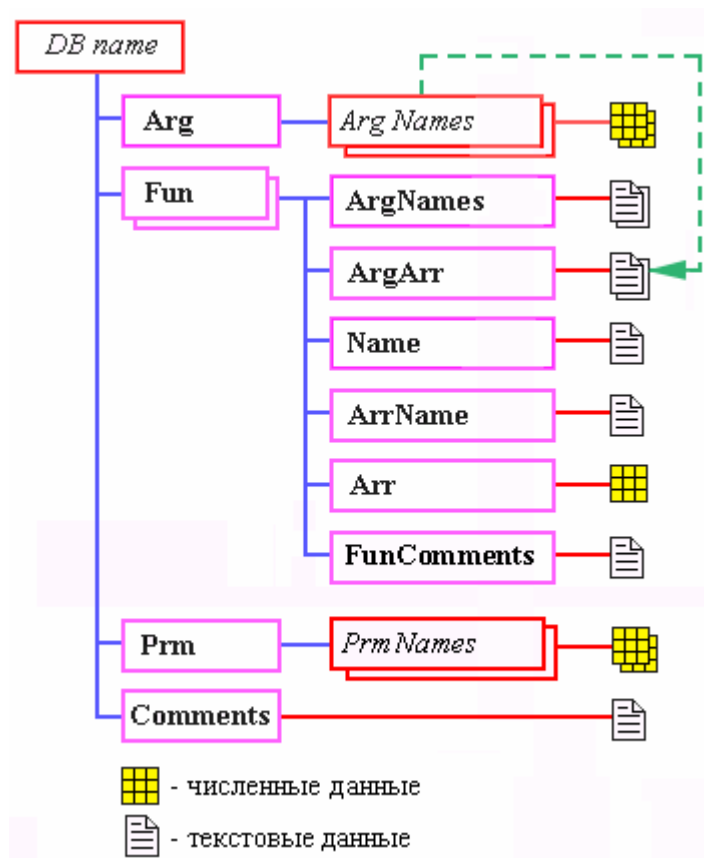


Рис. 1. Структура базы данных.

Следует отметить, что никаких программных ограничений на количество параметров, аргументов и функций в структуре не существует, и базы могут быть довольно большими. Например, база данных, включающая 200 параметров, 200 аргументов и 200 функций, содержится в **mat**-файле размером около 800 килобайт.

2. Описание управляющей программы поддержки баз данных

Для обработки структуры с данными рассмотренного выше типа в среде MATLAB разработана программа **Database Manager** (функция **dbm** для среды MATLAB).

Запуск программы поддержки баз данных:

При вызове этой функции без аргументов пользователю предлагается выбрать **mat**-файл с расширением *.dbm, содержащий базу данных (структуру), в стандартном окне выбора файлов операционной системы Windows. Для выбранного файла открывается окно редактирования, представленное на рис.2. Имя и путь файла с базой данных выносятся в заголовок окна. Сама структура считывается и хранится в буфере окна, и все ее изменения происходят там же, пока пользователь не нажмет на кнопку «Сохранить файл базы данных» (в этом случае перезаписывается **mat**-файл базы данных на диске).

Для создания новой структуры данных (в новом окне) нужно вызвать функцию **dbm** с параметром **new**: **dbm('new')**.

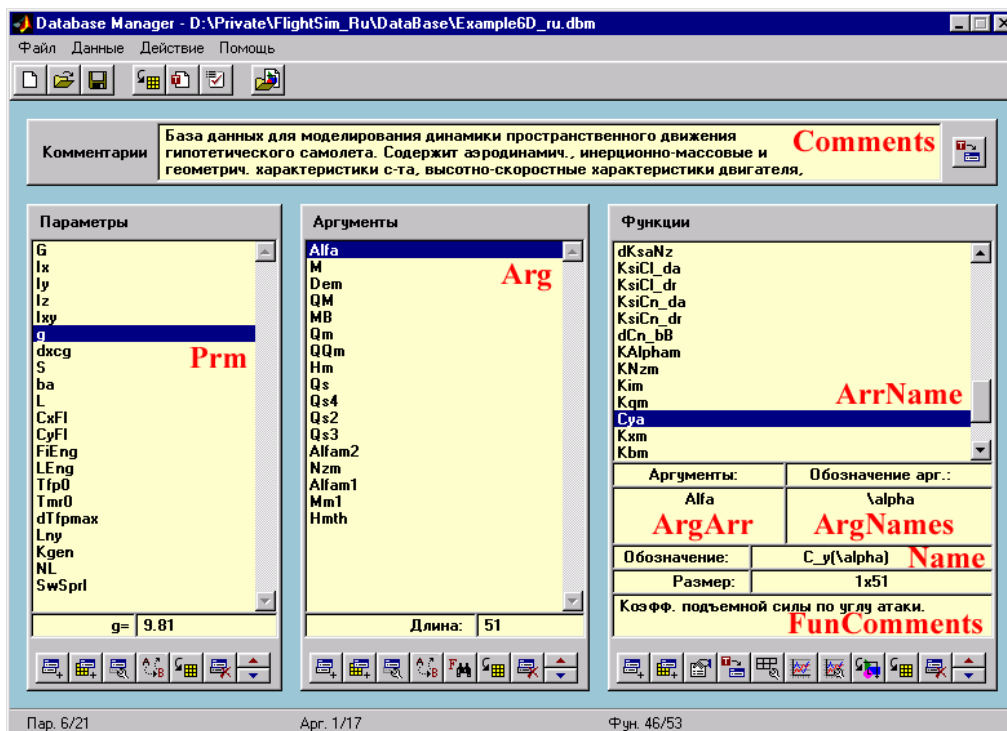









Рис. 2. Главное окно управляющей программы поддержки базы данных.

Для правильной работы программы папка с файлом dbm.m должна быть добавлена в MATLAB Path. Программа использует также некоторые функции в отдельных файлах. Эти файлы должны быть помещены в ту же самую папку, что и основная программа dbm.m.

Основные элементы окна управляющей программы:

Вверху окна обработки базы данных содержится панель инструментов и ниспадающих (pulldown) меню, предназначенных для выполнения общих операций с базой данных и сервисных функций. На панели инструментов расположены следующие кнопки (pushbuttons) (по порядку слева направо):

1.  — «Создать новую базу данных» создает в новом окне новую базу с одним параметром, аргументом и функцией, которая не содержится ни в каком mat-файле.
2.  — «Открыть файл базы данных» открывает окно выбора файла с расширением *.dbm и загружает в новое окно базу данных, считываемую из выбранного mat-файла.
3.  — «Сохранить файл базы данных» сохраняет базу данных в mat-файле под именем, содержащемся в заголовке окна. Если база данных не была считана из mat-файла, программа предлагает пользователю сохранить базу в виде файла с новым именем.
4.  — «Загрузить в Workspace» загружает в MATLAB Workspace все массивы параметров, аргументов и функций из базы данных, где они становятся доступным для других приложений среды MATLAB.
5.  — «Создать текстовый файл» создает текстовый файл специального формата, в котором записываются все комментарии, параметры, аргументы и функции базы данных.
6.  — «Проверка базы данных» проверяет соответствие всех аргументов и функций базы данных и выдает сообщения об ошибках различного типа и предупреждения.
7.  — «Открыть модель Simulink» открывает окно выбора файла с расширением *.mdl и затем открывает модель системы Simulink, соответствующую выбранному файлу.

В результате выполнения команды «Проверка базы данных» открывается окно, содержащее сообщение об отсутствии ошибок в базе данных или перечисление найденных ошибок следующего вида (рис. 3).

Выполняется проверка возможных ошибок следующих типов:

- Размерность массива функции не совпадает с количеством ее аргументов.
- Аргумент функции отсутствует в списке аргументов базы данных.
- Размер массива аргумента не соответствует размеру массива функции по этому аргументу.

- Две функции имеют одинаковые имена массивов (ArrName), либо совпадают имена двух массивов из списка функций, аргументов или параметров, что может привести к потере данных в рабочем поле Workspace.



Рис. 3. Пример окна отчета об ошибках.

Кроме сообщения об ошибках, программа выдает предупреждение в случае, если аргумент из списка нигде в функциях не используется.

Перечисленные выше действия доступны также из ниспадающих меню окна обработки базы данных.

Кроме этого, ниспадающие меню включают также некоторые дополнительные действия, например:


- сортировка набора параметров, аргументов, функций базы данных по разным признакам;
- слияние двух баз данных в одну базу (объединение двух файлов);
- очистка MATLAB Workspace (полная или частичная).

Для наиболее часто используемых операций в меню введены сочетания «горячих клавиш» (Ctrl+...).

Внизу окна расположена строка состояния, где приводятся номера текущих (выделенных) аргументов, функций и параметров, а также полное их число в базе данных.


Основная часть окна состоит из 4 областей (для комментариев, параметров, аргументов, функций). В этих областях приводятся списки данных для каждого раздела и информация о выбранном элементе из списка. Кроме этого, в каждой области расположены управляющие кнопки (pushbuttons) для работы только с данными из этого раздела базы данных.

1. Комментарии.

Текст комментариев к базе данных содержится в верхней части окна. Чтобы изменить комментарии в структуре базы данных (хранящейся в окне), нужно нажать кнопку  — «Прочитать/изменить комментарии», расположенную по правую сторону от текста комментариев.

2. Параметры.

В области параметров приводится список параметров (чисел или массивов). Элементы списка располагаются в нем в порядке создания. Ниже от списка находится окно, куда выводится значение текущего параметра (или его размер, если это вектор). В области параметров расположены следующие управляющие кнопки:

1.  — «Добавить новый параметр». При нажатии на эту кнопку появляется небольшое окно диалога, позволяющее пользователю ввести в структуру новый параметр (имя и значение задаются в окне диалога).

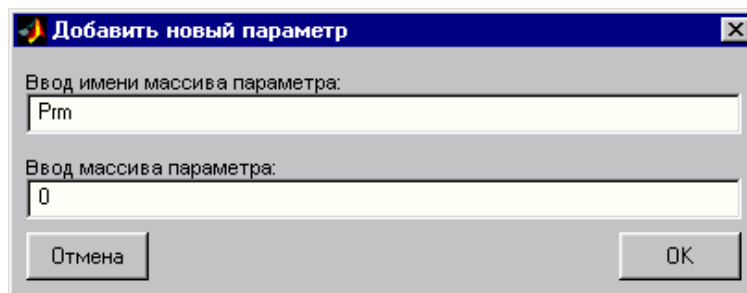





Рис. 4. Диалог ввода нового параметра.

2.  — «Добавить новый параметр из Workspace». Так как вектора параметров (как и аргументов) могут иметь большую длину, иногда их удобнее формировать вне базы данных и добавлять их в структуру через рабочее поле MATLAB Workspace. При нажатии на эту кнопку открывается диалог со списком всех переменных и векторов, содержащихся в данный момент в Workspace (см. рис. 5). Пользователь может выбрать любой из них. Далее открывается вспомогательный диалог, аналогичный тому, что вызывается при действии «Добавить параметр». В нем пользователь имеет возможность изменить имя и значение параметра из Workspace, после чего новый параметр заносится в структуру базы данных.
3.  — «Просмотреть/изменить параметр». Здесь появляется диалог, где можно просмотреть и изменить значение текущего параметра в списке.

4.  — «Переименовать параметр» позволяет изменить имя массива.

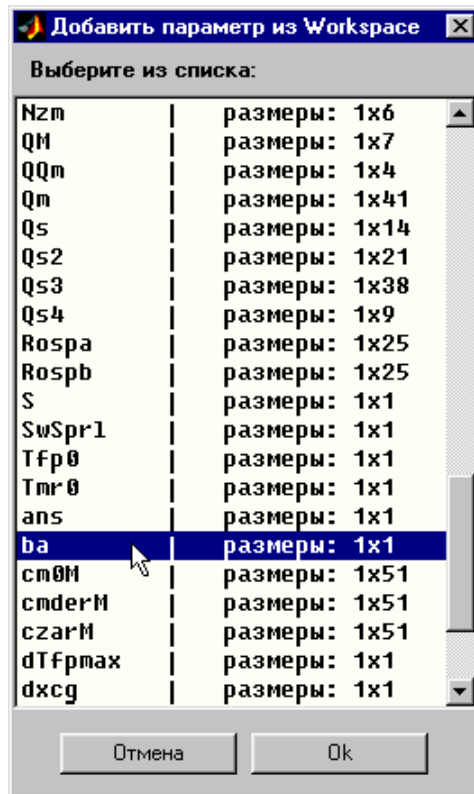



Рис. 5. Диалог ввода нового параметра из Workspace.

5.  — «Отправить параметр в Workspace». При нажатии на эту кнопку выбранный параметр загружается в MATLAB Workspace, где он становится доступным для других приложений среды MATLAB, например Simulink. При работе с моделями Simulink часто бывает удобным временно заносить в Workspace под именем данного массива нулевой (ZEROS) либо единичный (ONES) массив. Пользователь может осуществить это при помощи следующего диалога:

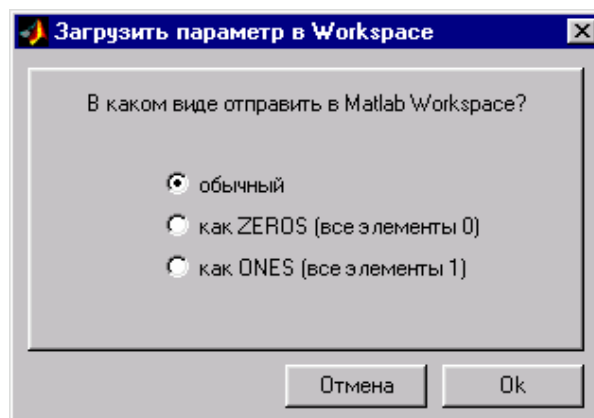












Рис. 6. Диалог выбора содержимого параметра, отправляемого в Workspace.

6.  — «Удалить выделенный параметр» удаляет выделенный параметр из структуры по предварительному подтверждению. Это означает, что до удаления программа вызывает простой диалог типа MessageBox для Windows, в котором просит подтвердить, действительно ли нужно удалить данный параметр:
7.  — «Переместить вверх» и «Переместить вниз» перемещают параметр по списку.

3. Аргументы.

Область аргументов аналогична области параметров, так как строение полей аргументов и параметров структуры одинаково, и над параметрами и аргументами производятся аналогичные действия.

В этой области приведен список массивов аргументов. Элементы списка располагаются в нем в порядке создания. Ниже от списка находится окно, куда выводится размер текущего аргумента (вектора). В области аргументов расположены следующие кнопки (по порядку слева направо):

1.  — «Добавить новый аргумент». При нажатии на эту кнопку появляется небольшой диалог (см. рис. 4), позволяющий пользователю ввести в структуру новый аргумент (имя и значение задаются в диалоге).
2.  — «Добавить новый аргумент из Workspace» позволяет добавить в базу данных новый вектор аргумента из рабочего поля подобно тому, как это описано в случае добавления параметра из Workspace.
3.  — «Просмотреть/изменить выделенный аргумент». Здесь появляется диалог (как на рис. 4), где можно просмотреть и изменить значение текущего аргумента в списке (но не его имя).
4.  — «Переименовать выделенный аргумент и обновить функции» позволяет изменить имя массива и соответствующие обозначения аргументов всех функций, в которых он используется.
5.  — «Найти функции выделенного аргумента» производит поиск функций в списке, у которых в массиве аргументов присутствует данный аргумент. Результаты выводятся в окно, аналогичное окну, приведенному на рис. 3.
6.  — «Загрузить выделенный аргумент в Workspace». При нажатии на эту кнопку выбранный аргумент загружается в MATLAB Workspace, где он становится доступным для других приложений среды MATLAB, например Simulink.
7.  — «Удалить выделенный аргумент» удаляет аргумент из структуры по предварительному подтверждению.
8.  — Кнопки «Переместить вверх» и «Переместить вниз» перемещают аргумент по списку.

4. Функции.

Поля функций в структуре имеют самый сложный вид, и программа позволяет осуществить многочисленные действия над функциями базы данных.




В области функций приводится список массивов функций (содержимое поля ArgName в подструктуре каждой функции — см. предыдущий раздел) в порядке возрастания номера функции в структуре. В этом списке можно выделять как одну, так и несколько функций, например для изображения на графике или для загрузки в Workspace.

Ниже в окне приводятся имена массивов аргументов для текущей функции, которые берутся из списка аргументов главной структуры и содержатся в поле ArgArg (см. рис.2). Справа от них выводятся их обозначения (содержимое поля ArgNames) для подписей на графиках.

Еще ниже списка аргументов находится окно, куда выводится обозначение (содержимое поля Name) текущей функции, которое также используется при построении графиков.

Под ним в другом окне приводятся размеры массива текущей функции (размеры содержимого поля Arg). Еще ниже располагается окно комментариев текущей функции (содержимое поля FunComments).

В области функций расположены следующие кнопки (по порядку слева направо):

1.  — «Добавить новую функцию» добавляет в базу данных одномерные, двумерные, трехмерные функции с аргументами, выбранными из списка. Пользователь может создать новую функцию на базе тех, что уже присутствуют в структуре, либо из произвольной матрицы.
2.  — «Добавить новую функцию из Workspace». Массивы функций иногда удобнее формировать вне базы данных и добавлять их в структуру через MATLAB Workspace. При нажатии на эту кнопку открывается диалог со списком всех одномерных, двумерных и трехмерных массивов, содержащихся в Workspace (см. рис. 5). Пользователь может выбрать любой из них. Далее открывается диалог, аналогичный тому, что вызывается при нажатии на кнопку «Свойства функции» (см. рис. 7). В нем пользователь имеет возможность выбрать для новой функции аргументы из списка аргументов, задать обозначения аргументов (для графиков), имя массива функции и ее обозначение (для графиков). Если размерность массива новой функции не соответствует размерностям массивов ее аргументов, при закрытии диалога функция в базу данных занесена не будет, программа выдаст сообщение об ошибке (MessageBox).
3.  — «Свойства функции» вызывает диалог, позволяющий пользователю изменить содержимое полей текущей функции: имя массива функции ArgName, обозначение функции Name, список аргументов функции ArgArg (из списка аргументов главной структуры), обозначение

ния аргументов из поля ArgNames, которые используются при построении графиков. Если размерность массива функции не соответствует размерностям массивов ее аргументов, при закрытии диалога изменения записаны не будут, программа выдаст сообщение об ошибке (MessageBox).

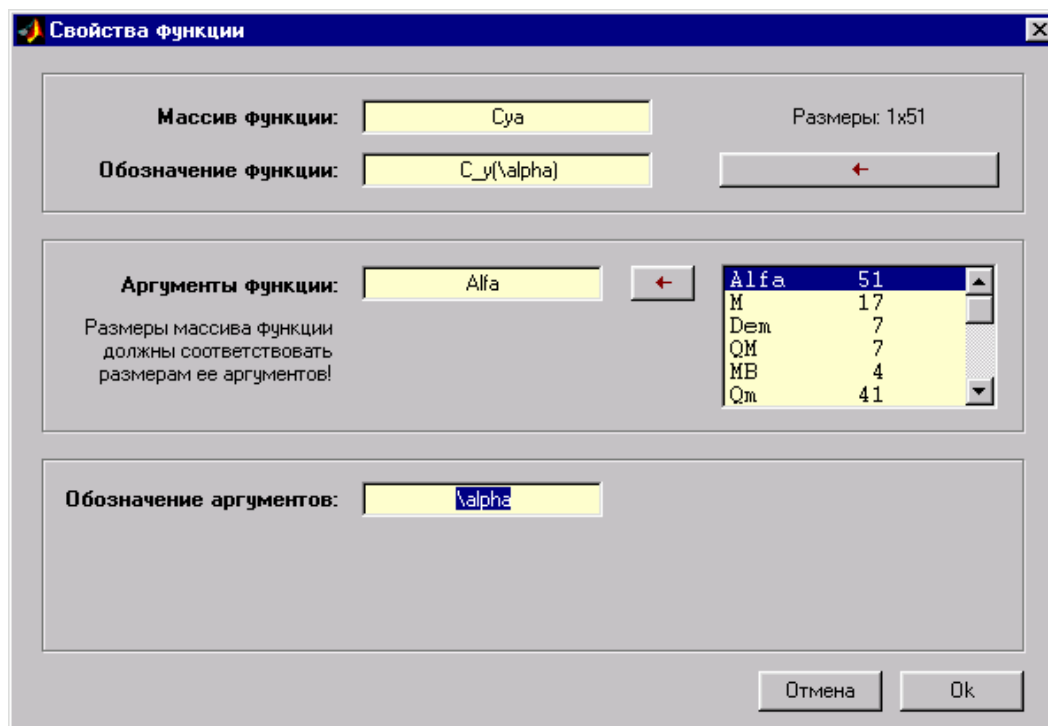





Рис. 7. Диалог, задающий свойства функции.

4.  — «Комментарии функции» вызывает диалог, позволяющий пользователю читать и редактировать содержимое поля комментариев для выбранной функции.
5.  — «Отправить в редактор» («Извлечь из редактора»). Залипающая кнопка (типа togglebutton) позволяет изменять численные значения функций. Массив выделенной в списке функции временно загружается в MATLAB Workspace и редактируется в нем при помощи встроенного редактора MATLAB Array Editor. Чтобы извлечь его оттуда и записать в структуре базы данных, необходимо еще раз нажать на эту же кнопку.
6.  — «Нарисовать график функции». В случае одномерных функций (1-D) на одном графике возможно изображение нескольких функций от одного аргумента. В случае 2-D функций из всех выделенных в списке изображается только первая функция $F(x,y)$ в проекциях XZ , YZ и в трехмерном виде XYZ . В случае 3-D функций одна функция $F(x,y,z)$ может изображаться по проекциям XZ или YZ или в виде XYZ для всех значений последнего аргумента z .

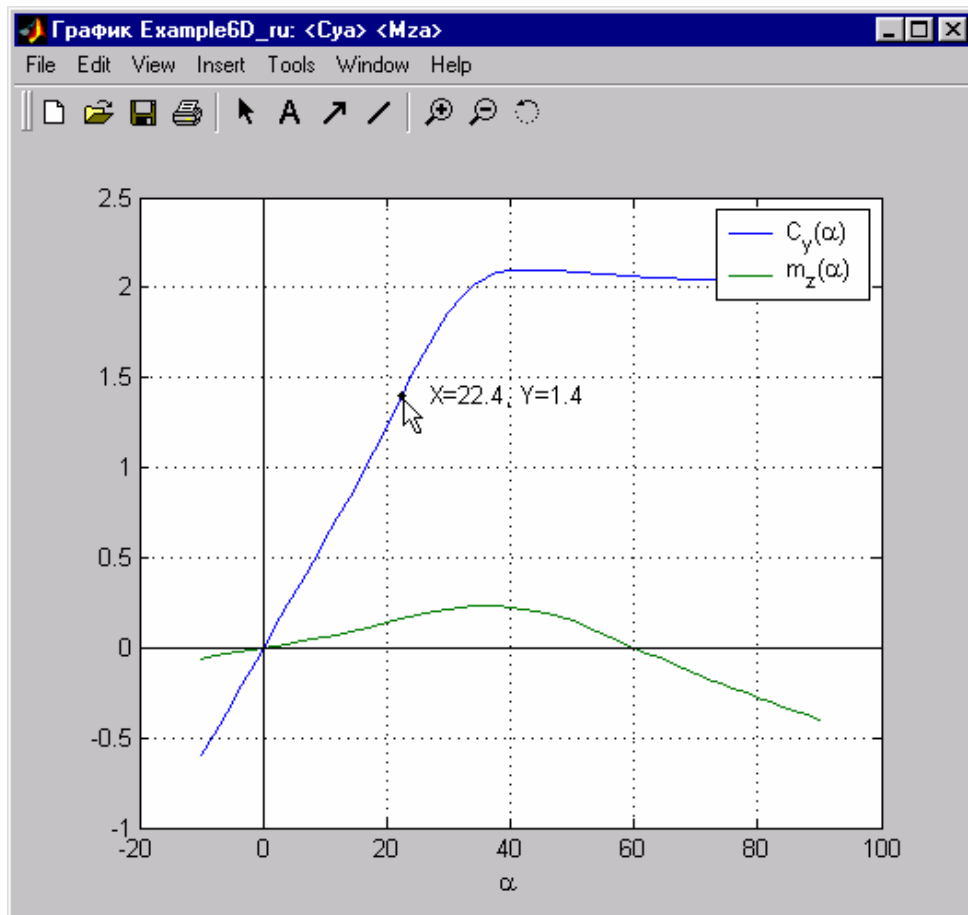




Рис. 8. Построение графика функции.

7.  — «Графическое редактирование функции». Часто бывает удобным редактировать функцию графически (методом «drag&drop»). Для этого в программу включены два графических редактора — для одномерных и двумерных функций, окна которых показаны на рис. 9 и 10. Вызвав их, можно редактировать несколько одномерных функций с одинаковым аргументом или одну двумерную функцию (первую в списке выделенных).

Рассмотрим окно редактора одномерных функций (рис. 9). В этом окне обрабатываемая функция представляется в виде узлов, соответствующих точкам массива функции, и отрезков, их соединяющих. При помощи мыши можно изменять положение ординаты узла на графике. Координаты двигаемой точки приводятся в окне в правом верхнем углу редактора. По желанию пользователя можно установить количество знаков после запятой в ординате при перемещении точки, то есть некий шаг перемещений. Редактор поддерживает числа с количеством знаков после запятой от 0 до 8. Эту величину можно выбрать при помощи первой кнопки на панели редактора  («Установить шаг перемещений узлов при помощи мыши»). Если значения функции изменились, то по закрытию окна редак-

тора программа предлагает пользователю сохранить изменения в численном значении функции в структуре базы данных.

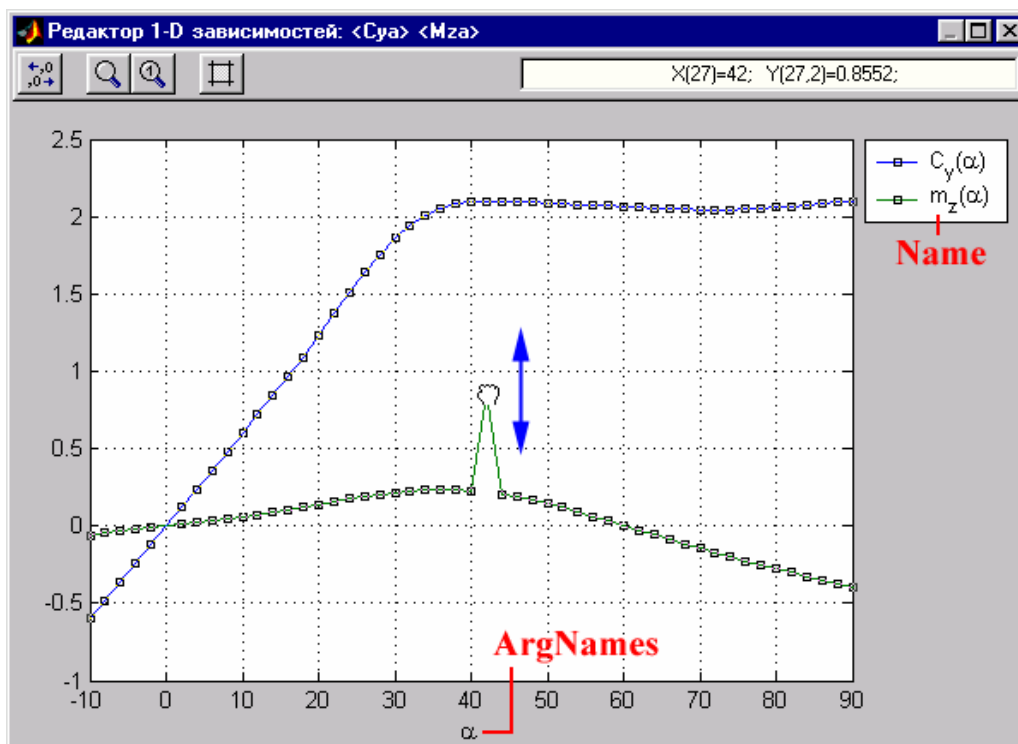


Рис. 9. Окно графического редактора одномерных функций.

Редактор двумерных функций $Z(X, Y)$ похож на редактор одномерных функций. В нем функция представима в 3 видах: в проекции XZ при фиксированных значениях Y , в проекции YZ при фиксированных значениях X , и в виде трехмерного изображения (рис.10). Линии проекций можно обрабатывать (как в одномерном редакторе), а трехмерное изображение приводится для наглядности. Как и в одномерном редакторе, здесь можно задать шаг изменения координаты Z .

Оба редактора обладают контекстным меню (рис. 10). В контекстном меню содержатся различные функции графической обработки изображения (установка цветовой гаммы, толщины линии, размера маркеров узлов и т. д.), так что при помощи этих редакторов можно получать графики различного вида по желанию пользователя.

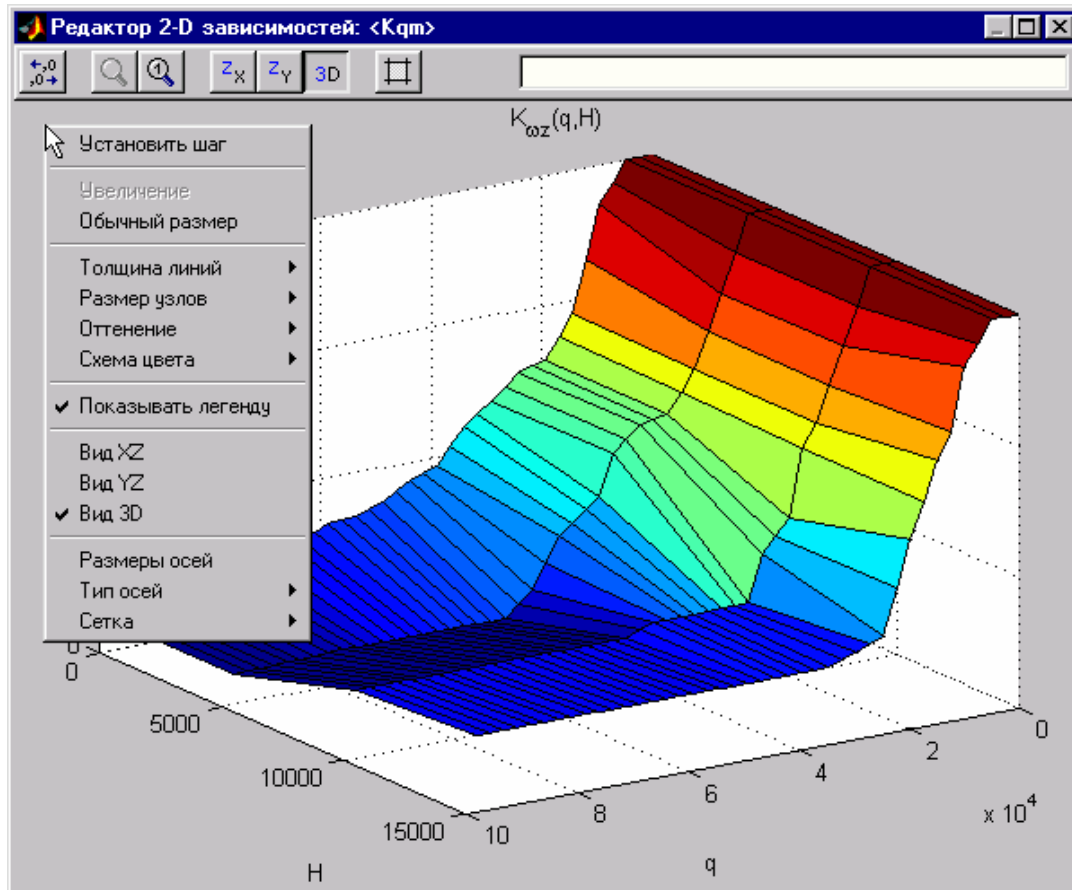



Рис. 10. Окно графического редактора двумерных функций.

8.  — «Создать таблицы (блоки) Simulink» служит для создания таблиц Simulink для выделенных в списке функций. Эти блоки в дальнейшем можно использовать для построения в Simulink модели динамики какого-либо устройства. При этом необходимые для построения таблиц массивы загружаются в рабочее поле Workspace.

В системе Simulink (для версии MATLAB 6.0 и выше) существует несколько типов встроенных элементов для описания табличных зависимостей:

- одномерная таблица (1-D Look-Up Table) с линейной интерполяцией и экстраполяцией;
- двумерная таблица (2-D Look-Up Table) с линейной интерполяцией и экстраполяцией;
- многомерная таблица (n-D Look-Up Table) с настройками, задаваемыми пользователем (ее можно использовать для построения 1-D, 2-D, 3-D зависимостей).

Кроме того, в пакет программы поддержки баз данных включена S-функция **Table3d.dll** с линейной интерполяцией и экстраполяцией, написанная специально для трехмерных зависимостей.

Пользователь может выбрать, таблицы какого типа создавать. При нажатии кнопки «Создать таблицы (блоки) Simulink» появляется диалог

(см. рис. 11), в котором осуществляется выбор типов, и задаются необходимые настройки:

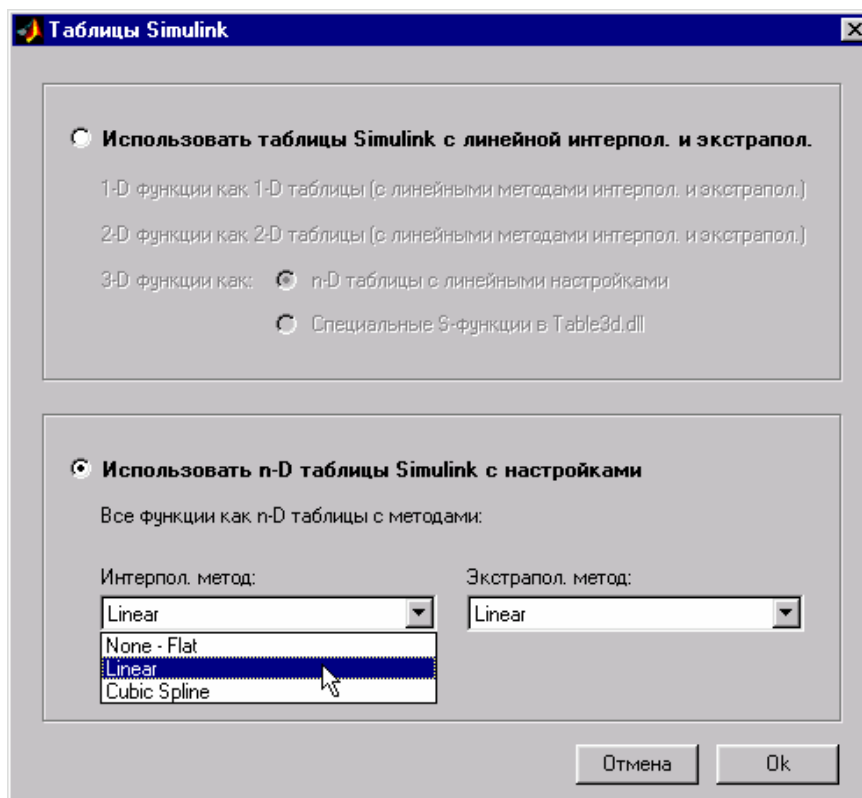


Рис. 11. Диалог для создания таблиц Simulink.

При нажатии кнопки «Ok» диалога создается новая модель, содержащая блоки таблиц с заданными именами массивов из базы данных для выделенных в списке функций. Пример новой модели приведен на рис. 12:

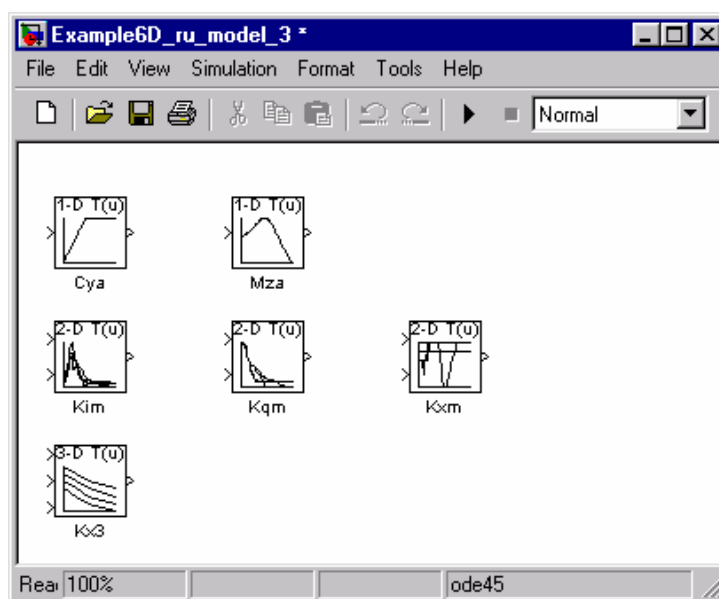





Рис. 12. Таблицы интерполяции в Simulink.

9.  — «Загрузить выделенные функции в Workspace». При нажатии на эту кнопку выбранные в списке функции загружаются в рабочее поле MATLAB Workspace под именами своих массивов, где они становятся доступным для других приложений среды MATLAB. Как и в случае с загрузкой параметров в Workspace, пользователь вместо обычного содержимого массива может отправить под его именем нулевой либо единичный массив, что определяется при помощи диалога, аналогичного тому, что показан на рис.6.
10.  — «Удалить выделенные функции» удаляет выбранные в списке функции из структуры по предварительному подтверждению.
11.  — Кнопки «Переместить вверх» и «Переместить вниз» перемещают функцию или несколько функций по списку.

3. Взаимодействие программы с моделями Simulink

Взаимодействие программы поддержки баз данных **Database Manager** с моделями Simulink организовано очень просто.

Пусть задана произвольная модель Simulink, в которой все необходимые численные данные в блоках типа Look-Up Table, Gain и других заданы параметрически, то есть в виде идентификаторов переменных и массивов, которые содержатся в базе данных. Тогда перед началом моделирования все массивы параметров, аргументов и функций, содержащиеся в базе, необходимо загрузить в рабочее поле MATLAB Workspace. В главном окне программы поддержки базы данных это действие можно выполнить нажатием одной кнопки (см. раздел 2). Программа обеспечивает также частичную загрузку некоторых массивов в рабочее поле и занесение массивов из рабочего поля в базу данных, как это описано в предыдущем разделе.

Общая схема взаимодействия баз данных и моделей приведена на рис. 13.

Такой способ взаимодействия является очень удобным, так как позволяет использовать для одной модели Simulink несколько баз данных, в которых значения массивов могут быть различными, и в зависимости от обстоятельств для данной модели загружать в рабочее поле ту или иную базу.

И наоборот, этот способ позволяет создавать одну общую базу данных для нескольких различных моделей или модификаций одной модели.

Программу поддержки баз данных можно также использовать для ускорения сборки моделей Simulink, так она позволяет автоматически создавать блоки таблиц типа Look-Up Table с заданными именами массивов из базы данных, как это описано в разделе 2.

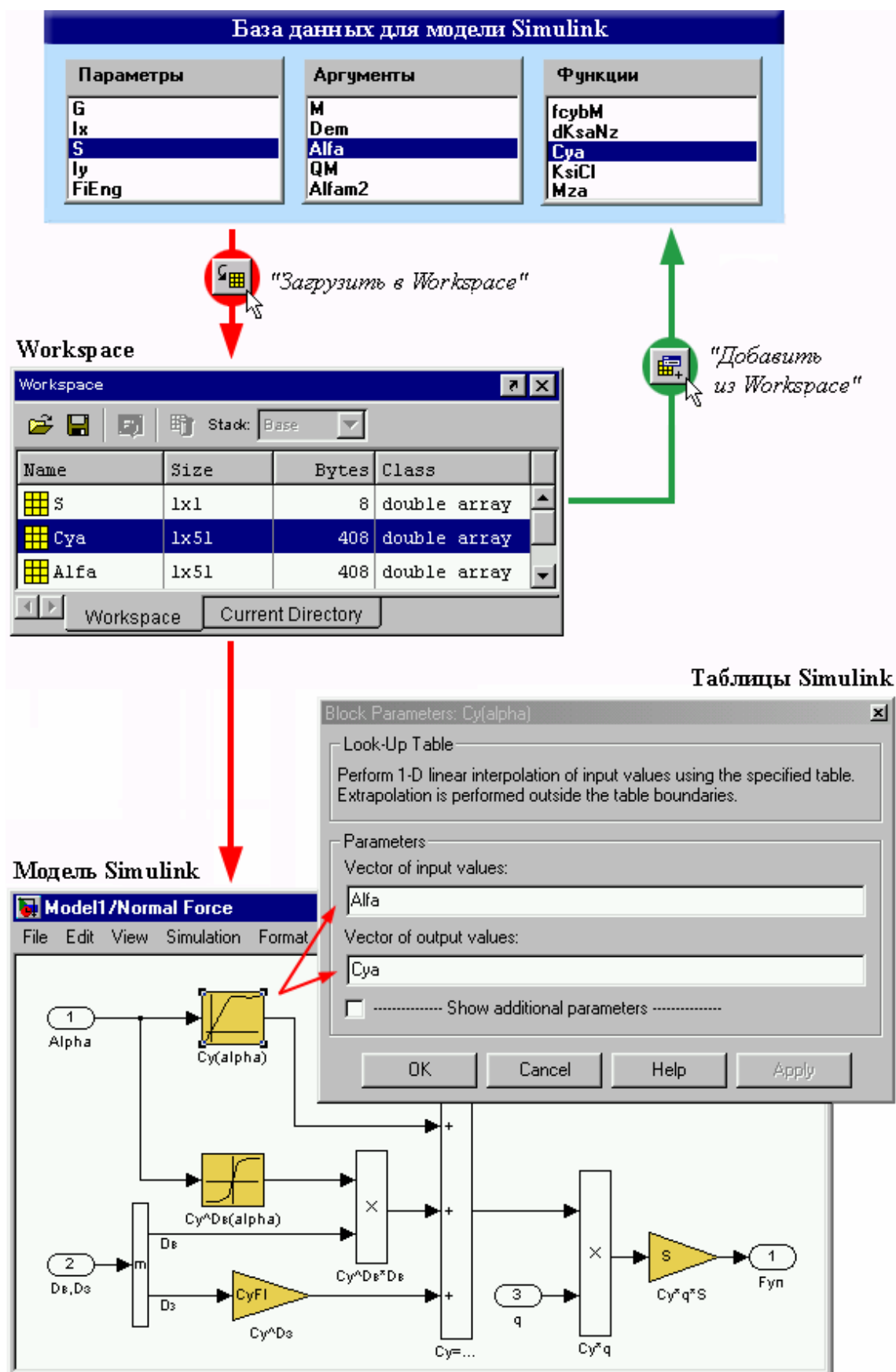


Рис. 13. Схема взаимодействия баз данных и моделей Simulink.

4. К вопросу использования русского языка в среде MATLAB/Simulink

Первоначальный вариант программы базы данных разрабатывался для версии MATLAB 6.0/6.1 (R12/R12.1). Как известно, эта версия не поддерживает русскую строчную букву «я», поэтому в программе **Database Manager** эта буква в интерфейсных элементах нигде не используется, хотя ее можно вводить в комментарии к базе данных и в наименования функций и аргументов для графиков.

Использование строчной буквы «я» при программировании на языке MATLAB в версии 6.1 вызывает ошибку в командном окне. Использование же этой буквы в модели Simulink (в подписях блоков и в масках подсистем) может вызвать потерю текстовой информации после нее, или, в худшем случае, сделать нормальное открытие модели невозможным. В последнем случае необходимо открыть текст модели любым текстовым редактором и в нем исключить букву «я» вручную или автоматически. После этого модель должна открыться нормально.

В версии MATLAB 6.5 (R13) ошибки, связанные с буквой «я», при программировании больше не возникают, но в Simulink 5.0 проблема буквы «я» сохранилась! Таким образом, если нет возможности избежать использования этой буквы в моделях Simulink, то нужно вводить ее в качестве прописной.

Как показывает практика, проблемы с русскими буквами могут возникнуть и в других приложениях среды MATLAB. Так, например, следует опасаться использования русской строчной буквы «э» при работе с Simulink Performance Tools версии 1.2 (R13).

Однако часто бывает так, что на компьютерах под управлением Windows 98 или Windows 2000 MATLAB/Simulink и некоторые другие программы вообще неправильно отображают все символы кириллицы.

Решение этой проблемы может быть следующим: в разделе реестра Windows
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage
нужно присвоить параметру «1252» значение «с_1251.nls» и перезагрузить компьютер.

Заключение

Программа **Database Manager** — мощное средство для работы с базами данных рассмотренного выше типа. Достаточно простая в применении, она обладает обширными возможностями и позволяет значительно упростить и ускорить создание и анализ моделей динамики технических устройств различного типа в среде MATLAB/Simulink.

УДК 677.054.755

МОДЕЛИРОВАНИЕ СИСТЕМЫ КОНТРОЛЯ ПЛОТНОСТИ ТКАНИ ПО УТКУ

Блинов О. В.

Ивановская государственная текстильная академия, Иваново,
e-mail: enka@igta.interline

Разработка устройства для контроля плотности ткани по утку является одной из существующих задач ткацкого производства текстильной промышленности и ведется с целью улучшения качества выпускаемого продукта. Есть несколько способов решения данной задачи. Ниже приведен один из возможных вариантов.

Теоретические сведения

Бердо посредством батанного механизма перемещается в крайнее правое положение, что соответствует прибитой уточной нити к опушке ткани. При перемещении берда в заднее положение, батанный механизм действует на товарный регулятор, вследствие чего он приводится в движение и отводит наработанную ткань, с помощью вальяна. В зависимости от скорости отвода ткани (определенной длины, за постоянный промежуток времени) меняется плотность ткани. Плотность ткани по утку определяется числом уточных нитей на определенной длине (чаще всего 10см.). Таким образом, контролируя отведенную длину наработанной ткани, при этом, считая количество приближений берда (прибитых уточных нитей), можно определить плотность ткани по утку. Структурная схема устройства, осуществляющего контроль плотности ткани по утку приведена на рис. 1.

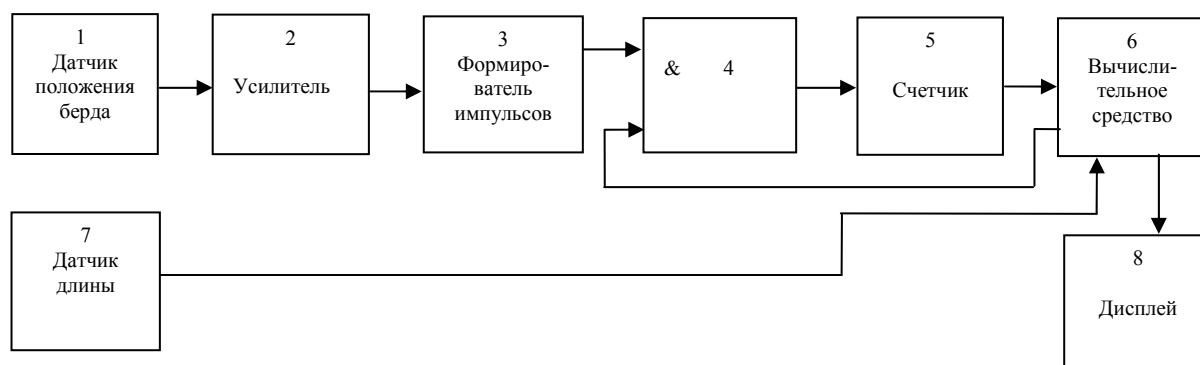


Рис. 1. Структурная схема устройства контроля плотности ткани по утку.

Имитационная модель системы контроля плотности ткани по утку

На основании структурной схемы, приведенной на рис. 1, а также алгоритма, отражающего взаимосвязанное поведение рабочих органов ткацкого станка, таких как бердо и товарный регулятор, составлена обобщенная модель системы контроля плотности ткани по утку. Модель представлена на рис. 2.

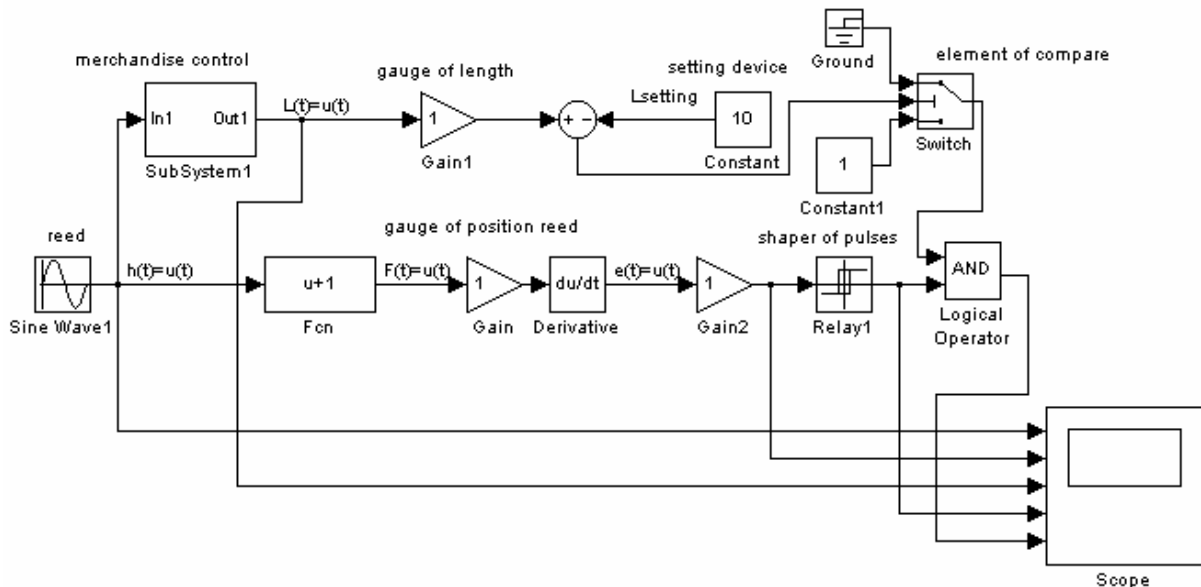


Рис. 2. Блок-схема, моделирующая контроль плотности ткани по утку.

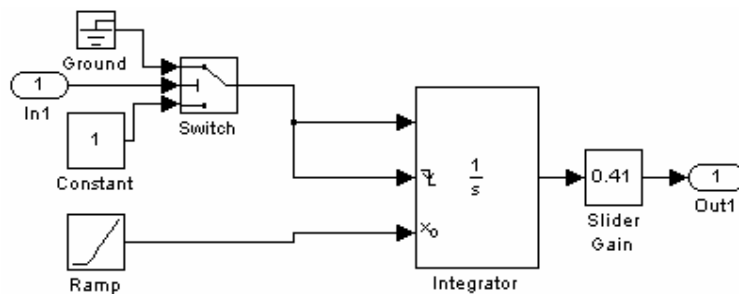


Рис. 3. Подсистема, моделирующая работу товарного регулятора.

Данные блок-схемы выполнены в системе моделирования Simulink, среды MATLAB.

Описание блоков модели

Блок Reed (Sine Wave1) — используется для моделирования движения берда (батана). Где максимум функции соответствует крайнему переднему положению берда (прибой уточины), минимум соответственно крайнему заднему положению, а полное колебание равно обороту главного вала.

Подсистема Merchandise control (Subsystem1) — в общем случае моделирует работу товарного регулятора. Выходным сигналом, которой является длина отводящейся ткани, а входным положение берда. Блок схема Subsystem1 представлена на Рис. 3. Где с помощью блока Slider Gain можно менять скорость отвода ткани, тем самым, изменяя плотность ткани по утку. Надо отметить, что такая модель отражает работу товарного регулятора периодического действия.

Блок gauge of length (Gain1) — моделирует датчик длины (преобразователь длины в электрический сигнал).

Блок Sitting device (Constant) — является задатчиком контролируемой длины.

Следующие три блока: Ground, Switch, Constant1 — служат для подачи сигнала на логический элемент «И», в соответствии с сигналом, поступающим из сумматора. Таким образом, три выше перечисленных блока в совокупности с сумматором и задатчиком моделируют работу вычислительного устройства.

Блоки Fcn, Gain и Derivative — моделируют работу датчика положения берда. Где блок Fcn отражает изменение входного параметра датчика в зависимости от изменения положения берда, а блоки Gain и Derivative являются передаточной функцией датчика.

Блок Gain2 — моделирует работу усилителя.

Блок Shaper of pulses (Relay1) — моделирует работу формирователя импульсов.

Результаты моделирования

Результаты моделирования системы контроля плотности ткани по утку представлены в виде графиков, полученных с помощью блока Scope, на рис. 4. Для наглядности полученных результатов значение блока Sitting device было принято равным единице.

На рис. 4. изображены (начиная сверху):

первое окно — зависимость положения берда от времени (частота колебания — рад/сек);

второе окно — сигнал с датчика приближения берда;

третье окно — зависимость отводимой длины товарным регулятором от времени — см/сек (так как коэффициент передачи датчика длины равен единице, то сигнал с датчика длины будет соответствующим);

четвертое окно — сигнал с формирователя импульсов;

пятое окно — выходной сигнал системы.

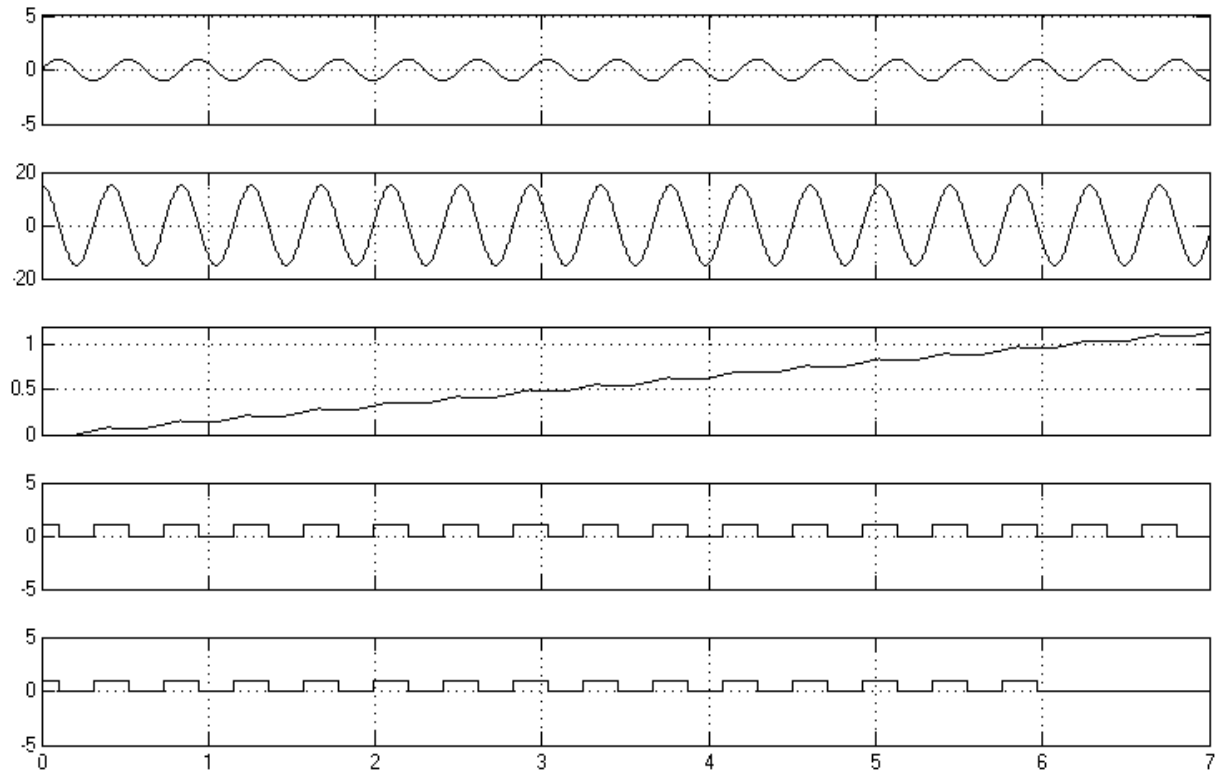


Рис. 4. Результаты моделирования системы контроля плотности ткани по утку.

Таким образом, исходя из графиков, плотность ткани равна 15 уточных нитей (15 импульсов) на единицу (1 см.) длины ткани.

Литература

1. *Расторгуев А. К., Власов Е. И.* Теория и расчет первичных измерительных преобразователей для текстильной промышленности.— Иваново: ИГТА, 1991.
2. *Николаев С. Д., Власов П. В.* Теория процессов, технология и оборудование ткацкого производства.— М.: Легпромбытиздат, 1995.
3. *Черных И. В.* Simulink: среда создания инженерных приложений.— М.: ДИАЛОГ-МИФИ, 2003.

УДК 621.317.785:621.314.333:621.314.52

ПРИМЕНЕНИЕ MATLAB/SIMULINK ДЛЯ ИЗМЕРЕНИЯ И ОЦЕНКИ КАЧЕСТВА ЭЛЕКТРОЭНЕРГИИ В ТРЕХФАЗНЫХ СИММЕТРИЧНЫХ СИСТЕМАХ С АКТИВНЫМИ ПРЕОБРАЗОВАТЕЛЯМИ

Борисов П. А.

*Санкт-Петербургский государственный университет информационных технологий,
механики и оптики, Санкт-Петербург,
e-mail: borisov@ets.ifmo.ru*

В настоящее время в связи с повсеместным распространением преобразовательной техники, ростом единичных мощностей преобразователей и повышением в целом доли нелинейной нагрузки проблемы обеспечения качества электроэнергии и энергосбережения стали приобретать значение приоритетных, требующих первоочередного решения. Требования международных стандартов, таких как IEC 61000-3, IEEE 519, к качеству энергопотребления с годами постоянно ужесточались. Темпы внедрения подобных стандартов в России до недавнего времени существенно отставали от международных, однако в последние годы ситуация в значительной степени изменилась. Пересмотрены не только требования, предъявляемые к показателям качества электроэнергии ГОСТ 13109-97 [1], но и введены новые показатели качества ГОСТ Р 51317.3.2.-99 (МЭК 61000-3-2-95). О государственной значимости этого вопроса свидетельствует принятие Закона «Об электромагнитной совместимости». В целом можно сказать, что проблема преимущественно сферы преобразовательной техники превращается в проблему электроэнергетики.

Решение указанных проблем средствами самой преобразовательной техники возможно, так как в настоящее время за счет создания и освоения промышленностью высокоэффективных силовых приборов и вычислительных устройств на базе программируемых микроконтроллеров существенно расширились функции силовых устройств. Реализация новых алгоритмов управления потоком электроэнергии от источника к приемнику и наоборот позволяет активно влиять на показатели качества электроэнергии.

Электротехнические комплексы и системы на базе полупроводниковых преобразователей электрической энергии разделяются на две подсистемы: энергетическую и информационную [2]. Энергоподсистема включает в себя силовые цепи источника питания, полупроводникового преобразователя и нагрузки и осуществляет двусторонний обмен энергией между первичным источником питания и нагрузкой определенными дозами, по-

средством модуляции того или иного вида. Информационная подсистема включает в себя систему управления полупроводниковым преобразователем с информационно-измерительной частью и осуществляет управление потоком электрической энергии между первичным источником питания и, как правило, нелинейной нагрузкой.

Решение задач обеспечения необходимых энергетических показателей и минимизации массогабаритных показателей сводится, в первую очередь, к определению составляющих полной мощности энергоподсистемы. При проектировании как питающих сетей, так и электротехнических комплексов и систем с полупроводниковыми преобразователями необходимо не только правильно учитывать и разделять все составляющие полной или кажущейся мощности, но и закладывать мероприятия по уменьшению мощностей обменного характера. Следует отметить, что эти мероприятия требуют трудоемкого расчета во временной и частотной областях. В этой связи в последнее время появилась практика проведения исследований в специализированных интегрированных пакетах. В данной работе задача решается с использованием пакета MATLAB/Simulink (версия 6.5) и его приложения SimPowerSystems.

Известно [3], что мгновенная мощность любой энергоподсистемы определяется произведением мгновенных значений напряжения и тока на ее входе, равна скорости поступления электромагнитной энергии в данный момент времени и в общем случае изменяется в течение периода переменного тока по амплитуде и знаку.

Активная мощность P равна среднему значению мгновенной мощности за период питающего напряжения и определяет количество электромагнитной энергии, необратимо преобразующейся в другие формы энергии. Активная мощность характеризует полезную работу в нагрузке, включая полезную мощность и мощность потерь в установке. Коэффициент полезного действия (к. п. д.) любой энергоподсистемы, характеризующий потери активной составляющей полной или кажущейся мощности S , определяется как отношение активной мощности отдаваемой в нагрузку P_{Load} к потребляемой активной мощности от сети P :

$$КПД = \frac{P_{Load}}{P}.$$

Полная, или кажущаяся, мощность S всегда больше фактически передаваемой нагрузке активной мощности из-за существования неактивных составляющих мощности, которые, не создавая полезного эффекта, приводят в то же время к увеличению потерь в питающей сети [3]–[7]. Известны три неактивные составляющие полной мощности [3]: реактивная мощность Q , или мощность сдвига, мощность искажения T и мощность несимметрии H .

Реактивную мощность или мощность сдвига Q , связывают со сдвигом по фазе основной гармоники тока относительно напряжения питающей сети [4]. Вследствие сдвига основной гармоники тока появляется реактивная составляющая тока, которая не участвует в передаче активной мощности нагрузке, так как среднее значение мгновенной мощности за период, обусловленное этой составляющей тока, равно нулю.

Ряд авторов [5], [7], в том числе и большинство специалистов по преобразовательной технике, считают мощность искажения T составляющей реактивной мощности, обусловленной высшими гармониками тока. Среднее значение мгновенной мощности, связанной с этими гармониками, за период также равно нулю, однако и они вызывают дополнительные потери энергии в сети.

Мощность несимметрии H учитывает дополнительные потери энергии, связанные с неравномерным распределением тока по фазам многофазной цепи. В однофазных и многофазных симметричных системах мощность несимметрии равна нулю [3].

В общем случае полная мощность связана со своими составляющими известным выражением [3] вида:

$$S = \sqrt{P^2 + Q^2 + T^2 + H^2}.$$

В симметричной системе полная мощность связана со своими составляющими соотношением [3], [5]–[7]:

$$S = \sqrt{P^2 + Q^2 + T^2}. \quad (1)$$

Определив полную мощность энергоподсистемы S и ее составляющие P , Q и T можно определить основные показатели качества энергопотребления [3]: $K_{\text{мощности}}$ — коэффициент мощности, $K_{\text{сдвига}}$ — коэффициент сдвига, $K_{\text{искажения}}$ — коэффициент искажения, $K_{\text{гармоник}}$ — коэффициент гармоник или K_{THD} , $K_{\text{несимметрии}}$ — коэффициент несимметрии.

Коэффициент мощности электротехнического комплекса характеризует способность этого комплекса потреблять электрическую энергию первичного источника питания:

$$K_{\text{мощности}} = \frac{P}{S}.$$

Коэффициент сдвига характеризует обмен энергией между приемником и источником, обусловленный способностью реактивных элементов электротехнического комплекса накапливать и отдавать энергию:

$$K_{\text{сдвига}} = \frac{P}{\sqrt{P^2 + Q^2}}.$$

Коэффициент искажения характеризует обмен энергией между источником и приемником, обусловленный высшими гармоническими составляющими тока:

$$K_{\text{искажения}} = \frac{\sqrt{P^2 + Q^2}}{\sqrt{P^2 + Q^2 + T^2}}.$$

Коэффициент гармоник характеризует соотношение между энергией, обусловленной высшими гармониками тока, и энергией обусловленной основной (первой) гармоникой тока:

$$K_{\text{гармоник}} = \frac{T}{\sqrt{P^2 + Q^2}} = \sqrt{\left(\frac{1}{K_{\text{искажения}}}\right)^2 - 1}.$$

На сегодняшний день международные стандарты качества энергопотребления, такие как IEC 61000-3, IEEE 519, EN 61000-3-2, предъявляют жесткие требования к значениям гармонических составляющих тока вплоть до 49 гармоники. В пакете MATLAB/Simulink для определения интегрального показателя гармонического состава тока следует использовать стандартный блок THD (Total Harmonic Distortion) из приложения SimPowerSystems, вычисляющий этот показатель по выражению:

$$K_{\text{THD}} = K_{\text{гармоник}} = \frac{\sqrt{\sum_{k=2}^N (I_{kRMS})^2}}{I_{1RMS}}.$$

Коэффициент несимметрии характеризует степень несимметрии системы и определяется из равенства:

$$K_{\text{несимметрии}} = \frac{\sqrt{(P^2 + Q^2 + T^2)}}{S}.$$

В симметричной системе коэффициент несимметрии равен единице.

Случаю трехфазной сети с симметричной синусоидальной системой напряжений соответствует предпосылка о наличии питающей сети переменного тока бесконечно большой мощности и позволяет получить необходимую для практики точность расчетов в случаях, когда мощность вентильного преобразователя невелика в сравнении с мощностью сети [3]. Известно [8], что в трехфазной сети с симметричной системой напряжений и с симметричной (равномерной) нагрузкой фаз полная мощность и ее составляющие определяются выражениями:

$$\begin{aligned} S_3 &= 3U_\phi I_\phi & (2), \\ P_3 &= 3P_\phi, \\ Q_3 &= 3Q_\phi, \end{aligned}$$

где U_ϕ — действующее значение напряжения фазы, I_ϕ — действующее значение тока фазы. При равномерной нагрузке независимо от способа ее соединения (звездой или треугольником) [9]:

$$3U_\phi I_\phi = \sqrt{3}U_L I_L,$$

где $U_{Л}$ — действующее значение линейного напряжения, $I_{Л}$ — действующее значение линейного тока. Поэтому вместо формулы (2) часто используют [9] следующую:

$$S_3 = \sqrt{3}U_{Л}I_{Л}. \quad (3)$$

Мощность искажения может быть определена из формулы (1):

$$T_3 = \sqrt{S_3^2 - (P_3^2 + Q_3^2)}. \quad (4)$$

Для измерения активной мощности в трехфазной системе без нейтрального провода как при симметричном, так и при несимметричном режимах эффективнее всего использовать способ двух ваттметров [9], аналогично для измерения реактивной мощности следует использовать два варметра. Сумма показаний двух ваттметров (варметров) при этом определяет активную (реактивную) мощность всей системы независимо от того, звездой или треугольником соединена нагрузка. В соответствии с указанным способом в пакете MATLAB/Simulink была разработана измерительная схема Three-Phase Active & Reactive Power (рис. 1), выполненная в виде подсистемы (subsystem). Для измерения активной и реактивной мощности в схеме используется стандартный блок Active & Reactive Power, входящий в раздел SimPowerSystems \ ExtraLibrary \ Measurements.

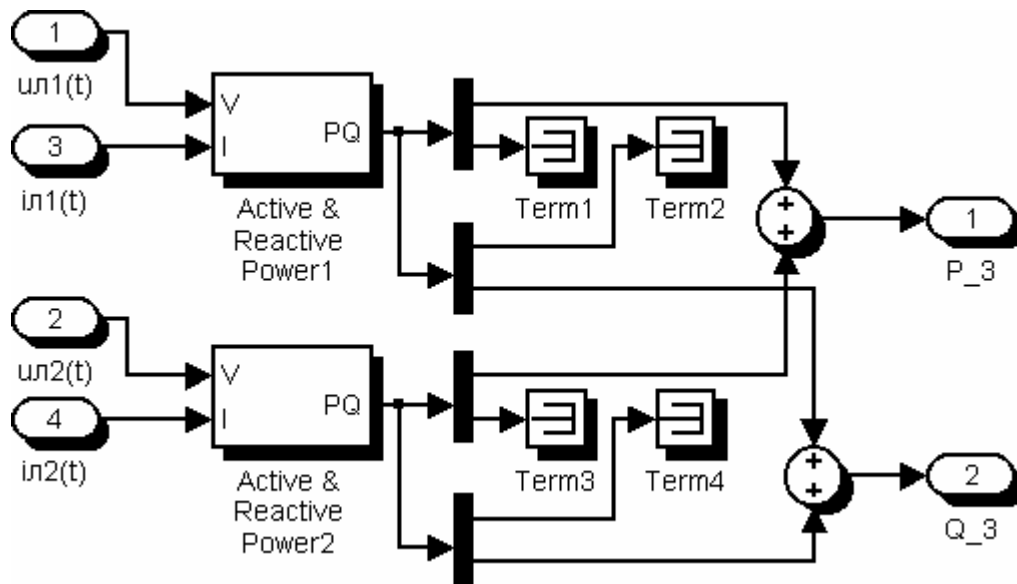


Рис. 1. Измерительная схема Three-Phase Active & Reactive Power.

Для измерения полной мощности трехфазной симметричной системы и мощности искажения в соответствии с формулами (3) и (4) была разработана измерительная схема Symmetric Three-Phase Power Measurement (рис. 2).

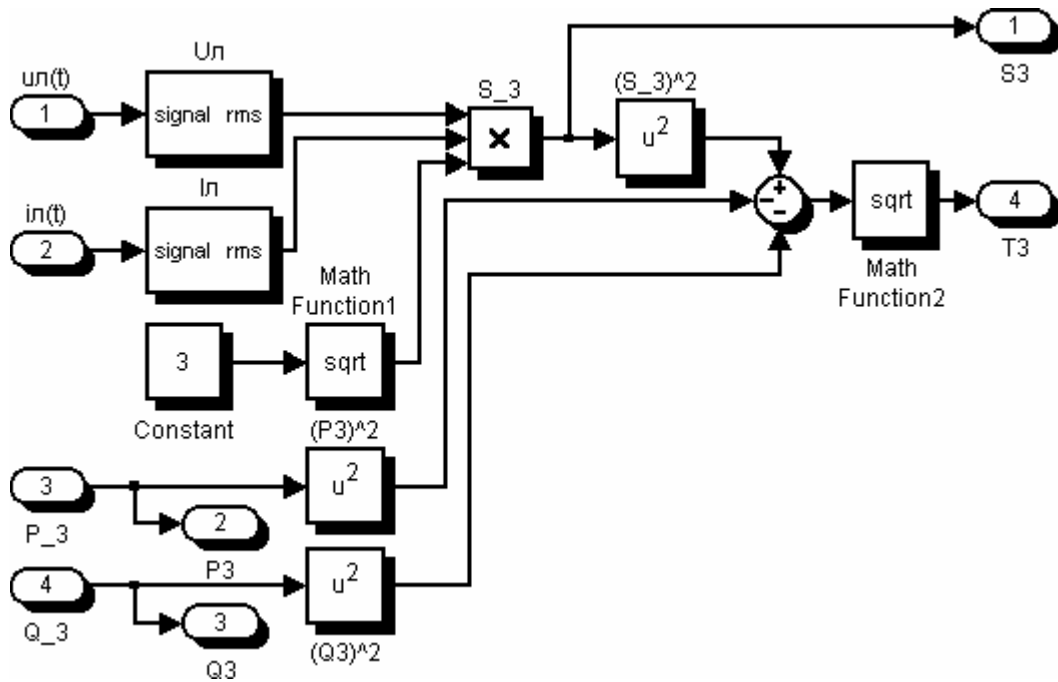


Рис. 2. Измерительная схема Symmetric Three-Phase Power Measurement.

Измерительные схемы (рис. 1) и (рис. 2) могут применяться независимо от способа соединения нагрузки, так на рис. 3 приведен пример трехфазной симметричной системы (схема соединений звезда-звезда). При моделировании использовались параметры силового трансформатора серии ТСЗ, ТСЗН номинальной мощностью 25 кВА. Остальные числовые параметры при моделировании были взяты следующими: действующее значение линейного напряжения источника $U_{л} = 6.3$ кВ, частота питающей сети 50 Гц, действующее значение линейного напряжения на вторичной стороне трансформатора $U_{2л} = 0.38$ кВ, сопротивление нагрузки в каждой фазе 8 Ом.

Среди стандартных блоков приложения SimPowerSystems раздела Extra Library имеются блоки Discrete 3-phase Total Power и Discrete 3-phase Positive-Sequence Active & Reactive Power, измеряющие составляющие полной мощности в трехфазной системе. Из полученных при моделировании результатов (Display и Display2 на рис. 3) видно, что величины составляющих полной мощности, измеренные разработанной схемой (рис. 1) и стандартными блоками, полностью совпадают в квазиустановившемся режиме. Это позволяет сделать вывод о том, что предложенная измерительная схема (рис. 1) может применяться наравне с используемыми в пакете.

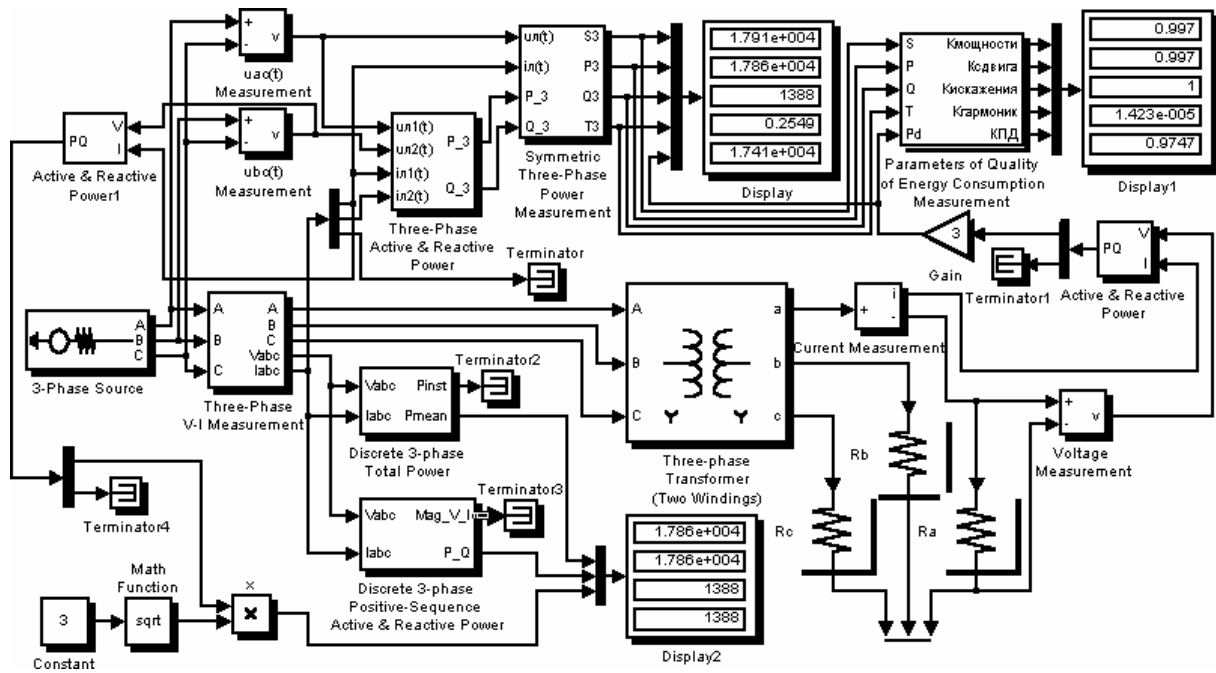


Рис. 3. Трехфазная симметричная система (схема соединений звезда-звезда).

Известно [10], что при симметричном режиме реактивную мощность можно измерить одним ваттметром:

$$Q_3 = \sqrt{3}P_w. \quad (5)$$

Величина реактивной мощности, измеренная с использованием стандартных блоков по формуле (5), также совпадает в квазиустановившемся режиме с величинами, полученными другими способами (Display2 на рис. 3).

Верификация используемого подхода производилась на моделях путем сравнения результатов моделирования с данными задач приведенных в литературе [9] — [10] и показала достоверность получаемых результатов. Так на рис. 4 приведен пример из [10], согласно данным задачи: $U_{\text{л}} = 0.38$ кВ, $\omega L = 2$ Ом, $r = 1/\omega C = 6$ Ом, активная мощность системы составляет 72.6 кВт, реактивная мощность системы равна нулю. Действительно, преобразовав в эквивалентную звезду группу элементов нагрузки соединенных в треугольник, получим, что суммарное реактивное сопротивление каждой из фаз равно нулю. Относительная погрешность измерения активной мощности системы на модели (рис. 4) не превышает 1%.

Показатели качества энергопотребления трехфазных систем предлагается определять измерительной схемой Parameters of Quality of Energy Consumption Measurement (рис. 5).

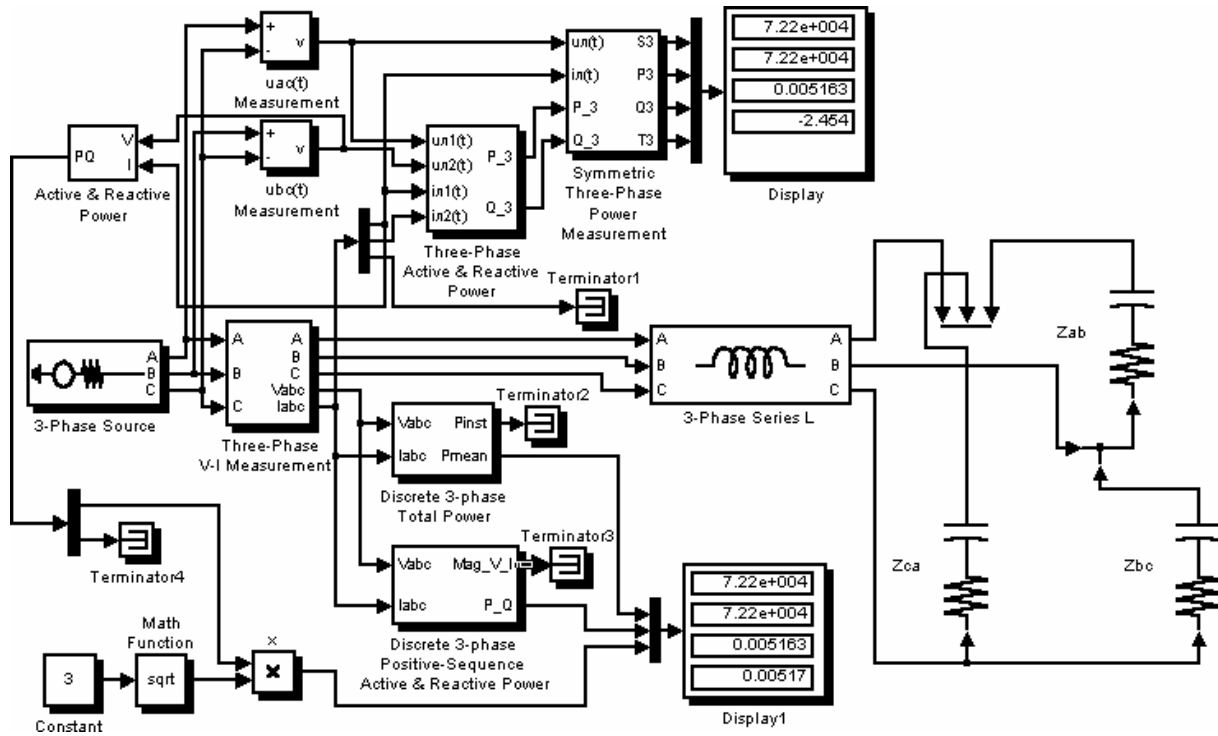


Рис. 4. Трехфазная симметричная система с активно-реактивной нагрузкой.

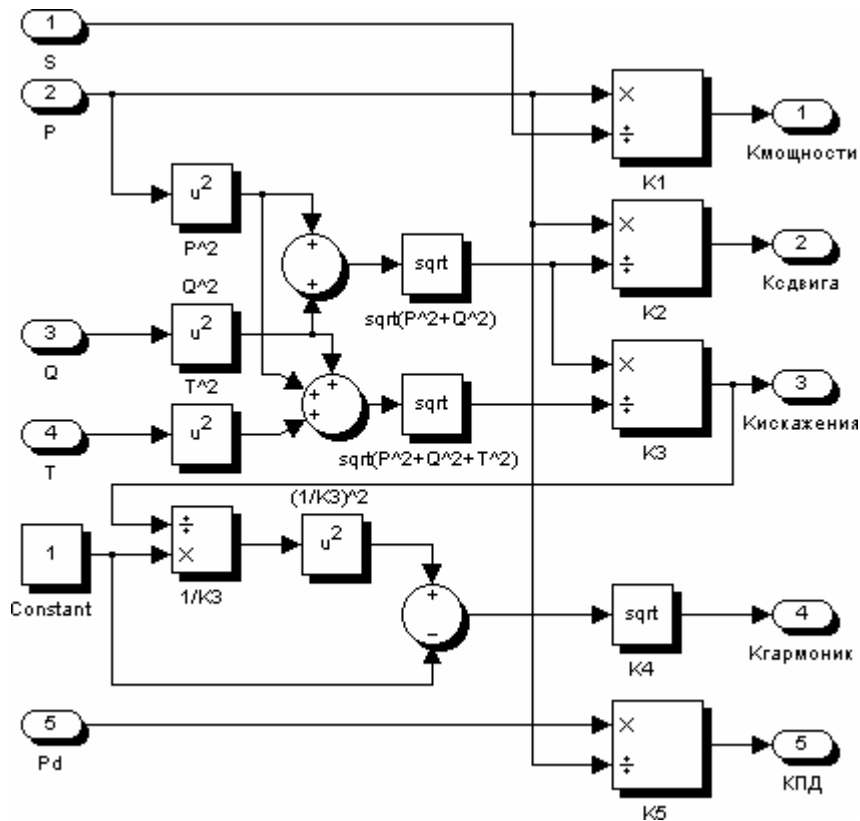


Рис. 5. Измерительная схема Parameters of Quality of Energy Consumption Measurement.

В наши дни большое внимание уделяется силовым схемам, которые позволяют наиболее перспективно решать проблемы повышения энергети-

ческой эффективности и экономичности использования электрической энергии в полупроводниковых преобразователях. Энергоподсистема с нагрузкой без противо-э. д. с., если нет необходимости в автономном источнике питания, реализуется по классической структуре: сеть переменного тока — неуправляемый выпрямитель — силовой фильтр — нагрузка. Наиболее эффективным из числа описанных в литературе [11] способов улучшения кривой потребляемого из сети переменного тока является способ, основанный на введении в схему неуправляемого выпрямителя дополнительного силового блока. В простейшем случае дополнительный блок выполняется по схеме импульсного повышающего регулятора [12], в состав которого входят токоограничивающий дроссель, ключевой прибор VT и отсекающий диод VD (рис. 6). Входной ток может быть получен близким к синусоидальному при неизменной длительности включенного состояния ключевого прибора, что существенно упрощает систему управления. Схемы, построенные по такому принципу, находят широкое применение благодаря простоте их реализации в электротехнических комплексах и системах, не требующих работы нагрузки в генераторном режиме, так как схемы обладают односторонней проводимостью.

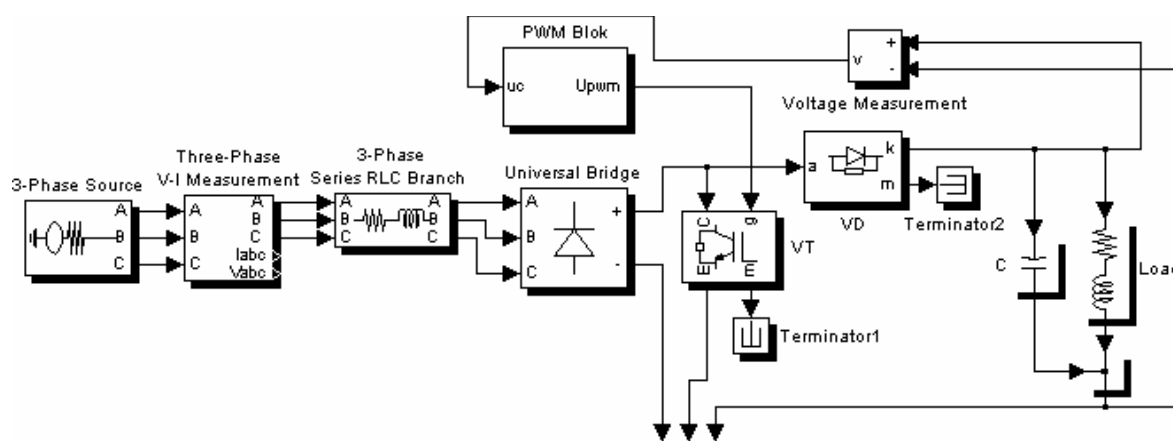


Рис. 6. Модель трехфазного мостового выпрямителя с емкостным фильтром и с дополнительным силовым блоком, выполненным по схеме импульсного повышающего регулятора.

Для систем полупроводниковых преобразователей и регулируемых электроприводов как постоянного, так и переменного тока с обратимым характером энергопотребления, в силовых схемах повышение энергетической эффективности достигается за счет применения полностью управляемых полупроводниковых приборов и импульсно-модуляционных алгоритмов управления, реализуемых на современных микроконтроллерных средствах вычислительной техники. Использование этого принципа позволяет при минимизации на достижимом уровне потерь в силовых полупроводниковых преобразователях активно влиять на их энергетическую и электромагнитную совместимость с питающей сетью. Схемы, построенные по та-

кому принципу, носят в отечественной литературе название активных преобразователей (АП) [13]. В англоязычной литературе используется термин преобразователь переменного/постоянного тока соответствующий терминологии стандарта МЭК 60050-551 (AC/DC Converter) [14], [15]. Управляемые силовые модули, на базе которых строится преобразователь, могут иметь [16] разное исполнение (рис. 7), обеспечивая полную управляемость преобразователем в режиме потребления нагрузкой энергии и (или) рекуперации энергии в сеть.

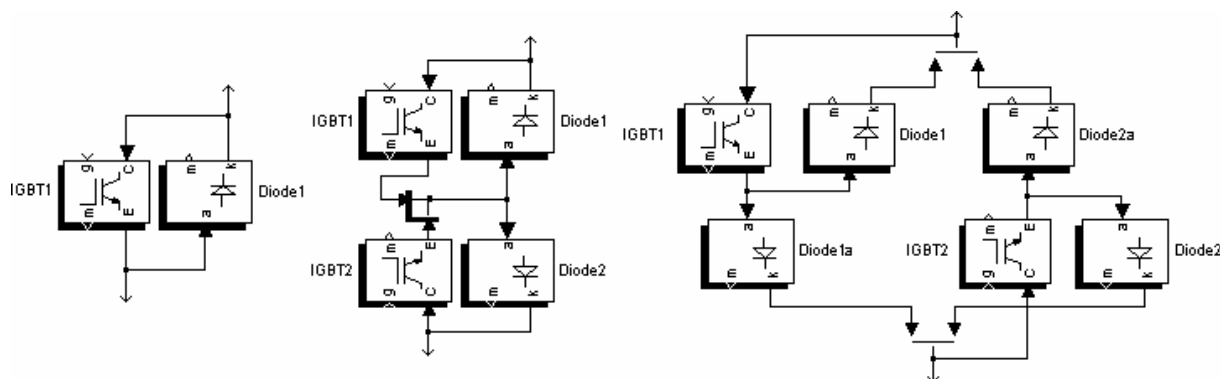


Рис. 7. Различные варианты исполнения управляемых силовых модулей.

Для обмена реактивной мощностью, включающей мощность высших гармоник, между сетью переменного тока и АП, очевидно, что со стороны постоянного тока к нему необходимо подключить накопитель реактивной мощности - индуктивный или емкостной. АП с емкостным накопителем энергии получили большее распространение по сравнению со схемами с индуктивным накопителем энергии. Это объясняется сложностью реализации систем управления на базе стандартных методов управления сетевыми токами из-за большой индуктивности накопителя [17].

В зарубежной литературе преобразователь переменного/постоянного тока, использующийся непосредственно для преобразования (выпрямления), именуется [16] ШИМ (активный) выпрямитель (АВ). ШИМ выпрямитель с емкостным фильтром или ШИМ (активный) выпрямитель напряжения, обеспечивающий двухсторонний обмен энергией между питающей сетью и нагрузкой, представляет собой обращенное относительно зажимов питания и нагрузки подключение схемы автономного инвертора напряжения к выпрямителю.

Для регулирования качества электроэнергии традиционно использовались компенсаторы реактивной мощности и пассивные фильтры [18]. Создание и освоение промышленностью высокоэффективных силовых приборов: биполярных транзисторов с изолированным затвором (IGBT) и запираемых тиристоров с интегрированным блоком управления (IGCT) позволило существенно расширить функции силовых устройств. Достоинством активного преобразователя является то, что он может работать в ре-

жиме активного силового фильтра (АСФ), компенсируя мощность основной гармоники и подавляя высшие гармоники, создаваемые не только их непосредственным потребителем, но и сторонним нелинейным потребителем. Активный силовой фильтр, по сути, представляет собой активный преобразователь, на который возложена функция регулирования качества энергопотребления в энергоподсистеме. АП в режиме АСФ может подключаться параллельно или последовательно нелинейному потребителю [14], [15]. Применение АП, в которых использовались оба режима АВ и АСФ, оказалось непрактичным и дорогим [19]. Основным недостатком АП в режиме АСФ является относительно большая установленная мощность. В этом отношении более перспективными являются гибридные фильтры [20], [21]. Последние являются компромиссным техническим решением, сочетающим достоинства традиционных пассивных фильтров из реактивных элементов и АСФ сравнительно малой мощности [14], [15].

Среди способов управления активными преобразователями следует выделить основные на базе: стандартных векторных алгоритмов [7], [22], [23]; прогнозирующего релейно-векторного управления [13]; нечеткого или фаззи-регулирования [24]–[26]; нейро-сетевого управления [24], [27]. Согласно работе [13] прогнозирующее релейно-векторное управление демонстрирует лучшее динамическое поведение, так же как фаззи-регулирование согласно [26], чем стандартные методы управления. При реализации различных алгоритмов управления в пакете MATLAB/Simulink для преобразования координат удобно применять стандартные блоки приложения SimPowerSystems раздел Extra Library, для реализации нечеткого регулятора следует использовать Fuzzy Logic Toolbox, при использовании нейро-сетевых технологий целесообразно применять Neural Network Blockset.

Основой любого активного преобразователя является полупроводниковый коммутатор [13]. Одна из возможных схем исполнения полупроводникового коммутатора для трехфазного активного преобразователя разработана в пакете MATLAB, выполнена в виде подсистемы (subsystem) и представлена на рис. 8. Используемые в схеме силовые модули обеспечивают полную управляемость преобразователем в генераторном режиме. Пример энергоподсистемы на базе трехфазного ШИМ — выпрямителя напряжения с RL нагрузкой, шунтированной конденсатором фильтра C , приведен на рис. 9. Преобразователь полагаем симметричным, представляющим собой в квазиустановившемся режиме симметричную нагрузку [3], что позволяет использовать при моделировании разработанные измерительные схемы (рис. 1) и (рис. 2).

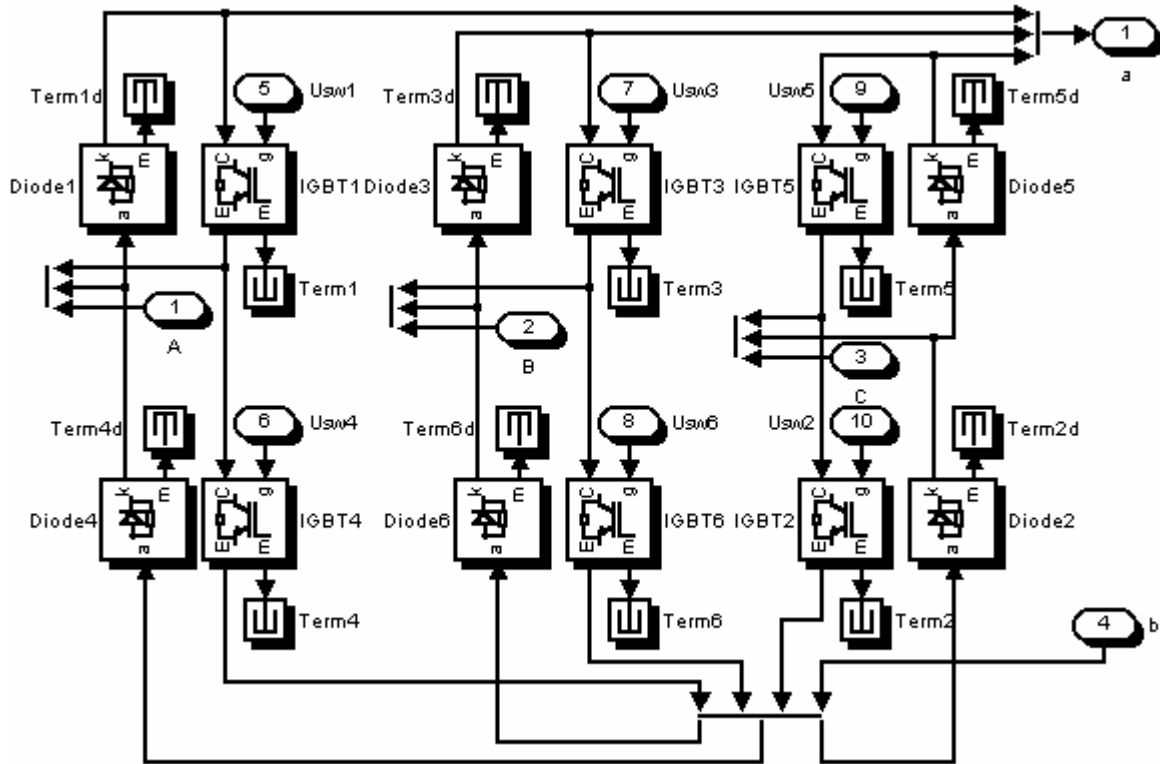


Рис. 8. Модель трехфазного мостового ШИМ коммутатора (Three Phase Bridge PWM Switchboard).

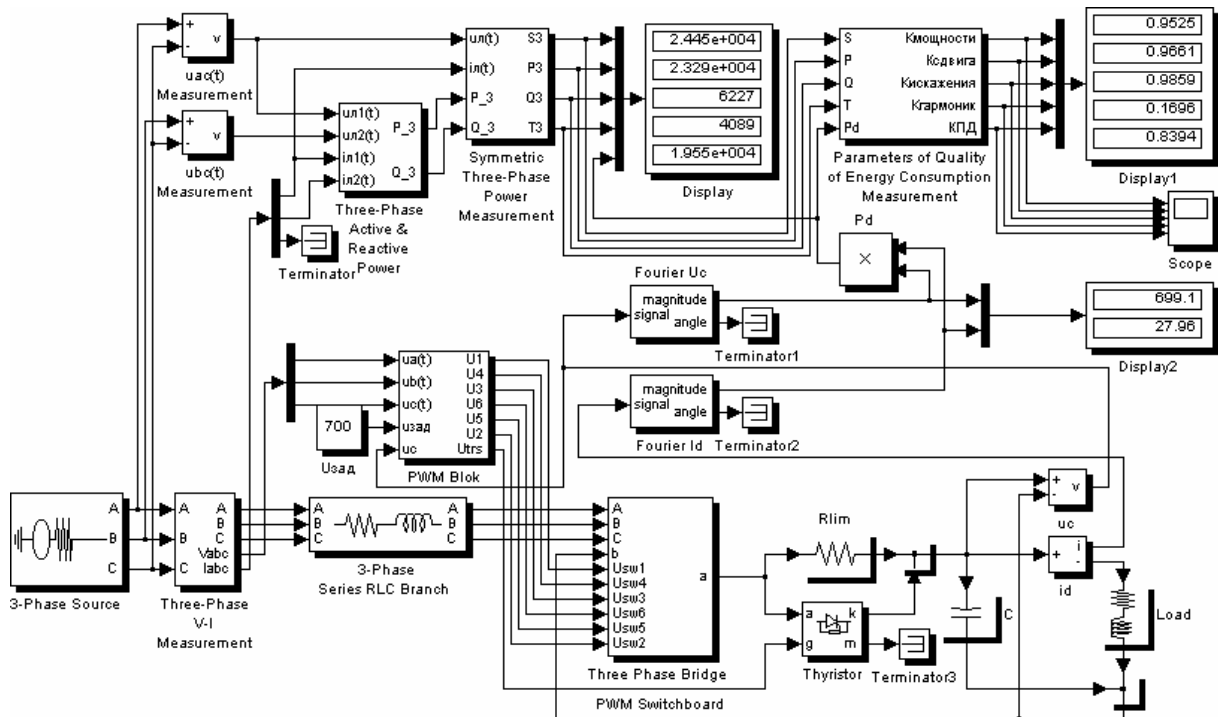


Рис. 9. Модель энергоподсистемы на базе трехфазного ШИМ — выпрямителя напряжения с RL нагрузкой, шунтированной конденсатором фильтра C .

В модели (рис. 9) используется алгоритм управления с одноконтурной структурой регулирования и стабилизации выходного напряжения

[28], который обеспечивает нормальную работу ШИМ - выпрямителя в случае несимметрии источника. Управление силовыми ключами осуществляется с двухсторонней ШИМ при частоте коммутации $f_k = 5$ кГц, тип регулятора пропорционально-интегральный. Числовые параметры при моделировании взяты следующими: $U_{л} = 380$ В, $f_s = 50$ Гц, $R_{Series} = 0.2$ Ом, $L_{Series} = 4$ мГн, $R_{lim} = 2$ Ом, $C = 1000$ мкФ, $R_{Load} = 25$ Ом, $L_{Load} = 1$ мГн, $U_{зАД} = 700$ В. Включение выпрямителя в питающую сеть является наиболее тяжелым режимом работы и, как правило, сопровождается увеличением электромагнитных нагрузок на элементы энергоподсистемы [29]. Для гашения переходного процесса при пуске выпрямителя используют цепь запуска (рис. 9), включающую в себя гасящий резистор и тиристор. Тиристор цепи запуска включается, когда напряжение на конденсаторе силового фильтра достигнет заданного значения (при моделировании выбрано 380 В), и шунтирует гасящий резистор R_{lim} .

Величины составляющих полной (кажущейся) мощности и показателей качества энергопотребления, полученные при моделировании, отражены в окнах блоков Display, Display1 (рис. 9). Коэффициент мощности в установленном режиме соответствует нормам, согласно которым для электротехнических комплексов указанного класса мощностей он должен быть в пределах 0.95.

Приведенные примеры позволяют в целом заключить, что в пакете MATLAB/Simulink используя стандартные или разработанные измерительные схемы можно решать следующие задачи:

- определять и анализировать составляющие полной (кажущейся) мощности и показатели качества потребляемой электроэнергии в трехфазных симметричных системах;
- реализуя различные алгоритмы управления полупроводниковым преобразователем, решать вопросы нахождения энергооптимального алгоритма управления;
- на основании результатов, полученных в ходе моделирования, определять зависимости массогабаритных показателей энергоподсистемы от составляющих полной мощности для квазистатического режима работы.

Работа выполнена при финансовой поддержке Министерства образования России в рамках плана работ по гранту А03-3.14-171.

Литература

1. *ГОСТ 13109-97. Электрическая энергия. Электромагнитная совместимость. Нормы качества электрической энергии в системах электроснабжения общего назначения.*— М.: Госстандарт, 1998.

2. Глазенко Т. А., Томасов В. С. Состояние и перспективы применения полупроводниковых преобразователей в приборостроении // Изв. вузов. Приборостроение.— 1996.— Т.39.— №3.— С.5.
3. Маевский О. А. Энергетические показатели вентильных преобразователей.— М.: Энергия, 1978.— 320 с.
4. Лурье Л. С. Коэффициент мощности несимметричной трехфазной нагрузки // Электричество.— 1952.— № 3.
5. Супронович Г. Улучшение коэффициента мощности преобразовательных установок: Пер. с польск. под ред. д. т. н., проф. В.А. Лабунцова.— М.: Энергоатомиздат, 1985.— 136 с.
6. Дрехслер Р. Измерение и оценка качества электроэнергии при несимметричной и нелинейной нагрузке: Пер. с чешск. А. А. Окина.— М.: Энергоатомиздат, 1985.— 112 с.
7. Зиновьев Г.С. Прямые методы расчета энергетических показателей вентильных преобразователей.— Новосибирск: Изд-во Новосиб. ун-та, 1990.— 220 с.
8. Нейман Л. Р., Демирчян К. С. Теоретические основы электротехники. В 2 т.— Л.: Энергия. Ленинградское отделение. 1967.
9. Бессонов Л. А. Теоретические основы электротехники: Электрические цепи: Учебник для электротехн., энерг., приборостроит. спец. вузов.— 8-е изд., перераб. и доп.— М.: Высш. шк., 1984. — 559 с..
10. Сборник задач и упражнений по теоретическим основам электротехники: Учеб. пособие для вузов / Под ред. проф. П. А. Ионкина.— М.: Энергоиздат, 1982.— 768 с.
11. Gyugyi L., Stricula E. C. Active AC power filters // Conf. Rec. Meet IEEE Ind. Appl. Soc.— 1976.— P.529–535.
12. Лабунцов В. А., Дайжун Ч. Трехфазный выпрямитель с емкостным фильтром и улучшенной кривой потребляемого из сети тока // Электричество.— 1993.— №6.— С.45–48.
13. Ефимов А. А., Шрейнер Р. Т. Активные преобразователи в регулируемых электроприводах переменного тока / Под общей ред. д. т. н., проф. Р. Т. Шрейнера.— Новоуральск: Изд-во НГТИ, 2001.— 250 с.
14. Розанов Ю. К., Рябчицкий М. В. Современные методы улучшения качества электроэнергии (аналитический обзор) // Электротехника.— 1998.— №3.— С.10–17.
15. Розанов Ю. К., Рябчицкий М. В., Кваснюк А. А. Современные методы регулирования качества электроэнергии средствами силовой электроники // Электротехника.— 1999.— №4.— С.28–32.
16. Vlatkovic V., Borojevic D., Lee F. C. A zero-voltage switched, three-phase isolated PWM buck rectifier // IEEE Trans. on Power Electronics.— V.10.— N.2.— March 1995.— P.148–157.

17. *Mao H., Lee F. C., Boroyevich D., Hiti S.* Review of high performance three-phase power-factor correction circuits // *IEEE Trans. Ind. Electron.*— V.44.— Aug. 1997.— P.437–446.
18. *Глинттерник С. Р.* Тиристорные преобразователи со статическими компенсирующими устройствами.— Л.: Энергоатомиздат. Ленингр. отделение, 1988. — 240 с.
19. *Katic V. A., Graovac D.* A method for PWM rectifier line side filter optimization in transient and steady states // *IEEE Trans. on Power Electronics.*— V.17.— N.3.— May 2002.— P.342–352.
20. *Rastogi M., Naik R., Mohan N.* A comparative evaluation of harmonic reduction techniques in three phase utility interface of power electronic loads // *Proc. IEEE—IAS Ann. Meeting.*— Toronto, Canada, Oct. 1993.— P.971–978.
21. *Malesani L., Tenti P.* Three-phase ac/dc PWM converter with sinusoidal ac currents and minimum filter requirements // *IEEE Trans. Ind. Applicat.*— V.23.— Jan./Feb. 1987.— P.71–77.
22. *Pouliquen H., Rioual P.* Vector control of shunt active filters // *Epe'95.*— Sevilla, 1995.— V.1.— P.880–885.
23. *Lee W.-C., Hyun D.-S., Lee T.-K.* A novel control method for three-phase PWM rectifiers using a single current sensor // *IEEE Trans. on Power Electronics.*— V.15.— N.5.— Sept. 2000.— P.861–870.
24. *Bose B. K.* Expert system, fuzzy logic, and neural network applications in power electronics and motion control // *Proc. IEEE.*— V.82.— Aug. 1994.— P.1303–1323.
25. *Ghazi R., Toliyat H. A., Rafiri S. M. R.* A fuzzy-genetic pulse width modulation for active power filters // *Stockholm power tech. conference.*— Stockholm, Sweden, 1995.— P.267–272.
26. *Dixon J. W., Contardo J. M., Moran L. A.* A fuzzy-controlled active front-end rectifier with current harmonic filtering characteristics and minimum sensing variables // *IEEE Trans. on Power Electronics.*— V.14.— N.4.— July 1999.— P.724–729.
27. *Carraso J. M., Quero J. M., Gomes R., Franquelo L. G.* An analog neural network controller for an active power filter on the instantaneous reactive power theory // *Epe'95.*— Sevilla, 1995.— V.1.— P.385–389.
28. *Томасов В. С., Борисов П. А.* Анализ электромагнитных процессов при включении активных выпрямителей напряжения в питающую сеть // *Научно-технический вестник СПб ГИТМО (ТУ). Выпуск 10. Информация и управление в технических системах. Санкт-Петербург.*— 2003.— С.218–222.
29. *Глазенко Т. А., Томасов В. С.* Методика анализа переходных процессов в выпрямителях с rC-фильтрами // *Изв. вузов. Приборостроение.*— 1994.— Т.37.— №11–12.— С.45–53.

УДК 621.3

ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ SIMULINK ДЛЯ ОЦЕНКИ ПОТРЕБЛЕНИЯ ЭЛЕКТРОЭНЕРГИИ АСИНХРОННЫМ ЭЛЕКТРОПРИВОДОМ

Браславский И. Я., Ишиматов З. Ш., Плотников Ю. В.

*Уральский государственный технический университет — УПИ, Екатеринбург,
e-mail: braslav@ep.etf.ustu.ru, ishmatov@mail.ru, plotnikovYV@mail.ru*

Введение

В последнее время наметилась устойчивая тенденция к использованию частотно-регулируемых асинхронных приводов в различных промышленных механизмах. Это обусловлено многими факторами, в том числе снижением потребления энергии при внедрении таких электроприводов. Очевидно, что при этом возникает задача оценки потребления электроэнергии в процессе работы электропривода для обоснования его экономической эффективности. В данной работе рассматриваются вопросы вычисления потребления энергии с использованием приложения Simulink пакета MATLAB. Использование предлагаемой методики в процессе моделирования асинхронного электропривода конкретного механизма позволяет оценить энергию, потребленную из сети, и потери энергии при его работе, т. е. энергетическую эффективность современных систем асинхронного электропривода. Кроме технико-экономического обоснования [1,2], методика вычисления потребленной энергии может также использоваться для проверки выбранного двигателя по нагреву; оба этих применения существенно упрощают процесс проектирования электроприводов.

Постановка задачи

В задачи разработанного блока вычисления потребления энергии входит:

1. Определение потребляемой из сети энергии за цикл работы при любых режимах работы асинхронного двигателя.
2. Определение потерь энергии в этих режимах.
3. Разделение потерь на потери в статоре и роторе.
4. Определение циклового КПД.
5. Определение мгновенных и средних потерь.

При разработке этого блока учтено, что при разделении потерь на потери в статоре и потери в роторе используется значение сопротивления статора, которое в некоторых случаях может меняться (например, в двух-

скоростном асинхронном двигателе лифта при переключении на высоко-полюсную обмотку [1]). При синтезе блока не учитываются потери в стали, добавочные и вентиляционные, что понятно при оценке переходных режимов, которые в механизмах циклического действия (например, в подъемно-транспортных механизмах) занимают большую часть времени.

Также необходимо учесть, что блок будет использоваться для определения потребляемой энергии в регулируемом электроприводе, представленном в относительных единицах, при этом уравнения блока претерпят некоторые изменения, приведенные далее.

Построение блока вычисления потребляемой энергии и потерь

Как было указано выше, при синтезе блока с использованием относительных единиц изменяется уравнение полной мощности, потребляемой из сети [3]. Поэтому будет рассмотрено два варианта построения блока.

Блок вычисления потребляемой энергии и потерь в абсолютных единицах. Уравнения для построения блока в этом случае имеют вид [4]:

Механическая мощность на валу двигателя:

$$P_{\text{мех}} = M \cdot \omega ,$$

где M и ω — момент и угловая скорость двигателя.

Полная мощность, потребляемая из сети:

$$P_s = \frac{3}{2} \cdot (U_{sx} I_{sx} + U_{sy} I_{sy}) ,$$

где I_{sx} , I_{sy} , U_{sx} , U_{sy} — проекции векторов тока и напряжения статора во вращающейся системе координат.

Баланс мощностей без учета потерь в стали:

$$P_s = P_{\text{мех}} + \Delta P_{s\text{э\text{э}}} + \Delta P_{r\text{э\text{э}}} .$$

Общие потери из баланса мощностей:

$$\Delta P_{\Sigma} = P_s - P_{\text{мех}} .$$

Потери в статорных цепях:

$$\Delta P_{s\text{э\text{э}}} = R_s I_{sx}^2 + R_s I_{sy}^2 ,$$

где R_s — сопротивление статорных цепей.

Потери в роторных цепях:

$$\Delta P_{r\text{э\text{э}}} = \Delta P_{\Sigma} - \Delta P_{s\text{э\text{э}}} .$$

Энергия, потребляемая из сети:

$$W_s = \int_0^{T_{\text{ц}}} P_s dt ,$$

где $T_{\text{ц}}$ — время цикла работы производственного механизма.

Энергия потерь:

$$\Delta W_{\Sigma} = \int_0^{T_u} (P_s - P_{mex}) dt .$$

Цикловой КПД как отношение механической энергии к энергии, потребляемой из сети за цикл работы:

$$\eta_u = \frac{W_{mex}}{W_s} = \frac{\int_0^{T_u} P_{mex} dt}{\int_0^{T_u} P_s dt} .$$

Блок вычисления потребляемой энергии и потерь в относительных единицах. В рамках данной работы используется система относительных единиц, ориентированная на переменные цепей статора и электромагнитную мощность машины. Это позволяет получить номинальные переменные статора и номинальный электромагнитный момент равными единице [3].

В этом случае изменится только выражение для потребляемой из сети мощности [3] и для получения энергии в относительных единицах нужно вводить базисную мощность.

Полная мощность, потребляемая из сети:

$$p_s = (u_{sx} i_{sx} + u_{sy} i_{sy}) .$$

Структурная схема блока, построенная на основании приведенных уравнений, показана на Рис. 1. Здесь приводится реализация этой схемы в приложении Simulink для блока с использованием относительных единиц. Схема блока с использованием абсолютных единиц не имеет принципиальных отличий. В схему введена постоянная величина, равная 0,01, для предотвращения деления на нуль в начале моделирования. Это не оказывает существенного влияния на точность вычисления.

На Рис. 2 показан внешний вид блока и окно установки параметров для маскированной подсистемы [5].

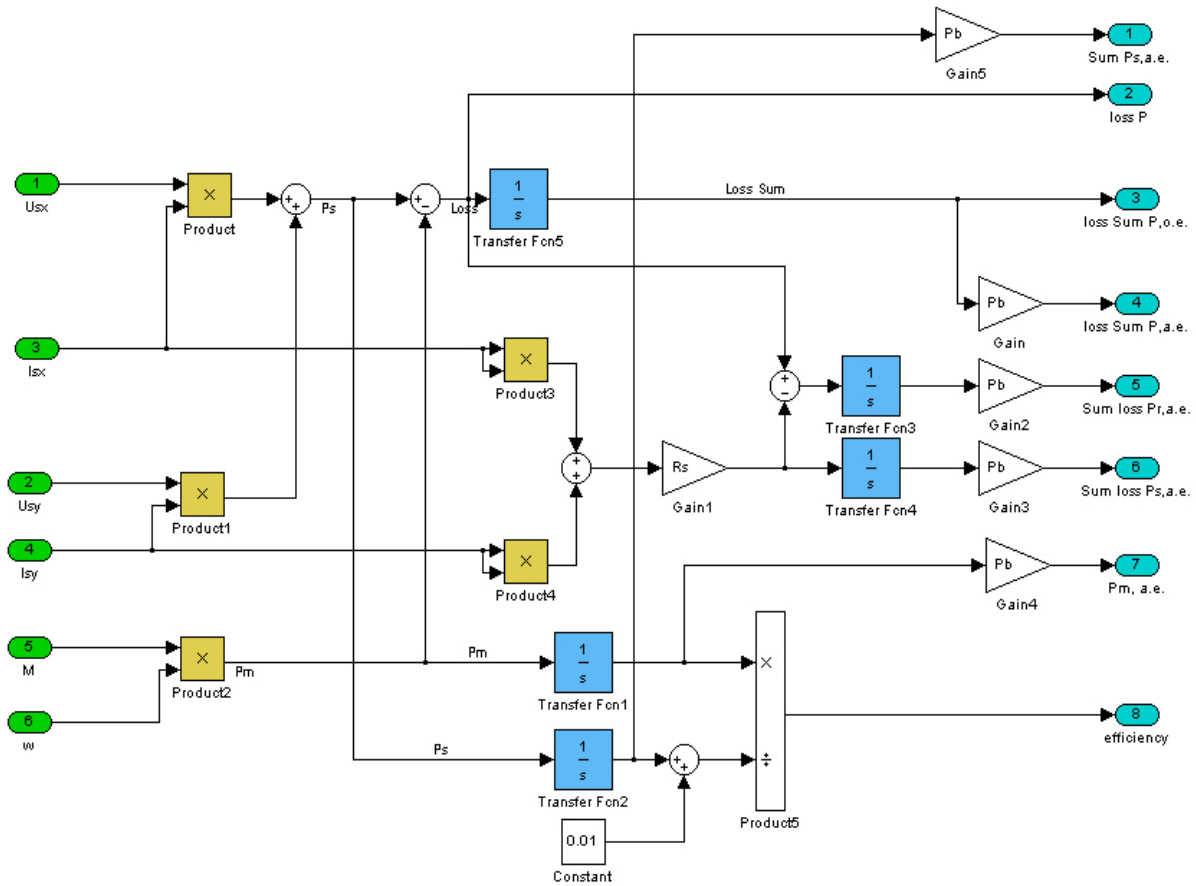


Рис. 1. Структурная схема блока вычисления потребляемой энергии и потерь в приложении Simulink пакета MATLAB.

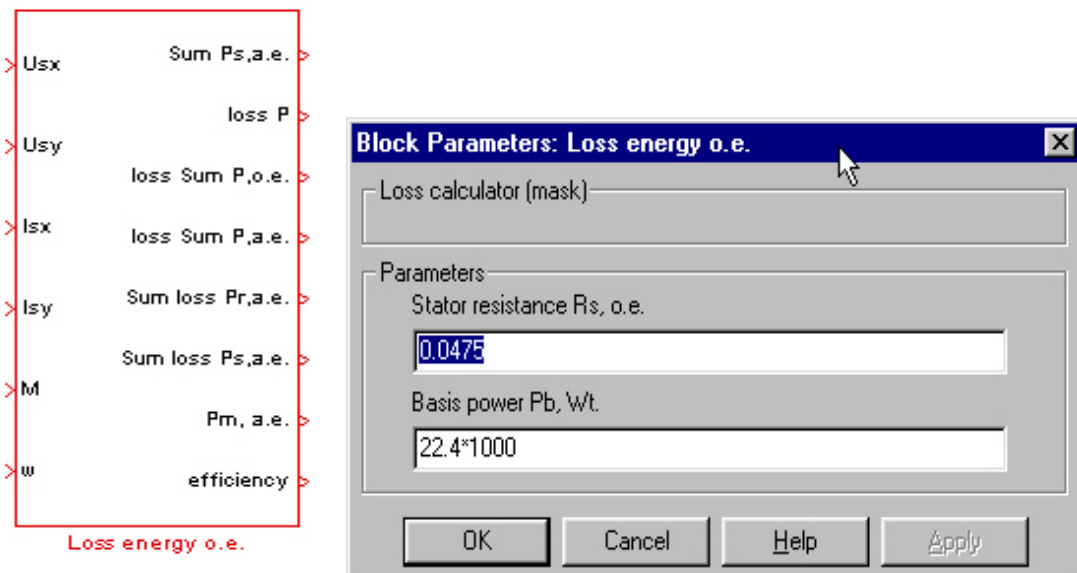


Рис. 2. Внешний вид блока вычисления потребления энергии (слева) и окно установки параметров (справа).

Результаты моделирования

Рассмотрим пример моделирования процесса пуска асинхронного двигателя 4A180S2У3 мощностью 22 кВт, данные которого можно найти в [6], при различных способах управления [7]:

1. Векторное управление с ориентацией поля по вектору потокосцепления ротора.
2. Скалярное управление с компенсацией падения напряжения на сопротивлении статора на низких частотах вращения.
3. Прямой пуск асинхронного двигателя.

Структурная схема для моделирования, реализованная в Simulink, представлена на Рис. 3.

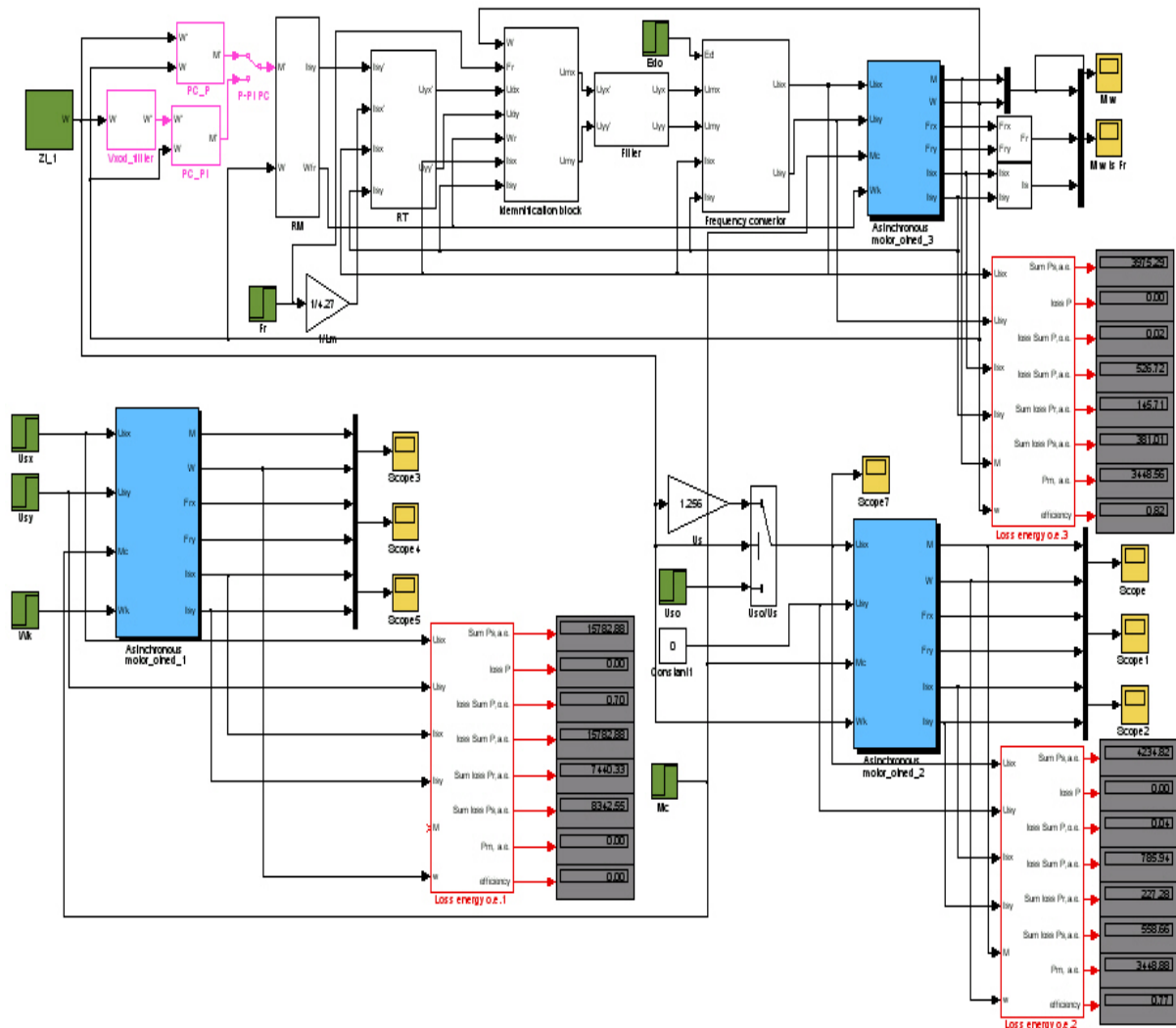


Рис. 3. Структурная схема для расчета потребляемой энергии при различных способах управления асинхронным двигателем в режиме пуска (вверху — векторное управление, в нижней части справа — скалярное управление, в нижней части слева — асинхронный двигатель, напрямую подключенный к сети).

Переходные процессы, соответствующие рассматриваемым системам, приведены на Рис. 4. Момент статического сопротивления в процессе моделирования принимался равным нулю. Механическая энергия для всех трех способов управления остается постоянной [8] и равной 3,4496кВт·с.

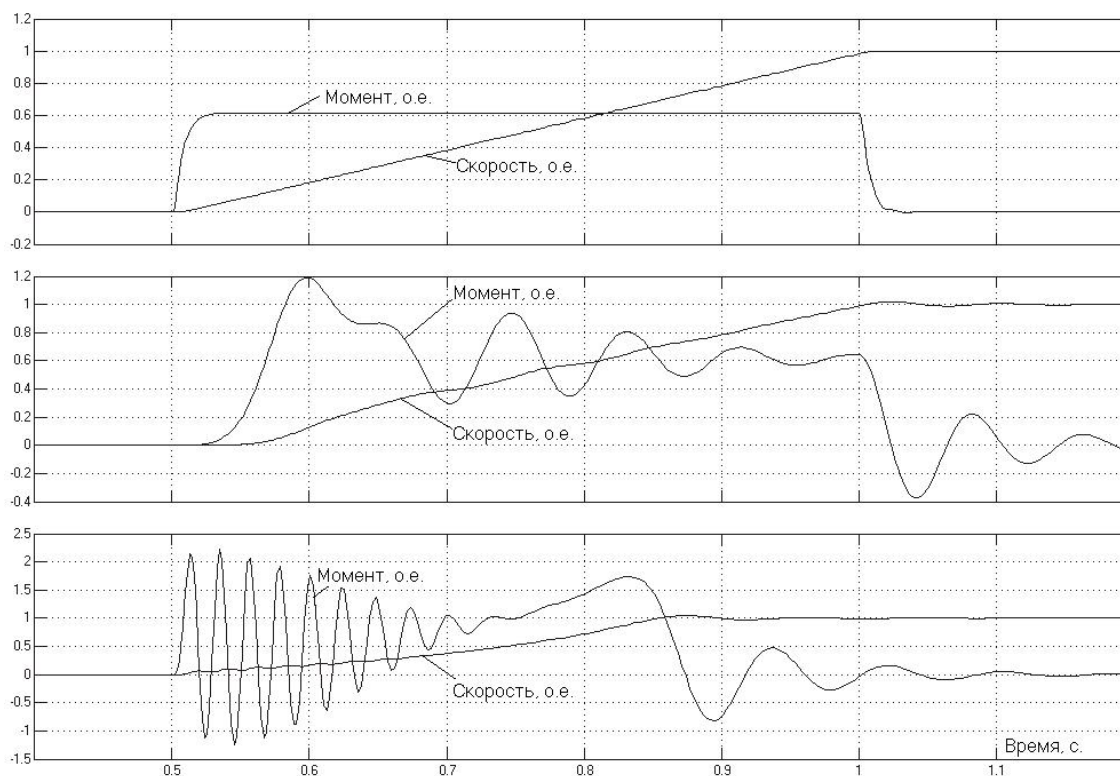


Рис. 4. Переходные процессы (вверху — векторное управление, в средней части — скалярное управление, внизу — асинхронный двигатель, напрямую подключенный к сети)

Результаты вычисления потребляемой энергии и потерь за время пуска сведены в табл. 1.

Таблица 1.

Потребляемая энергия и потери для различных способов управления асинхронным двигателем в режиме пуска.

Энергия, кВт·с	Прямой пуск	Частотно-регулируемый электропривод	
		Скалярное управление	Векторное управление
Потребленная из сети, W_s	15,78	4,23	3,93
Полных потерь в меди, W_{Σ}	12,33	0,79	0,48
Потерь в меди ротора, $\Delta W_{rэл}$	3,99	0,23	0,13
Потерь в меди статора, $\Delta W_{sэл}$	8,34	0,56	0,35
Механическая на валу, $W_{мех}$		3,45	

Для иллюстрации полученных результатов приведена диаграмма, изображенная на рис. 5, в которой потребляемая энергия и потери показаны в относительных единицах. За 100% принято значение потребляемой из сети энергии при прямом пуске асинхронного двигателя.

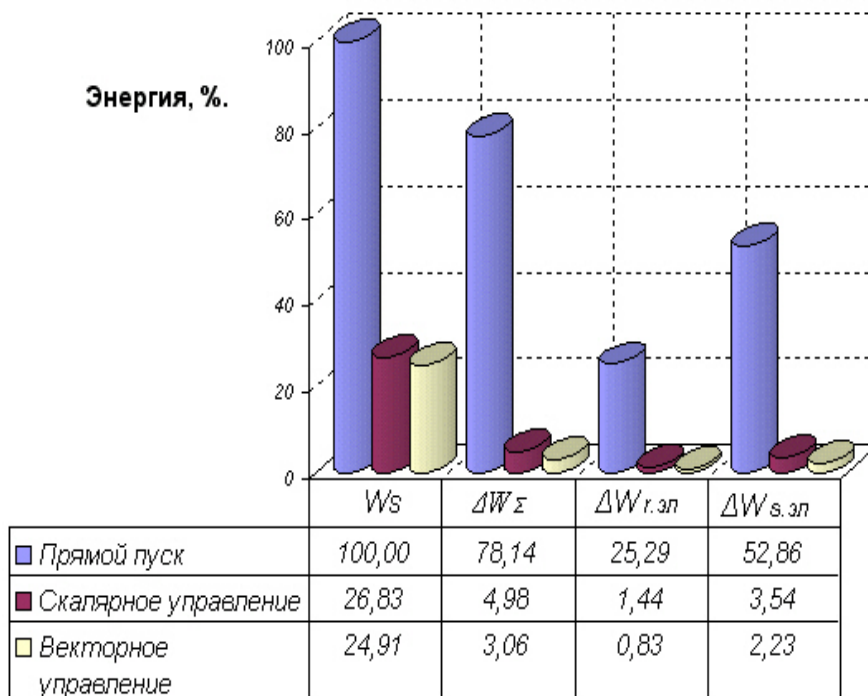


Рис. 5 Диаграмма потребляемой энергии и потерь при пуске асинхронного двигателя с различными способами управления.

Как видно из приведенного примера, использование блока вычисления потребляемой энергии позволяет получить достаточно полные данные для технико-экономического обоснования внедрения новой системы электропривода. При этом для оценки потребления электроэнергии необходимо построить модели сравниваемых систем электропривода. Применение разработанного блока упрощает процедуру расчета потребляемой энергии, поскольку не требует от разработчика дополнительных затрат.

Следует также отметить, что, как было указано выше, данный блок может применяться и для проверки выбранного двигателя по нагреву методом средних потерь за цикл работы [8].

Литература

1. Браславский И. Я., Ишиматов З. Ш., Плотников Ю. В. Об эффективности применения частотно-регулируемого электропривода в подъемно-транспортных механизмах // Вестник Национального технического университета «Харьковский политехнический институт». — 2003.

2. *Браславский И. Я., Ишиматов З. Ш., Авербах И. А., Барац Е. И., Киртичников Н. В.* Частотно-регулируемый асинхронный электропривод кранов // Энергетика региона.— №1.— 2004.
3. *Шрейнер Р. Т.* Математическое моделирование электроприводов переменного тока с полупроводниковыми преобразователями частоты.— Екатеринбург: УРО РАН, 2000.
4. *Копылов И. П.* Математическое моделирование электрических машин: Учебник для вузов.— 3-е изд., перераб. и доп.— М.: Высш. шк., 2001.
5. *Дьяконов В.* Simulink 4. Специальный справочник.— СПб: Питер, 2002.
6. Асинхронные двигатели серии 4А: Справочник / А. Э. Кравчик, Е. А. Соболенская.— М.: Энергоиздат, 1982.
7. *Поздеев А. Д.* Электромагнитные и электромеханические процессы в частотно-регулируемых электроприводах.— Чебоксары: Изд-во Чуваш. ун-та, 1998.
8. *Ключев В. И.* Теория электропривода: Учебник для вузов.— 2-е изд. перераб. и доп.— М.: Энергоатомиздат, 1998.

УДК 62–83.001.57

МОДЕЛИРОВАНИЕ ВЕНТИЛЬНЫХ АСИНХРОННЫХ ЭЛЕКТРОПРИВОДОВ

*Браславский И. Я., Костылев А. В., Мезеушева Д. В.,
Степанюк Д. П., Кириллов А. В.
ГОУ ВПО «УГТУ-УПИ», Екатеринбург,
e-mail:kafedra@ep.etf.ustu.ru*

Введение

Разработка математической модели исследуемой системы является первой задачей практически любого исследования. На этапе разработки модели закладываются основные допущения анализа, которые, по сути, являются ограничениями применимости модели. Трудозатраты на разработку и отладку модели, как правило, существенно превосходят трудозатраты на дальнейшие численные эксперименты, проводимые на ее основе. Еще совсем недавно создание моделей систем электроприводов начиналось многими начинающими исследователями «с нуля». Требовалось создавать модели двигателей, полупроводниковых элементов и т. п. на основе исходных уравнений объекта. Сегодня в помощь специалистам разработаны специализированные мощные пакеты моделирования, содержащие в своем составе как готовые элементы, так и уже собранные системы электроприводов различных типов.

Бесспорным лидером среди таких пакетов является MATLAB. Это обусловлено с одной стороны универсальностью и огромным разнообразием имеющихся библиотек моделей самых различных типов, а с другой их открытостью, позволяющей свободно осуществлять их модификацию и расширение. В настоящей статье представлены методы моделирования весьма широко распространенных сегодня вентильных асинхронных электроприводов, разработанные авторами в результате многочисленных прикладных исследований. По сути, предложена универсальная методика моделирования асинхронных приводов и их систем управления для широкого круга задач.

В общем случае можно выделить следующие задачи модельного анализа вентильных электроприводов с асинхронными машинами.

Расчет статических и квазиустановившихся режимов работы привода. Обычно эти исследования не сопряжены с особыми сложностями, а математические модели легко поддаются программированию.

Анализ динамических свойств систем, связанный с исследованием устойчивости, качества процессов, построением ЛЧХ и переходных функ-

ций. Такой анализ, как правило, выполняется на основе линеаризованной модели системы «преобразователь — асинхронный двигатель».

Моделирование процессов в системах электроприводов без учета преобразователей электрической энергии. Здесь решается большинство задач, связанных с начальными этапами синтеза систем. Следует отметить, что во многих случаях этого анализа достаточно для получения достоверных результатов.

Моделирование динамических и квазиустановившихся процессов в системах электроприводов с учетом преобразователей электрической энергии. В этом случае модели имеют наибольшую сложность и требуют максимальных ресурсов вычислительных систем.

Далее рассмотрены приемы разработки и анализа моделей электроприводов для указанного круга задач.

Исследование динамических свойств электроприводов с асинхронным электродвигателем в «малом»

При исследовании динамических свойств систем как объекта регулирования обычно проводится анализ устойчивости и качества процессов регулирования, строятся и анализируются ЛЧХ. Математические основы данных методов излагаются в учебной литературе по теории авторегулирования, например в [1]. Поскольку системы электроприводов являются нелинейными, для их анализа применяют линеаризацию, в частности, разложением в ряд Тейлора. В качестве примера рассмотрим модель роторных цепей асинхронной машины:

$$\begin{aligned} p\psi_{rx} &= k_r r_r i_{sx} - \alpha_r \psi_{rx} + \omega_k \psi_{ry} - \omega \psi_{ry}; \\ p\psi_{ry} &= k_r r_r i_{sy} - \alpha_r \psi_{yx} - \omega_k \psi_{rx} + \omega \psi_{rx}, \end{aligned} \quad (1)$$

где p — оператор дифференцирования; ψ_{rx} , ψ_{ry} — проекции вектора потока сцепления ротора в свободно ориентированной вращающейся системе координат; i_{sx} , i_{sy} — проекции тока статора; ω_k , ω — скорость вращения системы координат и угловая скорость ротора; k_r , r_r , α_r — коэффициент связи, активное сопротивление и коэффициент затухания роторной обмотки соответственно [2].

В результате разложения в ряд Тейлора система (1) преобразуется к виду:

$$\begin{aligned} p_{\Delta} \psi_{rx} &= k_r r_r \Delta i_{sx} - \alpha_r \Delta \psi_{rx} + \Delta \omega_k \psi_{ry}^0 + \omega_k^0 \Delta \psi_{ry} - \Delta \omega \psi_{ry}^0 - \omega^0 \Delta \psi_{ry}; \\ p_{\Delta} \psi_{ry} &= k_r r_r \Delta i_{sy} - \alpha_r \Delta \psi_{ry} - \Delta \omega_k \psi_{rx}^0 - \omega_k^0 \Delta \psi_{rx} + \Delta \omega \psi_{rx}^0 + \omega^0 \Delta \psi_{rx}. \end{aligned} \quad (2)$$

Здесь символ Δ указывает на приращение, а символ 0 — на центр разложения в ряд соответствующей величины.

Полученная система легко преобразуется к матричной форме или форме пространства состояний:

$$\begin{aligned} pX &= AX + BU; \\ Y &= CX + DU, \end{aligned} \quad (3)$$

где

$$U = \begin{pmatrix} \Delta i_{sx} \\ \Delta i_{sy} \\ \Delta \omega_k \\ \Delta \omega \end{pmatrix}; X = \begin{pmatrix} \Delta \psi_{rx} \\ \Delta \psi_{ry} \end{pmatrix}; Y = \begin{pmatrix} \Delta \psi_{rx} \\ \Delta \psi_{ry} \end{pmatrix}; \quad (4)$$

$$\begin{aligned} A &= \begin{pmatrix} -\alpha_r & \omega_k^0 - \omega^0 \\ -\omega_k^0 + \omega^0 & -\alpha_r \end{pmatrix}; B = \begin{pmatrix} k_r r_r & 0 & \psi_{ry}^0 & -\psi_{ry}^0 \\ 0 & k_r r_r & -\psi_{rx}^0 & \psi_{rx}^0 \end{pmatrix}; \\ C &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; D = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \quad (5)$$

Для дальнейшего анализа линеаризованной системы (3) следует создать ее программную динамическую модель, воспользовавшись функцией `ss` (от англ. state space), которая принадлежит набору инструментов Control System Toolbox. Ее параметрами являются непосредственно матрицы (5). В нашем случае это будет выглядеть так:

`sys=ss(A,B,C,D).`

В результате будет создана структура, удобная для дальнейшей обработки средствами Control System Toolbox.

Если имеется необходимость создать динамическую модель, описанную передаточной функцией, следует использовать функцию `tf` (от англ. transfer function), параметрами которой являются векторы, содержащие коэффициенты полинома числителя и знаменателя. Так, например, для описания фильтра 1-го порядка с постоянной времени T_f следует записать:

`sys=tf(1,[Tf 1])`

Для анализа свойств полученной динамической модели целесообразно воспользоваться функцией `Itiview(sys)`. Эта функция открывает окно интегратора Control System Toolbox, позволяющего воспользоваться разнообразными приемами анализа динамических моделей. Нажав правой кнопкой на графиках можно выбрать их тип, а также входные и выходные координаты. Например, ниже на рис. 1. приведены ЛЧХ роторных цепей:

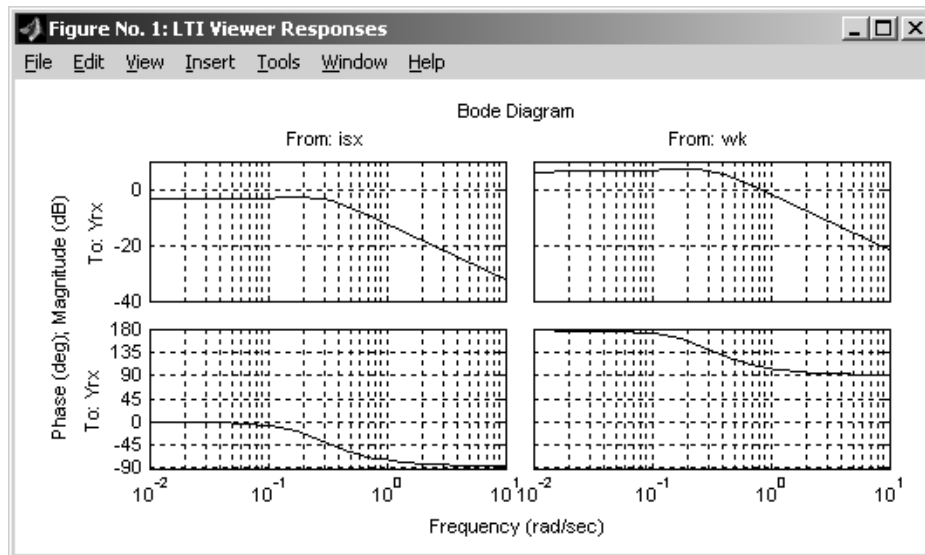


Рис. 1. ЛЧХ роторных цепей, построенные с помощью функции `ltiview`.

При анализе сложных систем, одни элементы которых могут быть описаны с помощью пространства состояний, а другие — с помощью передаточных функций удобно воспользоваться функциями `append` и `connect`. Первая из них формирует общую динамическую модель, создавая единый блок уравнений, а вторая формирует сколь угодно сложные связи между отдельными блоками. Подробное описание использования этих функций содержится в справочной системе.

Важно отметить, что задачу объединения моделей решает автоматически пакет `Simulink`. Для этого в окне модели, состоящей из стандартных блоков, нужно открыть меню «Tools–Linear analysis...». При этом поступит предложение указать входы и выходы исследуемой системы. После этого в открывшемся автоматически окне `LTI viewer` следует выбрать меню «Simulink–Get Linearized Model». Однако использовать этот прием следует аккуратно, если в модели присутствуют нелинейные звенья, т. к. `Simulink` выполняет их линеаризацию самостоятельно.

Моделирование систем с асинхронным электродвигателем

Составление модели асинхронной машины в среде `Simulink` по различным традиционным вариантам ее уравнений [2, 3] обычно не представляет сложности. Однако иначе обстоит дело при необходимости учета изменения параметров двигателя вследствие нагрева, эффекта вытеснения тока ротора, насыщения главной магнитной цепи. В этом случае параметры могут рассчитываться по достаточно сложным итерационным алгоритмам, которые крайне затруднительно реализовать как с помощью традиционного, так и с помощью расширенного набора блоков `Simulink`.

Для решения данной задачи целесообразно включить в модель двигателя блок расчета параметров, выполненный на основе S-функции. S-функция позволяет создать сколь угодно сложный динамический объект, описываемый в общем случае нелинейными дифференциальными или разностными уравнениями. Использование встроенного языка программирования пакета MATLAB, либо внешних языков, например С, позволяет программировать правые части дифференциальных уравнений с любым необходимым набором промежуточных вычислений, в том числе и итерационных. Впрочем, недостаток такого метода — отсутствие наглядности и сложность отладки модели. Поэтому полностью создавать модель двигателя как S-функцию не рекомендуется. Правила создания S-функций изложены в справочной системе.

В нашем случае входами S-функции являются выбранные переменные состояния двигателя, например, потокосцепления, а выходами — набор параметров уравнений системы электропривода.

В качестве примера использования данного приема приведем графики прямого пуска двигателя 4А132М6 без учета и с учетом эффекта вытеснения тока ротора. Расчетные соотношения приведены в [3].

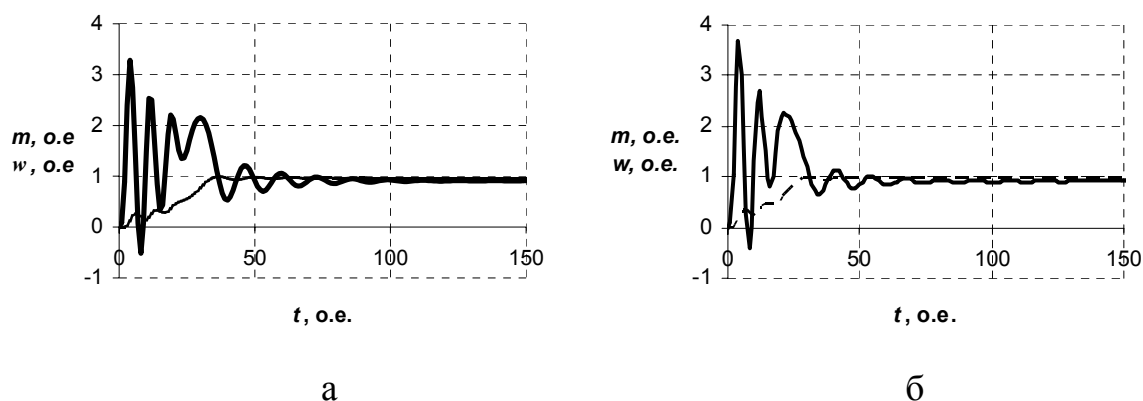


Рис. 2. Процессы пуска асинхронного двигателя:
а — без учета вытеснения; б — с учетом вытеснения
— момент двигателя, -- скорость

Для данного асинхронного двигателя эффект вытеснения незначительно изменяет вид процесса по сравнению с процессом, рассчитанным без учета этого явления. Тем не менее различия в графиках очевидны. С увеличением же мощности двигателя пренебрежение эффектом вытеснения в ряде случаев вообще не позволяет обеспечить процесс прямого пуска при номинальном моменте сопротивления, так как не обеспечивается требуемый пусковой момент.

Моделирование асинхронного электропривода с тиристорным преобразователем напряжения

Когда требуется моделировать процессы в системах электроприводов с учетом вентильных преобразователей, следует воспользоваться набором Power Systems Blockset. Разнообразный набор элементов электрических цепей, включая полупроводниковые элементы с реальными параметрами, позволяет достаточно уверенно создавать различные схемы силовых цепей. Цепи управления могут быть также собраны средствами Power Systems Blockset, либо, что удобнее, с помощью традиционных элементов Simulink.

В набор Power Systems Blockset входит также модель асинхронной машины. Однако данная модель построена без учета эффектов вытеснения тока ротора и насыщения главной магнитной цепи, что ограничивает ее применение в ряде задач. Поэтому представляет интерес встраивание модели двигателя, разработанной в традиционных элементах, в схему с элементами Power Systems. Пользовательские элементы для Power Systems Blockset создаются на основе модели управляемого источника тока с параллельно установленным сопротивлением достаточно большой величины. Последнее необходимо для обеспечения развязки с другими элементами схемы. Управляющим входом для источника тока является выход пользовательской модели, а напряжение на источнике в свою очередь является входом для нестандартного звена. В этом случае рассматривается трехфазная модель АД. С учетом вышесказанного модель двигателя выглядит так, как показано на рис. 3.

Тиристорный преобразователь напряжения не является стандартным элементом Power Systems. Поэтому его модель собирается из отдельных моделей силовых тиристоров. Пример модели приведен на рис.4.

Графики фазных токов и напряжений данной модели приведены на рис. 5. Расчет проведен при условии работы двигателя на подсинхронной скорости при пониженном напряжении. Угол открытия вентилей $\alpha=90^\circ$.

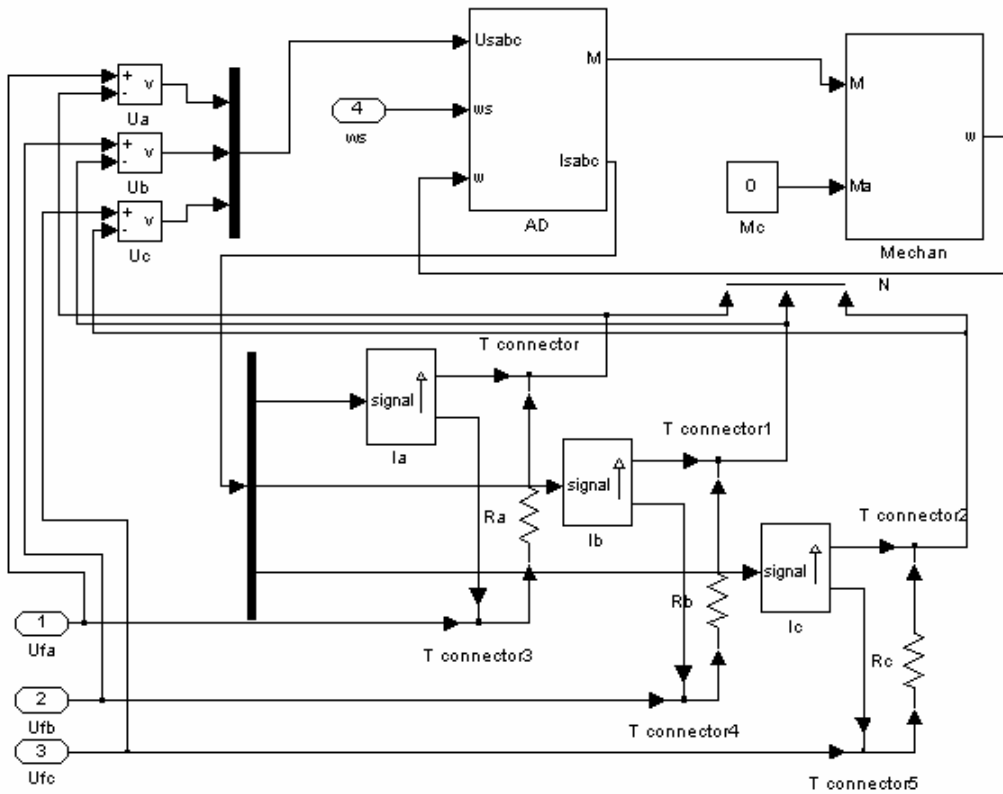


Рис.3. Модель асинхронной машины для Power Systems Blockset

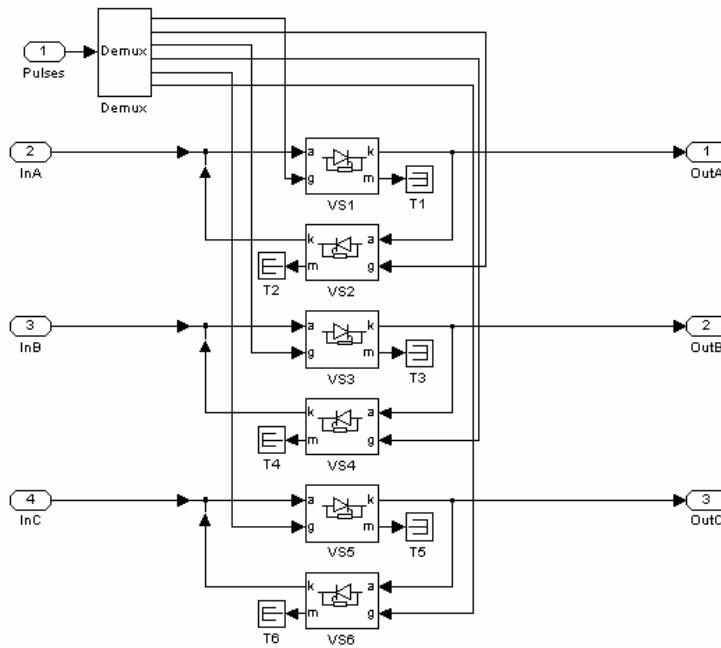


Рис. 4. Пример модели тиристорного преобразователя напряжения, выполненного с помощью блоков Power Systems

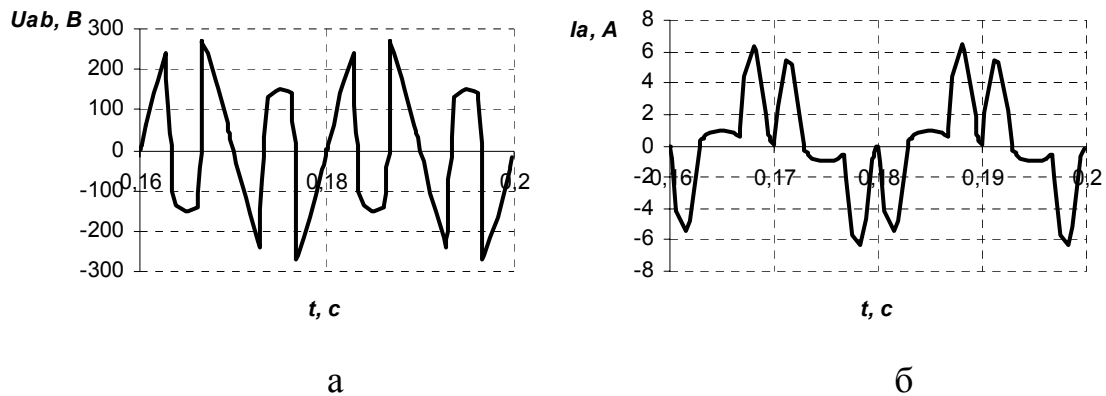


Рис. 5. Линейное напряжение (а) и ток фазы (б) в системе «Тиристорный преобразователь напряжения — асинхронный двигатель»

Моделирование асинхронного частотно-регулируемого электропривода на основе автономного инвертора напряжения с ШИМ

Модель автономного инвертора напряжения непосредственно встроена в Power Systems Blockset. Следует выбрать блок «Universal Bridge» — универсальный мост и сконфигурировать его следующим образом. Сделать точки A, B, C выходами модели, а также выбрать силовые ключи, например, IGBT. Для формирования управляющих импульсов предназначен блок PWM Generator. Данный блок может работать как автономно, так и в режиме отработки заданных сигналов управления. Наиболее простая схема системы с преобразователем частоты в этом случае приведена на рис. 6.

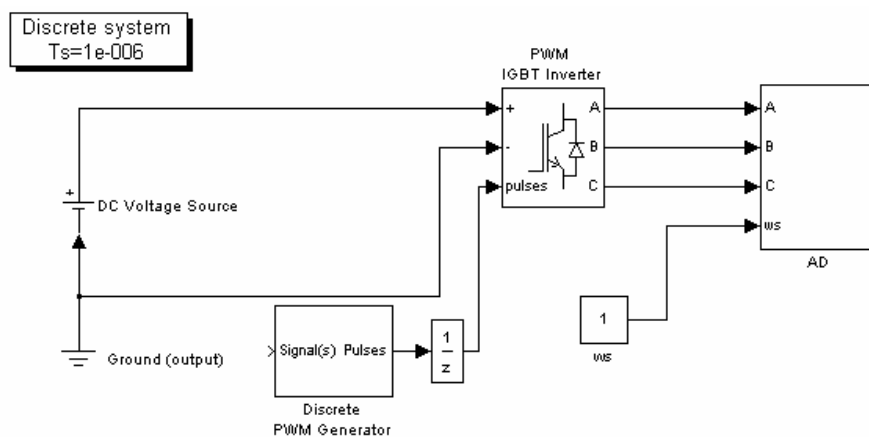


Рис. 6. Пример модели асинхронного электропривода с преобразователем частоты, выполненного с помощью блоков Power Systems.

Следует отметить, что при моделировании в качестве метода интегрирования целесообразно использовать дискретный метод. Однако тогда все непрерывные элементы схемы должны быть заменены на дискретные,

а также в модель должен быть добавлен блок Discrete System. Последний формирует заданное время дискретизации для всех элементов Power Systems.

Графики фазных токов и напряжений модели системы с преобразователем частоты приведены на рис. 7.

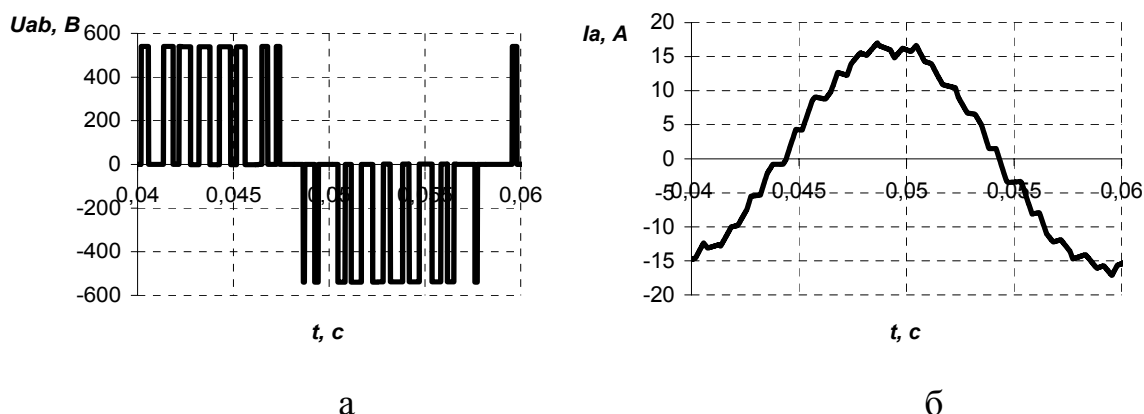


Рис. 7. Линейное напряжение (а) и ток фазы (б) в системе «Автономный инвертор напряжения с ШИМ — асинхронный двигатель».

Как уже было сказано, моделирование систем с преобразователями требует наибольших вычислительных ресурсов. Особенно это проявляется, когда требуется просчитывать многократно похожие процессы, например, при обучении нейронных сетей, решениях задач оптимизации и т. п. Здесь может оказаться полезным один простой прием — параллельное решение задачи на нескольких вычислительных машинах. Анализируемые вычислительные процессы запускаются с разными параметрами на нескольких машинах, объединенных в локальную сеть. Результаты расчетов сохраняются в файлах, которые расположены на так называемом сетевом диске главной машины, решающей задачу обработки процессов. Возможная коррекция параметров также передается через файл. Для централизованного управления процессами в этом случае можно использовать программы сетевого мониторинга.

Литература

1. Бесекерский В. А., Попов Е. П. Теория систем автоматического регулирования.— М. Наука, 1975.
2. Браславский И. Я. Асинхронный полупроводниковый электропривод с параметрическим управлением.— М.: Энергоатомиздат, 1988.
3. Шрейнер Р. Т. Математическое моделирование электроприводов переменного тока с полупроводниковыми преобразователями частоты.— Екатеринбург: УРО РАН, 2000.

УДК 681.3

МОДЕЛИ ДИНАМИЧЕСКИХ СИСТЕМ НЕЦЕЛОГО ПОРЯДКА В СРЕДЕ MATLAB/SIMULINK

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

Отделение гибридных моделирующих и управляющих систем в энергетике

Института проблем моделирования в энергетике

им. Г. Е. Пухова НАН Украины, Киев, Украина

e-mail: vsyv@visti.com, todorova@visti.com

Под динамическими системами понимают процессы и объекты любой физической природы, поведение которых описывается системой интегро-дифференциальных уравнений в обыкновенных и частных производных. Дробное интегро-дифференцирование является естественным обобщением обыкновенных производных и интегралов классического математического анализа. В настоящее время дробное исчисление, которое позволяет получить более точное представление реальных процессов, широко применяется при математическом моделировании различных явлений в средах с фрактальной структурой, физике, математической биологии и т. п. [1].

Наиболее часто встречается определение интеграла дробного порядка β , известное как интеграл Римана-Лиувилля [1]:

$$[D^{-\beta} x(t)]_0^t = \frac{1}{\Gamma(\beta)} \int_0^t (t-\tau)^{\beta-1} x(\tau) d\tau, \quad (1)$$

где через $\Gamma(*)$ обозначена Гамма-функция.

Операционные методы анализа широко применяются при моделировании динамических систем, в том числе, и нецелого порядка [2]. Операционный подход позволяет алгебраизировать интегро-дифференциальные уравнения, которыми обычно описываются динамические системы. Представление сигналов методами полиномиальной аппроксимации по некоторой системе базисных функций приводит к аппроксимирующим полиномиальным спектрам, на основе которых могут быть построены различные операционные исчисления. Особое место занимают локальные ортогональные системы базисных функций на основе смещенных полиномов Лежандра, в частности, методы блочно-импульсных функций и аппроксимирующих импульсных спектров [3]. Преимуществом таких методов аппроксимации является достаточно большое быстродействие и простота реализаций в программной среде и в аппаратном виде. Недостатком этих методов является то, что аппроксимации сигналов, полученные на основе метода наименьших квадратов, претерпевают разрывы на границах подинтервалов разбиения аргумента сигнала. Рассмотрим указанные методы опре-

деления аппроксимирующих полиномиальных спектров.

Интервал определения сигнала $[0, T]$ разбивается на m одинаковых отрезков длиной $h=T/m$ каждый. На полученной решетке аргумента вводится система блочно-импульсных базисных функций:

$$v_i(t) = \sigma(t - (i-1)h) - \sigma(t - ih), \quad i = 1, \dots, m, \quad (2)$$

где $\sigma(t)$ — функция единичного скачка, определяемая как

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

Сигнал $x(t)$ восстанавливается в виде кусочно-постоянной аппроксимации:

$$\tilde{x}(t) = \sum_{i=1}^m X(i)v_i(t), \quad (3)$$

где $X(i)$ — блочно-импульсный спектр сигнала, определяемый по формуле:

$$X(i) = \frac{1}{h} \int_{(i-1)h}^{ih} x(t) dt. \quad (4)$$

В работах Симак Л. А. [3] было предложено дополнить систему локально-импульсных функций (2) подсистемой блочно-треугольных функций, определяемых по формуле:

$$w_i(t) = \left(2t \frac{m}{T} + 1 - 2i \right) v_i(t), \quad i = 1, \dots, m. \quad (5)$$

Тогда аппроксимация сигнала будет иметь вид:

$$\tilde{x}(t) = \sum_{i=1}^m (X(i)v_i(t) + X'(i)w_i(t)), \quad (6)$$

а расчет коэффициентов *аппроксимирующего импульсного спектра* проводится по формуле (4) и

$$X'(i) = \frac{3}{h} \int_{(i-1)h}^{ih} x(t) \left(\frac{2t}{h} + 1 - 2i \right) dt. \quad (7)$$

Преобразования на основе аппроксимирующих импульсных спектров позволяют легко изменять величину интервала аргумента, на котором рассматривается решение и порядок m базисных функций, что дает возможность создавать эффективные аппаратные реализации. При этом точность вычислений существенно выше по сравнению с блочно-импульсными функциями.

В работе [4] был предложен интерполяционно-экстраполяционный метод, суть которого заключается в следующем. Рассмотрим фрагмент сигнала и его блочно-импульсную аппроксимацию (рис. 1).

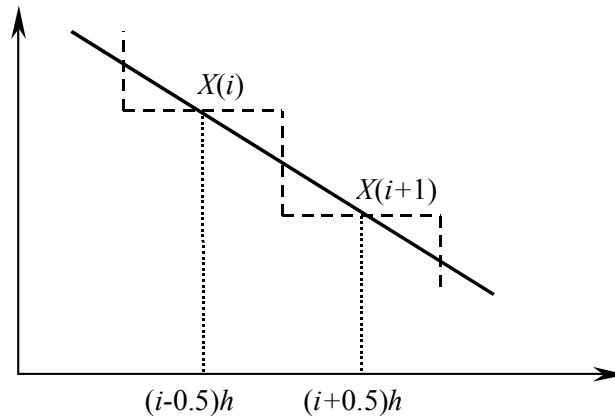


Рис. 1. Фрагмент сигнала (сплошная линия) и его блочно-импульсная аппроксимация (штриховая линия)/

При достаточно большом числе интервалов разбиения оси абсцисс m точки пересечения кривой с ее аппроксимацией находятся приблизительно посередине отрезков $(i-1)h$ и ih , т. е. имеют абсциссы $(i-0.5)h$. Это является следствием метода наименьших квадратов, а именно равенство площадей, ограниченных данной кривой и ее аппроксимацией. Зная элементы блочно-импульсного спектра, можно построить аппроксимацию сигнала на основе линейной интерполяции между серединами подинтервалов разбиения оси аргумента.

Уравнение для определения коэффициентов интерполирующей кривой будет иметь вид:

$$\alpha_i + \beta_i \cdot t = \begin{cases} X(i), & \text{если } t = (i-0.5)h, \\ X(i+1), & \text{если } t = (i+0.5)h. \end{cases} \quad (8)$$

Значения $X(i)$ и $X(i+1)$ рассчитываются по формуле (4). Разрешая (8) относительно α_i и β_i , получим:

$$\beta_i = \frac{X(i+1) - X(i)}{h},$$

$$\alpha_i = X(i) \cdot (i+0.5) - X(i+1) \cdot (i-0.5).$$

Уравнение аппроксимирующей прямой определяется формулой:

$$\tilde{x}(t) = \sum_{i=1}^{m-1} \left(X(i) \cdot (i+0.5) - X(i+1) \cdot (i-0.5) + \frac{X(i+1) - X(i)}{h} \cdot t \right). \quad (9)$$

Но аппроксимация (9) строится на интервале $\left[0 + \frac{1}{2}h, T - \frac{1}{2}h\right]$, по-

этому на интервалах $\left[0, \frac{1}{2}h\right]$ и $\left[T - \frac{1}{2}h, T\right]$ кривая экстраполируется. Тогда полное уравнение аппроксимирующей прямой будет иметь вид:

$$\begin{aligned} \tilde{x}(t) = & \sum_{i=1}^{m-1} \left(X(i) \cdot (i + 0.5) - X(i + 1) \cdot (i - 0.5) + \frac{X(i + 1) - X(i)}{h} \cdot t \right) + \\ & + \left(X(1) \cdot 1.5 - X(2) \cdot 0.5 + \frac{X(2) - X(1)}{h} t \right) \left(\sigma(t) - \sigma\left(t - \frac{h}{2}\right) \right) + \\ & + \left(X(m - 1)(m - 0.5) - X(m)(m - 1.5) + \frac{X(m) - X(m - 1)}{h} t \right) \times \\ & \times \left(\sigma(t - (m - 0.5)h) - \sigma(t - mh) \right). \end{aligned} \quad (10)$$

Так как в вышеизложенных методах аппроксимирующий спектр сигнала рассчитывается практически по одной формуле, то S-модель анализатора сигнала можно представить в следующем виде (рис. 2).

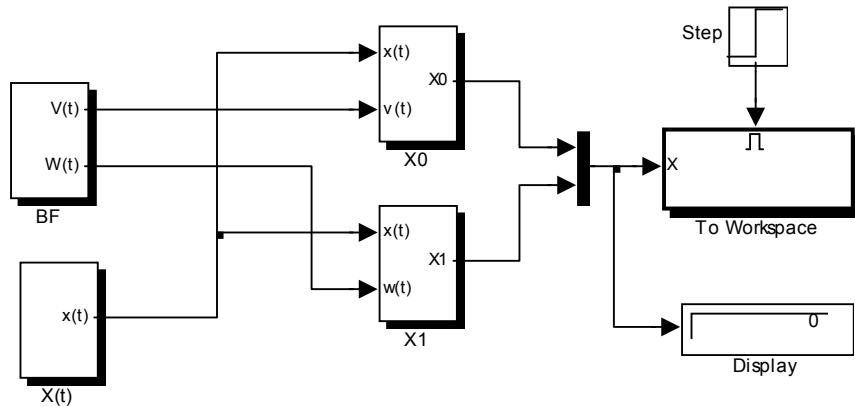


Рис. 2. Блок расчета коэффициентов спектра сигнала.

Здесь блок $X0$ определяет спектр сигнала $x(t)$ для метода блочно-импульсных функций и интерполяционно-экстраполяционного метода по формуле (4), а вместе с блоком $X1$ по формуле (7) — для метода аппроксимирующих импульсных спектров. При восстановлении сигнала по определенному методу выбирается необходимый вектор коэффициентов (либо $X0$, либо $X0$ и $X1$). Полученный спектр записывается в рабочую область для дальнейшей обработки.

При моделировании динамических систем нецелого порядка, операция дробного интегрирования сводится к операции интегрирования и заменяется операцией умножения на матрицу интегрирования соответствующего порядка.

Рассмотрим сигнал $x(t)$, который аппроксимируется обобщенным полиномом вида (3). Тогда операция дробного интегрирования по формуле (1) сводится к интегрированию образующих функций $v_i(t)$. Обозначим через $r = i - j$, $i, j = \overline{1, m}$, номер диагонали матрицы интегрирования P . Тогда после преобразований получим треугольную матрицу вида:

$$P(\beta, r) = \frac{h^\beta}{\Gamma(\beta + 2)} \begin{cases} 0, & r < 0 \\ 1, & r = 0 \\ [(r+1)^{\beta+1} - 2r^{\beta+1} + (r-1)^{\beta+1}], & r = 1, \dots, m-1. \end{cases} \quad (11)$$

S-модель, выполняющая расчет матрицы интегрирования, которая является теплицевой нижнетреугольной, приведена на рис. 3.

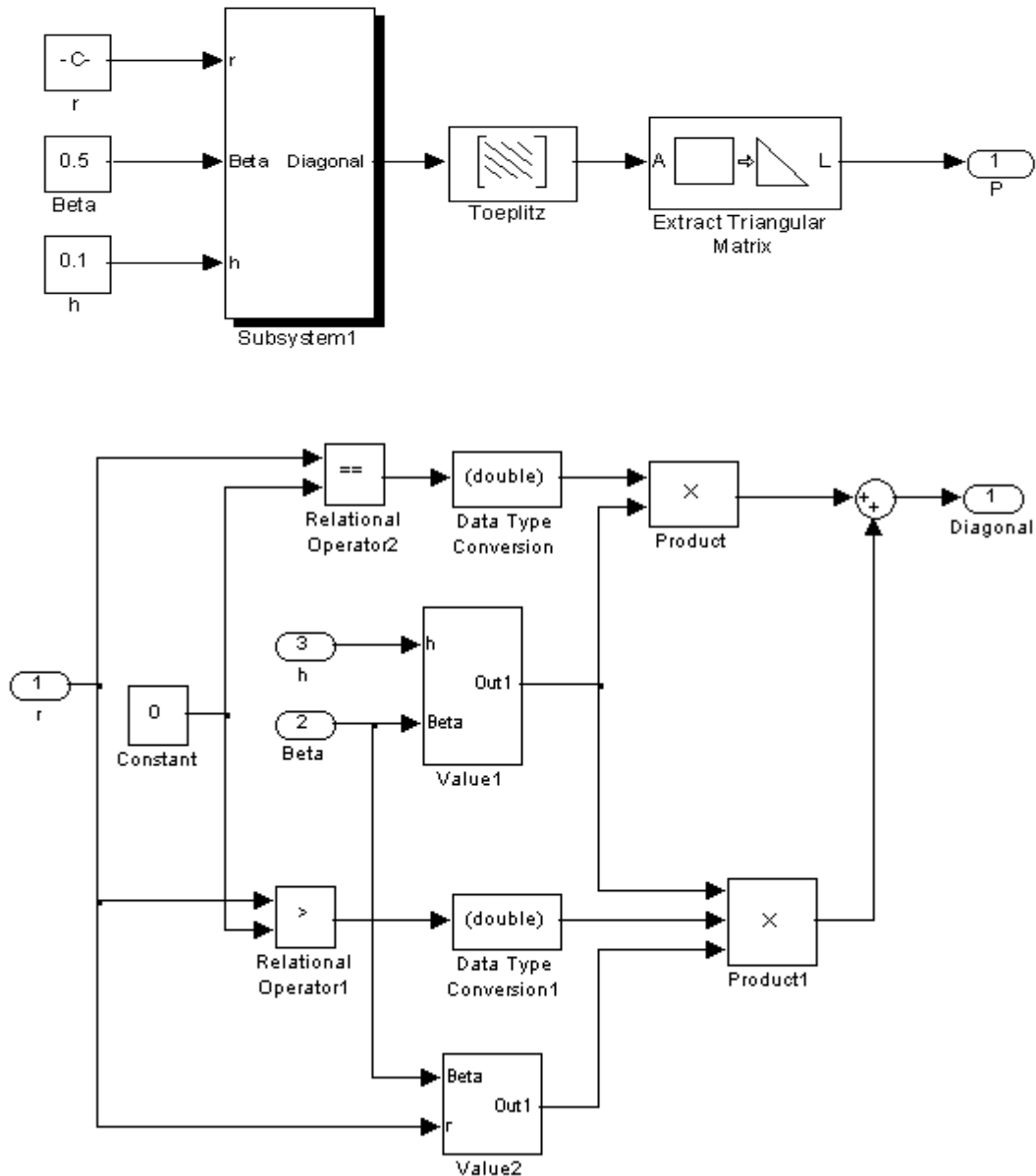


Рис. 3. Схема расчета матрицы интегрирования дробного порядка.

Моделирование динамической системы нецелого порядка в системе локально-импульсных функций по рассмотренным методам проводится по следующей схеме:

- интегро-дифференциальное уравнение, описывающее поведение системы, приводится к интегральному уравнению путем последовательного применения оператора интегрирования;
- по формулам (4) и (6) рассчитываются локально-импульсные спектры всех известных функций и констант, и по формуле (11) — матрицы интегрирования необходимого порядка;
- записывается преобразованное уравнение в операционной области, с учетом правил спектральной алгебры;
- в результате решения алгебраического уравнения в операционной области, находится блочно-импульсный спектр неизвестного сигнала;
- строится аппроксимация решения по выбранному методу (3), (5) или (10).

Проиллюстрируем изложенную схему на примере. Рассмотрим дробное дифференциальное уравнение вида [5]:

$$\begin{aligned} (D^\beta[x - x_0])(t) &= Ax(t) + f(t), & 0 \leq t \leq 1, \\ x(0) &= x_0, \end{aligned} \quad (12)$$

где $0 < \beta < 1$, f — заданная функция на интервале $[0, 1]$, $A \leq 0$ и x — неизвестная функция. Здесь $D^\beta x$ — дробная производная порядка β функции x по Риману-Лиувиллю. Заданный интервал изменения аргумента позволяет учесть начальные условия подобно обыкновенным дифференциальным уравнениям.

При $\beta = 1/2$ уравнение (12) описывает поведение модели затухания в механике, где x является перемещением, $A = -1/\xi$ (ξ — вязкость) и $f(t) = lN(t)/(EV\xi)$. Здесь l — длина исследуемого объекта, V — объем, E — модуль Юнга и $N(t)$ — внешняя сила.

Уравнение типа (12) описывает процессы диффузии, а также применяется в электродинамике, электрохимии, материаловедении, теории ультра-медленных процессов, теории специальных функций и др.

Рассмотрим случай, когда $f(t) = t^2 + 2t^{3/2}/\Gamma(2,5)$, $A = -1$, $\beta = 1/2$, начальные условия $x(0) = 0$. Точное решение уравнения $x(t) = t^2$. Проинтегрируем уравнение и с учетом нулевых начальных условий, перейдем к уравнению в операционной области:

$$\tilde{X}(E + P^{(1/2)}) = P^{(1/2)}F,$$

где E — единичная матрица порядка m ; $P^{(1/2)}$ — матрица интегрирования порядка $1/2$; F — спектр интеграла правой части уравнения.

На рис. 4 представлена S-модель решения дифференциального уравнения в операционной области (верхняя часть). Найденный спектр решения подается на входы блоков, восстанавливающих сигнал, в частности, для блочно-импульсной аппроксимации и интерполяционно-экстраполяционного метода. На рис. 5. показаны результаты работы моде-

ли — аппроксимации, точное решение и соответствующие ошибки аппроксимации.

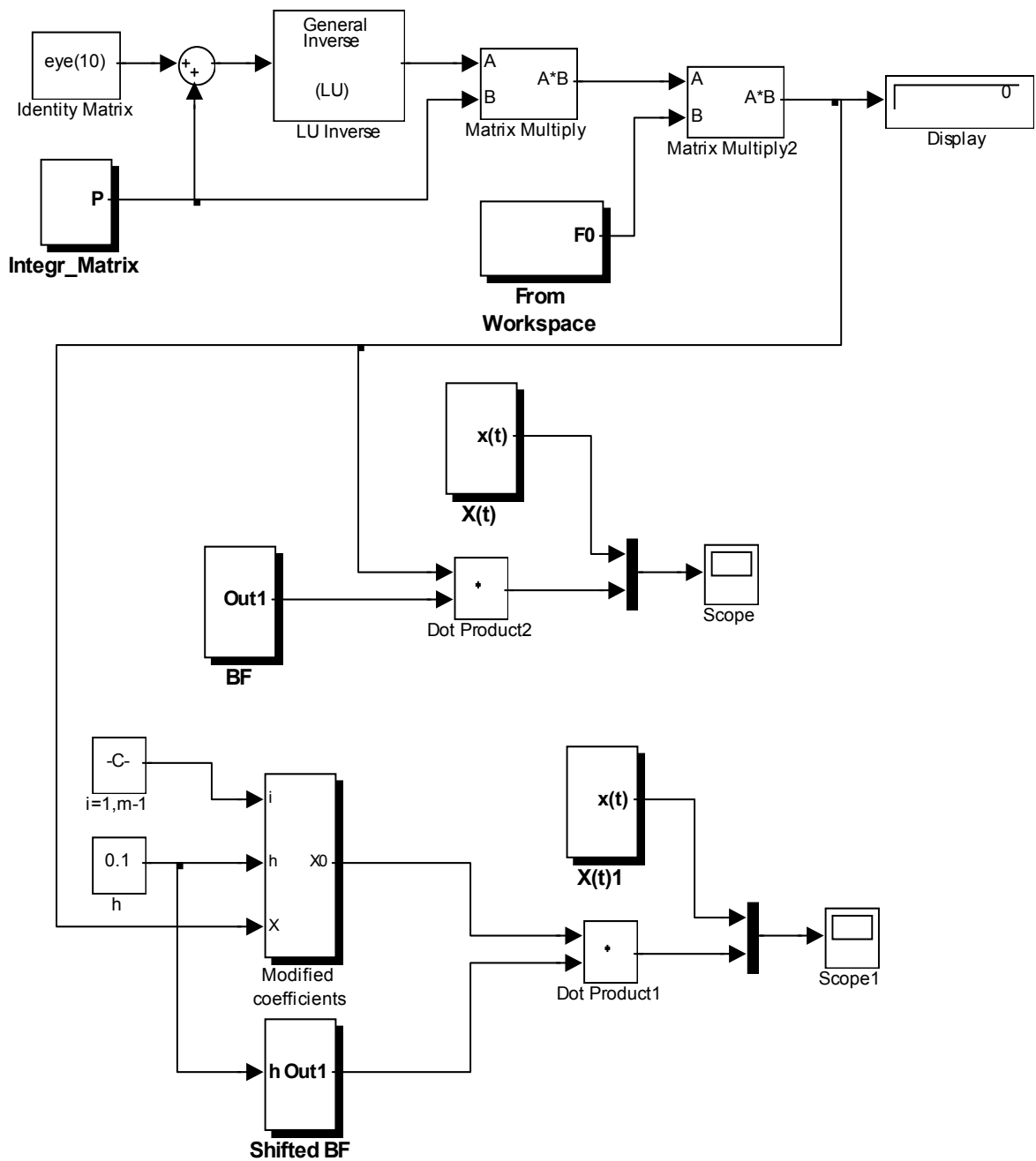
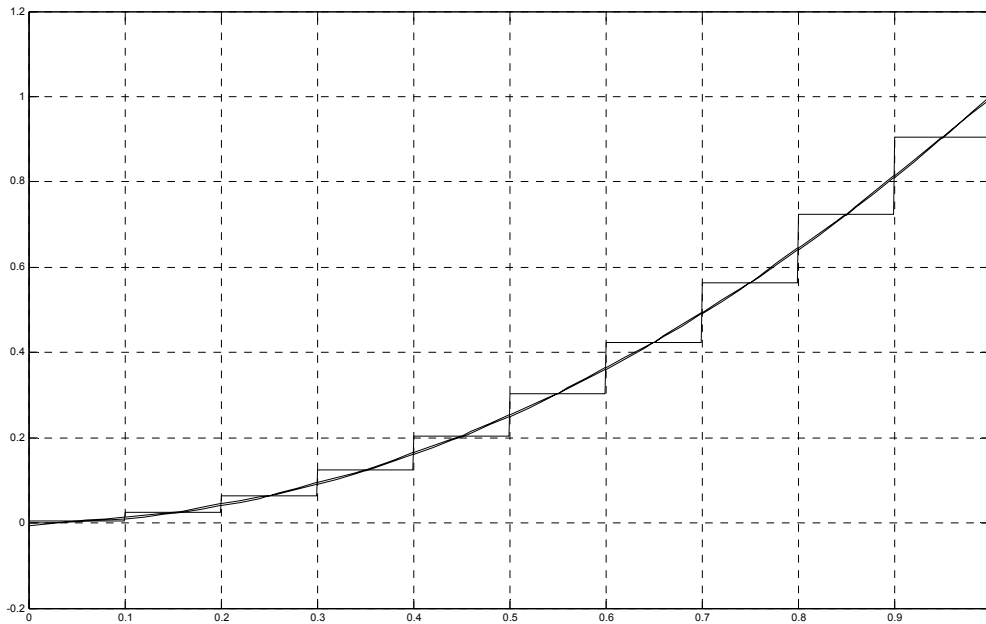
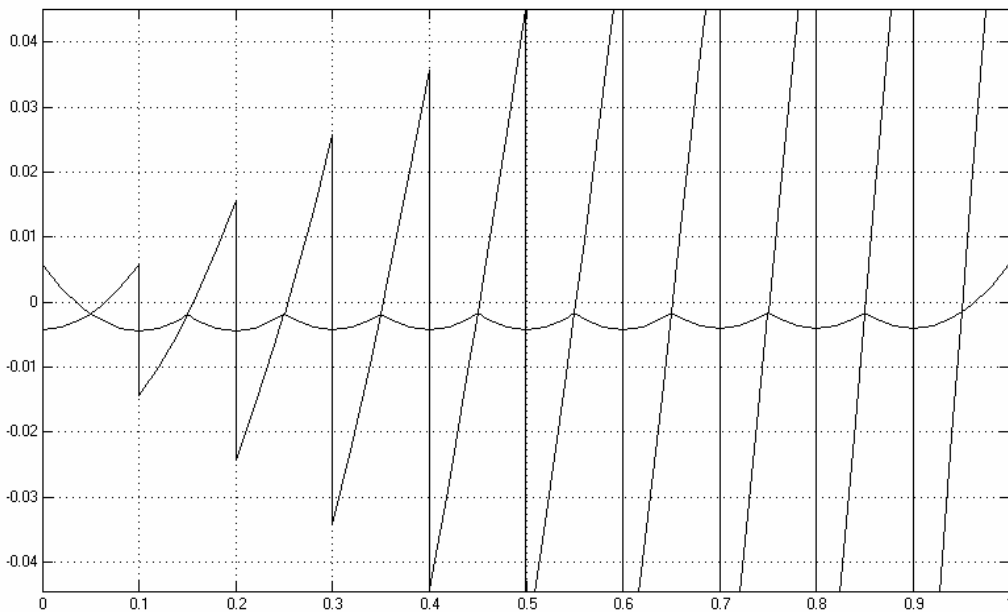


Рис. 4. S-модель решения дифференциального уравнения нецелого порядка.



а)



б)

Рис. 5. а) Точное решение уравнения, блочно-импульсная аппроксимация и аппроксимация по интерполяционно-экстраполяционному методу; б) соответствующие ошибки аппроксимации.

Операционный метод моделирования позволяет получать алгебраические аналоги динамических систем целого и дробного порядков, которые допускают эффективные реализации метода блочно-импульсных спектров. В свою очередь интерполяционно-экстраполяционный метод повышает точность блочно-импульсной аппроксимации без увеличения вычислительных затрат.

Литература

1. *Oldham K. B., Spanier O.* The fractional calculus.— New York&London: Academic Press, 1974.— 234 p.
2. *Васильев В. В., Грездов Г. И., Симак Л. А. и др.* Моделирование динамических систем: Аспекты мониторинга и обработки сигналов.— К.: НАН Украины, 2002.— 344 с.
3. *Симак Л. А.* Аппроксимирующие импульсные спектры в приложении к дробно дифференциальному анализу.— Киев, 1989.— 56 с. (Препр. / АН УССР. Ин-т проблем моделирования в энергетике, 89–8).
4. *Васильев В. В., Симак Л. А., Тодорова А. М.* Интерполяционно-экстраполяционный метод цифровой обработки сигналов на основе смещенных систем базисных функций // Моделювання та інформаційні технології.— 2004.— Вип.13.— С.13–23.
5. *Diethelm K.* An algorithm for the numerical solution of differential equations of fractional order // Electronic transactions on numerical analysis.— 1997.— V.5.— №5.— P.1–6.

УДК 519.6

ИМИТАЦИОННАЯ МОДЕЛЬ ЛЕСНОЙ ОТРАСЛИ ПРОИЗВОДСТВЕННО–ТЕХНОЛОГИЧЕСКОГО СЕКТОРА ЭКОНОМИКИ

Васильева И. Л.

*Пермский государственный университет, Пермь,
e-mail: irka17@hotmail.com*

В последнее время имитационное моделирование получило большое развитие и становится все более широко используемым. В основном, его используют для создания модели и анализа сложных систем, для которых воспользоваться аналитическими средствами для решения и анализа не представляется возможным. Некоторые экономические проблемы можно формализовать с помощью математических моделей. Иногда возникают более сложные задачи, решение которых с помощью математических моделей не отвечает существу поставленной проблемы. Кроме того, самым главным преимуществом является то, что это стохастический подход, позволяющий учитывать влияние множественных случайных факторов. Этот метод позволяет построить наиболее адекватную модель сложной системы. С появлением ЭВМ одним из наиболее важных и полезных орудий анализа структуры сложных систем становится имитационное моделирование.

Имитационное моделирование применяется в любой отрасли науки. Первоначально имитационное моделирование использовалось в ходе реализации авиакосмической программы, но на самом деле область его применения гораздо обширнее. В настоящее время оно используется для исследования разнообразных систем в различных областях, в частности в коммерческой деятельности, экономике, маркетинге, в исследовании проблем городов, транспортных системах, в системе образования и здравоохранения, политике и обществоведении, в кадровой политике, а также в теории управления и исследования операций, в управлении промышленными предприятиями и организациями, в стратегическом анализе и др.

Остановимся немного подробнее на понятии, целях и преимуществах выбранного подхода.

Под имитационным моделированием будем понимать процесс конструирования модели реальной системы и постановки эксперимента на этой модели с целью понять поведение системы, либо оценить (в рамках ограничений, накладываемых некоторыми критериями или их совокупностью) различные стратегии, обеспечивающие функционирование данной системы. Вообще, имитационное моделирование применяется для иссле-

дования границ и структур системы с целью решения конкретной проблемы; определения и анализа критических элементов, компонент и точек в исследуемых системах и процессах; для синтеза и оценки предполагаемых решений; прогнозирования и планирования будущего развития исследуемых систем.

Цели имитационного моделирования следующие:

- описать поведение системы;
- построить теорию и гипотезы, которые могут объяснить наблюдаемое поведение;
- использовать это для предсказания будущего поведения системы, т. е. тех воздействий, которые могут быть вызваны изменениями в системе или способе ее функционирования.

Укажем несколько основных положений определения имитации.

1. Это численный метод, используемый в крайнем случае, когда аналитических способов для анализа модели нет. Для многих проблем экономики и теории управления предприятием имитация — единственный пригодный метод исследования.
2. Это эксперимент. Особое внимание следует обратить на его планирование и обработку результатов.
3. Хотя ЭВМ не является необходимым элементом проведения имитационного эксперимента с моделью экономической системы, она ускоряет процесс, уменьшая возможность ошибки (будем рассматривать только машинные эксперименты).
4. Машинная имитация позволяет исследовать поведение модели как в определенный момент времени, так и в течение длительного периода (статическая и динамическая имитация).
5. Большинство имитационных экспериментов с моделями экономических систем являются стохастической имитацией (содержит случайные функции времени). Если в модель включить случайные величины, то с помощью имитационного эксперимента можно сделать выводы о поведении системы в целом, основываясь на вероятностном распределении случайных величин (метод Монте-Карло).

Каковы же преимущества имитационного моделирования? Машинная имитация дает возможность экспериментировать с большим числом переменных, с большим числом правил принятия решений и с более сложными моделями, точнее описывающими реальное поведение систем. Можно выделить следующие преимущества:

1. Имитация позволяет исследовать сложные внутренние взаимодействия в рассматриваемой системе.
2. С помощью имитации можно изучать воздействие на функционирование системы некоторых информационных и органических изменений, а также изменений во внешней обстановке.

3. Детальное наблюдение имитируемой системы (знания, полученные при ее обработке) позволяет лучше ее понять и разработать предложения по улучшению.
4. Имитация сложных систем дает представление о том, какие переменные наиболее существенны и как они взаимодействуют.
5. Имитация служит для изучения новой ситуации, для проверки новых стратегий и правил принятия решений перед проведением эксперимента на реальной системе.
6. Для некоторых типов статистических моделей особо значима последовательность событий. Данные только об ожидаемых значениях и моментах могут быть недостаточны для описания процесса. Тогда единственный способ для получения информации — метод Монте-Карло.
7. Если имитационное моделирование применять в течение длительного периода, то есть возможность создавать модели с периодическим циклом. Также оно помогает при составлении прогнозов относительно возможного поведения системы в будущем.
8. Имитация типа Монте-Карло может служить и для проверки аналитических решений.
9. Имитация позволяет изучать динамические системы в реальном или приведенном времени.

Основной метод имитационного моделирования — метод статистических испытаний (метод Монте-Карло, о котором уже часто упоминалось выше). Вследствие этого и, кроме того, сложности моделей, одним из необходимых средств имитационного эксперимента является ЭВМ. Поэтому любая имитационная модель представляет собой более или менее сложный программный продукт. Как и любая другая программа, имитационная модель может быть разработана на любом универсальном языке программирования, кроме того, существует довольно много специальных языков программирования для разработки исследуемого подхода (например, такие как СЛАМ, GPSS, ФОРТРАН, СИМУЛА и пр.). Поэтому на пути разработчика модели возникают следующие проблемы: требуется знание не только области исследования, но и хорошее знание языков программирования, причем на достаточно высоком уровне. Помимо этого, интерес представляет не столько эффективности системы, сколько ее поведение в той или иной ситуации, а для этого нужно изменять характеристики системы в достаточно широких пределах (причем не дожидаясь окончания текущего эксперимента), сохранять внесенные изменения до тех пор, пока они необходимы.

В настоящее время есть вычислительная система, позволяющая очень эффективно решать эти проблемы — пакет MATLAB, а точнее содержащийся в его составе инструмент визуального моделирования Simulink. Важным достоинством Simulink является то, что это открытая систе-

ма, т. е. она позволяет пользователю пополнять библиотеку блоков и дорабатывать ее в соответствии с их потребностями. Кроме того, в ходе моделирования пользователь имеет возможность наблюдать за процессами, происходящими в системе, для этого используются специальные «смотровые окна», входящие в состав библиотеки Simulink. Интересующие характеристики могут быть представлены как в числовом, так и графическом варианте, а еще можно включить в состав модели анимацию.

Для иллюстрации имитационного подхода и применения широких возможностей Simulink нами была рассмотрена модель лесоперерабатывающей отрасли производственного сектора. Целью было построение и анализ имитационной модели средствами инструмента Simulink для нахождения общей тенденции развития отрасли и для прогноза в будущем. Модель включает уравнения для производства пиломатериалов, для производства бумаги, целлюлозы, картона, уравнение, моделирующее количество рабочих, занятых в производстве и заработную плату. Коэффициенты модели получены с помощью метода наименьших квадратов (как уравнения линейной и нелинейной регрессии) по статистическим данным, опубликованным в открытой печати. Кроме того, была проведена оценка достоверности полученных уравнений. В результате мы получили следующую модель:

Переменные модели

$D(t)$ — годовой объем древесины

$\Pi(t)$ — годовой объем производства пиломатериалов

$\Psi(t)$ — годовой объем производства целлюлозы

$B(t)$ — годовой объем производства бумаги

$K(t)$ — годовой объем производства картона

$N(t)$ — количество рабочих, занятых в производстве

$E(t)$ — средняя заработная плата

$$\Pi(t) = 0,402D(t) - 0,007\Pi(t - 1) - 52100,47 \quad R = 0,888$$

$$B(t) = -0,047D(t) + 0,644B(t - 1) + 18233,67 \quad R = 0,793$$

$$K(t) = 0,012D(t) + 0,815K(t - 1) - 4339,24 \quad R = 0,983$$

$$\Psi(t) = 0,013D(t) + 0,837\Psi(t - 1) - 3705,37 \quad R = 0,979$$

$$N(t) = -12,661D(t)^2 + 122,51D(t) + 169,9 \quad R = 0,985$$

$$E(t) = 1,011N(t) + 0,996 \quad R = 0,938$$

R — выборочный коэффициент корреляции, позволяющий оценить достоверность полученных уравнений.

По описанной модели были просчитаны траектории значений эндогенных переменных с 1960–1985 годы (интервал — 5 лет). Были построены диаграммы реальных и имитируемых траекторий и проведен дисперсионный анализ полученных решений для оценки пригодности модели.

Хотя взятая нами модель довольно проста (многие факторы, влияющие на производство не учитываются вследствие нехватки статистических данных

по отрасли) и лишь частично описывает реальную экономическую систему, нашей целью было научиться применять технику имитационного моделирования, показать возможности пакета MATLAB для построения и анализа таких моделей. Взяв пока простейшую модель и отработав на ней технику проведения имитационного эксперимента, в будущем мы планируем применение полученного опыта к построению и исследованию более сложных моделей, более адекватных и описывающих реальные системы. Ведь техника имитационного моделирования довольно молода и очень перспективна и с помощью пакета MATLAB довольно просто и эффективно реализуется, кроме того он дает возможность очень хорошей визуализации результатов.

УДК 519.6

РЕШЕНИЕ ЗАДАЧ ГИДРАВЛИЧЕСКОГО РАСЧЕТА СИСТЕМ ЗАПРАВКИ РАКЕТЫ КОСМИЧЕСКОГО НАЗНАЧЕНИЯ СРЕДСТВАМИ ПАКЕТА SIMULINK

Веселов М. В.

Конструкторское бюро общего машиностроения им. В. П. Бармина, Москва,
e-mail: kbom@rosaviakosmos.ru

Система заправки ракеты космического назначения (РКН) компонентами топлива представляет собой гидравлическую систему, содержащую средства вытеснения и (или) насосной закачки компонента, основную магистраль и магистрали отдельных баков РКН с запорно-технологической арматурой, а также собственно баки [1].

На рис. 1 в качестве примера приведена упрощенная расчетная гидравлическая схема системы заправки насосом трех баков РКН, на которой обозначено: H_{i1} , H_{i2} — напоры в начале и в конце i -й магистрали соответственно; H_1 — напор на входе насоса; ξ_i — приведенные коэффициенты гидравлического сопротивления i -ой магистрали; Q_i — расходы в i -ой магистрали.

Принимая, что потери напора пропорциональны величине скоростного напора в магистрали, а напоры H_{12} , H_{22} и H_{32} постоянны, получим систему уравнений

$$\left. \begin{aligned} Q_0^2 \xi_0 &= H_{01} - H_{02} \\ Q_1^2 \xi_1 &= H_{11} - H_{12} \\ Q_2^2 \xi_2 &= H_{21} - H_{22} \\ Q_3^2 \xi_3 &= H_{31} - H_{32} \end{aligned} \right\} . \quad (1)$$

Из условия равенства напоров в узловой точке

$$H_{11} = H_{21} = H_{31} = H_{02},$$

равенства расходов

$$Q_0 = Q_1 + Q_2 + Q_3,$$

и заданной напорной характеристики насоса $H_n = f(Q_0)$, при которой

$$H_{01} = f(Q_0) + H_1,$$

получаем уравнение для Q_0

$$Q_0 = \sqrt{\frac{f(Q_0) + H_1 - H_{12} - Q_0^2 \xi_0}{\xi_1}} + \sqrt{\frac{f(Q_0) + H_1 - H_{22} - Q_0^2 \xi_0}{\xi_2}} + \sqrt{\frac{f(Q_0) + H_1 - H_{32} - Q_0^2 \xi_0}{\xi_3}}$$

Остальные расходы при известном Q_0 определяются из системы (1).

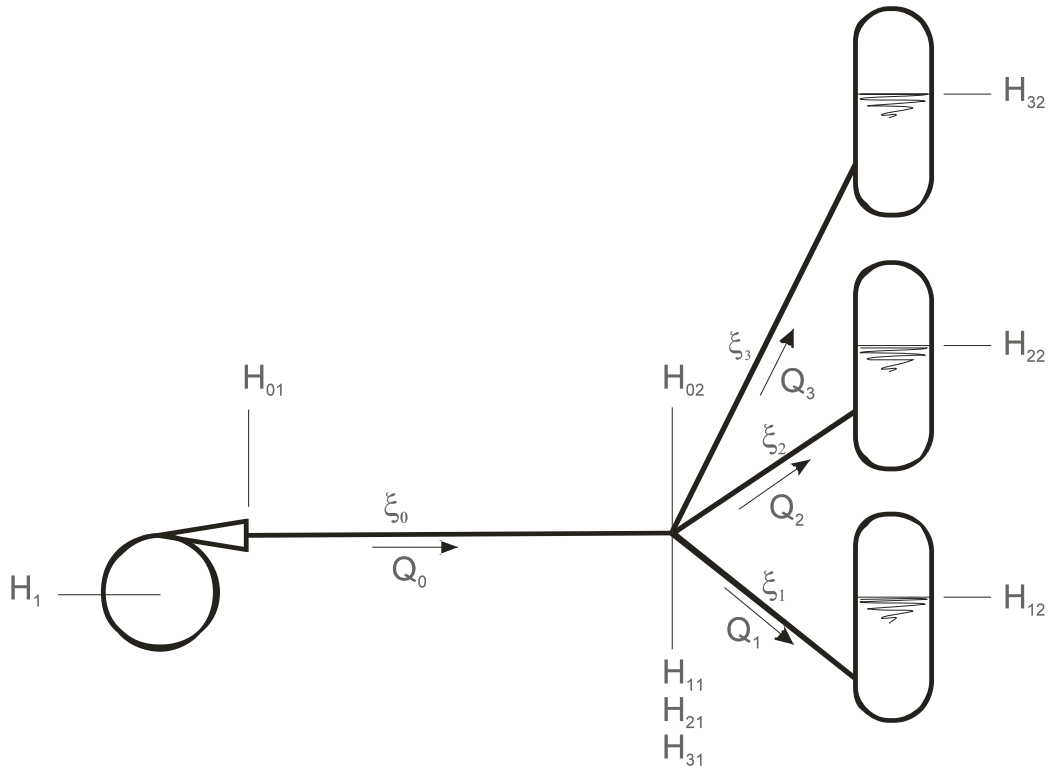


Рис. 1. Гидравлическая схема заправки насосом трех баков.

Рассмотренная выше задача, ввиду того, что H_{12} , H_{22} и H_{32} принимались постоянными, являлась стационарной.

На рис. 2 приведена блок-схема решения стационарной задачи для произвольного количества баков средствами пакета Simulink [2].

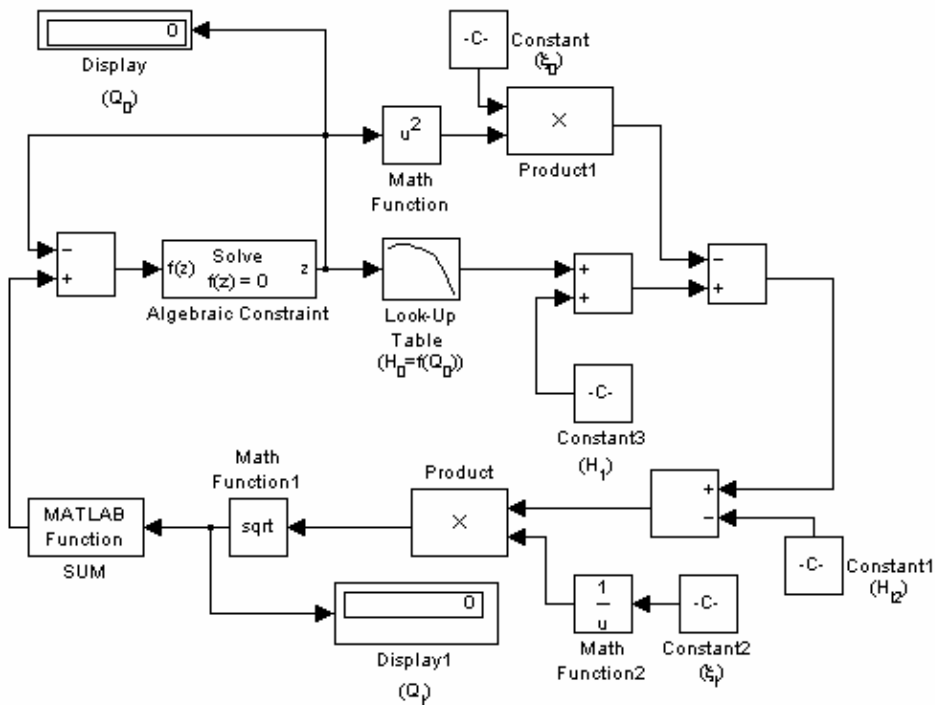


Рис. 2. Блок-схема задачи расчета системы заправки.

Если уровень заполнения бака зависит от времени, то задача является нестационарной.

Схема простой нестационарной задачи заправки насосом одного бака приведена на рис. 3.

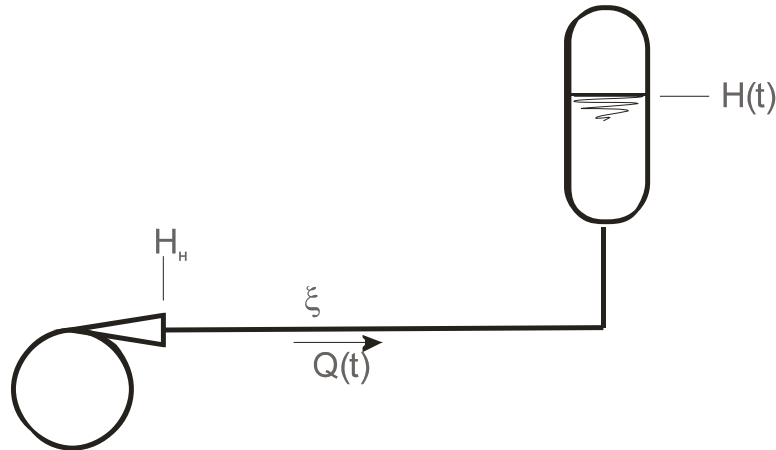


Рис. 3. Гидравлическая схема нестационарной задачи заправки бака насосом.

Дифференциальное уравнение для расхода по магистрали в этом случае имеет вид

$$\frac{1}{Q} \frac{df(Q)}{dt} - 2\xi \frac{dQ}{dt} = \frac{1}{F},$$

где F — площадь поперечного сечения бака.

Если $H_n = Const$ и при начальных условиях $t=0$; $H(0)=H_0$

$$Q(t) = \sqrt{\frac{H_n - H_0}{\xi} - \frac{t}{2\xi F}},$$

$$H(t) = H_n - \left(\sqrt{\frac{H_n - H_0}{\xi} - \frac{t}{2\xi F}} \right)^2 \xi.$$

На рис. 4 приведена блок-схема решения нестационарной задачи для произвольного количества баков средствами пакета Simulink.

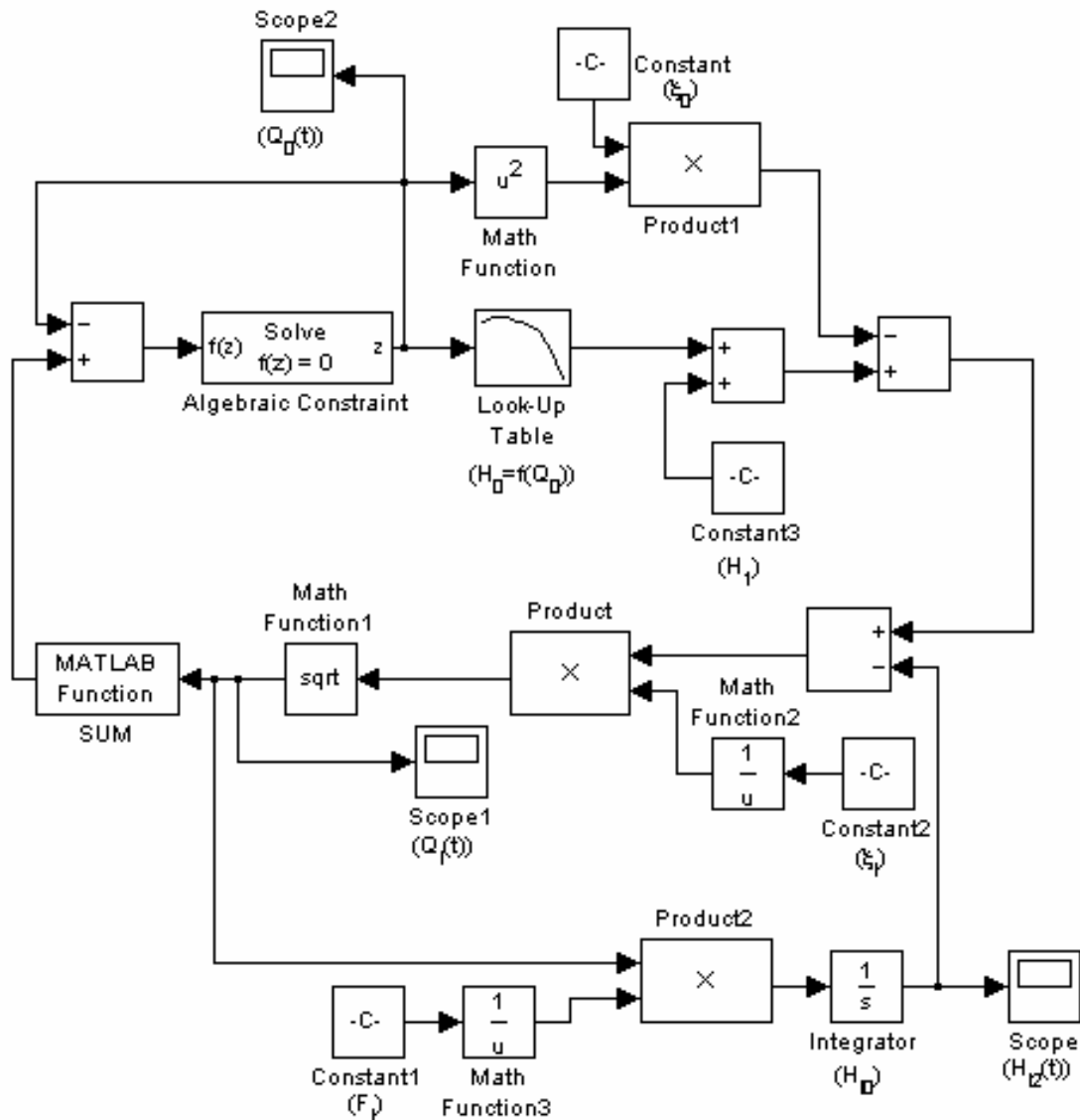


Рис. 4. Блок-схема задачи расчета нестационарного процесса заправки.

Приведенные на рис. 2 и 4 блок-схемы несложно дополнить и изменить при наличии двух насосов включенных параллельно или последовательно, при необходимости получения времени заправки отдельных баков и др.

Литература

1. *Космодром* / Под ред. А. П. Вольского.— М.: Воениздат, 1977.— 310 с.
2. *Дьяконов В. Simulink 4. Специальный справочник.*— СПб.: Питер, 2002.— 518 с.

УДК 519.6

ИНТЕРАКТИВНАЯ АППРОКСИМАЦИЯ ФУНКЦИЙ В СРЕДЕ MATLAB/SIMULINK

Выдолоб Г. М.

*Московский государственный институт электронной техники
(технический университет), Москва,
e-mail: gevy@mail.ru*

При проектировании вычислительных устройств и систем управления часто возникает задача нахождения аппроксимирующих функций для зависимостей, которые не могут быть вычислены с помощью простых арифметических операций: сложение, вычитание, умножение и деление. Для решения этой задачи исходная функция заменяется полиномом или рациональной дробью, параметры которых находятся в процессе минимизации ошибки приближения по выбранному критерию, например, среднеквадратичному или равномерному [1]. В некоторых случаях в качестве аппроксимирующей функции применяется совокупность полиномов (сплайн) или рациональных дробей.

Имеющиеся в среде MATLAB средства для приближения функций (команда **polyfit**, Curve Fitting Toolbox, пакет numapprox из ядра Maple, интегрированного в MATLAB) решают поставленную задачу в классическом виде и не позволяют управлять поведением аппроксимирующей функции на интервале приближения и его краях, а также вводить взаимную зависимость параметров аппроксимирующей функции.

Цель данной работы — создание в среде MATLAB инструмента для гибкой и наглядной аппроксимации функций полиномами или рациональными дробями с возможностью интерактивного управления ошибкой приближения. Это позволяет при поиске аппроксимирующих функций более полно учитывать особенности технической задачи и возможности практической реализации полученных приближений в приборах и системах.

В связи с тем, что полином можно рассматривать как частный случай рациональной дроби с полиномом знаменателя нулевого порядка, рассмотрим алгоритм нахождения параметров рационального приближения

$$R(x) = \frac{a_0 + a_1x + \dots + a_Nx^N}{b_0 + b_1x + \dots + b_Mx^M}$$

для функции $F(x)$ на интервале $x_{min} \leq x \leq x_{max}$. Заметим, что хотя бы один из коэффициентов знаменателя $b_0 \dots b_M$ отличен от нуля. Следовательно, можно сделать этот коэффициент равным единице путем деления на него всех коэффициентов числителя и знаменателя рациональной дроби. В

дальнейшем будем считать $b_0 = 1$.

1 этап. Интерполируем функцию $F(x)$ рациональной дробью $R(x)$ на интервале $x_{min} \leq x \leq x_{max}$ в $N+M+1$ точках

$$R(x_i) = F(x_i)$$

$$0 \leq i \leq N + M$$

Число точек интерполяции соответствует количеству независимых переменных $R(x)$. Значения узлов интерполяции выберем произвольно, например, равномерно. В результате интерполяции получим систему $N+M+1$ линейных уравнений

$$a_0 + \dots + a_N x_0^N - b_1 x_0 F(x_0) - \dots - b_M x_0^M F(x_0) = F(x_0);$$

.....

$$a_0 + \dots + a_N x_i^N - b_1 x_i F(x_i) - \dots - b_M x_i^M F(x_i) = F(x_i); \tag{1}$$

.....

$$a_0 + \dots + a_N x_{N+M}^N - b_1 x_{N+M} F(x_i) - \dots - b_M x_{N+M}^M F(x_i) = F(x_{N+M}).$$

Решением данной системы уравнений является набор параметров $a_0 \dots a_N$, $b_1 \dots b_M$ аппроксимирующей функции $R(x)$.

2 этап. Построим график ошибки приближения $\Delta(x)$ с весом $q(x)$

$$\Delta(x) = q(x)(R(x) - F(x)).$$

Для абсолютной ошибки $q(x)=1$, для относительной ошибки $q(x)=1/F(x)$.

3 этап. Проанализируем график ошибки приближения и изменим положение узлов интерполяции там, где необходимо изменить величину ошибки приближения. Сближение узлов интерполяции приводит к уменьшению ошибки аппроксимации между узлами, а удаление — к увеличению ошибки соответственно.

После этого переходим к этапу 1 с новыми узлами интерполяции. В результате повторения этапов 1,2 и 3 до достижения требуемого вида ошибки аппроксимации получаем итоговые значения параметров $a_0 \dots a_N$, $b_1 \dots b_M$ аппроксимирующей функции $R(x)$.

Выполнение этапов 1 и 2, связанных с большим объемом вычислений, возложено на компьютер. Этап 3 проводит человек. Такое распределение обязанностей делает процесс аппроксимации интерактивным и позволяет наиболее полно использовать возможности человека и компьютера.

Для реализации изложенного алгоритма в среде MATLAB/ Simulink разработана модель approximation.mdl, приведенная на рис. 1.

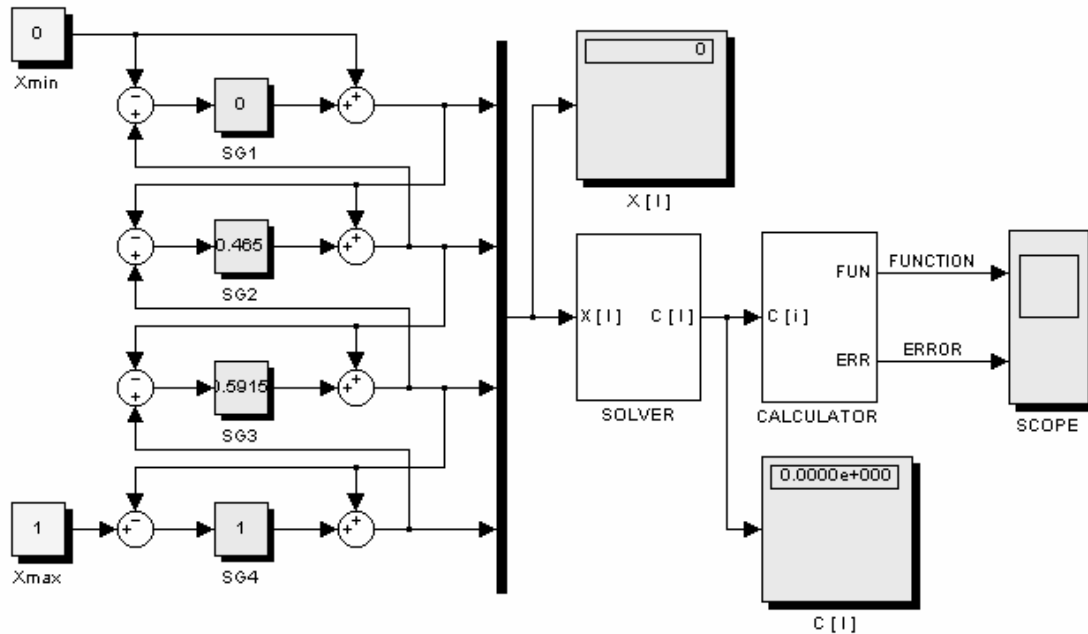


Рис.1. Модель интерактивной аппроксимации функций.

Данная модель интерполирует функцию

$$F(x) = \sin\left(\frac{\pi}{2} x\right);$$

$$0 \leq x \leq 1$$

рациональной дробью в виде отношения полиномов второго порядка с взаимосвязанными коэффициентами знаменателя

$$b_2 = -b_1.$$

Такие условия возникают при аппроксимации синусной и косинусной функций рациональными дробями с одинаковыми знаменателями.

Модель включает генератор четырех управляемых узлов аппроксимации, выполненных на основе стандартных блоков Slider Gain SG1 ... SG4, формирователь и решатель системы линейных уравнений (1) SOLVER, вычислитель аппроксимирующей функции и ошибки приближения CALCULATOR и дисплеи для отображения результатов моделирования: X[I] — вектор узлов аппроксимации, C[I] — вектор параметров аппроксимирующей функции, блок SCOPE отображает аппроксимирующую функцию и ошибку приближения. Установив бесконечное время моделирования и предел временного интервала в блоке Scope, равный единице, можно непрерывно наблюдать графики аппроксимирующей функции и ошибки приближения в процессе коррекции узлов аппроксимации. Изменяя положения узлов аппроксимации с помощью блоков Slider Gain SG1 ... SG4, добиваемся желаемой формы ошибки аппроксимации. На дисплеях X[I] и C[I] отображаются вектора узлов аппроксимации и параметров ап-

проксимирующей функции.

Реализация блоков SOLVER, CALCULATOR и блока ERROR изображена на рис. 2, 3 и 4.

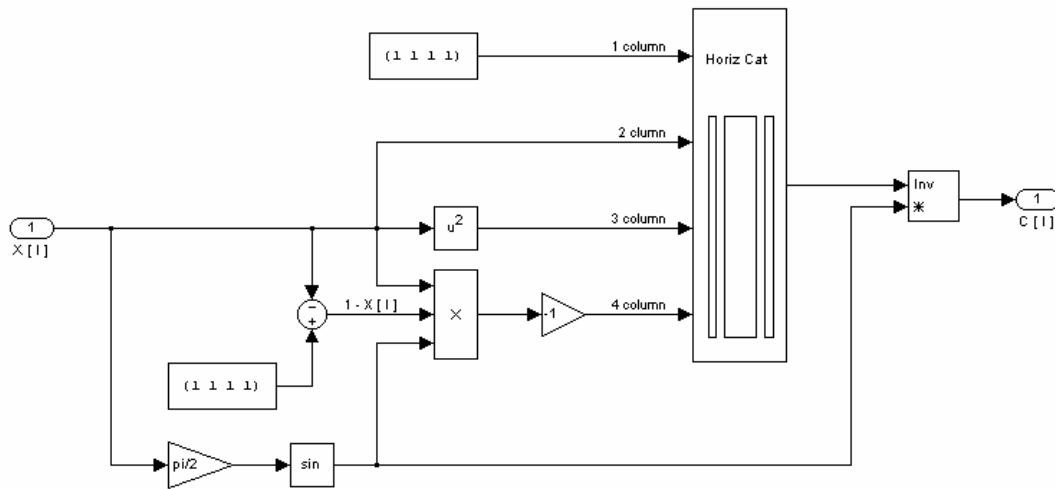


Рис.2. Модель блока SOLVER.

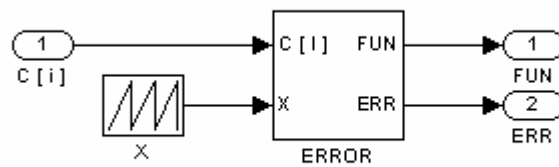


Рис.3. Модель блока CALCULATOR.

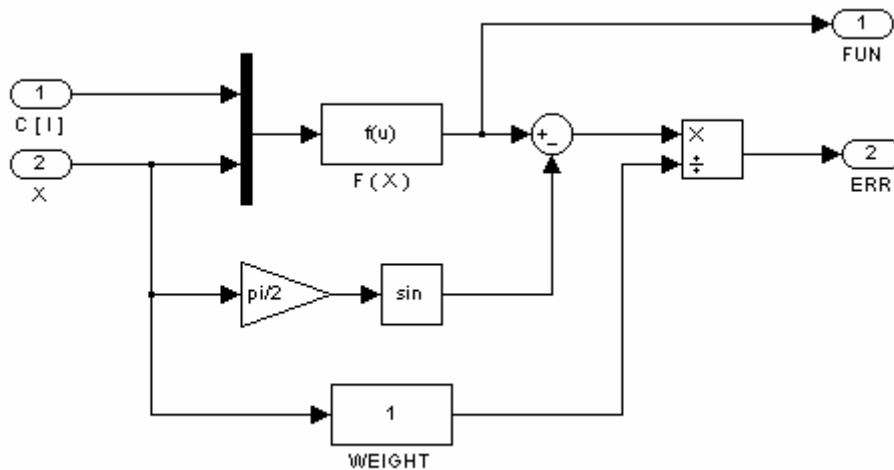


Рис.4. Модель блока ERROR.

В результате моделирования получено три варианта аппроксимирующей функции:

- при произвольном расположении узлов аппроксимации

$$R_1(x) = \frac{0.0015 + 1.5360x - 0.5360x^2}{1 - 0.4123x + 0.4123x^2}$$

$$\Delta_{\max} = 0.0015$$

- при расположении одного из узлов аппроксимации в начале диапазона изменения аргумента функции

$$R_1(x) = \frac{1.5486x - 0.5470x^2}{1 - 0.4014x + 0.4014x^2}$$

$$\Delta_{\max} = 0.0016$$

- при расположении двух узлов аппроксимации в начале и конце диапазона изменения аргумента функции

$$R_1(x) = \frac{1.5420x - 0.5420x^2}{1 - 0.4151x + 0.4151x^2}$$

$$\Delta_{\max} = 0.0021$$

Модель approximation.mdl является удобным инструментом для интерактивной аппроксимации функций в среде MATLAB/Simulink. Она наглядно показывает процесс поиска желаемого приближения и может быть легко настроена на особенности конкретной рациональной дроби или полинома.

Литература

1. *Калиткин Н. Н.* Численные методы.— М.: Наука, 1978.— 512 с.

УДК 621.313.012

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ АДАПТИВНОГО УПРАВЛЕНИЯ ДОБЫЧЕЙ НЕФТИ В ПАКЕТЕ MATLAB/SIMULINK

Ганеев А. Р., Шаньгин Е. С.

*Уфимский государственный авиационный технический университет, Уфа,
e-mail: nomad@tc.ugatu.ac.ru*

Современное состояние развития методов и технических средств добычи нефти из малодебитных высокообводненных скважин переживает драматический период, обусловленный, во-первых, тем, что все скважины на заключительной стадии эксплуатации переходят в разряд малодебитных по нефти, а технология добычи и технологическое оборудование остается прежним. Во-вторых, изменение режимов продуктивного пласта и добывающих скважин требует соответствующего изменения режимов добычи на основе повышения экономичности и снижения себестоимости добычи нефти, а технические средства, которые могут осуществить это, отсутствуют.

Наиболее экономичными с точки зрения затрат энергии являются наземные приводы штанговых скважинных насосных установок (ШСНУ) маятникового типа (осцилляторы) [1, 2].

В современной технике практически не используются колебательные системы маятникового типа (кроме часовых механизмов), обладающие всеми признаками маятника. Поэтому представляет интерес использование маятника в системе привода ШСНУ.

Рассмотрим привод насоса, в работе которого используется принцип действия крутильного маятника Максвелла (рис. 1).

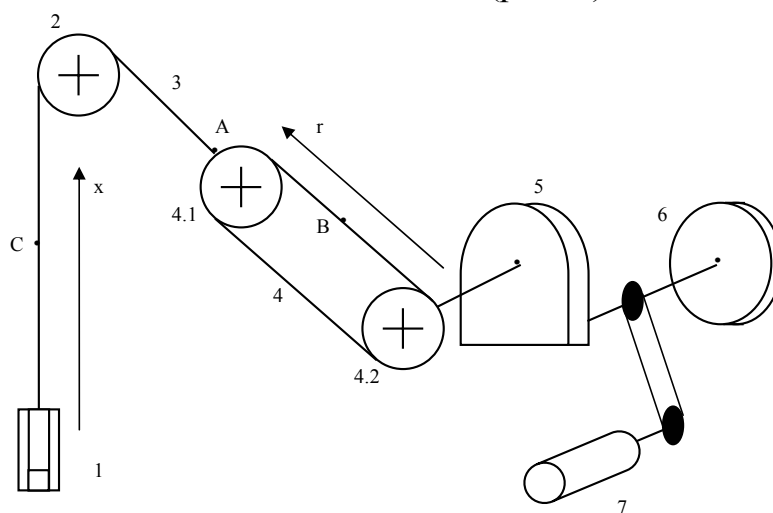


Рис. 1. Кинематическая схема наземного привода ШСНУ с крутильным маятником.

В начале рабочего цикла (колонна штанг опущена) электропривод 7 через редуктор 5 вращает маховик 6 и барабаны 4.1 и 4.2, соединенные цепью 4, и с помощью троса 3, соединенного с цепью 4 в точке А, поднимает колонну штанг в верхнее положение. В точке А происходит перемена направления движения штанг насоса.

Рабочий процесс начинается с опускания колонны штанг под действием гравитационных сил. Потенциальная энергия штанг при опускании преобразуется в кинетическую энергию вращения маховика 6. Кинетическая энергия вращения маховика, запасённая в период опускания колонны штанг, используется для поднятия груза в период хода вверх. Потери, неизбежные при движении всех элементов глубинно-насосной установки, не позволяют поднять штанги на исходную высоту только за счёт энергии маховика.

Когда кинетическая энергия маховика иссякает, электропривод 7 продолжает подъём штанг до достижения ими первоначального положения, тем самым происходит компенсация потерь в колебательной системе.

Для определения оптимального закона управления приводом ШСНУ разработана имитационная модель в математическом пакете фирмы MathWorks — MATLAB Simulink 4.0. Привод ШСНУ маятникового типа с электродвигателем представляет собой сложную нелинейную колебательную динамическую систему. Свободные колебания крутильного маятника описываются следующим дифференциальным уравнением:

$$\ddot{x} + \omega^2 x = 0, \quad (1)$$

где ω^2 — циклическая частота свободных колебаний маятника, x — координата перемещений маятника.

Циклическая частота свободных колебаний системы зависит от жесткости и массы маятника:

$$\omega^2 = c / M, \quad (2)$$

где c — жесткость колебательной системы; M — масса маятника.

Из (2) следует, что циклическая частота свободных колебаний системы изменяется обратно пропорционально квадратному корню массы системы.

Благодаря заполнению насоса жидкостью при движении его вверх подвижная масса увеличивается, а при движении вниз — уменьшается. Так как клапанная система насоса срабатывает в течение некоторого времени, вследствие чего нагрузка на подвижную систему насоса изменяется постепенно, то и изменение циклической частоты колебаний системы также происходит постепенно.

Приближенно оценить циклическую частоту системы позволяет следующее соотношение:

$$\omega = 2\pi f, \quad (3)$$

где f — частота колебаний системы.

Диапазон изменения частоты качаний ШСНУ — от 1 до 5 качаний в минуту. Поэтому диапазон изменения квадрата циклической частоты примерно $0,0167 \div 0,274 \text{ рад}^2/\text{с}^2$. При движении насоса вверх скорость положительна, масса насоса с поднимаемой жидкостью выше, следовательно, ω^2 ниже. При движении насоса вниз скорость отрицательна, масса насоса ниже, следовательно, ω^2 больше. Поскольку насос наполняется/освобождается от жидкости постепенно, то и изменение ω^2 происходит постепенно (предположительно по линейному закону).

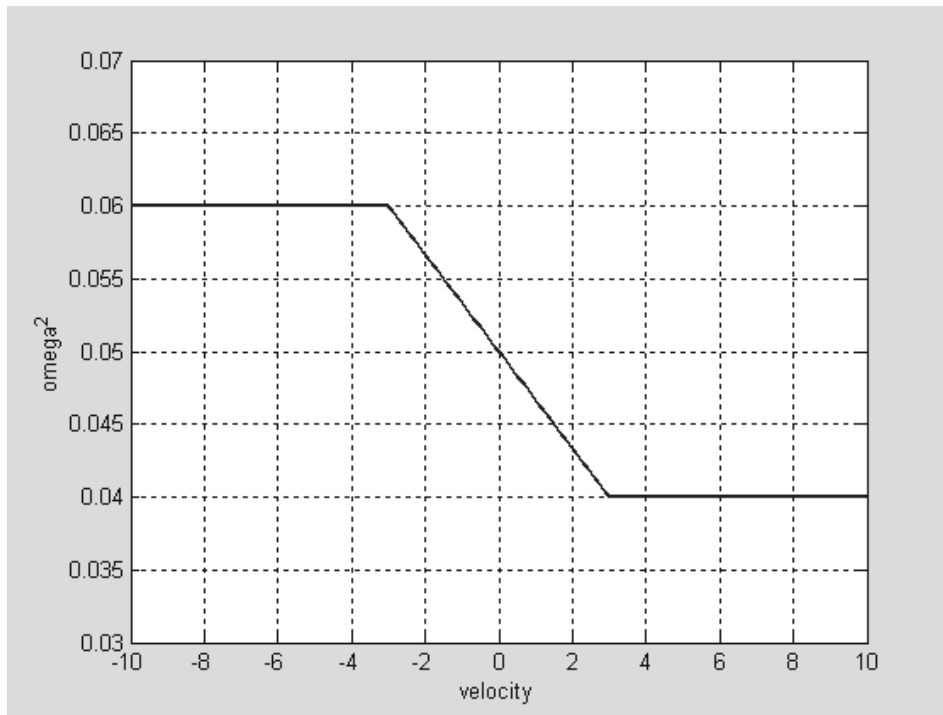


Рис. 2. График изменения ω^2 от скорости перемещения штанг.

На рис. 2 представлена зависимость $\omega^2 = f(\dot{x})$, где \dot{x} — скорость перемещения точки подвески штанг (ТПШ). В модели эта зависимость реализуется функцией `interp.m`.

Решением уравнения (1) являются синусоподобные затухающие колебания; искажения формы синусоиды происходят в результате изменения ω , а затухание в результате потерь энергии при подъеме жидкости насосом. На рис. 3 представлены затухающие свободные колебания станка-качалки для точки В (рис. 1) из положения +3м, при условии, что ось координат r направлена вдоль цепи 4.

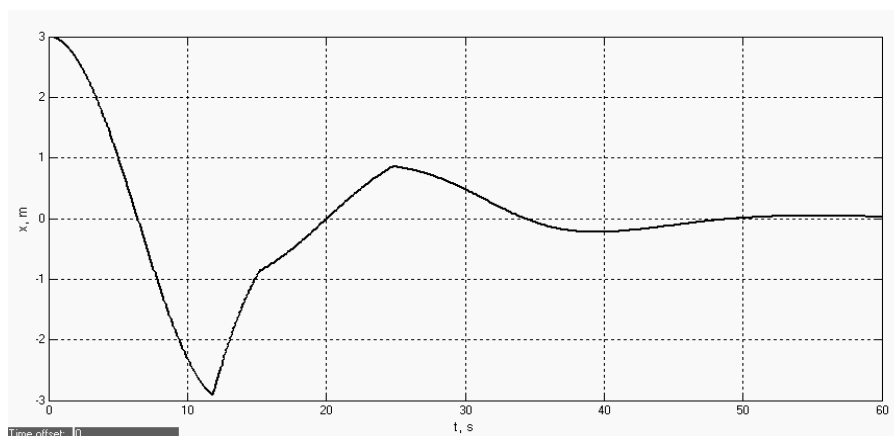


Рис. 3. Свободное движение системы из положения ТПШ +3м.

Изломы кривой объясняются кусочно-ломаным характером характеристики $\omega^2 = f(\dot{x})$, (рис. 1). Модель данной системы представлена на рис. 4. Изменение ω реализует блок MATLAB Function, который, в свою очередь, обращается к функции `multip.m`.

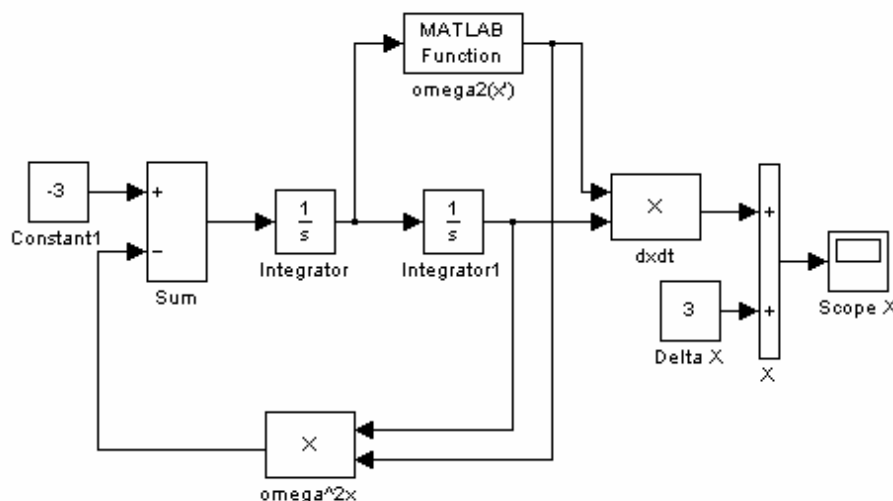


Рис. 4. Модель колебательной системы, совершающей свободные колебания.

В случае линейной системы схема (рис. 4) являлась бы просто решением уравнения (1), записанным на графическом языке Simulink — решение представляло бы собой незатухающую синусоиду (вся система в целом представляла бы собой колебательное консервативное звено).

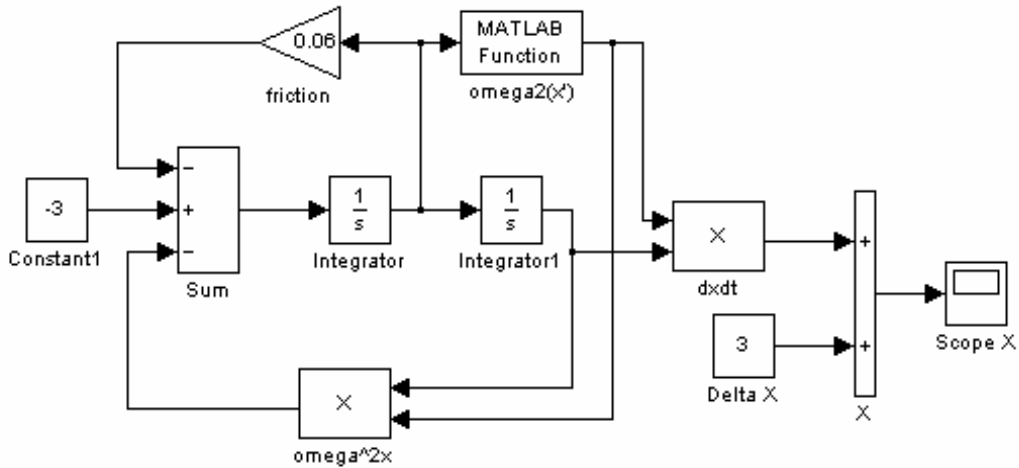


Рис. 5. Модель колебательной системы с затуханиями.

Введем в модель отрицательную обратную связь по скорости для моделирования диссипативных потерь в результате вязкого и жидкостного трения в виде усилительного звена с коэффициентом усиления 0.06. Схема такой системы в Simulink представлена на рис. 5.

На рис. 6 представлен график движения точки В (рис. 1) из положения +3м с диссипативными потерями на трение.

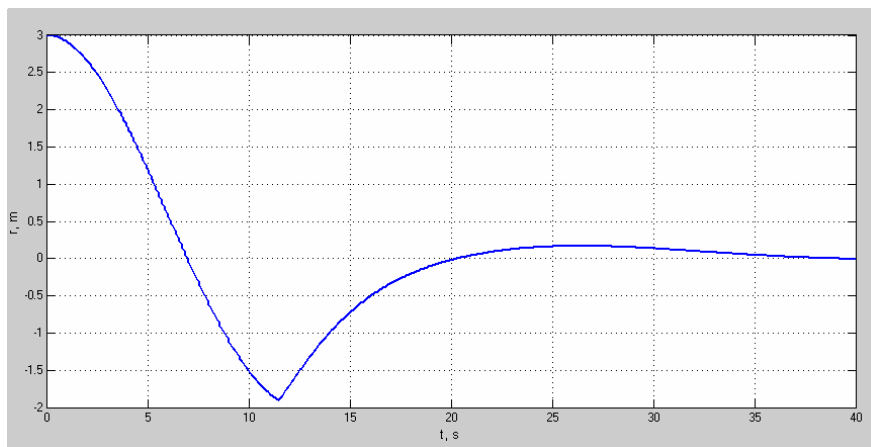


Рис. 6. Движение колебательной системы с диссипативными потерями.

Что касается закона движения ТПШ, то в результате крепления троса, на который подвешивается колонна штанг к цепи между блоками 4.1 и 4.2 (рис. 1), движения ТПШ вдоль оси координат, направленной вверх от ШСНУ, будут подчиняться следующему закону: когда точка крепления троса А движется по оси r , тогда $x = |r|$, когда А движется против направления оси r , тогда $x = -|r|$. При движении точки А по поверхности блока (примем его радиус за 0,3 м), x будет изменяться по синусоидальному закону (рис. 7).

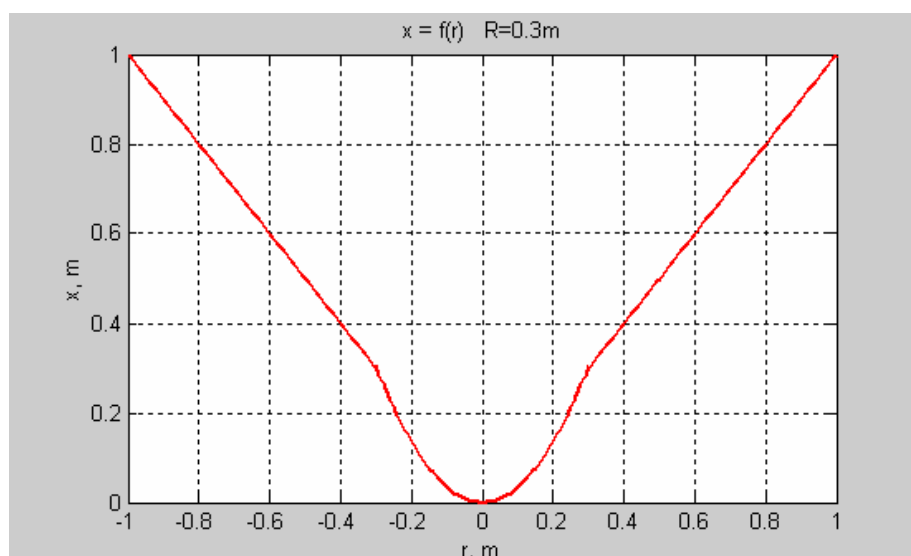


Рис. 7. Закон изменения перемещения ТПШ.

Указанную зависимость $x = f(r)$ реализует блок MATLAB Function, который использует функцию `wheel.m`. Движение из положения +3м ТПШ представлено на рис. 8. Схема модели в Simulink представлена на рис. 9.

Анализируя траекторию движения ТПШ (рис. 7), можно заключить, что в результате накопления кинетической энергии маховиком 6 (рис. 1) ТПШ преодолевает более половины хода насоса вверх по инерции. Таким образом, путем приложения дополнительной энергии в точке А (рис. 8) и принудительного перевода ТПШ в точку В можно достичь устойчивых рабочих колебаний насоса.

Регулируя наклон прямой восстановления АВ можно регулировать период качания насосной установки, и, таким образом, производительность насоса.

Следовательно, энергия электропривода будет употребляться только на компенсацию потерь энергии в результате жидкостного и вязкого трения, а также потерь энергии, затраченной на подъем скважинной жидкости.

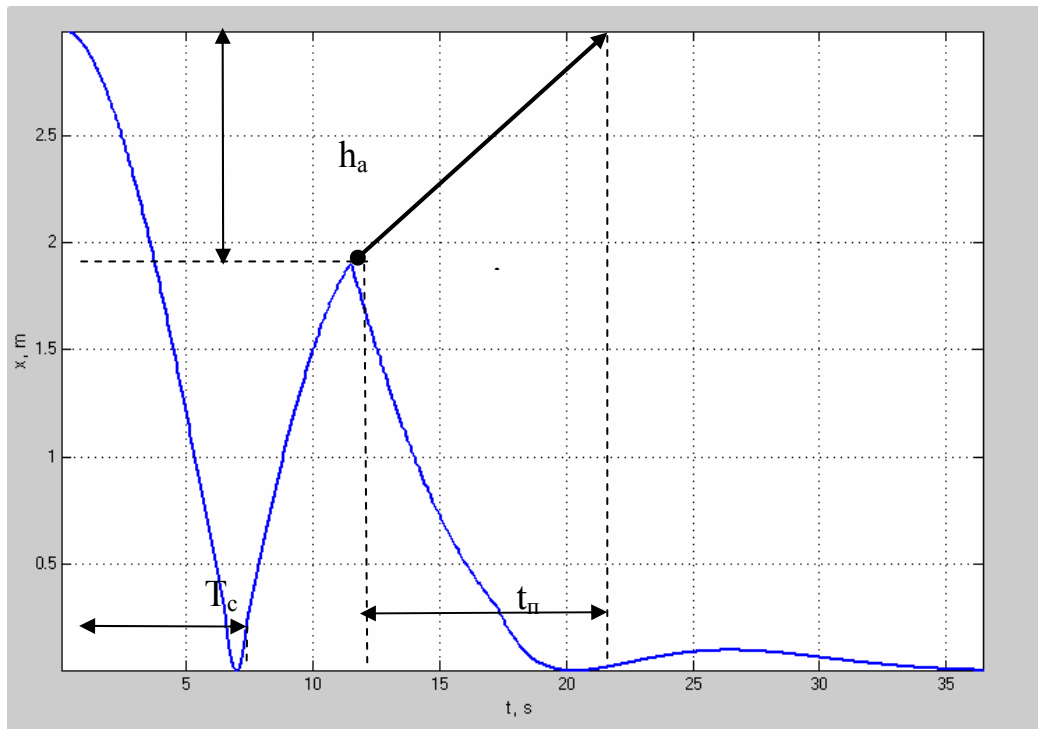


Рис. 8. Траектория движения ТПШ из положения +3м.

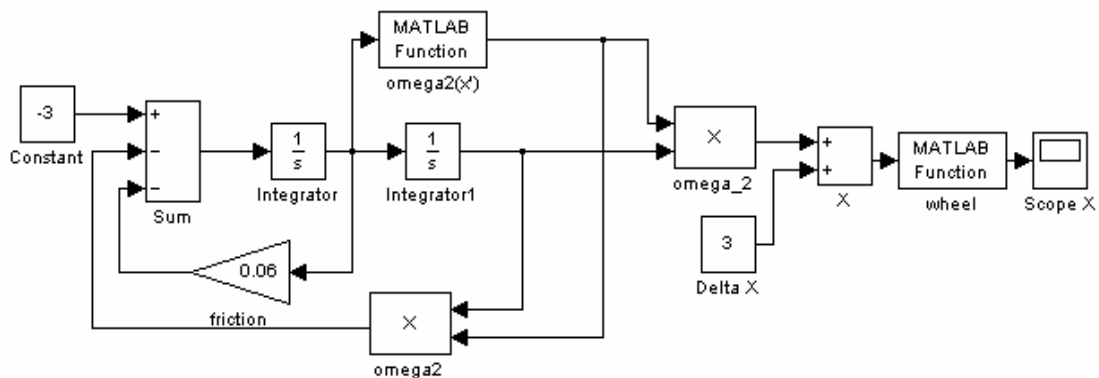


Рис. 9. Схема модели в Simulink.

Перемещение ШСНУ под действием электропривода можно упрощенно описать прямой, проходящей из точки А в точку В (рис. 8), уравнением в координатах x и t :

$$x = -\frac{x_b - x_a}{t_b - t_a} \cdot (t - t_b) + x_a. \quad (4)$$

Примем момент включения электродвигателя равным $4/3T_c$ (рис. 8).

Для реализации блока «Электропривод» создадим подсистему Simulink «Drive out». Поскольку в (4) входит время t , воспользуемся блоком Simulink «Clock» — на выходе которого формируется текущее время моделирования (рис. 10).

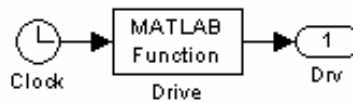


Рис. 10. Подсистема Simulink «Drive out».

Формирование сигнала по формуле (4) в моменты времени, отстоящие от начала подъема на $T_c/2$, реализуется функцией Drive.m.

После подъема электроприводом штанг в положение +3 м происходит повторение цикла спуска насоса и раскручивания маховика. При моделировании требуется «сброс» блоков интеграторов для «возвращения» модели в нулевые начальные условия (состояние +3 м) при достижении точки В (рис. 8). Для сброса интеграторов создадим подсистему «Reset intgs» (рис. 11), и зададим порты сброса для интеграторов Integrator и Integrator1 (рис. 12).



Рис. 11. Подсистема Simulink «Reset intgs».

Блок «Ground» используется для предотвращения предупреждений Simulink о свободном входе блока MATLAB Function. Сброс системы производится выработкой ненулевого выходного сигнала в зависимости от состояния флага — глобальной переменной bReset_model (функция reset_model.m).

Установка флага bReset_model производится внутри функции Drive.m.

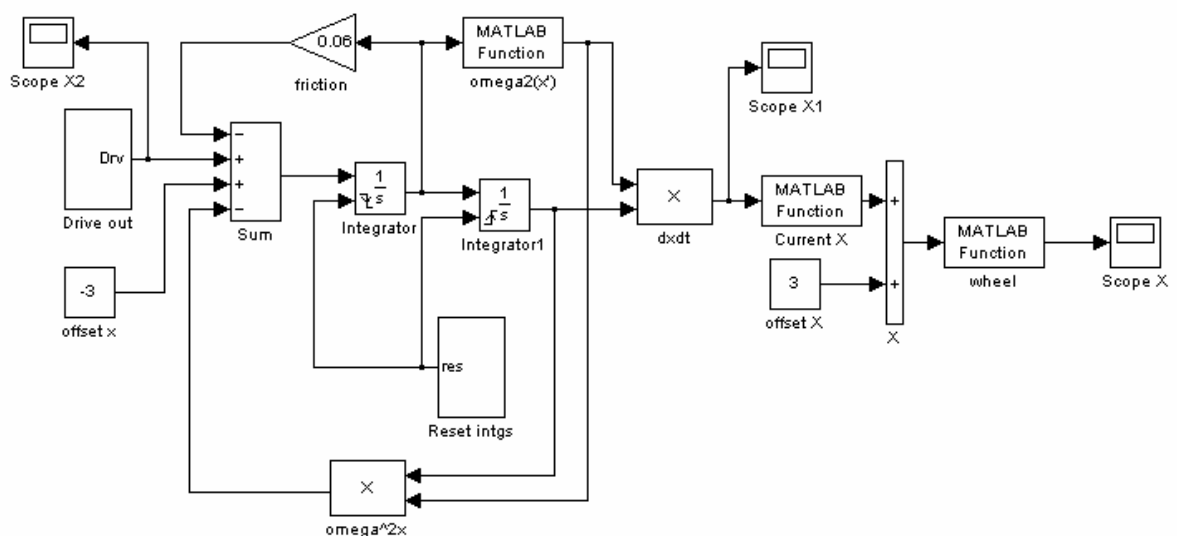


Рис. 12. Модель Simulink с подсистемами Drive out и Reset intgs.

Уравнение (4) реализуется в Drive.m следующим образом:

$$\text{res} = -((3 - \text{Local_max_z}) / \text{Delta_t}) * (t - (\text{T_counter} * \text{T_half})) + \text{Local_max_z}.$$

Электродвигатель включается периодически в определенные моменты времени, таким образом, на выходе подсистемы «Drive out» формируется трапецеидальный сигнал (рис. 13).

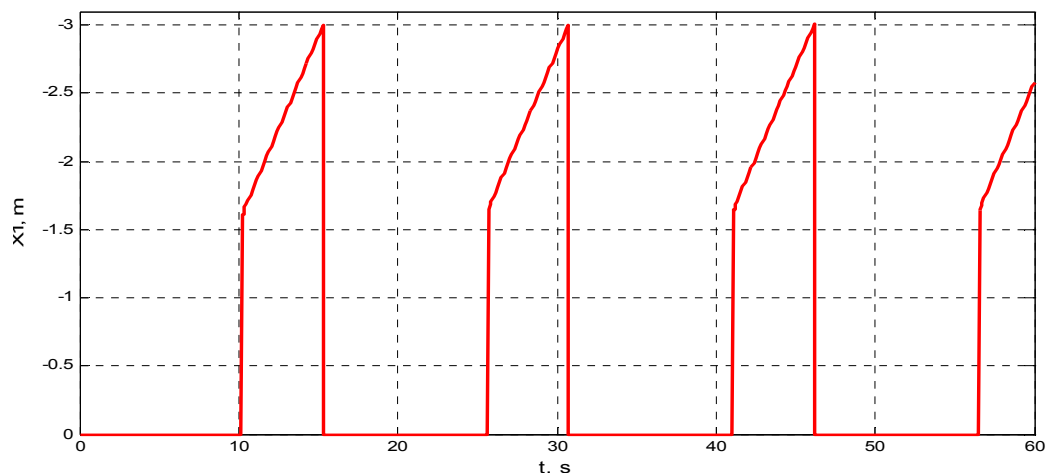


Рис. 13. График работы подсистемы Drive out от времени моделирования.

Рассмотрим некоторые параметры функции Drive.m.

Delta_t — время подъема электроприводом ТПШ в положение +3 м.

T_counter — множитель для определения времени включения двигателя. Инкрементируется каждый цикл работы двигателя переменной T_counter_increment.

T_half — время спуска ТПШ из положения +3 м (фиксируется автоматически).

Local_max_z — наивысшее положение ТПШ, достигнутое за счет запасенной энергии маховика (фиксируется автоматически).

Путем подбора установлены значения параметров, обеспечивающие работу ШСНУ для различных скоростей электропривода. Путем подбора параметров T_counter и T_counter_increment построены графики движения ТПШ для Delta_t = 5, 7, 10 с (рис. 14÷16).

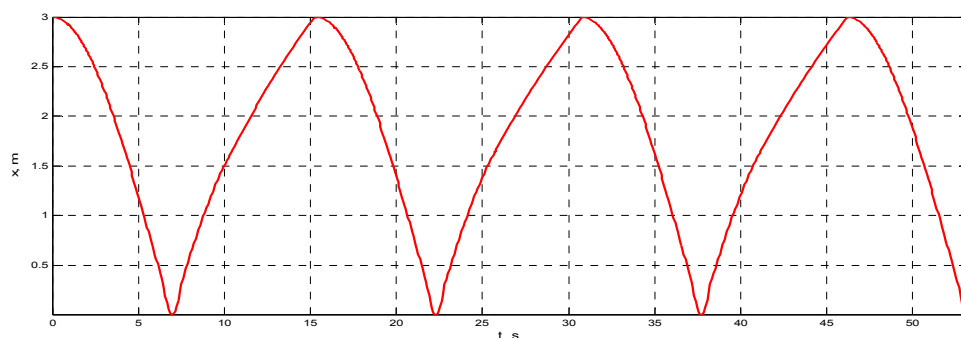


Рис. 14. Траектория движения ТПШ при Delta_t = 5с.

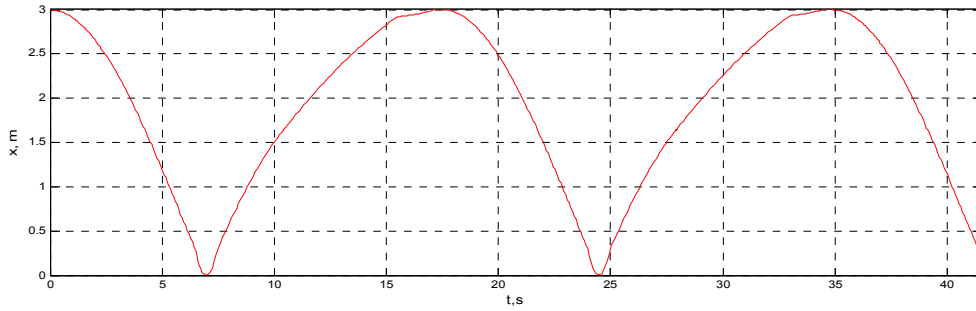


Рис. 15. Траектория движения ТПШ при $\Delta t = 7\text{c}$.

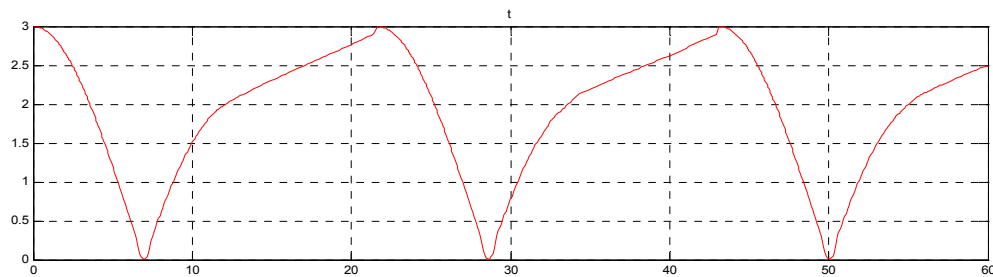


Рис. 16. Траектория движения ТПШ при $\Delta t = 10\text{c}$.

На основании полученных значений параметров T_{counter} и $T_{\text{counter_increment}}$ составим графики их зависимости от Δt (рис. 17).

Таким образом, при использовании зависимости, приведенной на рис. 17, появляется возможность регулировать период качаний ШСНУ в широких пределах, что позволяет создать автоматическую систему управления производительностью насоса.

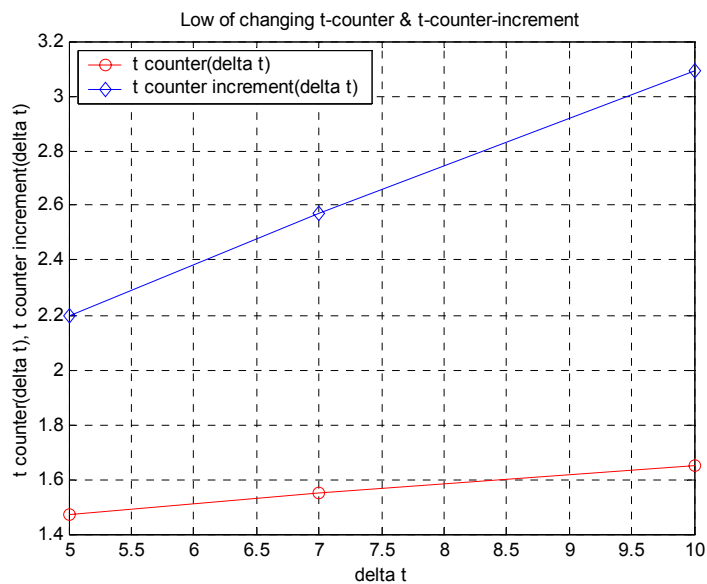


Рис. 17. Закон изменения $\Delta t = 10\text{c}$.

Положим, что производительность насоса Q прямо пропорциональна частоте качаний привода f :

$$f = \frac{1}{T_c + t_n}, \quad (5)$$

где T_c — константа для каждого отдельно взятого станка-качалки и насосного оборудования. Изменяя t_n (время подъема электроприводом штанг и насоса с жидкостью), можно производить регулирование f в пределах $1/T_c \div 0$ Гц.

На рис. 18 представлена схема системы автоматического управления (САУ) привода ШСНУ маятникового типа.

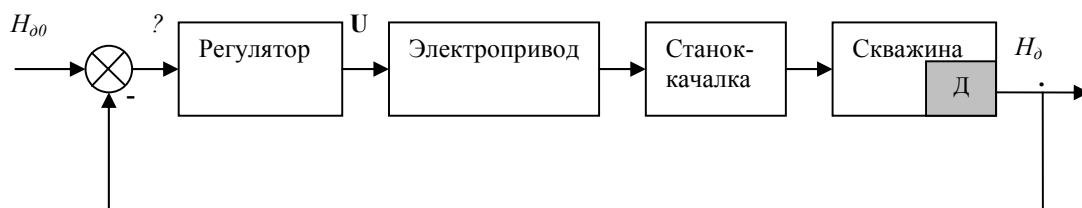


Рис. 18. САУ технологическими режимами ШСНУ:

H_{00} — заданный (расчетный) динамический уровень жидкости в скважине, ε — ошибка управления, U — управляющее воздействие регулятора (контроллера), Q — производительность насоса, H_0 — текущий динамический уровень жидкости в скважине.

Таким образом, используя возможности моделирования в пакете MATLAB Simulink, можно с достаточной для практического применения точностью провести подбор технологических режимов работы ШСНУ, обеспечив тем самым снижение энергозатрат и повышение эффективности нефтедобычи.

Возможность автоматизированного управления режимом ШСН в соответствии с текущим значением дебита продуктивной скважины позволяет по-новому организовать процесс нефтеизвлечения. На малодебитной по нефти высокообводненной стадии эксплуатации в скважине существенно изменяется характеристика среды — вместо заполненного нефтью ствола (характерного для высокодебитной стадии эксплуатации скважины) значительную высоту нижней части скважины занимает вода, в верхней части скапливается нефть. Изменившаяся характеристика внутрискважинной среды требует применения других технологических приемов нефтеизвлечения. Схема организации подъема нефти показана на рис. 19. Для исключения воды из добываемой скважинной жидкости прием насоса должен быть помещен в слой нефти в верхней части скважины, причем уровень приема насоса совпадает со статическим уровнем воды в скважине при отсутствии нефти.

Пусть в рассматриваемый момент времени скважина заполнена водой, а величина пластового давления постоянна — $P_{пл} = const$. Тогда статический уровень воды, зависящий от величины пластового давления, определяется выражением:

$$h_{см.в} = P_{пл} / \rho_в \cdot g = const,$$

где $\rho_в$ — плотность воды ($1,016 \text{ г/см}^3$); g — ускорение свободного падения ($9,8062 \text{ м/с}^2$).

При накоплении слоя нефти в верхней части скважины, высота которой h_n , необходимо определить положение нижнего уровня слоя нефти — границу раздела нефти и воды:

$$h_в = h_{см.в} - \frac{\rho_n}{\rho_в} \cdot h_n,$$

где ρ_n — плотность нефти.

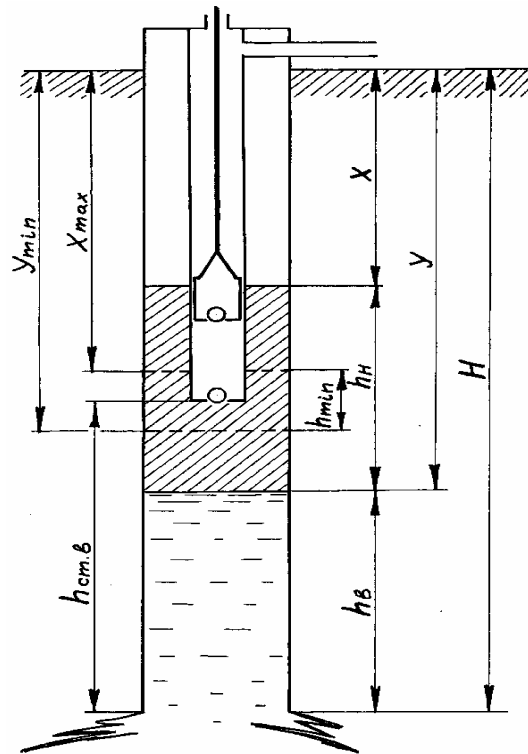


Рис. 19. Схема размещения глубинного насоса на поздней стадии эксплуатации скважин

Обозначим динамический уровень скважинной жидкости (верхнюю границу слоя нефти) через x , а нижнюю границу слоя нефти (границу раздела нефти и воды) — через y . Тогда $h_n = y - x$, $h_в = H - y$, а $\rho_n / \rho_в = k$, где H — глубина скважины (рис. 20). Полная глубина скважины будет определяться выражением:

$$H = h_{см.в} - ky + kx + y.$$

Отсюда определяем функциональную зависимость положения границы раздела нефти и воды y от положения динамического уровня x $y = f(x)$:

$$y = \frac{H - h_{cm.в} - kx}{1 - k}.$$

Зная эту зависимость (рис. 21), для целей управления режимом насоса можно ограничиться измерением только динамического уровня, хотя созданный для этой цели прибор может измерять обе границы нефтяного слоя.

В начале процесса нефтеизвлечения, когда слой нефти достаточно велик, насос работает в режиме интенсивной откачки. По мере уменьшения слоя нефти верхняя и нижняя границы нефтяного слоя приближаются с двух сторон к приему насоса. Для того чтобы насос работал эффективно, необходимо ограничить уменьшение слоя нефти уровнем «страхового слоя» — $h_{min} = y_{min} - x_{max}$ (рис. 19).

В случае уменьшения слоя нефти до нуля, уровень воды поднимется до уровня приема насоса. График изменения уровней жидкостей в скважине представлен на рис. 21.

Учитывая предложенную схему нефтеизвлечения, автоматизированная система управления, показанная на рис. 18, приобретает вид, представленный на рис. 22.

При изменении динамического уровня и связанного с ним функциональной зависимостью нижнего уровня слоя нефти (границы раздела нефти и воды) с помощью САУ производится изменение подачи насоса Q :

$$\frac{dx}{dt} = +\xi_1 Q;$$

$$\frac{dy}{dt} = -\xi_2 \frac{dx}{dt} + \xi_3 P_{пл},$$

где ξ_1, ξ_2, ξ_3 — коэффициенты, учитывающие влияние подачи насоса, взаимовлияние x и y и давления пласта соответственно на уровень слоя нефти.

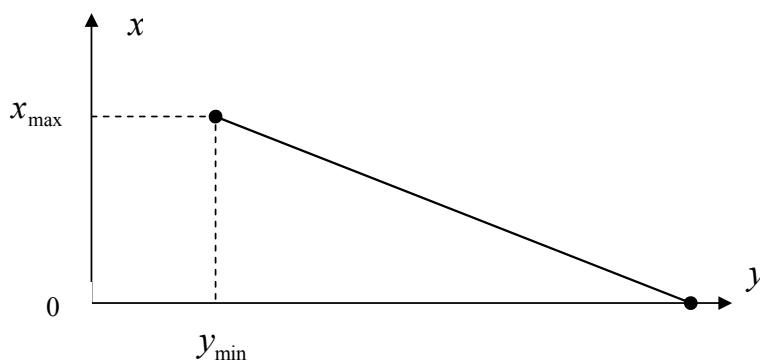


Рис. 20. Функциональная зависимость $y = f(x)$ уровня границы раздела нефти и воды от динамического уровня скважинной жидкости.

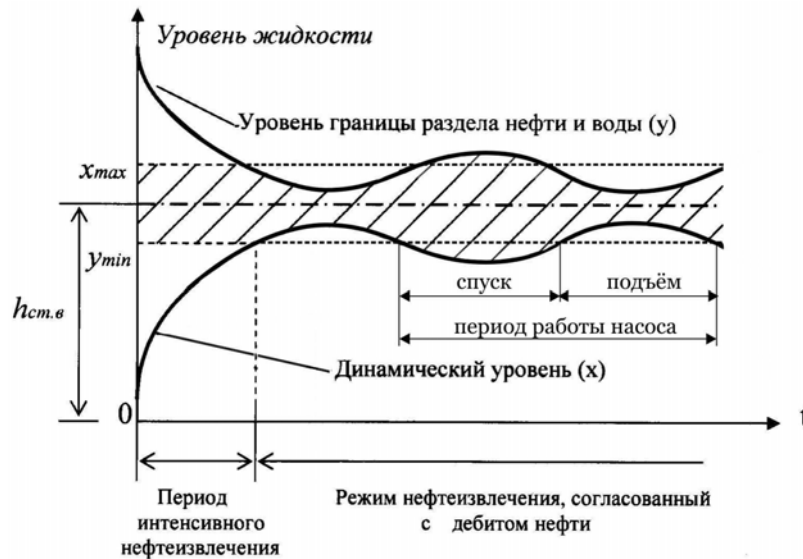


Рис. 21. График изменения уровней жидкостей в скважине на поздней стадии эксплуатации продуктивного пласта.

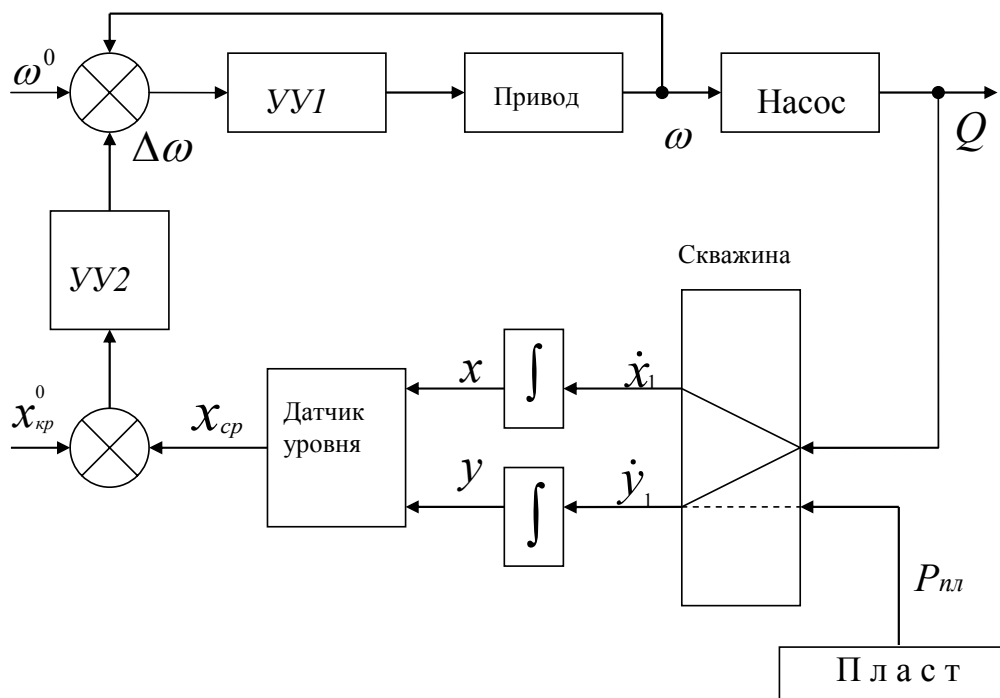


Рис. 22. Система автоматического управления технологическим режимом ШСНУ в соответствии с текущим значением дебита

На рис. 23 представлена схема САУ добычей нефти в MATLAB Simulink.

Уравнение для уровня жидкости в скважине:

$$x = x_0 + \frac{Q \cdot t}{S},$$

где Q — производительность насоса: $Q = W \cdot h \cdot k_{зан} \cdot S_{пл}$; W — число двойных ходов привода насоса: $W = 1 / (T_{собст} + T_{восст})$; $T_{собст}$ — период собственных

колебаний привода; $T_{восст} = 0,1$ мин — период восстановления начального положения точки подвеса штанг — управляемый параметр, который зависит от электропривода насоса (блок Drive); h — высота хода (3 м); $k_{зан}$ — коэффициент заполнения насоса (0,7); $S_{пл}$ — площадь плунжера насоса (блок Pump).

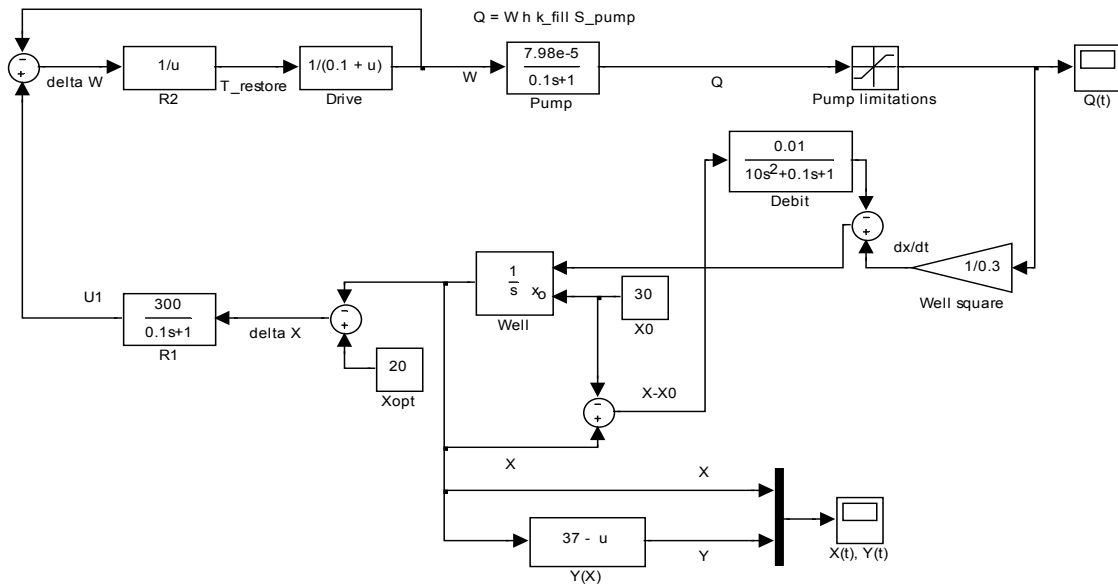


Рис. 23. САУ добычей нефти.

Для реализации указанной зависимости в Simulink воспользуемся интегратором с входом (x_0), реализующим начальные условия (блок Well).

Разделив производительность насоса на площадь горизонтального сечения забойной зоны, получим скорость изменения уровня x .

Блоки R1 и R2 являются регуляторами. Блок Debit реализует дебит скважины (он пропорционален разнице начального и текущего уровней жидкости $x-x_0$).

Блок Хорт — реализует заданный оптимальный уровень жидкости в скважине. В результате моделирования получены зависимости уровня жидкости в скважине $x(t)$ и уровня раздела вода-нефть $y(t)$ от времени (рис. 24).

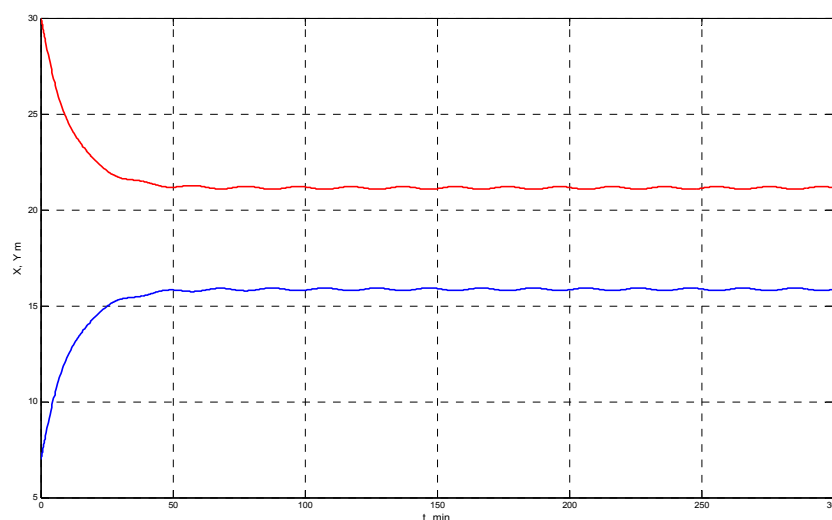


Рис. 24. Графики зависимостей $x(t)$ и $y(t)$.

Таким образом, предложенная модель нефтеизвлечения позволяет наиболее полно использовать запасы нефти и при этом исключить оборот воды, снизить металлоемкость ШСНУ за счет уменьшения глубины подвески насоса, уменьшить энергозатраты в 2,5-4 раза за счет снижения нагрузки на штанги и применения маятникового принципа работы наземного привода ШСНУ.

Литература

1. Шаньгин Е. С., Тагирова К. Ф. Система адаптивного управления режимами работы штанговых глубинных насосных установок // Мехатроника.— 2001.— №6.
2. Шаньгин Е. С. Автоматизированный привод глубинных насосов.— Уфа: Изд-во УТИС, 2001.

УДК 004

ИМИТАЦИОННАЯ МОДЕЛЬ ОПРЕДЕЛЕНИЯ ТОВАРНОГО СОРТА ПЛОДОВ

Гордеев А. С., Гурьянов Д. В., Рындюк К. Д.

Мичуринский государственный аграрный университет, Мичуринск, Тамбовская обл.,
e-mail: Duke10616@mail.ru

1. Процедура имитации сортирования плодов

Для окончательного определения качества плода, которое выражается категорией «товарный сорт», при автоматическом сортировании необходимо иметь также блок определения товарного сорта по результатам анализа изображения. Для приближения моделирующего алгоритма к реальному процессу сортирования плодов он должен содержать в себе процедуры (рис. 1):

- ввода информации (изображения плодов);
- отнесения каждого пикселя к одному из показателей качества поверхности плода (16 видов);
- определения площади, занимаемой на плоде каждым показателем качества;
- расчета товарного сорта;
- выдачи результата расчета товарного сорта на исполнительные механизмы.



Рис. 1. Процедуры сортирования плодов.

Целью моделирования является отработка методов распознавания качества, а также реализация методов определения товарного сорта.

2. Имитационная модель устройства сортирования плодов

Создание средств сортирования плодов по качеству является достаточно сложной задачей не только с точки зрения создания средств распознавания качества, но и в не меньшей степени — при создании соответствующих устройств подачи плодов в зону осмотра и их отвода к исполнительным механизмам для разделения потока на товарные сорта.

Имитационная модель устройства сортирования плодов приведена на рис. 2. Она состоит из пяти блоков:

Б1 — генератора изображений показателя качества;

Б2 — блока весовых коэффициентов;

Б3 — блока порогов показателей качества;

Б4 — блока расчета товарных сортов;

Б5 — блока исполнительных механизмов.

Блок Б5 исполнительных механизмов собственно не относится к имитационной модели, но его наличие предполагает законченность функционирования имитационной модели. Этот блок должен быть принадлежностью сортировочной машины.

Генератор изображений показателей качества

Генератор изображений показателя качества Б1 предназначен для формирования пикселей цветовых RGB составляющих изображения поверхности плода различного качественного состояния и подсчета их площади. С этой целью был использован банк изображений плодов, рис. 2, в котором находятся математические ожидания RGB составляющих различных показателей качества поверхности плодов и фона, а также их среднеквадратические отклонения. Вышеуказанные характеристики изображений плодов обозначены на рисунке 2 как элемент 1.

В блоке Б1 по математическим ожиданиям и среднеквадратическим отклонениям генерируются RGB составляющие одновременно по 16-ти каналам (показателям качества):

$$\begin{aligned} R &= MR \pm 3SR; \\ G &= MG \pm 3SG; \\ B &= MB \pm 3SB. \end{aligned} \quad (1)$$

Случайное значение из разрешенных базой данных с доверительными интервалами $\pm 3S$ генерируются с помощью генератора случайных чисел 2 и расчетного элемента 3, на выходе которого имеются три сигнала для каждой RGB составляющей для всех пикселей изображения плода всех 16-ти каналов. На рисунке 2 показан только один канал, остальные каналы блока Б1 — идентичны.

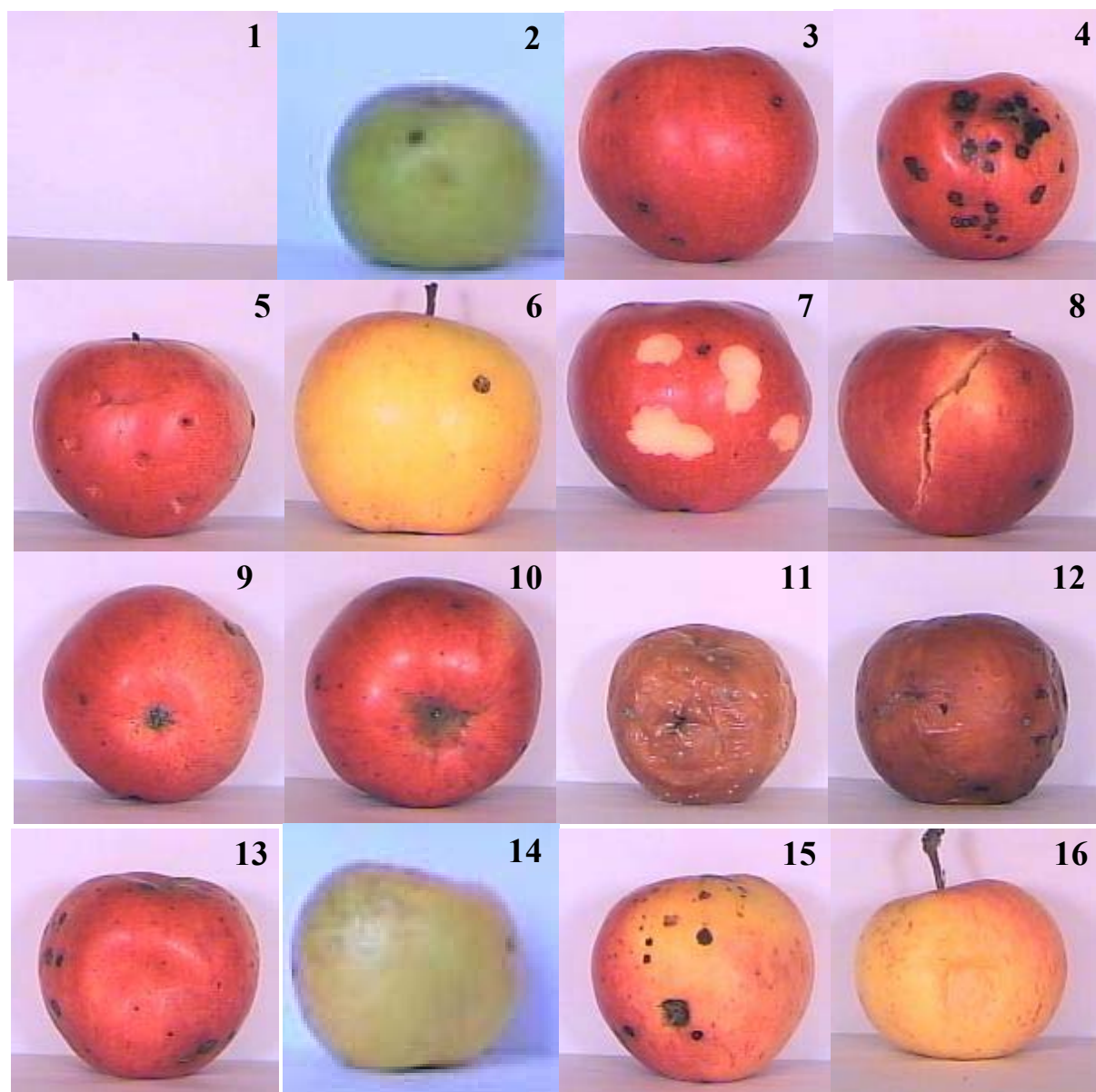


Рис. 2. Изображения плодов с различными показателями качества:
 1 — фон; 2 — зеленая здоровая; 3 — красная здоровая; 4 — парша; 5 — прокол свежий;
 6 — прокол заживший; 7 — сдир кожицы; 8 — растрескивание; 9 — чашечка; 10 —
 воронка; 11 — гниль плодовая; 12 — грибная гниль; 13 — ушиб; 14 — сетка; 15 —
 повреждение плодовой; 16 — загар.

Сгенерированная таким образом тройка чисел может быть пропущена через фильтр, ограничивающий по желанию исследователя размах выборки в процессе эксперимента. Это осуществляется с помощью специального блока 35 — ширина окна показателя качества. В этом элементе находятся вектора чисел, определяющие желаемые границы изменения каждой из RGB составляющих, которые подаются на входы сравнения элементов 4–9. На выходе каждой из пар элементов 4–5, 6–7, 8–9, подсоединенных соответственно к схемам совпадения 10–12, появляются логические 1 в случае, если сгенерированная RGB составляющая находится внутри ин-

тервала, определяемого элементами 4–9. Математически это выражается следующим образом:

$$\begin{aligned} &\text{если } R \leq R_{\max} \ \& \ R > R_{\min} \ , \ \text{то } F_R = 1, \ \text{иначе } F_R = 0; \\ &\text{если } G \leq G_{\max} \ \& \ G > G_{\min} \ , \ \text{то } F_G = 1, \ \text{иначе } F_G = 0; \\ &\text{если } B \leq B_{\max} \ \& \ B > B_{\min} \ , \ \text{то } F_B = 1, \ \text{иначе } F_B = 0. \end{aligned} \quad (2)$$

где: $(R_{\max} - R_{\min})$ — ширина окна R составляющей; $(G_{\max} - G_{\min})$ — ширина окна G составляющей; $(B_{\max} - B_{\min})$ — ширина окна B составляющей; F_R, F_G, F_B — сигналы на выходе блоков 10-12.

На выходе элемента совпадений 13

$$F_{13} = 1, \ \text{если } (F_R = 1 \ \& \ F_G = 1 \ \& \ F_B = 1), \\ \text{иначе } F_{13} = 0. \quad (3)$$

Это означает, что сгенерирован пиксель в заданном окне, соответствующем номеру канала — показателю качества. В элементе 38 записаны, вводятся исследователем, величины площадей показателей качества, которыми необходимо ограничить моделируемый процесс в силу тех или иных причин. В изображение вводится столько пикселей данного показателя качества, сумма которых соответствует площади данного показателя S_i , сколько введено в элемент 38.

Сигнал $F_{13} = 1$ подается на счетчик пикселей — элемент 14, который подсчитывает их количество за определенное время T или кадр изображения (В случае если ограничением является время, то оно задается пакетом Simulink MATLAB, а если ограничением является кадр изображения — то количеством пикселей в одном кадре. Полученная сумма пикселей S_i на выходе элемента 14 соответствует площади того i -го показателя, к которому относится канал:

$$S_i = \sum F_{13}. \quad (4)$$

Таким образом с выхода блока Б1 выводится площади до 16 показателей качества поверхности изображения плода.

Simulink — модели некоторых элементов блока Б1 приведены на рис. 3–5.

4. Блок весовых коэффициентов Б2

Блок весовых коэффициентов Б2 предназначен для свертки количества показателей и учета влияния разных показателей качества на величину искомого товарного сорта. На его входы поступают площади показателей качества S_i с блока Б1. Снижение количества показателей качества обусловлено необходимостью упрощения модели, и особенно ее эксплуатации — большое количество одновременно встречающихся показателей в одной партии маловероятно. Поэтому все показатели разделены на пять групп:

1. Механических повреждений, пятен (элемент 15, входы 4, 7, 10);

2. Ушибов от механических повреждений, нажимов (элемент 16, входы 13,14);
3. Болезни сада и хранилища, физиологические заболевания (элемент 17, входы 11,12,16);
4. Свежие проколы (элемент 18, входы 5, 6);
5. Здоровые плоды (элемент 19, входы 2, 3,8, 9).

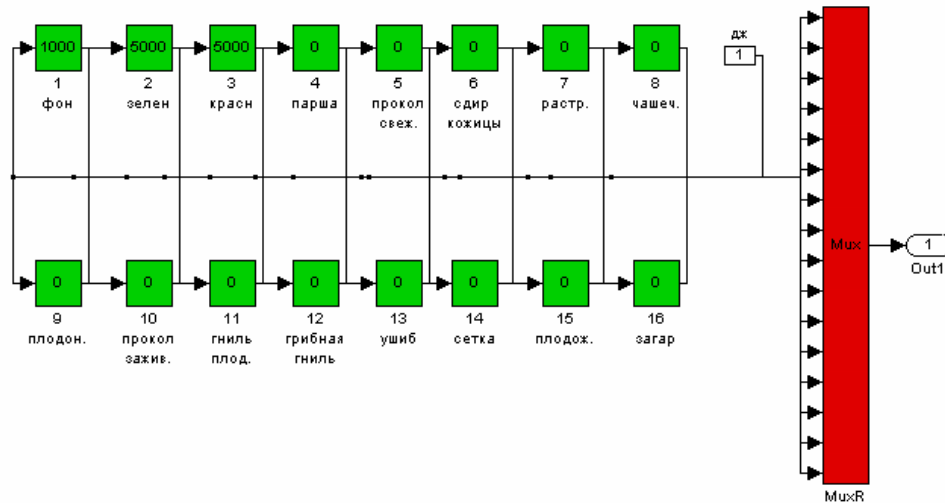


Рис. 3. Имитация вектора входных параметров.

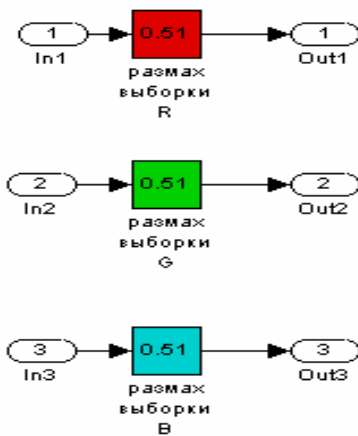


Рис. 4. Установка размаха выборки.

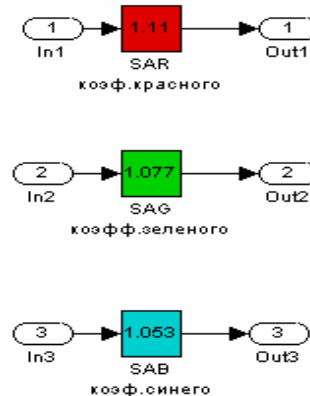


Рис. 5. Установка ширины окна.

Предполагается, что влияние показателей качества в каждой группе на конечный результат - товарный сорт одинаковые и что они имеют одинаковые весовые коэффициенты w_i . В каждой группе площади отдельных показателей суммируются:

- на выходе первой группы (элемент 15)

$$S_1 = S_4 + S_7 + S_{10}; \quad (5)$$

- на выходе второй группы (элемент 16)

$$S_2 = S_{13} + S_{14}; \quad (6)$$

- на выходе третьей группы (элемент 17)

$$S_3 = S_{11} + S_{12} + S_{16}; \quad (7)$$

- на выходе четвертой группы (элемент 18)

$$S_4 = S_5 + S_{76}; \quad (8)$$

- на выходе пятой группы (элемент 19)

$$S_5 = S_2 + S_3 + S_8 + S_9. \quad (9)$$

Выход элемента 19- площадь здоровой части плода. Сумма всех площадей дает размер плода:

$$S = S_1 + S_2 + S_3 + S_4 + S_5. \quad (10)$$

Если площади болезней стремятся к нулю, то площадь здоровой поверхности характеризует размер плода.

Так как влияние каждой группы показателей на величину товарного сорта различно, то площади с выходов умножаются в элементах 20–24 на соответствующие весовые коэффициенты w_i :

$$S_i = w_i * S_i, \quad i=1, 2, 3, 4, 5. \quad (11)$$

Весовые коэффициенты задаются и хранятся в специальном блоке, Simulink- модель которого приведена на рис.6. Выбор коэффициентов осуществляется путем экспертных оценок в интервале 0–1.

5. Блок порогов показателей качества

Блок Б3 порогов показателей качества предназначен для определения частного товарного качества по каждой группе показателей. Частный товарный сорт — это уровень площади повреждения отдельной группы показателей S_i относительно некоторых границ - пороговых значений площади. Задаются пороги площадей групп показателей, не более 3 - $S_{i1} < S_{i2} < S_{i3}$, с помощью которых можно разделить поток на 4 товарных сорта T_s по правилу:

$$\begin{aligned} \text{Если } 0 \leq S_i < S_{i1}, \text{ то } T_s &= 1; \\ \text{Если } S_{i1} \leq S_i < S_{i2}, \text{ то } T_s &= 2; \\ \text{Если } S_{i2} \leq S_i < S_{i3}, \text{ то } T_s &= 3; \\ \text{Если } S_{i3} \leq S_i, \text{ то } T_s &= 4. \end{aligned} \quad (12)$$

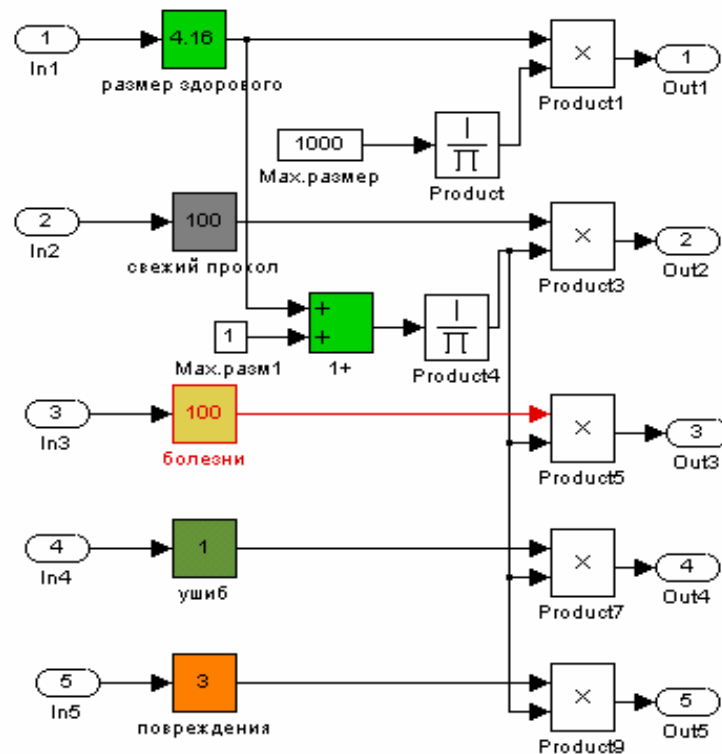


Рис. 6. Задание весовых коэффициентов показателей.

Шаг порогов ΔS , т. е. расстояние между ними определяются стандартами качества. Шаг как правило переменный. Пороги задаются и хранятся в элементе 37. Они по каждой группе 20-24 показателей качества подаются на элементы сравнения 25 — 29, на вторые входы которых с элементов 20-24 подаются величины площадей S_i , рис.2. Если поступившая, например, на входы элемента 25 с элемента 20 величина площади первой группы S_1 не превысила первый, самый маленький порог S_{11} , то на всех выходах элемента 25- логические нули, $T_s=1$. Если S_1 превысила первый порог S_{11} , то на первом выходе элемента 25 появляется логическая единица- $T_s = 2$, и т. д. Диаграмма преобразования площади S_i группы показателей качества в частный товарный сорт F_{ij} ($j=0, 1, 2, 3$ — количество частных товарных сортов), осуществляемый в элементах 25-29, приведена на рис.7.

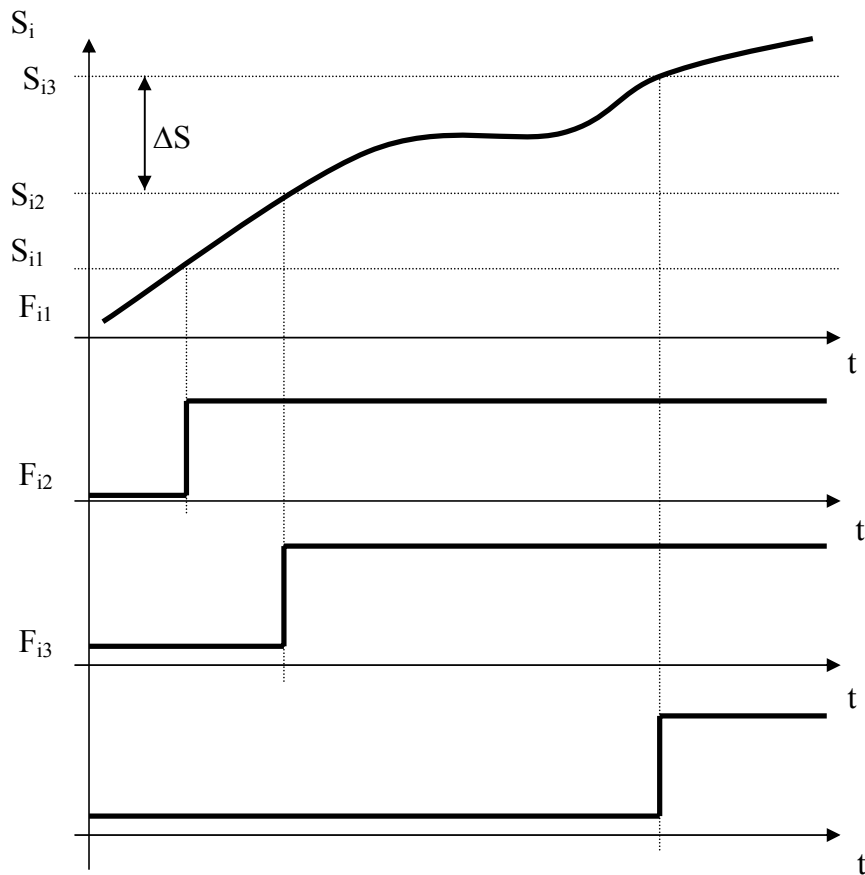


Рис. 7. Преобразование площади группы показателей качества в частный товарный сорт, $S_i \rightarrow F_{ij}$.

В каждом из элементов 25-29, рис.2., осуществляется обработка площади S_i отдельной группы показателей качества. Результатом работы блока Б3 является набор логической информации — последовательности нулей и единиц на выходах элементов 25-29 о частных товарных сортах, которые каждый со своей стороны характеризуют товарный сорт плода.

Simulink модель задания пороговых значений приведена на рис. 8.

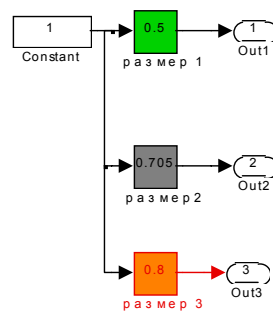


Рис. 8. Simulink модель задания пороговых значений.

6. Блок расчета товарного сорта

Блок расчета товарного сорта Б4 предназначен для получения номера товарного сорта T_s плода с учетом всех групп показателей качества. Входом блока расчета товарного сорта Б4 являются сигналы с элементов 25-29 блока Б3- логические последовательности частных товарных сортов F_{ij} . Они подаются на три схемы совпадения 30, рис 2. На каждую из этих схем подаются с блока Б3 одноименные частные товарные сорта F_{ij} со всех групп показателей качества, т. е. на элемент 30.1 подается только первый частный товарный сорт F_1 , на элемент 30.2- второй F_2 , на элемент 30.3.- третий F_3 . Таблица истинности логического преобразования в элементе 30 приведена в таблице 1.

Элемент 30, состоящий из трех пятиходовых (по числу групп показателей качества) схем повторений своими входами подключен к схеме выбора максимума 31. В соответствии с таблицей 2 на выходе схемы элемента 31 появляется номер товарного сорта T_s , который определяется максимальным частным товарным сортом F_{ij} из всех групп показателей качества ($j=5$). Информация с элемента 31 является результатом моделирования, который может быть использован для управления исполнительными механизмами 32-34. Эта же информация используется нами для анализа результата моделирования.

7. Окно управления имитационной модели

Для удобства исследования имитационной модели процесса сортирования создано окно управления, рис 9.

В окно управления модели выведены следующие блоки:

Блок 1 — вектор входных параметров — максимальных площадей показателей качества, которые возможны в процессе эксперимента $S = \{S_1, S_2, \dots, S_{16}\}$;

Блок 2 — размах выборки;

Блок 3 — ширина окна;

Блок 4 — весовые коэффициенты показателей;

Блок 5 — пороги товарных сортов;

Блок 6 — система имитационного моделирования распознавания товарного сорта плодов;

Блок 7 — дисплей 1 для наблюдения за показателями качества;

Блок 8 — дисплей 2 для наблюдения за показателями качества;

Блок 9 — величина показателя;

Блок 10 — наблюдение за величинами во времени;

Блок 11- товарный сорт продукции.

В блоке 1 устанавливается вектор $K = \{K_1, K_2, \dots, K_{16}\}$, состоящий из 16-ти чисел (максимальные пороговые значения по всем показателям качества), который затем представляется в виде последовательности чисел при помощи мультиплексора (Mux). В блоке 2 устанавливается размах определенной выборки по R, G и B. В блоке 3 устанавливаем ширину окна (коэффициент) по каждой из составляющих цвета. Установка необходимых значений в блоках 1, 2 и 3 производится с помощью пороговых элементов. В блоке 4 происходит оценка размеров, как здоровой поверхности, так и проколов, болезней, ушибов и повреждений на ней. В блоке 5 производится установка значений трех размеров по каждой группе поверхностей: здоровая (блок 5.1), свежий прокол (блок 5.2), болезни (блок 5.3), ушиб (блок 5.4), повреждения (блок 5.5) для каждого из трех товарных сортов. В блоке 6 происходит моделирование распознавания товарного сорта плодов. Блоки 7 и 8 служат для индикации показателей качества. Блок 9 позволяет посмотреть величину каждого показателя, а блок 10 — графическое окно изменения величин показателей во времени (имеется в виду модельное время эксперимента). В блоке 11 имеется возможность следить непосредственно за товарным сортом продукции.

Работа модели происходит следующим образом. Полученные массивы данных из математического ожидания и среднеквадратического отклонения по каждой составляющей цвета и каждому показателю качества ($M_R, M_G, M_B, S_R, S_G, S_B$) загружаются в рабочую область MATLAB. В блоке 6 производится считывание этих массивов данных и последующий расчет по всем показателям.

8. Методика проведения имитационного эксперимента

Методика проведения имитационного эксперимента заключается в следующем:

1. Задаем в блоке 1 максимальную площадь определенного показателя качества, который возможен в процессе эксперимента $S = \{S_1, S_2, \dots, S_{16}\}$;
2. Задаем в блоках 2, 3, 4, 5 параметры показателей качества (размах, ширину окна, весовые коэффициенты показателей, пороги товарных сортов).
3. Запускаем имитационную модель с панели окна управления (рис. 9) и на дисплее 11 наблюдает товарный сорт.

УДК 621.317.7

SIMULINK МОДЕЛЬ МИКРОМОЩНОГО ПРОГРАММИРУЕМОГО ОПЕРАЦИОННОГО УСИЛИТЕЛЯ

Грачев А. Д.

*Пензенский государственный университет, Пенза,
e-mail:graf@sura.ru*

Среда Simulink является удобной для проведения расчетов и проектирования измерительных преобразователей благодаря наличию большого набора моделей приборов и электронных компонентов. Этапы проектирования включают в себя последовательность построения структур измерительных преобразователей, начиная с идеальных компонентов из тулбокса Simulink (идеальный усилитель, сумматор, интегратор, дискретизатор и т. д.), и затем переход к структурам, содержащим все большую детализацию, отражающую особенность практической реализации. На последнем этапе моделирования используется набор компонент из тулбокса SimPowerSystems (диоды, резисторы, конденсаторы).

Проверка правильности расчетов проектирования осуществляется путем использования базы данных входных сигналов, полученных экспериментальным путем. Это позволяет обеспечить максимальную адекватность математической модели проектируемого измерительного преобразователя. Однако в тулбоксе SimPowerSystems отсутствуют модели микро-мощных программируемых операционных усилителей (ОУ), которые являются наиболее важным компонентом построения измерительных преобразователей.

При использовании специализированных программ для расчета электронных узлов, например в системе Pspice корпорации MicroSim, имеющей обширные библиотеки макромоделей ОУ ведущих фирм разработчиков и производителей аналогов интегральных микросхем, не возникает проблемы расчета технических характеристик, временных диаграмм и т. д. Но при этом теряется возможность строить комбинированные структуры с произвольной детализацией элементов и вычислительных алгоритмов, которые обеспечивают MATLAB.

Для расширения возможностей расчета и моделирования ИП предлагается макро-модель микро-мощного программируемого ОУ, в которой моделируется скорость нарастания выходного напряжения как для положительного фронта (V_+), так и для отрицательного фронта (V_-), зависимость скорости нарастания от управляющего тока I_y , ограниченность динамического диапазона выходного напряжения и коэффициент усиления по постоянному току (K). Другие параметры ОУ, такие как входное сопротивление, выходное сопротивление, входной ток смещения, напряжение смеще-

ния, тепловой шум могут быть учтены путем введения дополнительных компонентов.

Ограниченность скорости нарастания ОУ вызвана наличием специальной корректирующей емкости, которая задает полюс передаточной функции ОУ. При уменьшении тока потребления происходит увеличения постоянной времени интегрирующей цепи, что и проявляется в виде пропорционального уменьшения скорости нарастания выходного напряжения. Для задания числовых значений модели необходимо знать паспортные или экспериментальные данные о скорости нарастания V_0 при фиксированном токе управления I_0 . Тогда для произвольного тока управления скорость нарастания определяется как функция управления:

$$V = I \frac{V_0}{I_0}. \quad (1)$$

Ограниченность скорости нарастания будет иметь место, если будет ограничено напряжение (ток) на входе инерционного звена с передаточной функцией $\frac{1}{1 + \tau p}$, поскольку реакция на ступенчатое воздействие описывается линейной функцией:

$$U_{\text{вых}} = \frac{U_{\text{вх}} \cdot t}{\tau}$$

Таким образом, скорость нарастания можно изменять путем изменения τ по закону:

$$\tau = \frac{I_0}{I}$$

При условии, что $U_{\text{вх}}$ численно равно U_0 с точностью до масштабного коэффициента, будет выполнена функция управления (1).

Таким образом, модель функции управления током потребления и скоростью нарастания ОУ, должна содержать последовательно включенные ограничитель уровня «Saturation», модель инерционного звена с управляемой постоянной времени, входной и выходной усилители и ограничитель амплитуды выходного сигнала.

Вариант структуры микромощного программируемого ОУ приведен выше на рис. 1. Параметры модели соответствуют техническим характеристикам микромощного программируемого ОУ 1407 УД2. На рис. 2 приведена модель входного усилителя с ограничителем, на рис. 3 — модель инерционного звена с управляемой постоянной времени. Управление осуществляется током, протекающем по цепи «Упр». Величина тока преобразуется элементом «Current Measurement» в функцию управления.

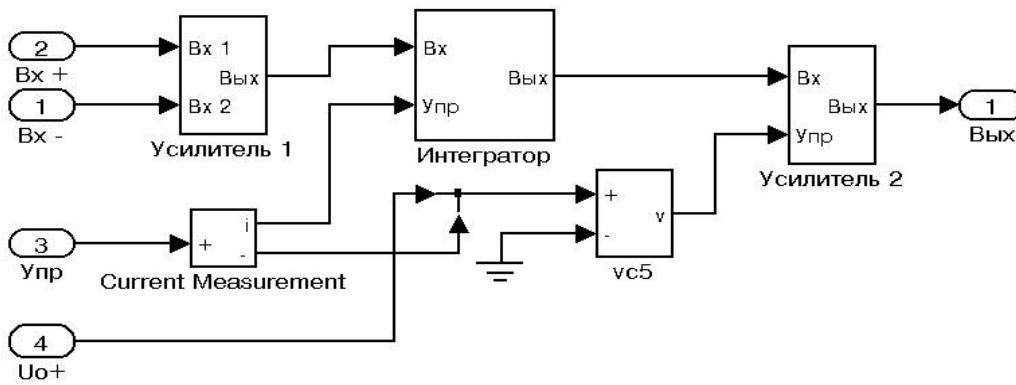


Рис. 1. Структура микромощного программируемого ОУ.

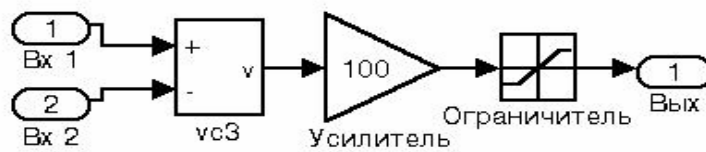


Рис. 2. Структура усилителя 1.

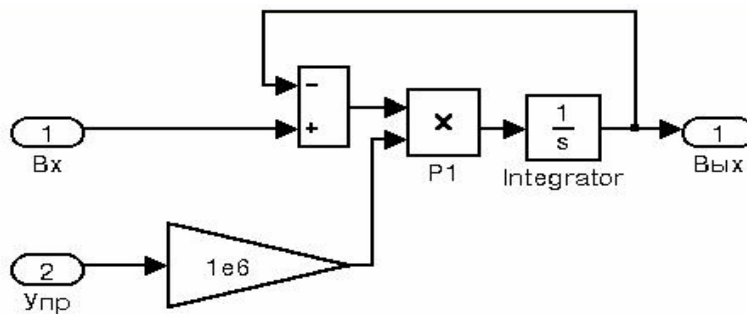


Рис. 3. Структура интегратора с управляемой постоянной времени.

На рис. 4 представлена модель выходного усилителя с ограничителем амплитуды выходного сигнала. Уровень ограничения задается значением напряжения питания U_{o+} . В выходной цепи включен резистор «R_вых», моделирующий выходное сопротивление ОУ.

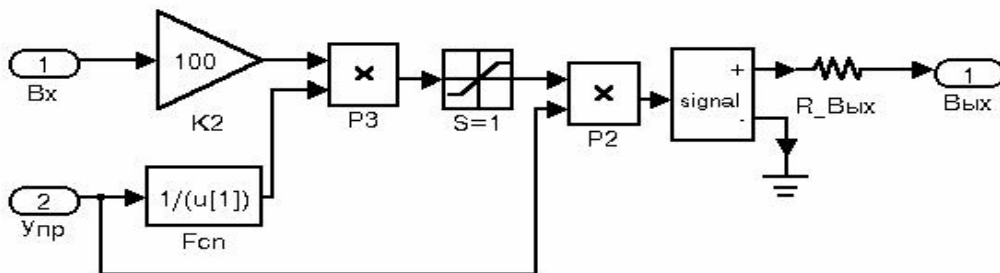


Рис. 4. Структура усилителя 2 с управлением ограничения выходного

На рис. 5 в качестве примера приведена структура генератора прямоугольных импульсов с использованием программируемого микромощного ОУ в режиме минимального тока потребления. На рис. 6 представлена временная диаграмма его работы при запуске по цепи управления генератором ступенчатого сигнала.

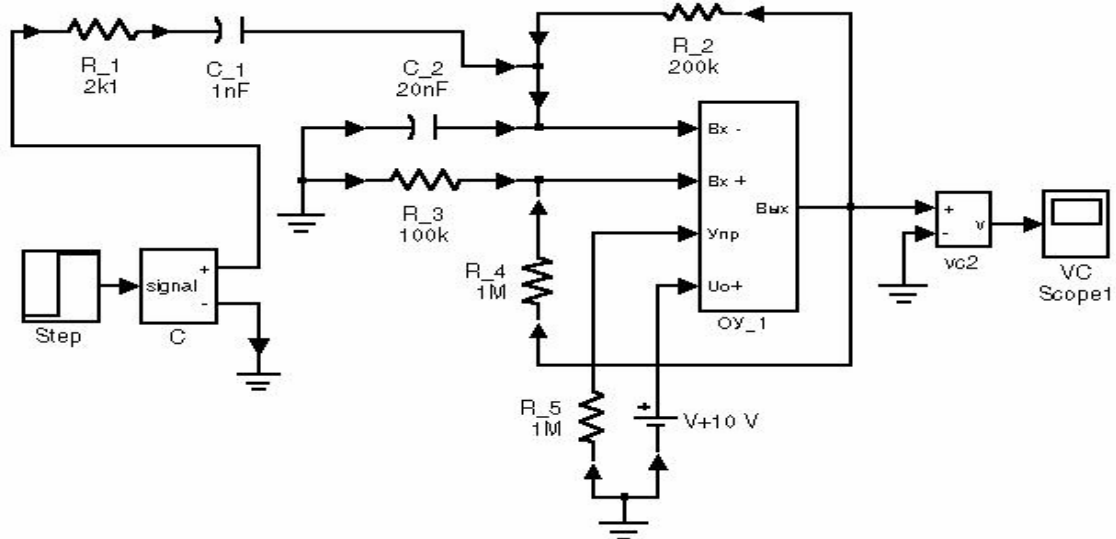


Рис. 5. Структура генератора прямоугольных импульсов.

Как видно из рис. 6 ограниченность скорости нарастания приводит к трапецевидной форме сигналов.

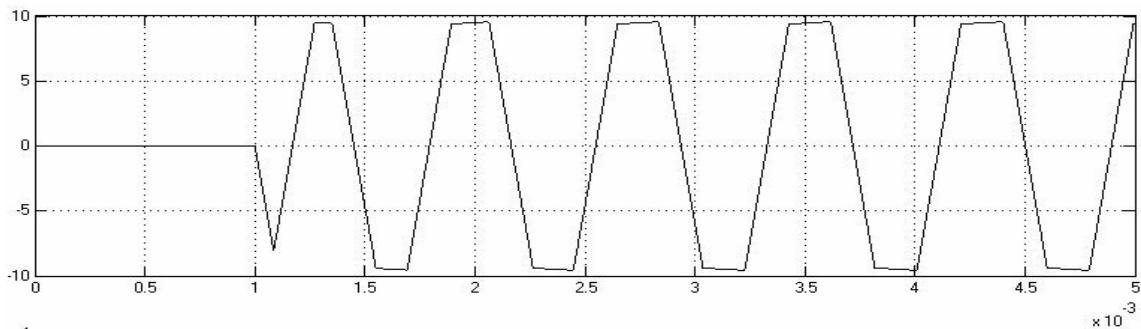


Рис. 6. Временная диаграмма выходного напряжения генератора при запуске фронту прямоугольного импульса в момент времени 10^{-3} с.

Использование предложенной модели микромощного программируемого операционного усилителя в сочетании с возможностями моделирования в среде Simulink позволяет более точно решать задачу численного расчета параметров электронных схем с минимальными затратами на экспериментальные исследования.

Литература

1. *Грачев А. Д.*, Системное проектирование микроощных измерительных преобразователей для распределенных информационно-измерительных систем // Материалы Междунар. юбилейного симп. «Актуальные проблемы науки и образования».— Пенза: Пензенский гос. ун-т, 2003.
2. *Дьяконов В. П.* Simulink 4. Специальный справочник.— СПб.: Питер, 2001.— 528 с.

УДК 621-521.001.57(076)

ПОСТРОЕНИЕ НА ЭЛЕМЕНТАХ SIMULINK ИЗМЕРИТЕЛЕЙ ФАЗЫ, ЧАСТОТЫ И АМПЛИТУДЫ

Данилов А. И.

Московский государственный университет пищевых производств, Москва,
e-mail: atp@mgupp.ru

В работе описываются два виртуальных прибора: измеритель затухания гармонических колебаний, а также измеритель фазы и частоты.

Автоматический пиковый измеритель затухания — это виртуальный прибор в среде Simulink, разработанный для снятия амплитудно-частотных характеристик некоторого неизвестного динамического объекта, см. рис. 1.

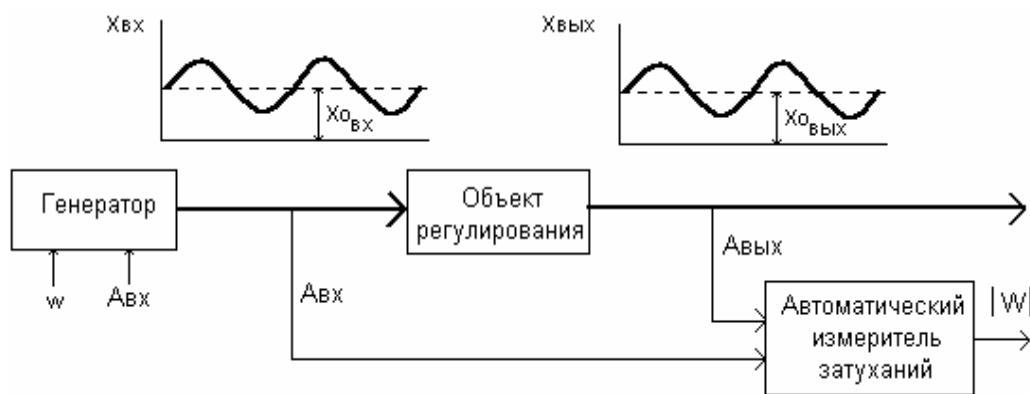


Рис. 1.

Принцип действия

При синусоидальном испытательном сигнале достаточно определить и запомнить значения сигнала в точках X_{\min} и X_{\max} для входа и выхода, см. рис.2.

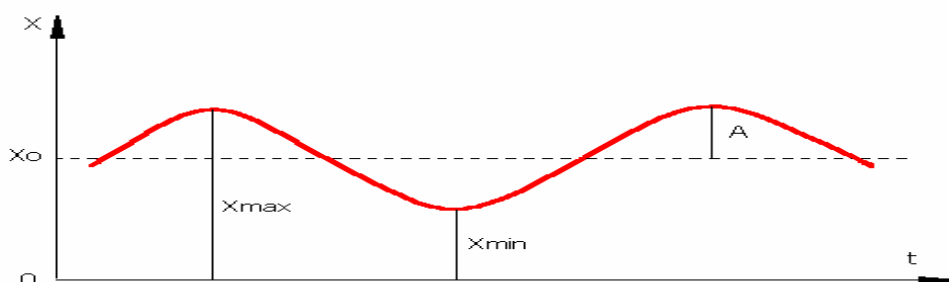


Рис. 2.

Независимо от частоты получаем значение амплитуды:

$$A = 0.5 * (X_{\max} - X_{\min}) .$$

Модуль передаточной функции(затухание):

$$|W| = A_{\text{ВЫХ}} / A_{\text{ВХ}} .$$

Прибор выполняет также дополнительные вычисления постоянных составляющих испытательного и выходного сигналов (смещения $X_{0\text{ВХ}}$ и $X_{0\text{ВЫХ}}$ на рис.1):

$$X_0 = 0.5 * (X_{\max} + X_{\min}) .$$

Устройство сконструировано на базе линейных, логических а также переключательных элементов Simulink, см. рис. 3, 4.

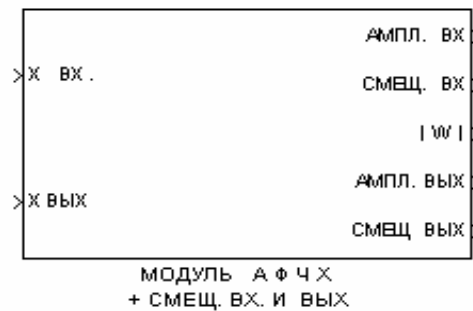


Рис.3.

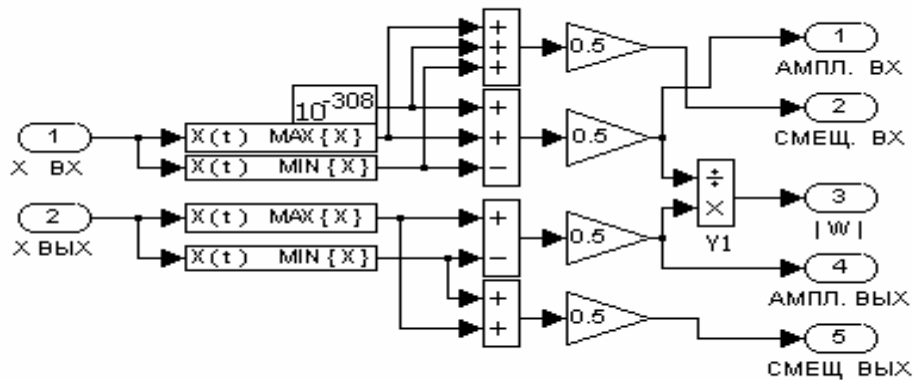


Рис.4.

Следует заметить, что при вычислении затухания в начальный момент времени в элементе Y1 возникает ситуация «деление на ноль». Чтобы этого избежать, ко входному сигналу $X(t)$ прибавляется бесконечно малое число 10^{-308} .

Также заметим, что в наборе элементов Simulink отсутствуют простые устройства для нахождения минимума или максимума скалярной функции. Поэтому их пришлось построить дополнительно, используя существующие стандартные элементы.

Схема для нахождения минимума показана на рис. 5, 6:



Рис.5.

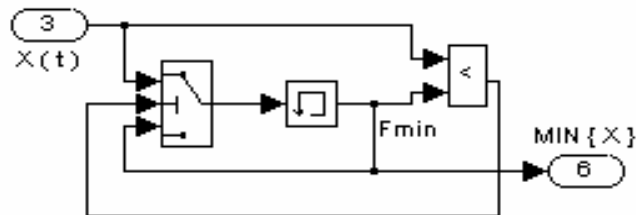


Рис.6.

Функционирование схемы, рис. 6.

Входной сигнал $X(t)$ поступает в блок анализатора и сравнивается с его предыдущим значением. (В самый начальный момент времени $X(t)$ сравнивается с числом, записанным в качестве начальных условий блока памяти). В схеме нахождения минимума, рекомендуется установить в качестве начальных условий блока памяти число 10^{307} (или $+\text{inf}$), а при нахождении максимума, см. рис. 9 — число минус 10^{307} (или $-\text{inf}$).

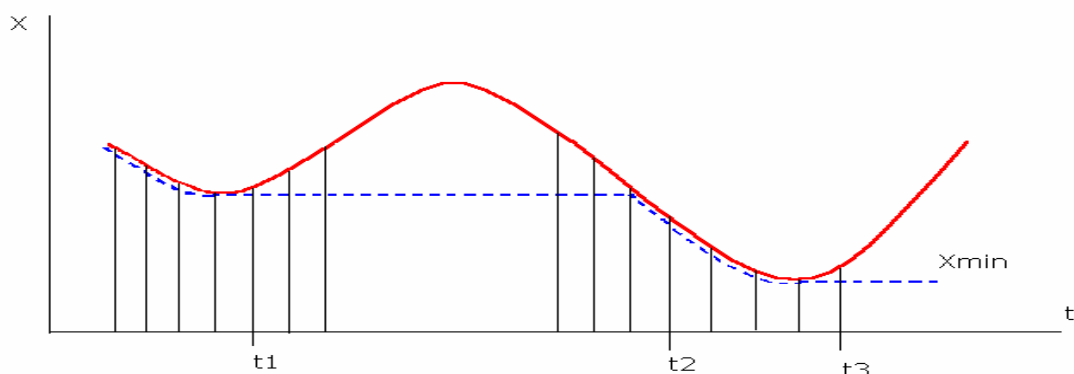


Рис.7.

Если неравенство выполняется (т. е. текущий сигнал X_i меньше предыдущего X_{i-1}), то это означает, что найдено еще меньшее значение, см. рис. 7, и на средний вход переключателя, см. рис. 6, подается *единица*. Он становится в верхнее положение и поиск продолжается.

В момент времени $t1$, см. рис. 7, текущее значение X_i станет больше предыдущего X_{i-1} (т. е. мы уходим от локального минимума), неравенство не выполняется, на вход переключателя подается *ноль* и он переходит в

нижнее положение. При этом блок памяти замыкается сам на себя и тем самым запоминает последнее минимальное значение. Далее следующее текущее значение $X(t)$ сравнивается с найденным локальным минимумом.

В момент времени t_2 текущее значение станет меньше ранее найденного минимума, — неравенство снова выполняется и переключатель перейдет в верхнее положение. Поиск продолжается, пока не будет найден очередной минимум в момент времени t_3 .

Устройство для поиска максимума работает аналогично, см. рис. 8, 9.

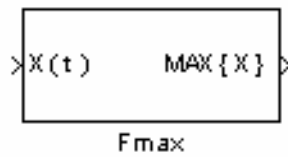


Рис.8.

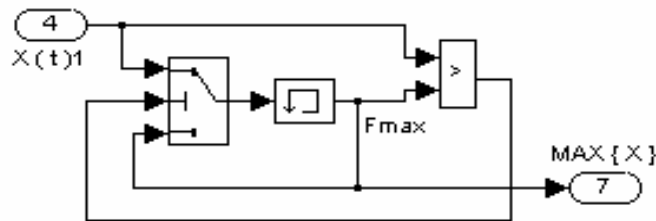


Рис.9.

Работа с прибором

Прибор имеет два входа, см. рис. 3:

- первый — для входного сигнала;
- второй — для сигнала на выходе объекта.

Время измерения должно быть таковым, чтобы переходные процессы в объекте закончились и образовались бы стационарные колебания.

Прибор имеет пять выходов:

- амплитуда входа (испытательного сигнала);
- смещение входа $X_{0вх}$;
- модуль АФЧХ, $|W|$;
- амплитуда выходного сигнала;
- смещение выходного сигнала $X_{0вых}$.

Величины смещений будут использоваться в качестве исходных данных для описываемого ниже прибора.

Фазочастотомер гармонических колебаний — это виртуальный прибор, предназначенный для измерения фазочастотных характеристик любых динамических объектов, рассматриваемых как «черный ящик».

Принцип действия

Чтобы вычислить фазу, достаточно измерить интервалы времени L и D , см. графики на рис. 11.

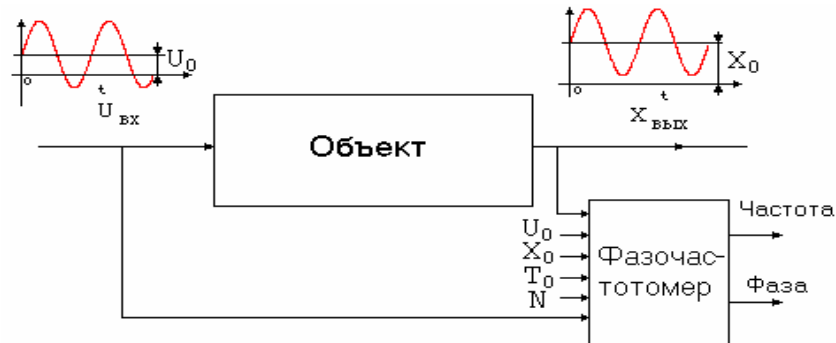


Рис.10.

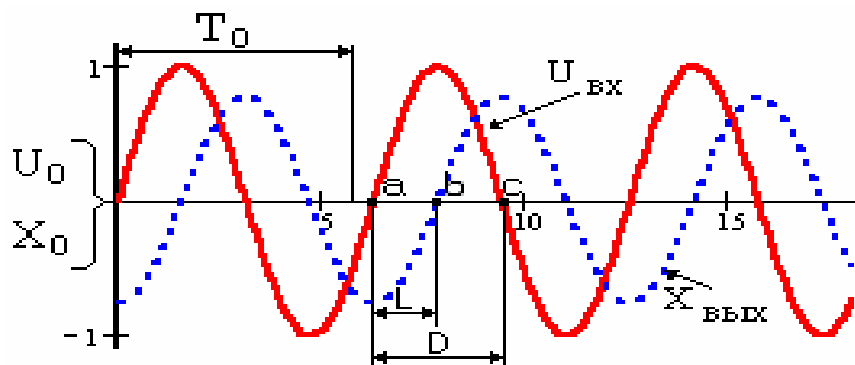


Рис. 11.

Прибор выполняет следующие этапы измерения:

- Считывает со входа интервал времени, за пределами которого установятся стационарные колебания (интервал T_0 на рис. 11).
- Измеряет длительность D одного полупериода входного сигнала и смещение L выходных колебаний по отношению ко входным, см. рис. 11.
- После получения значений L и D вычисляет фазу, а именно:

$$\Phi = (L/2D) \cdot 2\pi.$$
 (Знание постоянных смещений входных U_0 и выходных X_0 колебаний необходимо для вычисления фазы, поскольку на практике колебания происходят относительно не нулевого уровня).
- Побочный эффект. Зная D прибор вычисляет частоту колебаний:

$$f_{\text{рад}} = \pi/D \quad \text{или} \quad f_{\text{герц}} = 1/(2 \cdot D).$$

Устройство и функционирование фазочастотомера

Вид прибора представлен на рис.12 и рис.13. У него имеется 6 входов и 5 выходов.



Рис.12.

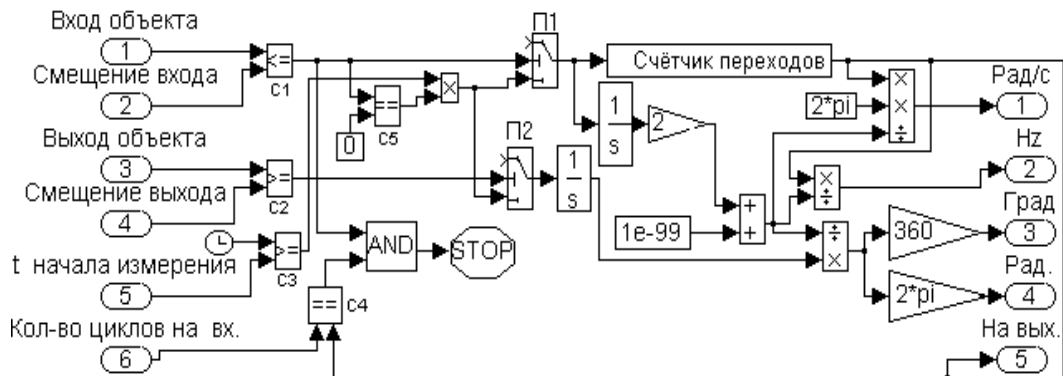


Рис.13.

Фазочастотомер использует логические элементы для сравнения постоянного смещения (U_0) относительно самого сигнала (см. рис.13, элемент C1),— если условие выполняется, то «кривая вниз», если нет — «кривая вверх» (на выходе 1 или 0 соответственно).

При измерении величины D , прибор рассматривает положительную полуволну (но только по прошествии интервала T_0).

Блок C3 используется совместно с элементом **Clock** для сравнения текущего времени моделирования со значением T_0 и далее для выдачи разрешающего сигнала на начало измерения отрезков L и D .

На точность измерения L и D влияет не только время T_0 (окончания переходных процессов), но и не нулевые значения постоянных составляющих U_0 и/или X_0 .

Для учета этого, значение сигнала сравнивается с величиной смещения U_0 (или X_0) и, как только $U_{вх}$ окажется в точке «а», т. е. на уровне U_0 (или в точке «b», т. е. на уровне X_0 для $X_{ввых}$), сразу начинается вычисление интервалов L и D . Для входного сигнала измеряется только длительность D полупериода, поскольку сигнал симметричен.

Собственно измерение интервалов L и D осуществляется на интеграторах.

Константа $1e-99$ позволяет избегать ситуаций «деление на ноль» в начальный период $[0-T_0]$.

Важным элементом прибора является блок под названием **Счетчик переходов**, см. рис. 14 и рис. 15.



Рис.14.



Рис.15.

Он служит для подсчета количества пересечений входным сигналом уровня U_0 , произошедших за все время измерения. Основную роль здесь играют стандартные элементы **Memory** и **Hit Crossing**, см. рис.15.

Последний представляет собой блок-датчик, который сообщает о пересечении входным сигналом установленного уровня.

Здесь следует заметить, что его входной сигнал представляет собой уже прямоугольные колебания в диапазоне $[0-1]$.

В одном полном периоде пересечение возможно: одно «на подъеме» и одно «на спуске». Таким образом, выставляя в настройках блока **Hit Crossing** значение ‘на подъем», на выходе получаем сигнал о начале нового периода колебаний. Для запоминания количества переходов используется блок **Memory**, который хранит в памяти его предыдущее значение.

Прибор спроектирован таким образом, что за время измерения величины L и D подсчитываются N раз (и суммируются). Это полезно для усреднения измеренных значений по нескольким периодам, особенно при наличии в сигналах искажений и шумов.

Рассмотрим подробнее организацию окончания процесса измерения.

С помощью логического элемента **C4**, установленное нами требуемое количество циклов N , сравнивается с числом уже случившихся переходов «снизу-вверх». Если сразу направить сигнал с выхода **C4** на элемент **Stop**, вычисления остановятся. Однако это будет неверным решением. Действительно, последний цикл не будет завершен, поскольку положительное полупериодное колебание только что началось и измерения еще не завершились.

По этой причине воспользуемся логическим **AND**, на второй вход которого подадим сигнал с выхода элемента **C1**, что позволит «задержать»

остановку измерений до тех пор, пока входной сигнал не перейдет в отрицательную область.

В заключении заметим, что предлагаемая на рис. 13 схема измеряет фазу в пределах $0\text{--}180^\circ$. Небольшая ее доработка в части завершения процедуры измерения позволит легко расширить этот диапазон до 360° .

Литература

1. *Данилов А. И.* Компьютерный практикум по курсу «Теория управления». Simulink–моделирование в среде MATLAB.— М.: МГУИЭ, 2002.
2. *Потемкин В. Г.* MATLAB 6. Среда проектирования инженерных приложений.— М.: Диалог-МИФИ, 2003.
3. *Черных И.В.* Simulink: Среда инженерных приложений.— М.: Диалог-МИФИ, 2004.

УДК 519.6

МОДЕЛИРОВАНИЕ СРЕДСТВАМИ ПАКЕТА SIMULINK ИНТЕГРИРУЮЩИХ АНАЛОГО-ЦИФРОВЫХ ПРЕОБРАЗОВАТЕЛЕЙ

Жесткова Ю. Е., Чернецов В. И.

*Пензенский филиал Российского государственного университета инновационных технологий и предпринимательства, Пенза,
e-mail: rgu@penza.net*

Интегрирующие АЦП (ИАЦП) занимают важное место в ряду всего многообразия аналого-цифровых преобразователей благодаря наличию у них таких полезных качеств как простота схемной реализации, высокая помехоустойчивость и точность преобразования [1, 2]. Вместе с тем многообразие требований предъявляемых к метрологическим характеристикам и условиям эксплуатации ИАЦП, заставляет возвращаться к созданию более совершенных преобразователей реализующих современные достижения микроэлектроники. Эффективным средством решения указанных задач в настоящее время является использование при проектировании методов моделирования на ЭВМ. В настоящем докладе рассматриваются вопросы применения пакета Simulink для моделирования различного типа ИАЦП.

Как показали исследования, непосредственно для моделирования необходимо применять не только пакет Simulink, но и сопряженные с ним пакеты моделирования Power System Blockset (в версии MATLAB 6.5 это пакет SimPower System) и DSP Blockset. Это позволяет быстро и оперативно создавать макромодели ИАЦП, которые вместе с тем с высокой степенью подробностей позволяют имитировать режимы работы АЦП, влияние помех и внешних дестабилизирующих факторов, а также влияния внутренних источников погрешностей.

В качестве примера рассмотрим модель ИАЦП (рис. 1) реализующего известный способ преобразования с так называемым двухтактным интегрированием [2], которая представлена в приложении в файле adc_2int. В данной модели голубым цветом выделены компоненты из пакета Power System Blockset, остальные компоненты относятся к пакету Simulink. Рассматриваемая модель содержит ряд подсистем моделирующих основные блоки АЦП.

Подсистема **Subsystem OP AMP** (рис. 2) моделирует работу операционного усилителя (ОУ) активного интегратора и учитывает влияние входного сопротивления **R_{in}**, конечного значения коэффициента усиления (компонента **Gain**), уровней насыщения выходного напряжения ОУ (компонента **Saturation**). Выходное напряжение ОУ задается управляемым источником напряжения **E** (компонента **Controlled Voltage Source**).

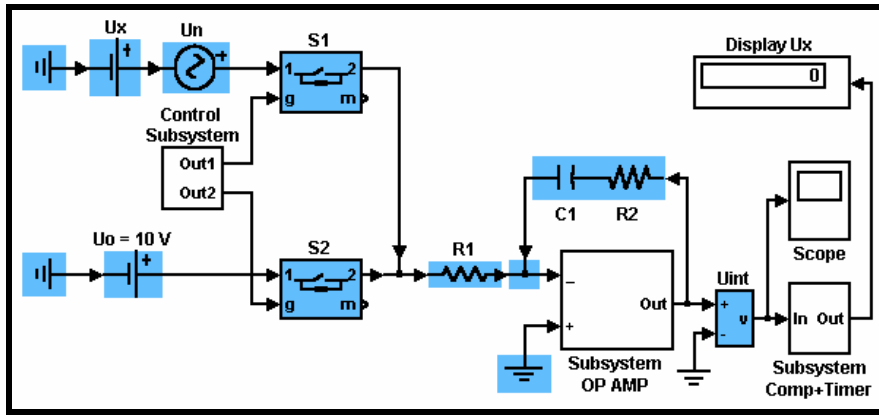


Рис. 1. S-модель ИАЦП с двухтактным интегрированием.

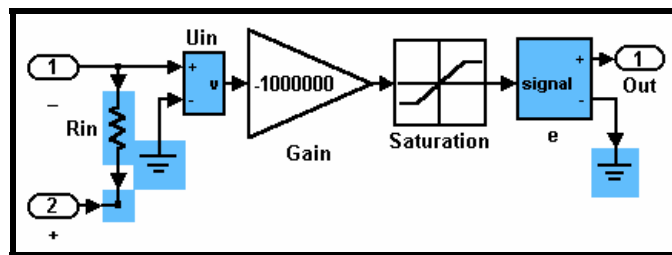


Рис. 2. Подсистема Subsystem OP AMP.

Подсистема **Subsystem Coup + Timer** (рис. 3) выполняет функции компаратора (блоки **Product** и **Hit Crossing**), а также преобразователя выходного информативного интервала времени в код (блоки **Integrator** и **Gain**).

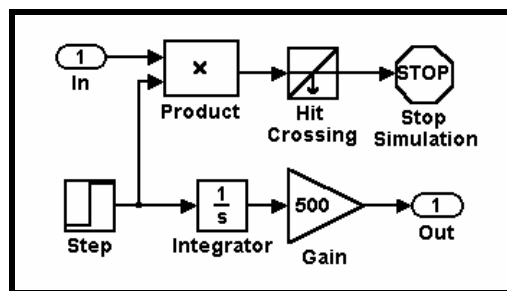


Рисунок 3. Подсистема Subsystem Comp+Timer.

Управление работой модели осуществляется посредством **Control Subsystem** (рис. 4), которая осуществляет коммутацию ключей **S1** и **S2** согласно логике работы схемы.

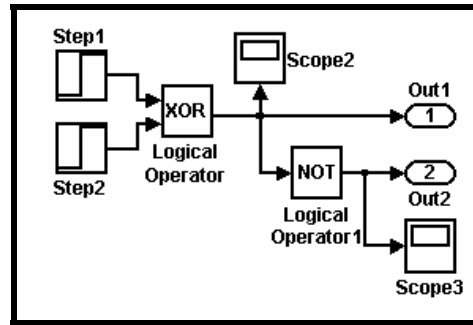


Рис. 4. Подсистема Control Subsystem.

В процессе работы схемы (см. временные диаграммы работы модели ИАЦП на рис. 5.) в течение интервала времени $[0, t_1]$, когда ключ $S1$ открыт, а ключ $S2$ — закрыт, осуществляется интегрирование входного преобразуемого напряжения U_x и аддитивной синусоидальной помехи с амплитудой, равной U_{Π} . В момент времени t_1 ключ $S1$ закрывается, а ключ $S2$ открывается, и начинается интегрирование опорного напряжения U_0 до момента t_2 равенства нулю входного напряжения интегратора.

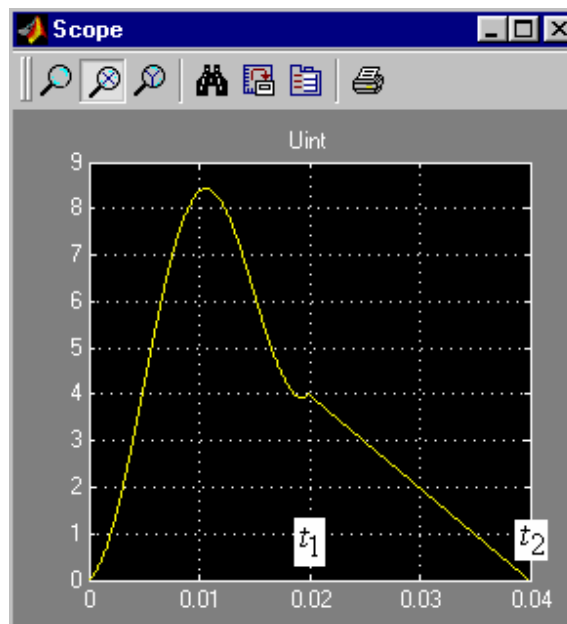


Рис. 5. Осциллограмма выходного напряжения интегратора для модели рис. 1.

Соответствующее уравнение преобразования ИАЦП описывается выражением:

$$\frac{U_x t_1}{R_1 C_2} + \frac{1}{R_1 C_1} \int_0^{t_1} U_{\Pi} \sin \frac{2\pi t}{T_{\Pi}} dt = \frac{U_0(t_2 - t_1)}{R_1 C_1}, \quad (1)$$

где T_{Π} — период помехи, $R_1 C_1$, — постоянная времени интегратора.

Очевидно, что при $t_1 = T_{\Pi}$

$$\int_0^{t_1} U_{\Pi} \sin \frac{2\pi t_1}{T_{\Pi}} dt = 0.$$

В этом случае функция преобразования ИАЦП принимает вид

$$t_2 - t_1 = \frac{U_x \cdot t_1}{U_0},$$

т. е. результат преобразования не зависит от размера помехи.

При записи выражения (1) не учитывалось влияние резистора R_2 , который применяется в схемах интеграторов для коррекции погрешностей нелинейности [3, 4]. Обычно $R_1 \gg R_2$, которое имеет значения порядка 10-50 Ом. В рассматриваемой модели применение добавочного резистора R_2 имеет и другую роль, связанную с обеспечением корректности работы модели. Если при настройке модели принять $R_2 = 0$, то при компиляции система выдает сообщение о наличие сингулярности в решении системы уравнений описывающих модель. Задание сколь угодно малого $R_2 \neq 0$ снимает все проблемы связанные с наличием сингулярности.

Следует особо подчеркнуть, что при настройке визуально-ориентированных моделей, для устранения сингулярностей в решениях систем уравнений полезно применять и другой прием, который предусматривает введение в схему шунтирующих резисторов с большими (практически не влияющими на работу схемы) номиналами.

Рассмотренная модель ИАЦП с двухтактным интегрирование позволяет эффективно исследовать свойства помеховосприимчивости проявляющееся во влиянии размера помехи на помехоустойчивость АЦП [5]. Действительно, если помеха по амплитуде превысит некоторый предел, то интегратор будет входить в насыщение, что в итоге приведет к резкому ухудшению помехоустойчивости.

Для устранения указанного недостатка была предложена структура ИАЦП [6], которая позволяет преобразовывать сигналы в случаях, когда помеха в десятки и сотни раз превышает размер полезного измеряемого сигнала. Модель такого ИАЦП приведена на рис. 6. Отличие данной структуры от рассмотренной выше заключается во введении дополнительного интегрирующего конденсатора C_2 , который в течение 1-го цикла интегрирования $[0, t_1]$ подключается посредством ключей S_3 параллельно основному интегрирующему конденсатору C_1 , а во втором такте через резистор R_3 соединяется с шиной земля. Структура ключей S_3 и схемы управления показаны на рис. 7, а временные диаграммы работы схемы приведены на рис. 8.

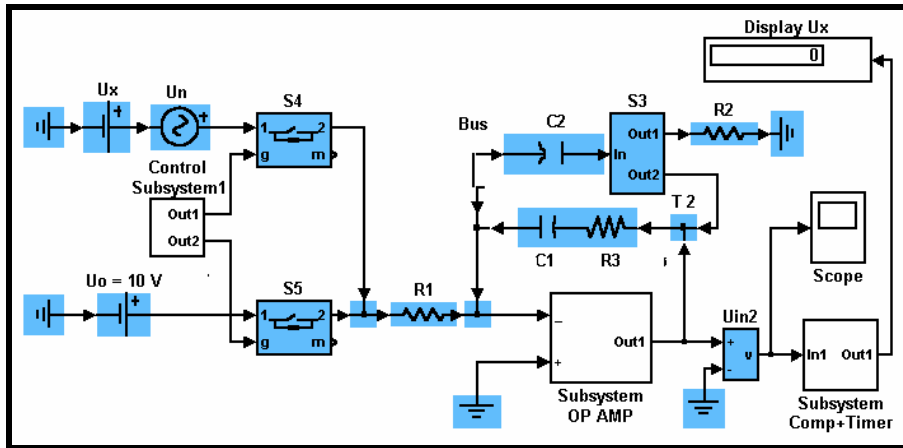


Рис. 6. S-модель ИАЦП с улучшенными характеристиками помехоустойчивости.

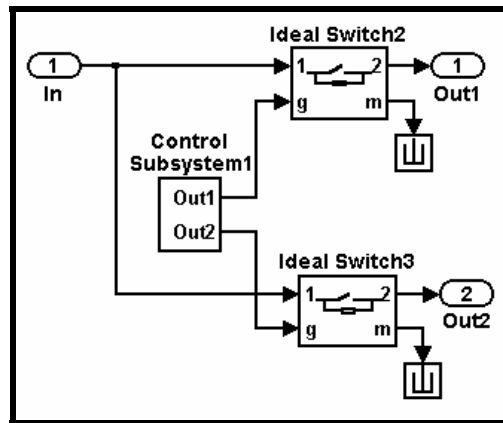


Рис. 7. Модель подсистемы S3.

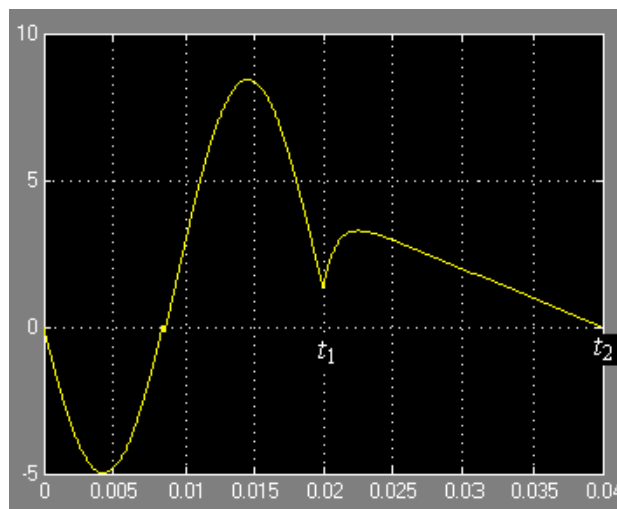


Рис. 8. Осциллограмма выходного напряжения интегратора для модели рис. 6.

Рассмотрим работу схемы, полагая, что $R3 = 0$ и влияние помехи отсутствует.

В первом такте преобразования к моменту времени t_1 на выходе ин-

тегратора накопится напряжение

$$U_{и}(t_{1-}) = \frac{U_x \cdot t_1}{(C_1 + C_2) \cdot R_1}. \quad (2)$$

После перекоммутации ключей $S3$ и $S4$ конденсатор $C2$ подключается к шине «земля» и накопленный на нем заряд передается на конденсатор $C1$ в результате этого напряжение на конденсаторе будет определяться выражением

$$U_{и}(t_{1+}) = U_{п}(t_{1-}) + \frac{U_x(t_{1-}) \cdot C_2}{C_1}. \quad (3)$$

С учетом этого уравнение преобразования принимает вид

$$U_{п}(t_{1+}) = \frac{U_0(t_2 - t_1)}{R_1 C_1},$$

или, учитывая (2) и (3), после соответствующих подстановок и преобразований получим:

$$(t_2 - t_1) = \frac{U_x t_1}{U_0},$$

т. е. функция преобразования имеет тот же вид, как и у классического ИАЦП с двухтактным интегрированием. При этом значения $C1$, $C2$ и $R1$ не входят в функцию преобразования. Принципиальное отличие заключается в том, что в первом такте преобразования есть возможность более рационального использования динамического диапазона ОУ интегратора, поскольку постоянная интегрирования велика и равна $(C1+C2)R1$. Из-за этого при большой амплитуде помехи интегратор не будет входить в насыщение, следовательно, улучшаются показатели помеховосприимчивости. Например, при $C2 = C1$ удастся в 4 раза повысить помеховосприимчивость ИАЦП при сохранении помехоустойчивости и точности преобразования.

Для имитации и исследования влияния источников погрешностей в дополнение к моделям ИАЦП была разработана в виде подсистем библиотека основных блоков интегрирующих аналого-цифровых преобразователей: операционных усилителей (см. рис. 9), компараторов и ключей.

В качестве примера: модель рис. 9,а учитывает только влияние конечного значения коэффициента усиления, модель рис. 9,б учитывает влияние входного R_{in} и выходного R_{out} сопротивлений ОУ, модель рис. 9,в дополнительно учитывает влияние эффектов насыщения выходного напряжения операционного усилителя.

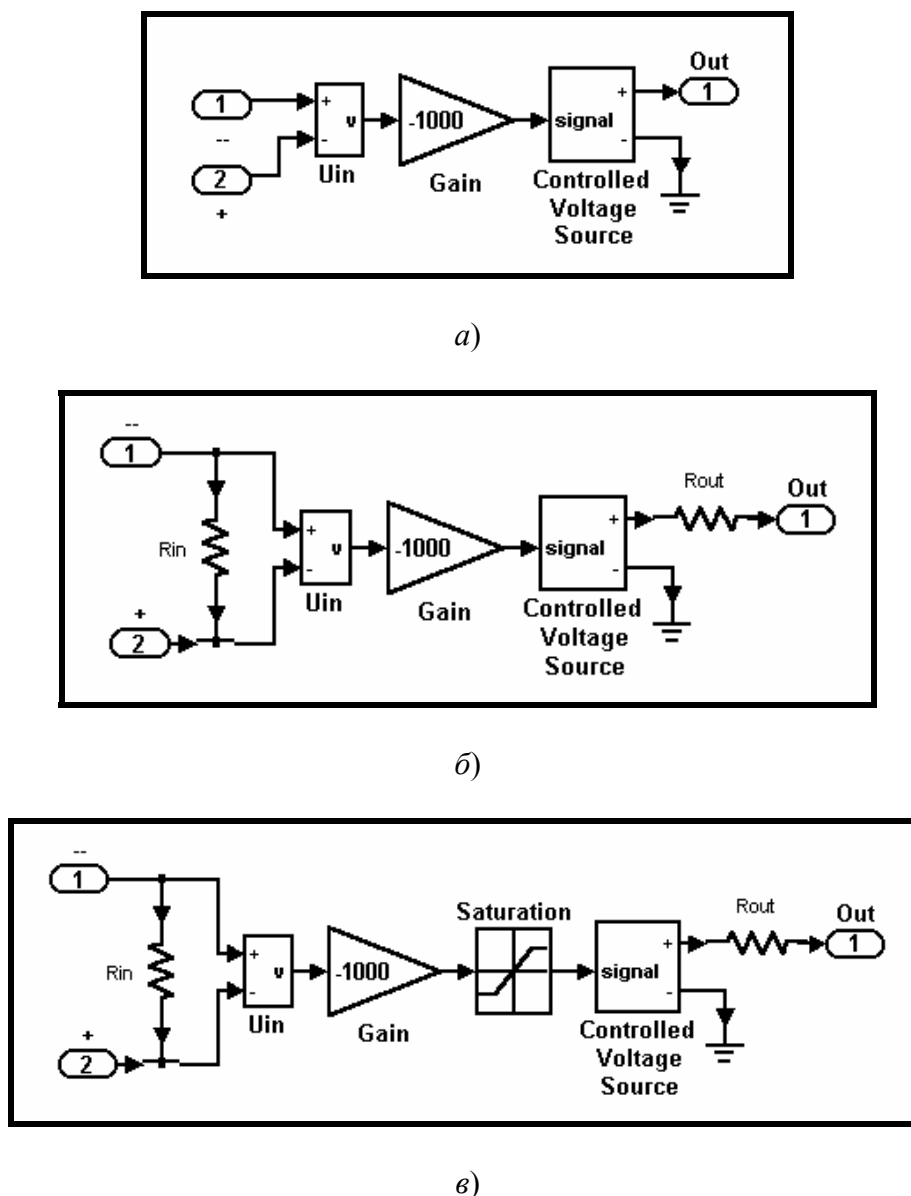


Рис. 9. Простейшие макромодели операционного усилителя.

Как показали исследования, использование пакета Simulink позволяет простыми и наглядными средствами строить достаточно адекватные модели ИАЦП, которые учитывают широкий спектр всевозможных влияющих факторов воздействующих на работу реальных АЦП.

Литература

1. Шахов Э. К., Михотин В. Д. Интегрирующие развертывающие преобразователи.— М.: Энергоатомиздат, 1986.— 142 с.
2. Шляндин В. М. Цифровые измерительные устройства.— М.: Высшая школа, 1981.— 335 с.
3. Гутников В. С. Измерительная электроника в измерительных устройствах.— Л.: Энергия, 1980.— 248 с.

4. *Остапенко А.Г.* Анализ и синтез линейных электронных цепей с помощью графов.— М.: Радио и связь, 1985.— 280 с.
5. *Карташев И.И.* Качество электроэнергии в системах электроснабжения. Способы его контроля и обеспечения / Под ред. М. А. Калугиной.— М.: Изд-во МЭИ, 2000.— 120 с.
6. *Жесткова Ю. Е., Мамыкин А. П., Тимофеев В. М.* Способ расширения динамического диапазона преобразователей с двухтактным интегрированием // Сб. докл. межд. науч.-техн. семинара «Датчики и преобразователи информации, системы измерения, контроля и управления».— М.: МГИЭМ, 2003.— С.98–100.

УДК 681.32:181.48.06

ПРИМЕНЕНИЕ ПАКЕТА FIXED–POINT BLOCKSET В РАЗРАБОТКЕ УСТРОЙСТВ РЕАЛЬНОГО ВРЕМЕНИ

Жуков К. Г.

*Санкт-Петербургский государственный технологический
университет растительных полимеров, Санкт-Петербург,
e-mail: crkt@gturp.spb.ru*

1. Введение

Широкое внедрение микропроцессоров (МП) — массовой продукции, которая нашла применение практически во всех отраслях современного технически развитого общества, вовлекает в работу с МП пользователя, часто не имеющего высокой квалификации в области цифровой вычислительной техники. В то же время особенности микропроцессорной техники (МПТ), возросшая сложность, как электронных компонентов, так и систем на их основе ставят новые проблемы перед разработчиками, производителями и обслуживающим персоналом микропроцессорных систем (МПС). Настоятельной необходимостью становится уменьшение сложности и трудоемкости работ на всех этапах проектирования и эксплуатации систем на базе микропроцессоров. Успешное решение возникающих проблем возможно только на основе широкого набора инструментальных средств для разработки, отладки и проверки МПС. Все виды работ с МПС должны обеспечиваться оптимальными по своим технико-экономическим характеристикам инструментальными средствами. Все множество инструментальных средств необходимо рассматривать как единую систему, отдельные компоненты которой покрывают некоторые участки единого, непрерывного по своему характеру пространства потребностей, возникающих при работе с МПС.

В докладе рассматривается специфика задач разработки и отладки устройств на гибридных микропроцессорах DSP 56800 фирмы Motorola. Описываются методы их решения и применяемые для этих целей технические и программные средства, которые собирательно можно назвать инструментальными средствами разработки и отладки.

2. Выбор средств для реализации МПС

При проектировании цифровых систем на базе МП важно понимать, имеющий место дуализм: аппаратурные средства — программное обеспечение. Суть этого дуализма сводится к следующему: любой процесс, rea-

лизуемый программным путем, может быть преобразован в эквивалентный процесс, реализуемый аппаратными средствами, и наоборот[1].

Ни аппаратные средства, ни программное обеспечение не существуют независимо, поэтому указанное преобразование никогда не может быть завершено полностью. Всегда существуют программы управления аппаратными средствами и аппаратура, при помощи которой реализуются программные алгоритмы. Это утверждение основано на опыте, и его принимают как аксиому.

Можно привести ряд разнообразных характеристик аппаратных и программных средств. Изменения алгоритма по-разному сказывается на аппаратуре и программном обеспечении. Обычно программная реализация алгоритма позволяет вносить значительные изменения, в то время как возможность изменения существующего аппаратного представления алгоритма практически отсутствует. Зачастую различия и способ выполнения алгоритма при этих подходах: обработка посредством аппаратных средств является непрерывным параллельным одношаговым процессом, программная же обработка, как правило, состоит из ряда последовательных дискретных шагов. В качестве примера здесь может служить важнейшая операция — умножение двух двоичных чисел. Эта операция аппаратно может быть выполнена за один шаг без каких-либо промежуточных результатов. Программная реализация алгоритма умножения с использованием аппаратно выполненных сумматоров распадается на несколько отдельных шагов, на каждом из которых получается промежуточный результат.

Описанное различие является одной из причин того, что обычно аппаратная обработка информации осуществляется значительно быстрее программной и обеспечивает наилучшую реализацию МПС реального времени.

Взаимосвязь аппаратных средств и программного обеспечения первоначально основывалась на соображениях экономического характера. В этой связи определяющим было правило: аппаратные средства обеспечивают быстрое действие, программное обеспечение — дешевизну, вследствие чего доминировали простые, преимущественно программные системы с относительно низким быстродействием. В процессе развития за последние годы, указанные соображения претерпели существенные изменения. Динамическая связь (экономического характера) между аппаратными и программными средствами явилась следствием быстрых технологических изменений, происшедших в рассматриваемой области. Объединение аппаратных и программных средств (firmware) привело к изменению, как самих МПС, так и инструментальных систем для их разработки и отладки.

Для успешной разработки цифровых систем на основе МП необходимо сочетать знание современных технологий проектирования с пониманием возможностей аппаратных средств и техники программирования.

3. Методология объектно-ориентированного анализа и проектирования

Наиболее существенным обстоятельством в развитии методологии объектно-ориентированного программирования (ООП) явилось осознание того факта, что процесс написания программного кода может быть отделен от процесса проектирования структуры программы. Действительно, до того как начать программирование классов, их свойств и методов, необходимо определить, чем же являются сами эти классы. Более того, нужно дать ответы на такие вопросы, как: сколько и какие классы нужно определить для решения поставленной задачи, какие свойства и методы необходимы для придания классам требуемого поведения, а также установить взаимосвязи между классами.

Эта совокупность задач не столько связана с написанием кода, сколько с общим анализом требований к будущей программе, а также с анализом конкретной предметной области, для которой разрабатывается программа. Все эти обстоятельства привели к появлению специальной методологии, получившей название методологии объектно-ориентированного анализа и проектирования (ООАП).

Появление методологии (ООАП) потребовало, с одной стороны, разработки различных средств концептуализации предметной области, а с другой — соответствующих специалистов, которые владели бы этой методологией. На данном этапе появляется относительно новый тип специалиста, который получил название аналитика или архитектора. Наряду со специалистами по предметной области аналитик участвует в построении концептуальной схемы будущей программы, которая затем преобразуется программистами в код. При этом отдельные компоненты выбираются таким образом, чтобы при последующей разработке их было удобно представить в форме классов объектов. В этом случае немаловажное значение приобретает и сам язык представления информации о концептуальной схеме предметной области.

Разделение процесса разработки на отдельные этапы способствовало становлению концепции жизненного цикла программы. Под жизненным циклом (ЖЦ) программы понимают совокупность взаимосвязанных и следующих во времени этапов, начиная от разработки требований к ней и заканчивая полным отказом от ее использования. Стандарт ISO/IEC 12207, хотя и описывает общую структуру ЖЦ программы, не конкретизирует детали выполнения тех или иных этапов. Согласно принятым взглядам ЖЦ проекта состоит из следующих фаз (этапов):

- анализа предметной области и формулировки требований;
- разработки структуры;
- реализации проекта (программно-аппаратной/программной);
- внедрения;

- сопровождения;
- отказа от использования.

На этапе анализа предметной области и формулировки требований осуществляется определение функций, которые должна выполнять разрабатываемая система (устройство), а также концептуализация предметной области. Результатом данного этапа должна явиться некоторая концептуальная схема, содержащая описание основных компонентов и тех функций, которые они должны выполнять.

Этап разработки структуры системы заключается в разработке детальной схемы будущей реализации, на которой указываются объекты, их свойства и методы, а также различные взаимосвязи между ними. Как правило, на этом этапе участвуют в работе аналитики и программисты.

Согласно методологии ООАП, именно данная схема должна служить исходной информацией для выбора целевой аппаратуры и написания программного кода.

Этап программирования выполняется в инструментальной среде быстрой разработки приложений (Rapid Application Development, RAD). Появление RAD — инструментариев позволило существенно сократить сроки и затраты на выполнение этого этапа. Результатом данной фазы является прикладное обеспечение, которое обладает требуемой функциональностью для решения необходимых задач в конкретной предметной области.

Этапы внедрения и сопровождения проекта связаны с необходимостью настройки и конфигурирования аппаратно-программного комплекса, а также с устранением возникших в процессе его использования ошибок. Часто в качестве отдельного этапа выделяют тестирование целевого оборудования, под которым понимают проверку его работоспособности на некоторой совокупности исходных данных или при некоторых специальных режимах эксплуатации.

Для создания проекта обычно используется спиральная модель ЖЦ (рис. 1).

На начальных этапах ЖЦ реализуемость технических решений проверяется путем создания прототипов.

Каждый виток спирали соответствует созданию фрагмента или версии системы, на нем уточняются цели и характеристики проекта, определяется его качество, и планируются работы следующего витка спирали. Таким образом, углубляются и последовательно конкретизируются детали проекта, и в результате выбирается обоснованный вариант, который доводится до реализации.

Методология ООАП тесно связана с концепцией автоматизированной разработки программного обеспечения/систем (Computer Aided Software/System Engineering, CASE) CASE — средства позволяют проектировать практически любые системы на компьютере.

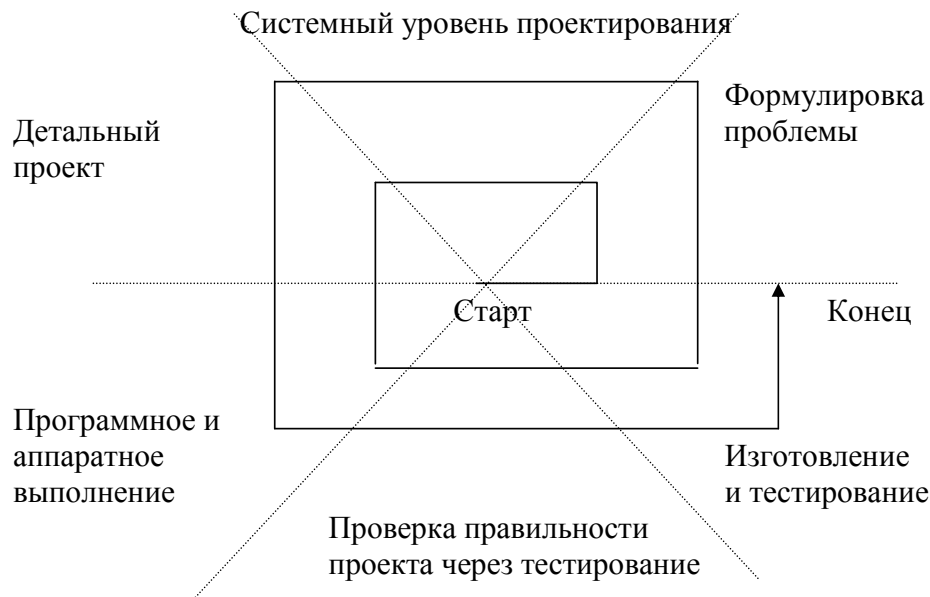


Рис.1 Спиральный процесс проектирования.

Инструментальные CASE — средства и диапазон их практического применения в большей степени зависят от удачного определения семантики и нотации соответствующего языка моделирования. Специфика унифицированного языка моделирования (Unified Modeling Language, UML) заключается в том, что он определяет семантическую метамодель, а не способы представления или реализации компонентов с конкретным интерфейсом.

Из более чем 800 компаний и организаций, входящих в настоящее время в состав консорциума OMG (Object Management Group) особую роль продолжает играть Ration Software Corporation, которая стояла у истоков разработки языка UML. Эта компания разработала и выпустила в продажу одно из первых инструментальных CASE — средств Ration Ruse 98, в котором был реализован язык UML [2, 3, 4, 5].

Встраиваемые системы реального времени, изначально описываемые языком моделирования ROOM (Real — Time Object — Oriented Modeling — объектно-ориентированное моделирование систем реального времени), в настоящий момент специфицированы с использованием стандарта UML. Это достигается путем использования мощных механизмов расширения UML.

4. Преимущества технологий MathWorks

Резкое расширение сферы применения вычислительной техники (ВТ) с развитием МПС привело к появлению большой группы пользовате-

лей, малоквалифицированных в области ВТ и практически не знакомых с программированием на языке низкого уровня.

Эта ситуация особенно остро поставила проблему автоматизации программирования, создания таких систем программирования, которые позволили бы пользователю формулировать алгоритмы на его профессиональном языке, а не в терминах незнакомого языка программирования. Результатом такой спецификации решения требуемой задачи должна стать прикладная программа МПС.

Одним из быстро развивающихся путей решения данной проблемы является использование *автоматических генераторов*, генерирующих прикладную программу на основе описания задачи пользователем.

Идеология автоматических генераторов программ является логическим продолжением и развитием методов, используемых ранее в гибридных вычислительных комплексах. Автоматические генераторы могут производить программы, выполняемые не только на кросс-машине, но и на любой другой машине способной принимать программы на выходном языке системы генерации.

Выбор CASE- средств для разработки устройств (систем) реального времени представляет еще более сложную проблему. Инструментальные пакеты MathWorks используют интуитивно понятный интерфейс и дают возможность профилирующему специалисту, надеясь только на свои силы, наиболее полно реализовать свои замыслы. Пакеты расширения MathWorks, включая Simulink, Fixed-Point Blockset и Real-Time Workshop, новое поколение средств автоматизированной разработки систем на базе МП.

Simulink использует язык программирования сверхвысокого уровня (VHLL). Он является языком программирования следующего поколения [6], удовлетворяющим требованиям ООП.

Real-Time Workshop (RTW) автоматически генерирует С- код из S - модели проектируемого цифрового устройства и практически не требует его дальнейшей коррекции.

Основные преимущества использования технологий MathWorks можно охарактеризовать следующими доминирующими положениями:

- Инструментальные средства используют методологию ООАП.
- Являются высокоэффективной CASE- средой для разработки инженерных и научно-технических приложений реального времени.
- Обеспечивают интерактивный, графический процесс разработки устройств реального времени.
- Имеют уровень абстракции для описания устройств наиболее понятный разработчику по сравнению с многочисленными диаграммами UML.
- Строятся по принципу открытой архитектуры, что гарантирует поддержку аппаратных средств большинства ведущих производителей DSP.

- Обеспечивают полный спиральный цикл (рис.1) разработки устройств реального времени.
- Функциональные возможности могут быть расширены в соответствии с требованиями конкретной разработки.
- Большинство разработанных новшеств, как правило, повторяются конкурентами с естественным значительным отставанием.

5. Практические результаты

На кафедре «Прикладной математики и информатики» разработан для учебных и научных целей пилотный вариант инструментального комплекса (ИКРВ). Структурная схема ИКРВ представлена на рис. 2.

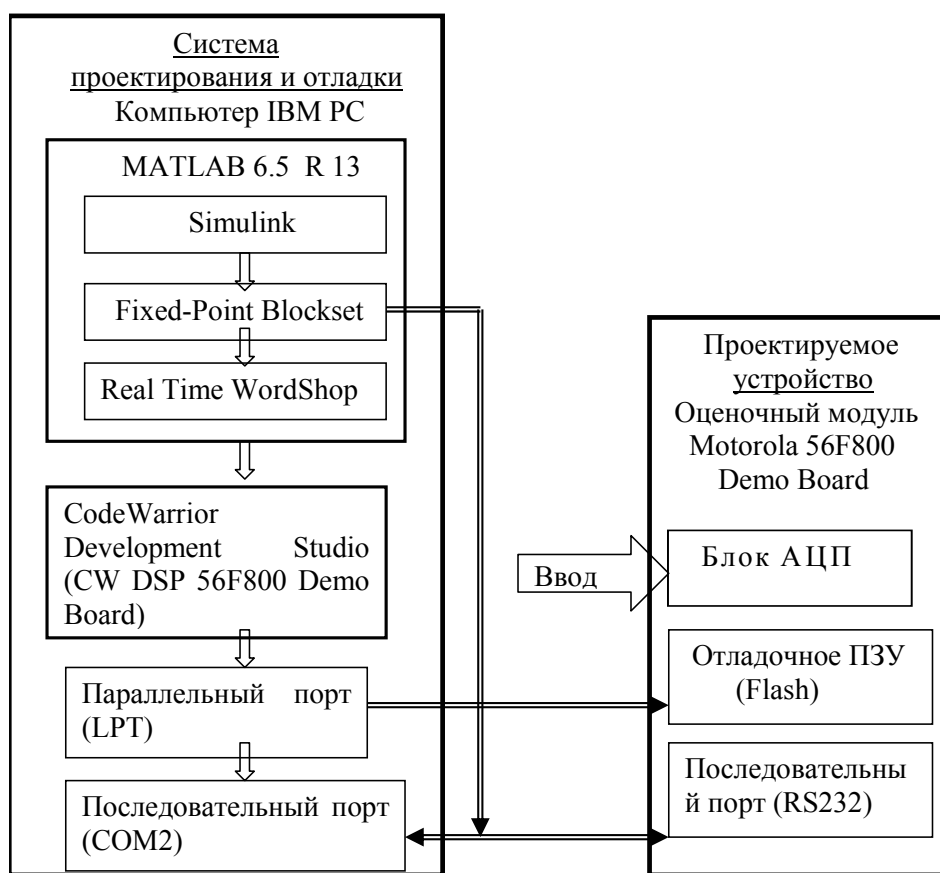


Рис. 2. Структурная схема инструментального комплекса.

В состав комплекса входят следующие аппаратные и программные средства:

- ПК.
- Оценочный модуль с целевым процессором (56F800 Demo Board).
- MATLAB 6.5 R13 с пакетами расширения (Simulink, Fixed-Point Blockset, Real-Time Workshop).

- Среда разработки сигнальных процессоров (CodeWarrior Development Studio for Motorola 56800/E Hybrid V6.1) фирмы Metrowerks.

Оценочный модуль и ПК соединяются двумя стандартными интерфейсными кабелями (LPT, COM). Через параллельный LPT — порт осуществляется загрузка исполняемого кода в целевой процессор. Обмен данными производится через стандартный интерфейс RS232. Структурная схема процесса разработки устройств реального времени показана на рис.3.

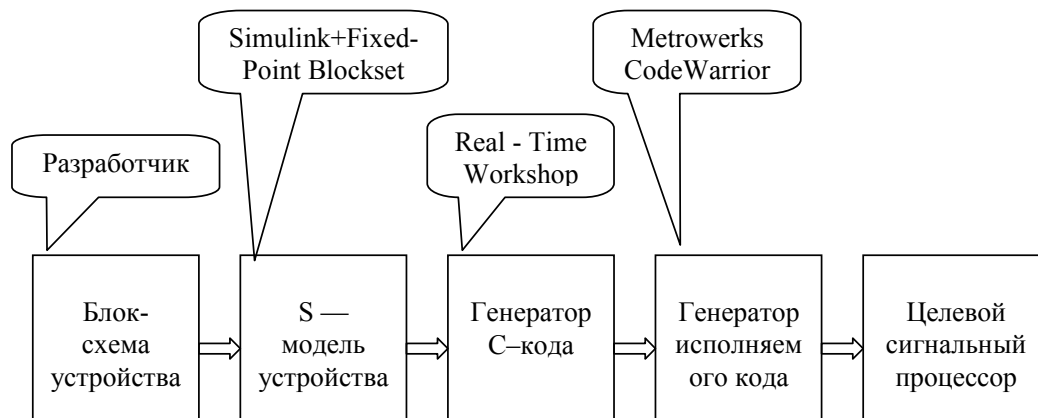


Рис. 3. Схема процесса разработки устройства.

Функциональные возможности ИКРВ позволяют осуществлять процесс проектирования по этой схеме. В настоящее время с помощью комплекса разработаны два устройства. Первое из устройств является цифровым настраиваемым интегратором. Принципы построения интегратора и оценка его методической погрешности рассмотрены в работе [7].

Второе - представляет генератор гармонических сигналов. Отличительной особенностью генератора является возможность одновременного воспроизведения синусоидального и косинусоидального сигналов заданной частоты.

Основная часть проектирования устройств выполнялась в Fixed-Point Blockset. Система команд целевого процессора ориентирована на представление данных в форме с фиксированной точкой. Было произведено совместное тестирование двух устройств и доказана их работоспособность.

В дальнейшем предполагается оценить возможности ИКРВ при проектировании многопроцессорных устройств.

Литература

1. Мясников В. А., Игнатьев М. Б. и др. Микропроцессоры: системы программирования и отладки.— М.: Энергоатомиздат, 1985.— 272 с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++.— 2-е изд.: пер. с англ.— М.: Изд-во Бинум, СПб.: Невский диалект, 1998.— 560 с.

3. Фаулер М., Скотт К. UML в кратком изложении. Применение стандартного языка объектного моделирования: Пер. с англ.— М.: Мир, 1999.— 191 с.
4. Трофимов С. А. CASE — технологии: практическая работа с Rational Rose.— М.: ЗАО «Издательство Бином», 2001.— 272 с.
5. Леоненков А. В. Самоучитель UML.— СПб.: БХВ–Петербург, 2001.— 304 с.
6. Product & Documentation CDs. The Real-Time Workshop Development Process.— The MathWorks, Inc.— P.D2–D34.
7. Жуков К. Г. Анализ погрешности цифровых интеграторов в Simulink // Тезисы докл. Всероссийской науч. конф. «Проектирование научных и инженерных приложений в среде MATLAB» (28–29 мая 2002).— М.: ИПУ РАН, 2002.— 207 с.

УДК 519.6

ИМИТАЦИОННАЯ МОДЕЛЬ ИЗМЕРИТЕЛЬНОГО ПРЕОБРАЗОВАТЕЛЯ ПО ОПРЕДЕЛЕНИЮ ПАРАМЕТРОВ ПЛАЗМЕННОГО ПОТОКА

Ионов Ю. Г., Чекмарев С. В.

*Московский институт радиотехники, электроники и автоматики
(технический университет), каф. Проблемы управления, Москва,
e-mail: Yu-ionov@yandex.ru*

Введение

В работе представлены результаты решения задачи построения имитационной модели измерительного преобразователя по определению параметров потока электродуговой низкотемпературной плазмы. Осуществляется обоснованное [1] описание логики функционирования исследуемой системы и взаимодействие во времени отдельных ее элементов, учитывающее присущие системе наиболее существенные причинно-следственные связи. Целью создания MATLAB-модели является разработка и отладка алгоритма работы ИП, определение требований к основным узлам реального преобразователя.

Данная модель разработана в программной среде MATLAB R13 с использованием пакета расширения Simulink и отображает логику работы измерительного преобразователя с учетом взаимосвязей узлов преобразователя и метрологических характеристик. В ней реализован алгоритм вычисления измеряемых параметров косвенным методом. Измерительный преобразователь (ИП) может быть рассмотрен как узел в составе системы управления (СУ) параметрами потока плазмы генераторов технологических установок. Для примера здесь использована медицинская плазменная хирургическая установка. Преобразователь может быть настроен и в СУ промышленных электродуговых установок, например, плазменной резки, напыления материалов и т. п..

Если рассматривать, как в данном случае, хирургическую установку, где плазмотрон используется в качестве скальпеля или коагулятора, то можно установить, что существует [3-7] зависимость показателей медико-биологического эффекта от многих параметров плазменного потока:

- род плазмообразующего газа;
- среднemasсовая температура ПП;
- тепловая мощность ПП;
- объемный расход плазмообразующего газа;
- динамический напор;
- вольтамперная характеристика электрического разряда.

Эти параметры находятся в тесном взаимодействии друг с другом, причем увеличение или уменьшение одного изменяет ряд других, что в свою очередь влияет на процессы, происходящие в биотканях [7]. Было выявлено, что одним из наиболее значимых параметров, с точки зрения медикобиологического эффекта, является среднемассовая температура. С другой стороны, результаты исследований генератора плазмы как объекта автоматического управления позволили установить, что среднемассовую температуру необходимо рассматривать в качестве оперативно контролируемого и управляемого параметра [2].

В этой связи представляется важным решить задачу разработки ИП.

Имитационная модель ИП

Модель представлена в виде блок-диаграммы и реализует метод косвенного определения температуры по основным параметрам — это ток, напряжение на дуге плазматрона, расход плазмообразующего газа, а также расход и разность температуры на входе и выходе канала охлаждающей плазматрон жидкости (воды). При создании модели в среде MATLAB использованы известные положения теории и практики имитационного моделирования [8–10].

На рис.1 в схеме представлены основные элементы: модель установки (Plasma Surgical System) — источника сигналов, информация о параметрах которых используется для определения температуры плазменного потока; модель измерительного преобразователя, состоящая из мультиплексора (Multiplexer), аналого-цифрового преобразователя (AD Converter), программируемого микроконтроллера (Controller), генератора тактовых импульсов (CLK) и индикатора (Scope). Элементы представлены в виде подсистем.

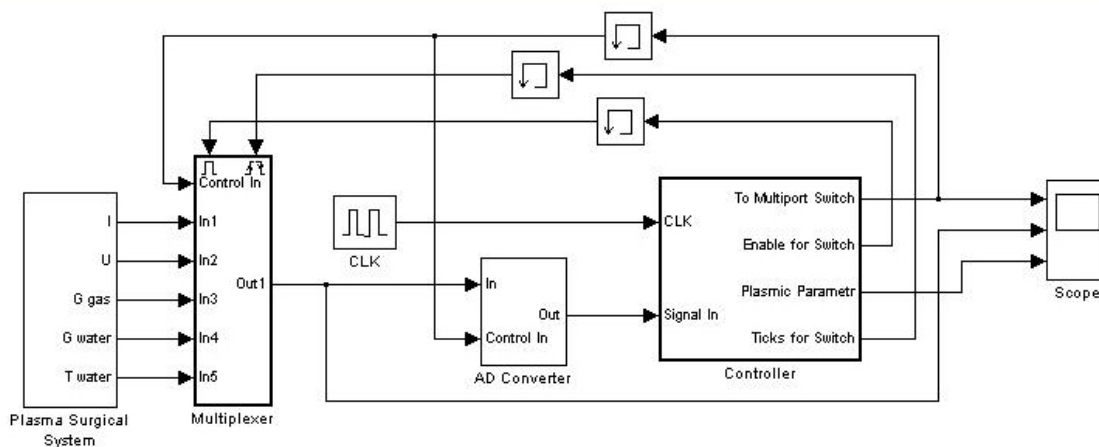


Рис. 1. Блок-диаграмма модели измерительного преобразователя.

Подсистема Plasma Surgical System (Плазменная Хирургическая Система) имитирует работу датчиков, подключенных непосредственно к плазменной хирургической установке (см. рис.2). На выходе данной подсистемы получают сигналы с датчиков.

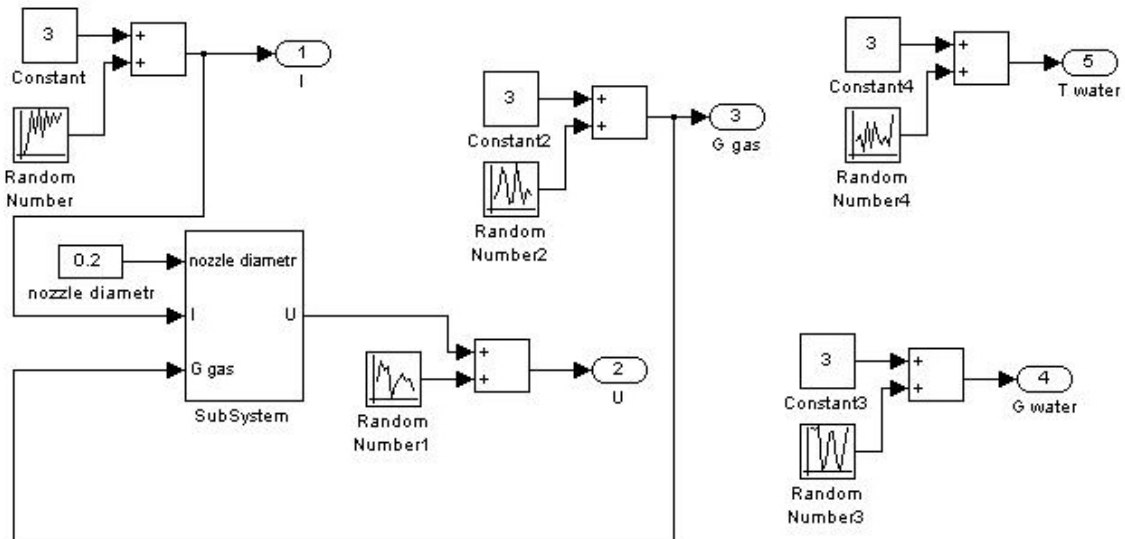


Рис. 2. Блок-диаграмма подсистемы Plasma Surgical System.

Каждый имитированный датчик (кроме датчика напряжения) состоит из блока источника сигнала постоянной величины — Constant, который складывается с сигналом, получаемым на выходе генератора случайных колебаний — Random Number, амплитуда и частота изменения которого задается окном настройки параметров определенного блока и приблизительно соответствует диапазону возможных значений параметров электродуговой плазмы конкретной установки (в данном случае были использованы паспортные данные установки «Факел»). Сигнал с датчика напряжения (U) скореллирован с сигналами датчиков тока (I), расхода плазмообразующего газа (G) и параметрами конструкции генератора т. к. напряжение на дуге плазмотрона зависит от этих факторов. Для описания динамики изменения параметров плазмы была использована также теория подобия и размерностей и данные экспериментов.

Далее (см. рис. 3) использована подсистема Multiplexer, которая состоит из стандартного блока Multiport Switch библиотеки Nonlinear, блока Enable и Trigger библиотеки Signals&Systems. Multiport Switch предназначен для поочередного переключения датчиков на один канал.

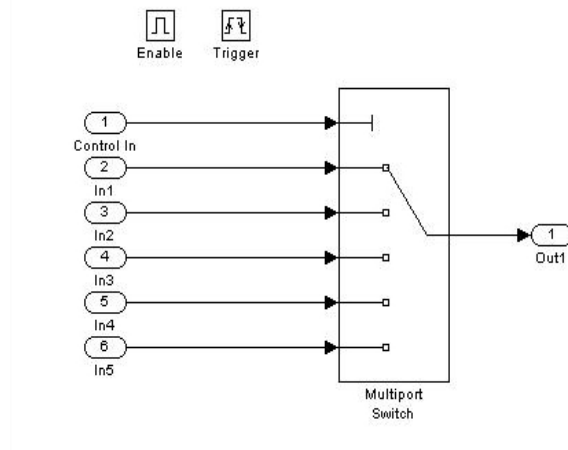


Рис. 3. Блок-диаграмма подсистемы Multiplexer.

Блок Enable и Trigger предназначены для создания управляемых подсистем., Активизация подсистемы происходит только при наличии некоторого внешнего управляющего события (или сигнала). Механизм управляемых подсистем обеспечивает эффективное и простое решение проблемы синхронизации параллельных процессов [1]. В нашем случае управляющие сигналы на Multiplexer подаются с блока Controller.

Сигнал с определенного датчика через Multiplexer поступает на блок AD Converter, который имитирует работу АЦП. Данная подсистема (см. рис.4) состоит из нескольких блоков.

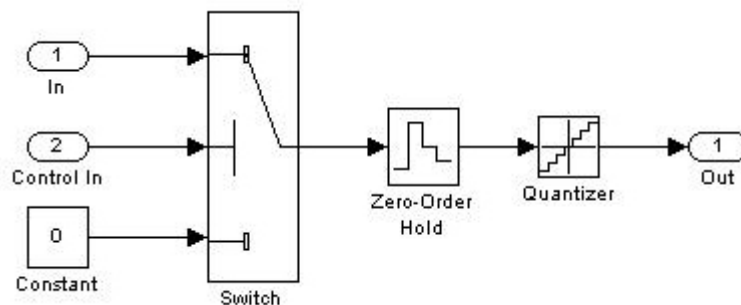


Рис. 4. Блок-диаграмм подсистемы AD Converter.

Switch — это переключатель с сигнала датчика на нулевой сигнал. Тем самым имитируется управление включением и выключением входного канала АЦП. Zero-Order Hold — блок выборки и хранения — осуществляет выборку мгновенного значения входного сигнала в заданный момент времени и фиксацию его на выходе вплоть до следующего момента выборки. Quantizer — выполняет квантование входного сигнала по уровню.

Блок CLK (рис. 1) предназначен для задания тактовой частоты, синхронизирующей процессы, протекающие в контроллере.

С АЦП оцифрованный сигнал поступает на блок Controller. Controller представляет собой сложную подсистему, с двумя подсистемами (см. рис.5).

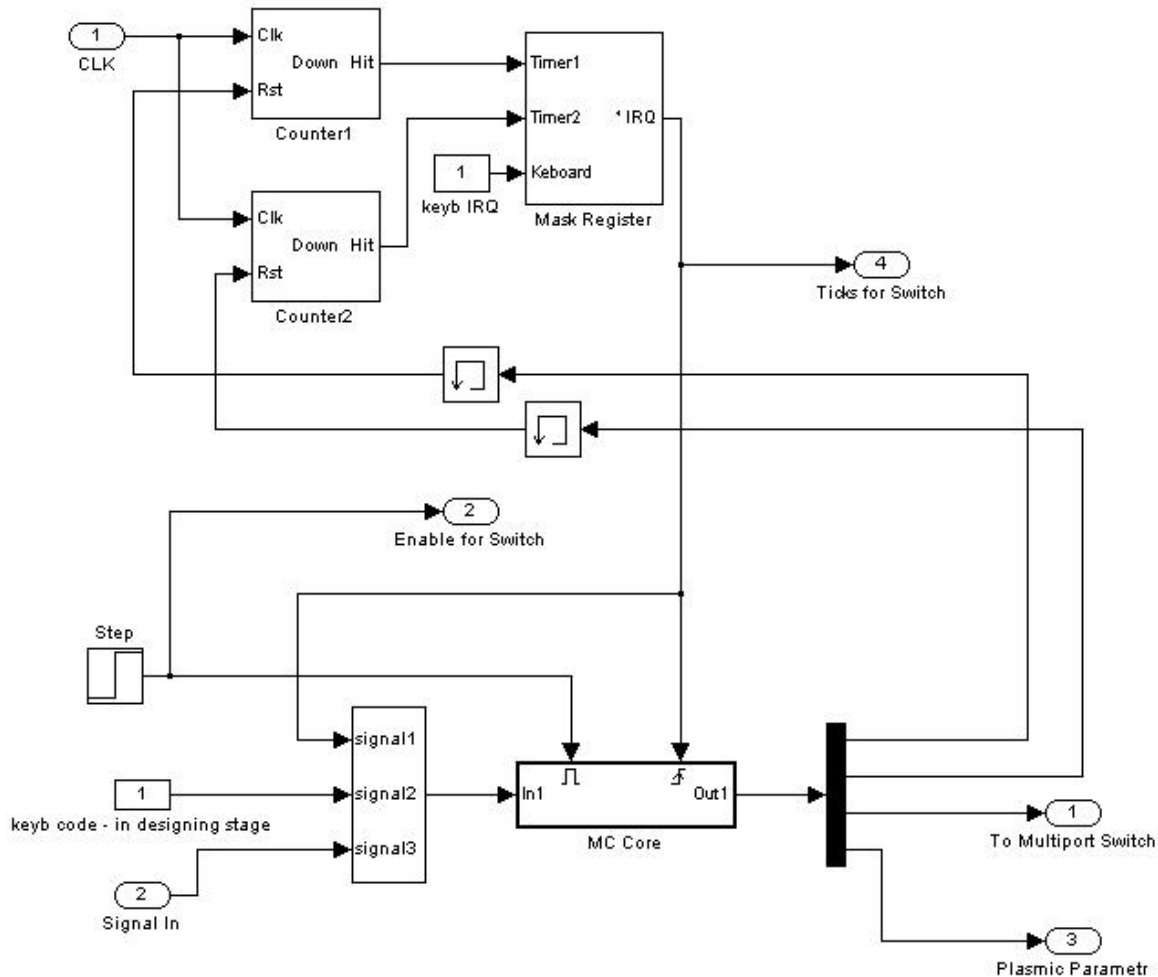


Рис. 5. Блок-диаграмма подсистемы Controller.

После того, как оцифрованный сигнал с определенного датчика поступает на вход Signal In, он смешивается с другими сигналами: signal1 — сигнал прерывания от таймеров или от клавиатуры, signal2 — сигнал, представляющий собой код нажатой на клавиатуре кнопки. С помощью технологии использования прерываний от таймеров осуществляется привязка к реальному времени и вся работа ИП делится на циклы, время выполнения которых строго соответствует времени реального прибора или модельному времени данной имитационной модели ИП.

Сигналы смешиваются в вектор данных, что требуется правилами использования S-функций пакета Simulink. Далее, этот вектор данных поступает на управляемую подсистему Core (ядро) (см. рис.6), где и обраба-

тываются согласно алгоритму работы программы хранящейся в файле «MC_Model.m» и написанной на внутреннем языке MATLAB.

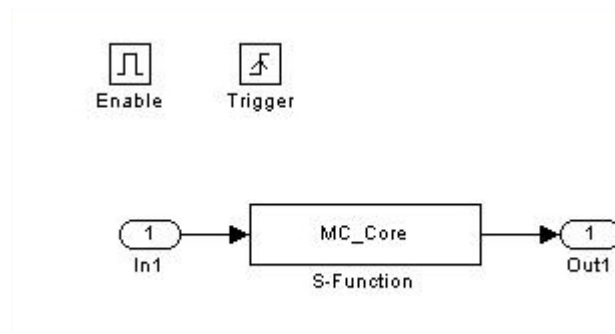


Рис. 6. Блок-диаграмма подсистемы Core.

На выходе блока S-function получается вектор значений, первый и второй элемент которого соответствует управляющему сигналу для сброса первого и второго таймера соответственно (вход Rst Counter1 и Counter2).

Третий элемент вектора соответствует выходу To Multiport Switch контроллера и подается на управляющий вход блока переключения канала Multiplexer (см. рис. 1).

Четвертый элемент соответствует выходу Plasmic Parametr и соответствует измеренному (посчитанному) значению параметра плазмы (в данном случае среднemasсовой температуры).

С выходов контроллера To Multiport Switch и Plasmic Parametr сигнал подается на вход блока Score. Для визуального контроля на вход блока Score подается также сигнал сразу с блока Multiplexer для отображения значения с коммутируемого датчика.

Вспомогательная подсистема Mask Register (см. рис.7) имитирует работу регистра маскирования прерываний и представляет собой S-функцию, текст программы которой хранится в файле «MC_Maskreg.m».

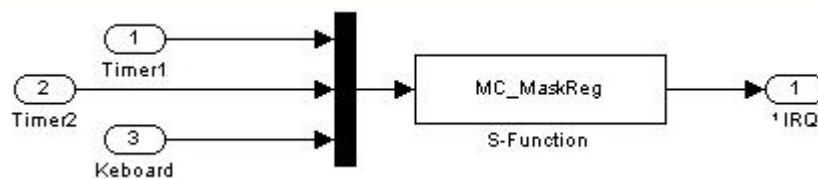


Рис. 6. Блок-Диаграмма подсистемы Mask register

Блоки Counter1 и Counter2 предназначены для генерирования сигнала прерывания по истечению определенного интервала времени. Counter1 — «отмеряет» интервал времени равный времени исполнения одного основного цикла алгоритма вычисления параметров (включая опрос датчиков, клавиатуры и т. д.). После генерации сигнала прерывания цикл вычисления начинается с самого начала (запускается программа обработки

прерывания — это и есть основная программа, в которой реализован алгоритм вычисления косвенным методом).

Counter2 — «отмеряет» время, в течении которого включается пауза необходимая для оцифровки АЦП сигнала с датчика, по истечению которой происходит чтение из АЦП оцифрованного сигнала.

Результаты моделирования и перспективы использования модели

Результаты моделирования можно просмотреть путем раскрытия блока Scope на основной блок-диаграмме (см. рис. 8).

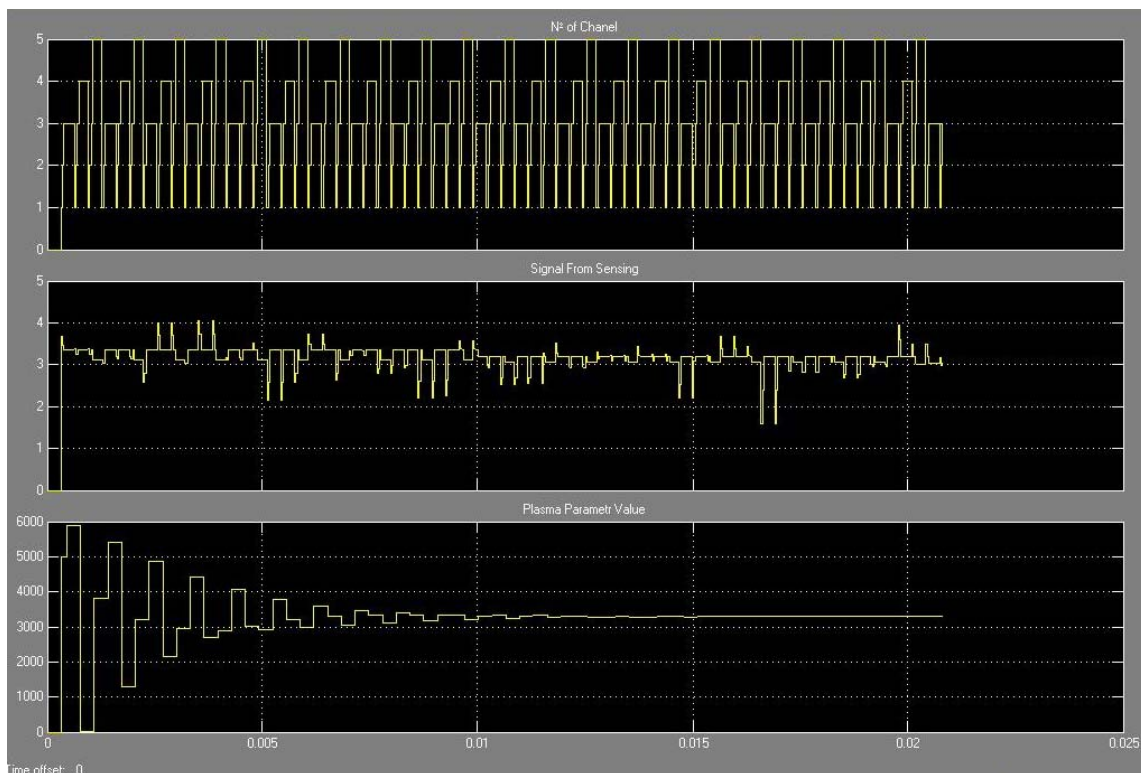


Рис. 8. Пример результата моделирования.

В нижнем окне показан переходной процесс в системе определения среднemasсовой температуры плазмы. Как видно, ее установившееся значение можно оценить величиной $\approx 3300^\circ\text{K}$. Эта оценка для определенного режима совпадает с экспериментальными данными, полученными для установки «Факел».

Описанная модель нами использовалась также для оценки других параметров потока плазмы: среднemasсовой энтальпии, среднemasсовой скорости потока на срезе сопла плазматрона, полезной мощности потока и коэффициента полезного действия генератора.

Следует отметить, что возможна адаптация ИП к СУ разного типа электродуговых установок. Для этого применяются инструменты совре-

менных интеллектуальных технологий, которые представлены в среде MATLAB. Соответствующий алгоритм реализуется контроллером.

Другое перспективное направление исследований, в котором предполагается использовать разработанную модель, связано с использованием Real-Time Workshop (RTW) — мастерской реального времени, т. к. ее основной механизм реализует универсальный способ сборки программного модуля реального времени на основе текущих данных моделирования.

Литература

1. *Ионов Ю. Г.* Автоматизированные плазмотехнологические комплексы обработки материалов (основные вопросы теории и проектирования). Дисс. ... д. т. н.— СПб.: СПбГЭТУ, 1999.
2. *Ионов Yu. G., Ионов A. Yu.* Statement and solution of a boundary value problem in a model of plasma generator as controlled system // Title of the book: Mathematical models of non-linear excitation, transfer, dynamics, and control in condensed systems and other media.— New York, Plenum Publishing Corporation, 1999.
3. *Вафин А. З.* Применение плазменных технологий в хирургии эхинококкоза легких // Хирургия.— 2002.— №1.
4. *Вафин А. З.* Применение плазменных технологий в хирургии эхинококкоза печени // Вестник хирургии.— 2002.— №4.
5. *Савельев В. С.* Перспективы использования плазменного скальпеля в хирургической практике // Хирургия.— 1986.— №10.
6. *Брехов Е. И.* Экспериментальное и клиническое изучение и перспективы применения плазменных потоков // Вестник хирургии.— 1986.— №1.
7. *Брехов Е. И.* Применение плазменных потоков в хирургии.— М., 1992.
8. *Гультяев А.* MATLAB 5.2. Имитационное моделирование в среде Windows.— СПб., 1999.
9. *Сергиенко А. Б.* Цифровая обработка сигналов.— СПб.: Питер, 2002.
10. *Кондрашев В. Е., Королев С. Б.* MATLAB как система научных-технических расчетов.— М.: МИР, 2002.

УДК 517.977

ИСПОЛЬЗОВАНИЕ MATLAB ДЛЯ СИНТЕЗА МИКРОПРОЦЕССОРНЫХ СИСТЕМ УПРАВЛЕНИЯ МЕТОДОМ ПОЛИНОМИАЛЬНЫХ УРАВНЕНИЙ

Ишматов З. Ш., Волков М. А.

Уральский государственный технический университет — УПИ, Екатеринбург,
e-mail: ishmatov@mail.ru

Сравнительный анализ известных способов синтеза цифровых регуляторов позволяет выделить среди них сравнительно простой и эффективный метод полиномиальных уравнений (π -уравнений), впервые представленный в [1], и получивший свое дальнейшее развитие в [2]. Активное применение метода полиномиальных уравнений началось на кафедре электропривода и автоматизации промышленных установок УГТУ–УПИ с 1986 года [3] и за это время накоплен богатый опыт его использования как в электроприводах постоянного [6-8] и переменного [9-14] тока, так и для управления различными технологическими процессами.

Дискретная передаточная функция (ДПФ) объекта управления представляется в виде:

$$W_o(z) = \frac{P(z)}{(z-1)^i Q(z)}, \quad (1)$$

где: $P(z)$ и $Q(z)$ — взаимно простые полиномы от z степени n_P и n_Q соответственно, не имеющие нулей в точке $z=1$, причем

$$n_P \leq n_Q + i, \quad (2)$$

i — количество интегрирующих звеньев в объекте. Характерным для объектов в цифровых системах управления является наличие запаздывания, что выражается неравенством (2), и в общем случае ненулевой порядок полинома $P(z)$, причем нули ДПФ (1) могут быть и устойчивыми, и неустойчивыми. Полюсы ДПФ (1) в электроприводе, как правило, устойчивы, но достаточно часто встречаются технологические агрегаты с неустойчивыми передаточными функциями.

При синтезе часть нулей и полюсов объекта могут компенсироваться регулятором. Неустойчивые нули и полюсы компенсировать недопустимо из условия работоспособности. Но, как показали исследования [4,5], компенсация устойчивых нулей также крайне нежелательна из-за возникновения скрытых (между моментами квантования) колебаний выходной координаты, и, что более важно, значительных колебаний управляющего воздействия и промежуточных координат электропривода на частотах, вдвое больших частоты квантования. Поэтому в цифровых системах целесооб-

разно ограничиться компенсацией лишь некоторых из устойчивых полюсов объекта. Представим полином знаменателя (1) в виде:

$$Q(z) = Q_+(z)Q_-(z) = Q_-(z)Q_{k+}(z)Q_{h+}(z),$$

где: $Q_+(z)$ и $Q_-(z)$ — полиномы от z степени n_{Q_+} и n_{Q_-} соответственно, содержащие устойчивые и неустойчивые полюсы объекта; $Q_{k+}(z)$ и $Q_{h+}(z)$ — полиномы от z степени $n_{Q_{k+}}$ и $n_{Q_{h+}}$ соответственно, содержащие компенсируемые и некомпенсируемые устойчивые полюсы объекта. Тогда ДПФ цифрового регулятора можно записать следующим образом:

$$D(z) = \frac{M(z)Q_{k+}(z)}{N(z)(z-1)^j},$$

где: $M(z)$ и $N(z)$ — искомые полиномы степени n_M и n_N соответственно; j — количество интегрирующих звеньев регулятора, обеспечивающих требуемый порядок астатизма $i+j$ замкнутой системы. Очевидно, что степени искомым полиномов должны удовлетворять условию:

$$n_M + n_{Q_{k+}} = n_N + j. \quad (3)$$

Анализ ДПФ разомкнутой системы

$$\Phi(z) = D(z)W_o(z) = \frac{P(z)M(z)}{(z-1)^{i+j}N(z)Q_{h+}(z)Q_-(z)}$$

показывает, что степень его числителя в силу (2) и (3) всегда не больше степени знаменателя:

$$n_M + n_P \leq n_N + i + j + n_{Q_{h+}} + n_{Q_-}.$$

Основой для синтеза служит ДПФ замкнутой системы по ошибке:

$$G_e(z) = \frac{1}{1 + \Phi(z)} = \frac{(z-1)^{i+j}N(z)Q_{h+}(z)Q_-(z)}{(z-1)^{i+j}N(z)Q_{h+}(z)Q_-(z) + P(z)M(z)}.$$

Степень множителя $(z-1)$ в числителе этой ДПФ определяет желаемый порядок астатизма системы по управляющему воздействию, при этом имеется свобода выбора j . Приравняв знаменатель передаточной функции к желаемому характеристическому уравнению замкнутой системы $A(z)$, получим полиномиальное уравнение синтеза:

$$(z-1)^{i+j}N(z)Q_{h+}(z)Q_-(z) + P(z)M(z) = A(z). \quad (4)$$

Методики решения этого полиномиального уравнения относительно искомым полиномов $M(z)$ и $N(z)$ достаточно подробно описаны в [3–7], выбор характеристического полинома в зависимости от требований к качеству регулирования представлен в [3, 5, 15]. Приведем лишь выражения для выбора степеней полиномов, соответствующих минимальному решению уравнения (4) с учетом условия (3) и обеспечивающих теоретически любое качество регулирования, определяемое $A(z)$:

$$n_A = 2n_Q - n_{Q_{k+}} + 2i + j - 1; \quad n_N = n_Q + i - 1; \quad n_M = 2n_Q - n_{Q_{k+}} + i + j - 1. \quad (5)$$

При решении практических задач иногда целесообразно понизить порядок полиномов на единицу по сравнению с (5). Это позволяет упростить регулятор, но накладывает ограничения на выбор $A(z)$.

В процессе синтеза этим методом решается полиномиальное уравнение (4) с двумя неизвестными полиномами, имеющее, как известно из линейной алгебры, бесконечное число решений. Кроме того, при синтезе систем управления для сложных объектов (например, многомассовые системы электропривода, частотно-регулируемые асинхронные электроприводы и т. д.) задача сводится к решению системы алгебраических уравнений седьмого и даже более высокого порядка. В процессе синтеза идет поиск минимального решения полиномиального уравнения, что не исключает перебора нескольких вариантов решения. Учитывая, что общая степень полиномиального уравнения может быть значительной, понятно, что его решение часто оказывается процессом достаточно трудоемким.

Все это значительно увеличивает сложность расчетов, вынуждает принимать ряд допущений, направленных на упрощение модели объекта, может приводить к ошибкам, связанным с использованием громоздких выражений в процессе решения полиномиального уравнения, даже с применением ЭВМ. Поэтому возникает задача автоматизации процесса синтеза цифрового регулятора, решение которой позволило бы относительно просто, при минимальных затратах времени и ручного труда выполнить синтез регулятора методом дискретных полиномиальных уравнений.

Для решения поставленной задачи в программной среде MATLAB была создана программа «**Regulator**».

Программа «Regulator»

Программа автоматически выполняет следующие основные этапы синтеза:

- факторизацию объекта регулирования;
- формирование полиномиального уравнения синтеза;
- расчет характеристического полинома;
- нахождение минимальных решений составленного уравнения;
- выбор необходимого решения;
- синтез физически реализуемого регулятора, не вносящего дополнительного запаздывания в контур регулирования;
- расчет ДПФ замкнутой системы $G(z)$ и передаточной функции ошибки $G_e(z)$, построение переходной характеристики замкнутой системы;
- проверку корректности решения уравнения.

Следует заметить, что если синтез регулятора без запаздывания невозможен, синтез производится при условии минимального запаздывания, вносимого в контур регулирования регулятором. В программе предусмотрен контроль корректности решения полиномиального уравнения и правильности выбора минимальных степеней полиномов $M(z)$ и $N(z)$, входящих в ДПФ регулятора. Это позволяет в некоторой степени оградить поль-

зователя от неверных результатов, появление которых возможно в силу ошибок вычисления MATLAB, связанных с несовершенством методик вычисления и алгоритмов, заложенных в саму программу.

В процессе синтеза задаются следующие параметры:

- -требуемого порядка астатизма по управляющему воздействию ;
- -периода дискретности системы T ;
- -величины среднегеометрического корня Ω , определяющего быстродействие системы;
- -степени и распределения корней характеристического полинома $A(z)$, обуславливающего вид переходного процесса.

В случае невозможности решения уравнения при заданной степени $A(z)$ предлагается изменить степень полинома. Также предоставляется возможность выбора требуемого варианта синтеза (низкая чувствительность, отсутствие скрытых колебаний и др.) и распределения корней характеристического уравнения (биномиальное, по Баттерворту и т. д.), которые определяют в итоге качество синтезируемой системы электропривода.

Поскольку процедуру синтеза можно разделить на отдельные этапы, такие, как факторизация, составление и решение полиномиального уравнения, проверка правильности решения и др., программу целесообразно выполнить как набор связанных между собой модулей, каждый из которых решает свою конкретную задачу. Кроме того, модульность позволяет просто выполнить отладку программы, включить дополнительный модуль в программу, применить уже используемые в других программах модули, использовать один модуль для выполнения какой-либо операции синтеза, когда применения самой программы не требуется.

На рис. 1. представлена структура программы. Как видно из рисунка, программа состоит из шести основных модулей.

1. *Модуль факторизации* объекта управления. Предлагается ввести векторы коэффициентов полиномов числителя и знаменателя ДПФ объекта, после чего выполняется операция разделения полиномов передаточной функции объекта (1) на полиномы с устойчивыми $P_+(z)$, $Q_+(z)$ и неустойчивыми $P_-(z)$, $Q_-(z)$ нулями и полюсами, то есть ДПФ объекта представляется в следующем виде:

$$W_0(z) = \frac{P_-(z)P_+(z)}{(z-1)^i Q_-(z)Q_+(z)},$$

что необходимо для составления π -уравнения синтеза.

2. *Модуль формирования уравнения синтеза*. Дает возможность выбрать один из вариантов синтеза N , после чего на основании степеней полиномов $P(z)$, $Q(z)$, $A(z)$ анализируется возможность получения регулятора $D(z)$, удовлетворяющего условию (3), определяются полиномы $B(z)$ и $C(z)$, составляющие уравнение синтеза. Основное уравнение синтеза в алгоритме программы представлено в общем виде

$$B(z)N(z) + C(z)M(z) = A(z),$$

что дает возможность без изменения решающего алгоритма находить решения для других возможных вариантов синтеза системы. В нашем случае, как видно из (4)

$$B(z) = (z-1)^{i+j} Q_{n+}(z) Q_-(z) \quad \text{и} \quad C(z) = P(z).$$

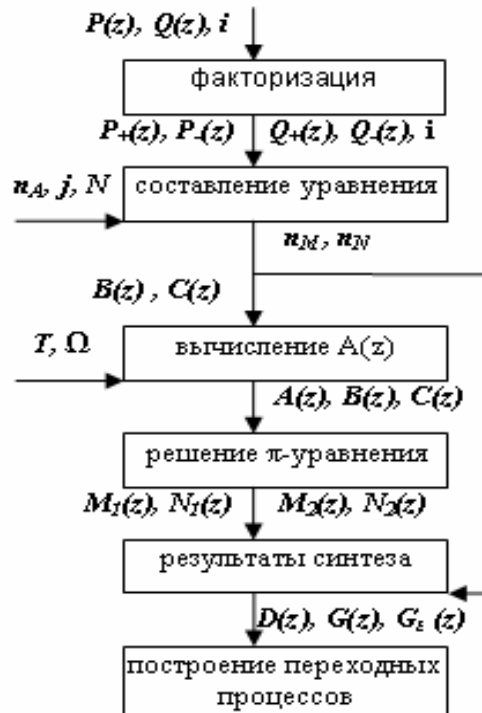


Рис. 1. Структура программы

Когда нет возможности выполнить условие (3), предлагается изменить степень $A(z)$. На этом этапе также вычисляются степени полиномов $N(z)$ и $M(z)$, входящих в передаточную функцию регулятора.

3. *Модуль вычисления характеристического полинома.* Предлагает задать желаемое распределение полюсов ДПФ замкнутой системы, период дискретности T , величину среднегеометрического корня Ω , после чего на основании введенной информации вычисляет характеристический полином системы $A(z)$.

4. *Модуль решения полиномиального уравнения.* Рассчитываются полиномы $M(z)$ и $N(z)$. В результате решения уравнения синтеза находятся минимальные решения π -уравнения с использованием алгоритма Евклида (в общем случае 2 решения) [1]. Под минимальным решением в данном случае понимаются полиномы $M(z)$ и $N(z)$ имеющие минимально возможные степени.

5. *Модуль проверки и вывода конечных результатов.* Выполняет выбор нужного решения на основании данных модуля 2, проверку правильности решения уравнения, формирование передаточной функции регуля-

тора $D(z)$, а также ДПФ замкнутой системы по управляющему воздействию $G(z)$ и по ошибке $G_e(z)$.

6. *Модуль построения переходных характеристик.* Производит расчет реакции системы на единичную ступенчатую функцию, после чего выводится график переходного процесса регулируемой координаты.

Пример синтеза регулятора

В качестве примера выполним синтез регулятора для двухмассовой электромеханической системы с вязким трением, структура которой представлена на рис. 2. Передаточная функция объекта может быть найдена методом направленных графов. При условии пренебрежения влиянием ЭДС двигателя передаточная функция объекта $W_o(p)$ принимает вид:

$$W_o(p) = \frac{K_o \left(p^2 + \frac{1}{T_c} p + \frac{\Omega_{12}^2}{\gamma} \right)}{\left(p^3 + \frac{1}{T_c} p^2 + \Omega_{12}^2 p + \frac{(\gamma - 1)\Omega_{12}^2}{\gamma T_c} \right) \left(p + \frac{1}{T_\gamma + T_\mu} \right)}, \quad (6)$$

где: $K_o = \frac{K_{mn} \cdot K\Phi \cdot K_{dc}}{(T_\gamma + T_\mu) J_1 \cdot R_\gamma}$ — коэффициент усиления объекта; $\gamma = (J_1 + J_2)/J_1$ —

коэффициент соотношения масс; $\Omega_{12} = \sqrt{\frac{c_{12}^2}{J_2}}$ — резонансная частота упругих колебаний; $T_c = J_2/\beta_c$ — механическая постоянная времени второй массы.

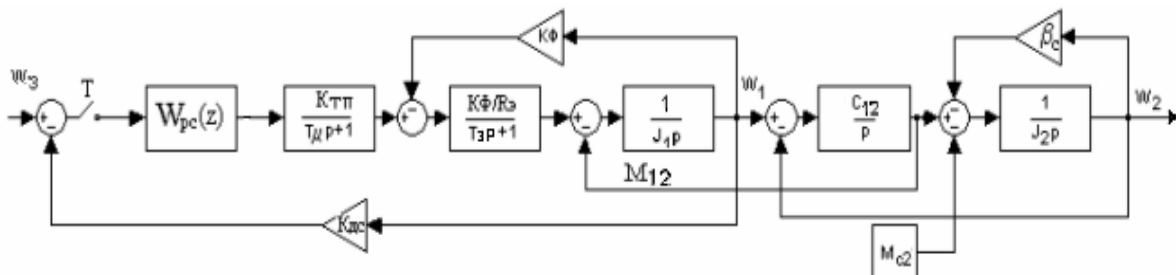


Рис.2. Структурная схема двухмассовой электромеханической системы с вязким трением.

Для синтеза цифрового регулятора выполним преобразование передаточной функции объекта регулирования из p -области в z -область, приняв период дискретности цифровой системы $T=0,01$ с и подставив конкретные параметры объекта. Для расчетов приняты двигатель 2ПН112: $P_H=2,2$ кВт; $U_H=220$ В; $K\Phi=1,03$; $J_1=0,018$ кг·м²; $T_\gamma=0,009$ с.; преобразователь: $K_{mn}=27,7$; $T_\mu=0,003$; $R_\gamma=4,36$ Ом; механическая часть: $J_2=0,021$ кг·м²; $C_{12}=72,5$; $\beta_c=0,6$ Нм·с. Коэффициент передачи датчика скорости $K_{dc}=0,065$ В·с.

$$W_0(z) = 0.12 \frac{z^3 - 1.45105z^2 + 0.745059z}{z^4 - 2.47868z^3 + 2.76030z^2 - 1.53696z + 0.31443}$$

В результате синтеза при выборе степени характеристического полинома $n_A = 4$ и настройки на скорейшее затухание [6] получим следующую передаточную функцию регулятора:

$$D(z) = \frac{10^{-7} (0.3357z^4 - 0.8322z^3 + 0.9267z^2 - 0.5160z + 0.1056)}{K_0 (z^4 - 3.9740z^3 + 5.9223z^2 - 3.9226z + 0.9743)}$$

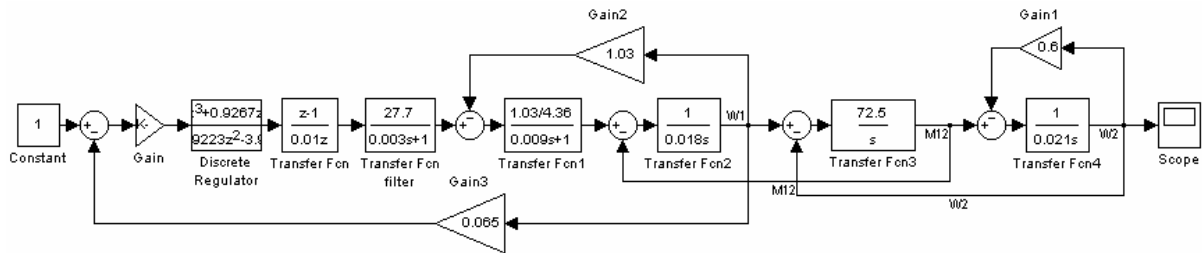


Рис. 3. Модель двухмассовой электромеханической системы с цифровым регулятором в Simulink.

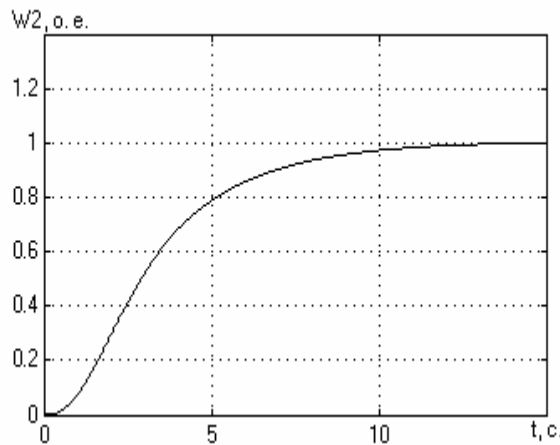


Рис. 4. Переходный процесс по скорости второй массы ω_2 .

На рис. 3. и 4. приведены модель системы в Simulink и переходная характеристика системы по скорости второй массы ω_2 .

Некоторое отличие переходного процесса от выбранной стандартной настройки связано с пренебрежением влияния ЭДС двигателя при нахождении передаточной функции объекта (6).

Таким образом, при применении программы либо ее отдельного модуля существенно уменьшается трудоемкость расчетов (необходимо лишь найти ДПФ объекта), значительно сокращается время синтеза регулятора, уменьшается вероятность ошибок синтеза, упрощается процедура синтеза регуляторов для сложных объектов управления.

Литература

1. *Волгин Л. Н.* Элементы теории управляющих машин.— М.: Сов. радио, 1962.— 164 с.
2. *Волгин Л. Н.* Оптимальное дискретное управление динамическими системами / Под ред. П. Д. Крутько.— М.: Наука, 1986.— 240 с.
3. *Ишматов З. Ш.* Тиристорный электропривод постоянного тока с прямым микропроцессорным подчиненным регулированием координат. Дисс. .. канд. техн. наук.— Свердловск, 1987.— 243 с.
4. *Ишматов З. Ш.* Метод полиномиальных уравнений для синтеза микропроцессорных систем электропривода // Тез. докл. 1-й международной (12-й Всероссийской) конф. по автоматизированному электроприводу.— СПб.: СПГЭУ, 1995.
5. *Ишматов З. Ш.* Использование метода полиномиальных уравнений для синтеза микропроцессорных систем управления электроприводами // Электротехника.— 2003.— №6.
6. *Волков А. И., Ишматов З. Ш., Кулесский Р. А. и др.* Предельное быстродействие электропривода постоянного тока с прямым микропроцессорным управлением // Автоматизация машиностроения на базе новых элементов и устройств при реализации программы «Интенсификация-90»: Материалы краткосрочного семинара.— Л.: ЛДНТП, 1988.
7. *Волков А. И., Ишматов З. Ш., Горецкий Я. Л. и др.* Принципы микропроцессорного управления электроприводами скоростных пассажирских лифтов // Электроприводы переменного тока с полупроводниковыми преобразователями: Тез. докл. к VIII научно-техн. конф.— Свердловск, 1989.
8. *Ишматов З. Ш., Казаков Е. Г., Куцин В. В. и др.* Электропривод постоянного тока с микропроцессорным управлением для механизмов типа экскаваторов-драглайнов // Инструктивные и информационные материалы по проектированию электроустановок.— М., ВНИПИ Тяжпромэлектропроект.— 1995.— №1.
9. *Ишматов З. Ш.* О некоторых особенностях синтеза алгоритмов управления частотно-регулируемым асинхронным электроприводом // Электротехника.— 1998.— №8.
10. *Браславский И. Я., Зюзев А. М., Ишматов З. Ш. и др.* Синтез микропроцессорных систем управления асинхронным электроприводом с применением метода полиномиальных уравнений // Электротехника.— 1998.— №6.
11. *Braslavsky I., Ishmatov Z., Shilin S.* Electrical drives microprocessor control systems synthesis method // Proc. of the symposium on power electronics, electrical drives, advanced machines, power quality.— Sorrento, Italy, 1998.
12. *Ишматов З. Ш., Кириллов А. В., Поляков В. Н.* Синтез регуляторов микропроцессорной системы управления частотно-регулируемым асин-

- хронным электроприводом // Вестник Уральского гос. техн. ун-та. Часть 2. Электромеханика и электротехнология.— 1995.
13. *Braslavsky I., Ishmatov Z., Shilin S.* The polinomial equations method for synthesis of microprocessor control system of electrical drives // Proc. of the 8th international conference «Power Electronics, Motion Control», V.4 «Motor Drives».— Prague, Czech Republic.— 1998.
14. *Ишматов З. Ш., Кириллов А. В.* Особенности синтеза микропроцессорных регуляторов для систем асинхронного электропривода с векторным управлением // Труды двенадцатой научно-технической конференции «Электроприводы переменного тока» .— Екатеринбург: УГТУ, 2001.
15. *Ишматов З. Ш., Казаков Е. Г., Кириллов А. В.* Методы синтеза микропроцессорных систем управления электроприводами.— Екатеринбург: УГТУ, 2000.— 48 с.

Приложение. Пример работы программы

--- УГТУ-УПИ, кафедра ЭАПУ ---
2004

Объект регулир. $W_0(z)=P(z)M(z-1)^v*Q_1(z)$
 Введите вектор коэфф. полинома $P(z)=[]$, начав со старшего.
 $P(z)=[1 -1.45101 0.745059 0]$
 Число интеграторов в ОР, $(z-1)^i$; $i=0$
 Введите вектор коэфф. полинома $Q_1(z)=[]$, начав со старшего.
 $Q_1(z)=[1 -2.47868 2.76030 -1.53696 0.31443]$
 Период дискретности $T=0.01$

ДПФ объекта регулирования $W_0(z)=$

Zero/pole/gain:
 $z (z^2 - 1.451z + 0.7451)$

 $(z-0.8409) (z-0.4348) (z^2 - 1.203z + 0.8601)$

Sampling time: 0.01

Требуемый пор_док астатизма по упр. воздействию $ru=1$

Вариант 1 - система регулир. обладает грубостью и устойчивостью,
 Вариант 2 - отсутствие скрытых колебаний регулируемой величины,
 Вариант 3 - низка_ чувствительность к изменению параметров ОР.

Желаемый вариант синтеза $N=2$
 Введите степень $A(z), nA(z)=4$

$N=1$ - биномиальное распределение,
 $N=2$ - настройка по Баттерворту,
 $N=3$ - скорейшее затухание,
 $N=4$ - конечна_ длительность.

Желаемый вариант настройки $N=3$

Величина стеднегеометрического корн_ OMEGA=1
 Характеристический полином A(z)=
 1.0000 -3.9740 5.9223 -3.9226 0.9743

D1=D11/D12

D11 =
 1.0e-007 *
 0.3357 -0.8321 0.9266 -0.5159 0.1055

D12 =
 1.0000 -3.9740 5.9223 -3.9226 0.9743

Козффициенты со старшего
 ДПФ регул_тора D(z)

Zero/pole/gain:
 3.3569e-008 (z-0.8409) (z-0.4348) (z^2 - 1.203z + 0.8601)

 (z-1) (z-0.9869) (z^2 - 1.987z + 0.9873)

Sampling time: 0.01

ДПФ замкнутой системы G(z)

Zero/pole/gain:
 3.3569e-008 z (z^2 - 1.451z + 0.7451)

 (z^2 - 1.986z + 0.9864) (z^2 - 1.988z + 0.9878)

Sampling time: 0.01

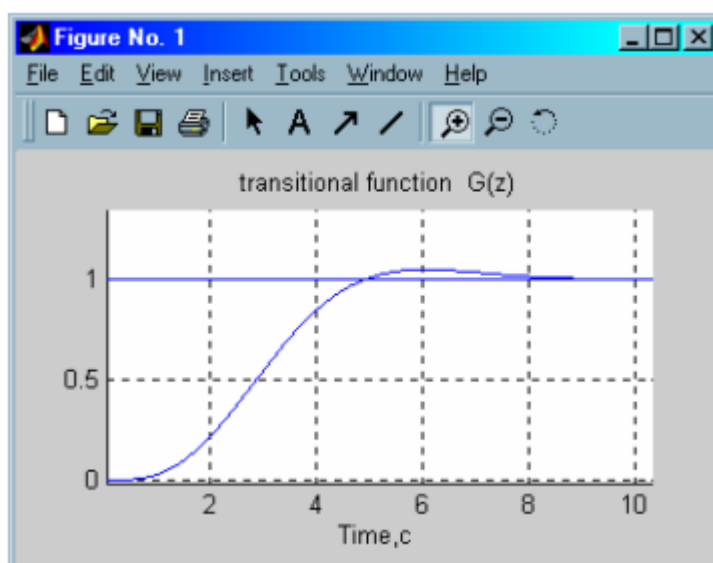
ДПФ замкнутой системы по ошибке 1-G(z)

Zero/pole/gain:
 (z-1) (z-0.9869) (z^2 - 1.987z + 0.9873)

 (z^2 - 1.986z + 0.9864) (z^2 - 1.988z + 0.9878)

Sampling time: 0.01

Врем_наблюдени_переходного процесса tnn=12



УДК 519.6

МОДЕЛИРОВАНИЕ СРЕДСТВАМИ SIMULINK/POWER SYSTEM BLOCKSET ПЕРЕМЕННЫХ RLC-ЭЛЕМЕНТОВ ЭЛЕКТРИЧЕСКИХ ЦЕПЕЙ

*Казаков В.А., Михотин В.Д., Тихонова Е.А.
Пензенский филиал Российского государственного
инновационных технологий и предпринимательства, Пенза,
e-mail: RGU@PENZA.NET*

Существуют целые классы задач моделирования электрических цепей и физических объектов представленных схемами замещения, в которых требуется имитировать изменения во времени значений RLC-элементов. Традиционно задачи такого класса решаются численными методами путем составления систем дифференциальных или интегральных уравнений и их решения с привлечением либо универсальных языков программирования (Fortran, Pascal и др.), либо специализированных математических пакетов MathCAD, Mathematica, Maple и т. п. Такой способ решения задач представляется трудоемким и требует большого внимания. В последние годы получило развитие новое направление использующее методы визуально-ориентированного программирования, которое нашло наиболее эффективное развитие в форме пакета MATLAB и его приложениях Simulink и Power System Blockset (PSB) и др. Например, известен предложенный Анохиным способ построения визуально-ориентированных моделей элементов электрических цепей с переменными параметрами [1]. Однако данный результат лишь ставит проблему и указывает общее направление решения и эффективен только для моделирования переменных сопротивлений.

В настоящее время признаком хорошего стиля считается использование при моделировании ряда (пучка) моделей, которые отличаются не только степенью подробностей в детализации исследуемого объекта, но и способами математического описания и алгоритмами реализующих их программ. Это позволяет, во-первых, иметь дополнительные степени свободы по расширению области исследования, во-вторых, оперативно по внутренним критериям оценивать адекватность моделей и результатов моделирования в целом, и, в-третьих, с точки зрения программирования, осуществлять более быструю отладку и настройку моделей.

В свете сказанного в настоящем докладе рассматриваются три способа построения моделей RLC-элементов с переменными параметрами, которые реализуются средствами входящих в систему MATLAB пакетов прикладных программ Simulink и Power System Blockset. Рассмотрим

предлагаемые способы подробнее на примере построения моделей переменных сопротивлений.

1 способ. Модели в базисе обобщенных сигнальных графов

Модели этого типа основываются на использовании описания свойств резисторов в виде взвешенных вершин обобщенных сигнальных графов (ОСГ) [2,3]. Для пояснения рассмотрим некоторый участок электрической цепи, который включает в себя резистор с переменным сопротивлением R_{var} (см. рис. 1.). Данному участку цепи, если условно зафиксировать потенциалы U_1 и U_2 , согласно теореме компенсации [4], соответствует источник тока со значением

$$i = \frac{U_1 - U_2}{R_{\text{var}}}. \quad (1)$$

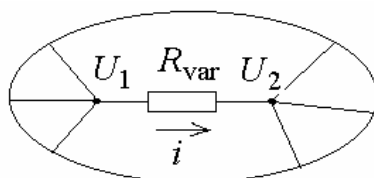


Рис. 1. Схема включения R_{var} .

Используя выражение (1), строится в виде подсистемы R_{var} средствами пакетов Simulink и Power System Blockset модель (PSB-модель), которая показана на рис. 2. Здесь и далее используются цветные обозначения блоков согласно их функциям и принадлежности к используемым пакетам программ визуально-ориентированного программирования: зеленый — блоки пакета Power System Blockset, желтый — функциональные блоки Simulink, голубой — средства отображения выходной информации пакета Simulink, розовый — подсистемы моделирующие переменные RLC-элементы электрических цепей.

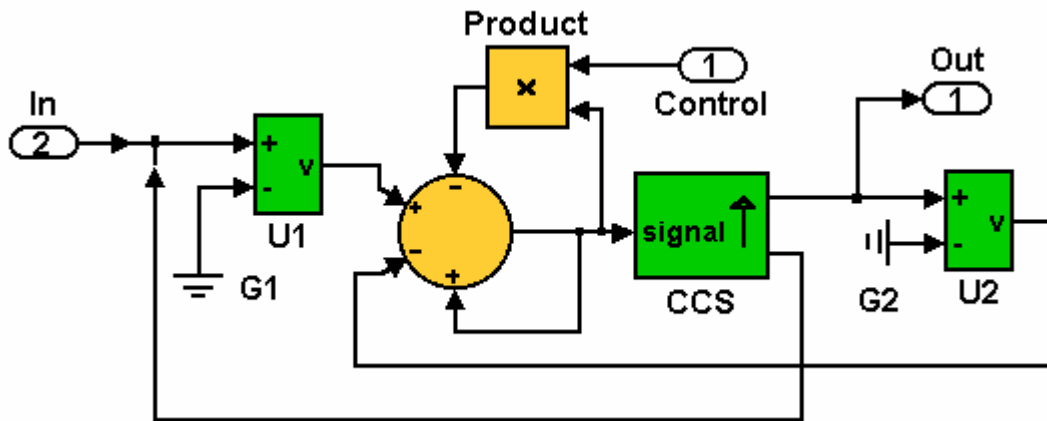


Рис. 2. 1-й вариант PSB-модели управляемого резистора (подсистема R_var на базе ОСГ).

Рассматриваемая PSB-модель содержит: два датчика напряжения U1 и U2 (PSB блоки Voltage Measurement), CCS — управляемый источник тока (блок Control Current Source), а также компоненты из пакета Simulink — сумматор и блок Product работающий в режиме умножителя. Последние из упомянутых блоков собственно и реализуют варьруемое сопротивление R_{var} , значение которого устанавливается сигналом подаваемым на вход Control подсистемы R_var.

Для тестирования и отладки подсистемы R_var удобно использовать PSB-модель представленную на рис. 3. В данной модели кроме блоков, назначение которых ясно без пояснения, используется датчик тока (PSB блок Current Measurement). Результаты тестирования показаны на рис. 4. При этом задавались следующие настройки компонент: напряжение DC V_0 равно 100 В, R_{var} изменялось по закону $R_{var} = 100 + 50\sin 4t$ [Ом].

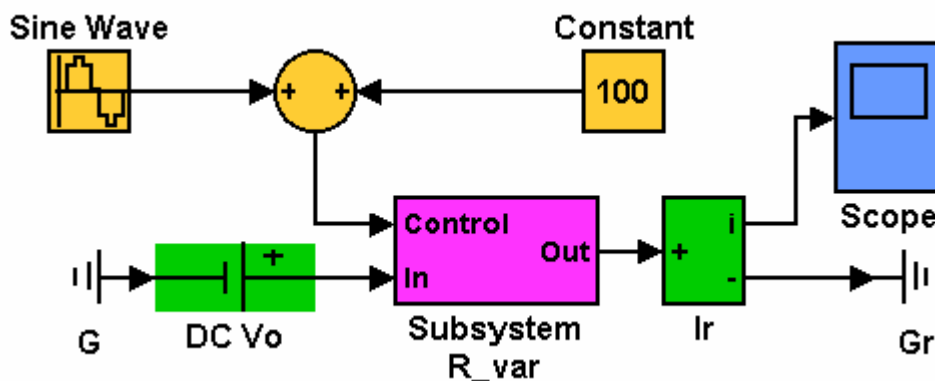


Рис. 3. PSB-модель для тестирования подсистемы R_var на базе ОСГ.

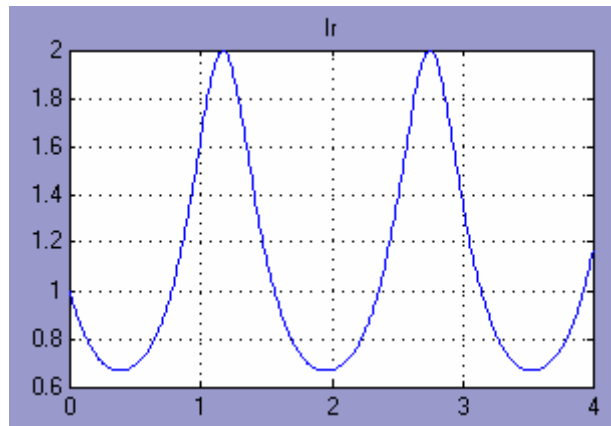


Рис. 4. Временные диаграммы работы PSB-модели рис. 3.

Как видно из рис. 4, представленная модель адекватно имитирует работу управляемого сопротивления R_{var} . Достоинством модели является высокая устойчивость решений проявляющаяся при различных режимах управляющих воздействий и при работе в составе PSB-моделей сложных электрических цепей. Вместе с тем рассмотренная модель представляется достаточно громоздкой, что создает определенные трудности при моделировании сложных по составу схем. Сложность, как по числу используемых блоков, так и выполняемым ими функциям, проявляется не только в увеличении времени программирования и настройки, но и в увеличении времени выполнения программ моделирования. От этих недостатков свободны рассматриваемые ниже модели.

2 способ. Модели на основе закона Ома с управляемыми источниками тока

Вариант построения PSB-модели переменного сопротивления R_{var} непосредственно реализующей закон Ома показан на рис. 5. При работе данной модели датчик напряжения U1 определяет падение напряжения на R_{var} , а значение тока источника CCS задается в соответствии с формулой $i = U_1 / R_{\text{var}}$, которая реализуется блоком Product настроенным для работы в режиме деления. Как и в рассмотренной выше PSB-модели рис. 2, на вход Control подается Simulink сигнал равный значению R_{var} . Естественно, при программировании следует следить, чтобы при работе PSB-модели не возникло ситуации, связанной с делением на ноль. В качестве превентивной меры на входе Control подсистемы R_{var} можно рекомендовать включение элемента Saturation из библиотеки Simulink.

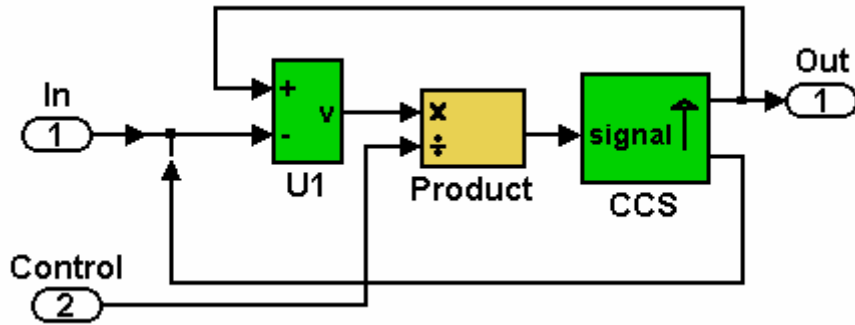


Рис. 5. 2-й вариант PSB-модели управляемого резистора.

Для тестирования рассмотренной модели рекомендуется использовать PSB-модель показанную на рис. 6, в которой применен ручной переключатель Manual Switch, позволяющий проводить испытания при постоянном значении сопротивления (нижнее положение ключа) и при синусоидальном изменении значения сопротивления R_{var} (верхнее положение переключателя). Кроме того, для оперативного изменения значений сопротивления применен блок ручной регулировки Slider Gain.

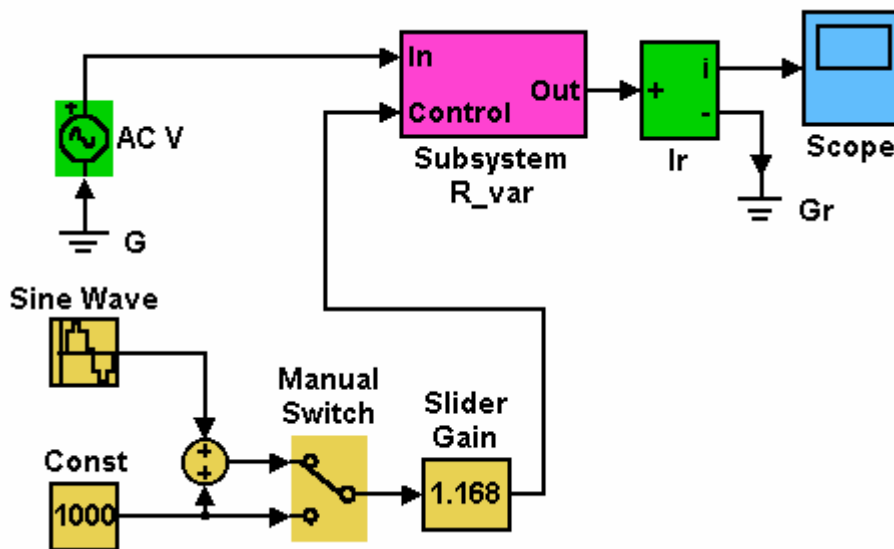


Рис. 6. PSB-модель для тестирования подсистемы R_{var} рис. 5.

Результаты тестирования рассмотренной модели показаны на рис. 7. При этом задавались следующие настройки параметров. Для генератора входного напряжения ACV:

$$U_{ACV} = 100 \cdot \sin 2\pi t \text{ [В]}.$$

Значение управляемого сопротивления менялось по закону

$$R_{var} = 1000 + 400 \sin 100t \text{ [Ом]}.$$

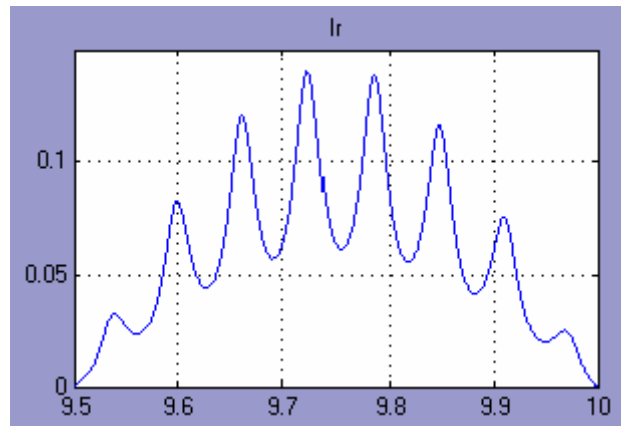


Рис. 7. Временные диаграммы работы модели рис. 6.

3 способ. Модели на основе закона Ома с управляемыми источниками напряжения

При реализации данного способа моделирования переменного сопротивления используется другая форма записи закона Ома $U = i \cdot R_{\text{var}}$, где U — напряжение снимаемое с управляемого источника; i — ток через R_{var} .

Соответствующая подсистема, моделирующая переменное сопротивление, показана на рис. 8. В рассматриваемой PSB-модели значение тока, определяемого датчиком тока CMeasur, умножается посредством блока Product на R_{var} , задаваемого по входу Control, и далее подается на управляемый источник напряжения CVS (Control Voltage Source). В цепи последовательно с CVS включен резистор R_0 , который служит для обеспечения устойчивой работы PSB-модели. При отсутствии R_0 во время компиляции система MATLAB выдает сообщение о наличии деления на ноль. Поэтому резистор R_0 в PSB-модели решает чисто формальную задачу. Реально при работе с рассматриваемой моделью значения R_0 задавались порядка единиц микроОм.

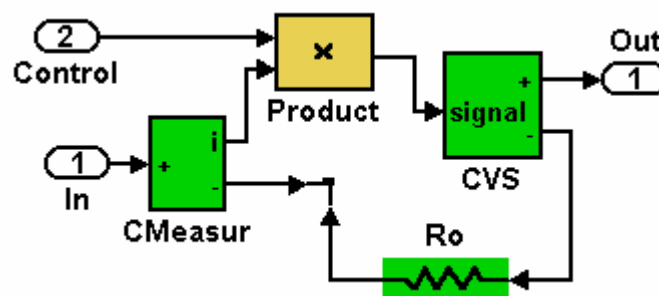


Рис. 8. 3-й вариант PSB-модели управляемого резистора.

Оценить влияние значения R_0 на точность моделирования можно по данным следующего примера. При $R_{var} = 100 \text{ Ом}$ и $R_0 = 1 \text{ мОм}$ погрешность воспроизведения R_{var} не превышает 0,001%. Как показали результаты моделирования R_{var} при помощи подсистемы рис. 8, в первом приближении погрешность моделирования может быть оценена по формуле $\delta = R_0/R_{var}$. Кроме того, при подстановке рассматриваемого варианта подсистемы R_{var} в PSB-модель рис. 6 и при прочих равных условиях тестирования, были получены результаты, полностью совпадающие с данными показанными на рис. 7.

Модели управляемых LC элементов

Представленные выше три варианта (способа) построения PSB-моделей управляемых сопротивлений являются базисом для синтеза моделей управляемых LC элементов. В качестве примера рассмотрим варианты PSB-моделей управляемых LC элементов, которые строятся как развитие базовой модели управляемого сопротивления, построенной по третьему способу. При построении таких моделей следует использовать рекомендации представленные в табл. 1, т. е. вместо блока Product, применяемого в PSB-моделях управляемого сопротивления R_{var} , в структуру модели дополнительно вводятся интегратор или дифференциатор в зависимости от того, какой реактивный элемент C или L моделируется.

Таблица 1.
Модели управляемых RLC элементов.

Элемент цепи	Тип управляющего звена
R	
L	
C	

Следует отметить, что для решения этих задач можно рекомендовать применение универсального блока PID Controller, который находится в библиотеке Simulink Extras\Additional Linear. Блок PID Controller выполняет функции параллельно включенных блоков масштабирующего преобразования, интегрирования и дифференцирования, что позволяет путем соответствующей настройки блока реализовывать функции требуемого активного или реактивного элемента электрической цепи.

Модель управляемой емкости C_{var} показана на рис. 9. Принцип работы модели не требует особых пояснений. Соответствующая PSB-модель для тестирования подсистемы C_{var} представлена на рис. 10, а результаты работы модели показаны на рис. 11. При настройке рассматриваемой PSB-модели задавались следующие параметры: ток (AC Current Source)

$$i = \sin 100\pi t, \quad (2)$$

где $R_0 = 1 \text{ мОм}$, $C = \text{ мФ}$.

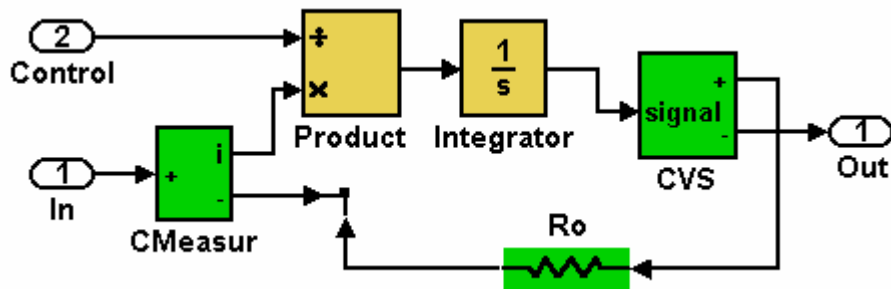


Рис. 9. PSB-модель подсистемы C_{var} .

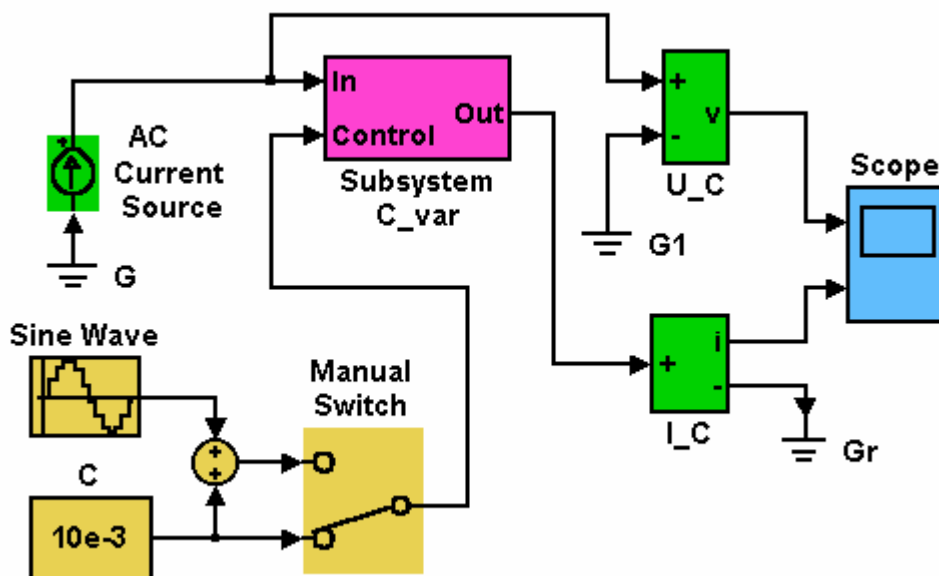


Рис. 10. PSB-модель для тестирования подсистемы C_{var} рис 9.

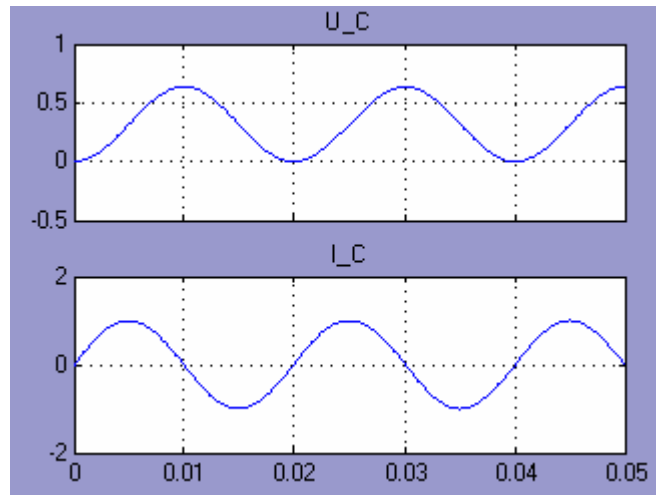


Рис. 11. Результат тестирования подсистемы C_var рис 10.

Как видно из рис. 11, напряжение на конденсаторе представляет собой интеграл от входного тока. Для проверки качества модели рассмотрим выражения для напряжения на C_{var} полученные согласно известной формуле [4], т. е.

$$U_c(t) = \frac{1}{C_0} \int_0^t \sin 100\pi \tilde{t} d\tilde{t} = \frac{1}{100\pi C} [1 - \cos 100\pi t]$$

Соответствующая амплитуда изменения напряжения на конденсаторе

$$U_m = \frac{1}{50\pi C} \Big|_{C=0,01} = \frac{2}{\pi} = 0,64 \text{ [В]},$$

что совпадает с результатами моделирования, показанными на рис. 11.

Модель управляемой индуктивности в виде подсистемы L_{var} показана на рис. 12. Для тестирования модели можно использовать схему аналогичную представленной на рис. 10. Результат тестирования при $L_{var} = 1 \text{ мГн} = const$ показан на рис. 13. При этом входной ток задавался согласно выражению (2) при $R_0 = 1 \text{ мОм}$.

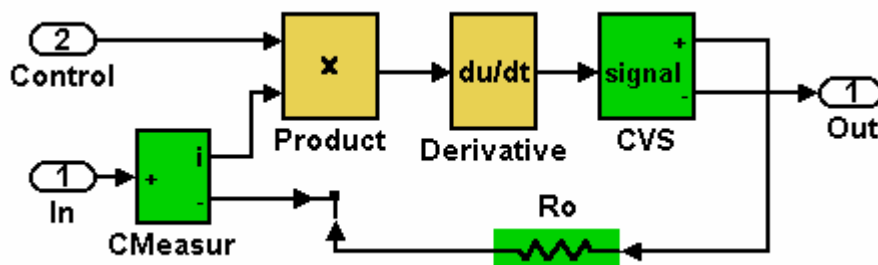
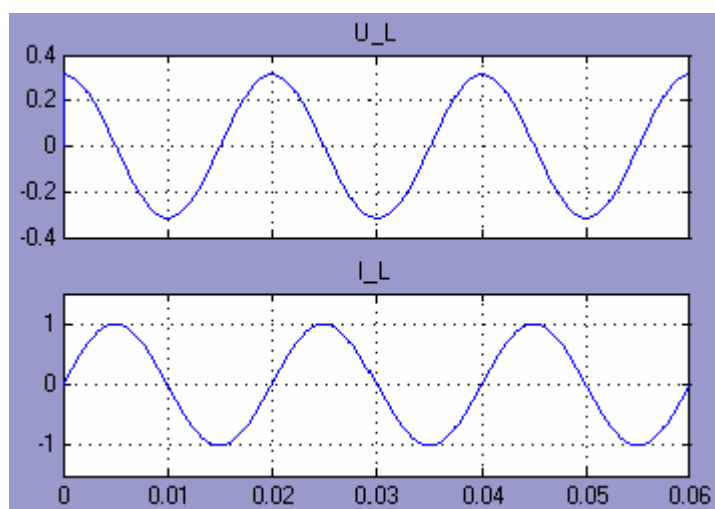


Рис. 12. PSB-модель подсистемы L_var.

Рис. 13. Результат тестирования подсистемы L_{var} .

Как видно из осциллограмм рис. 13, напряжение на индуктивности пропорционально производной от тока. Действительно

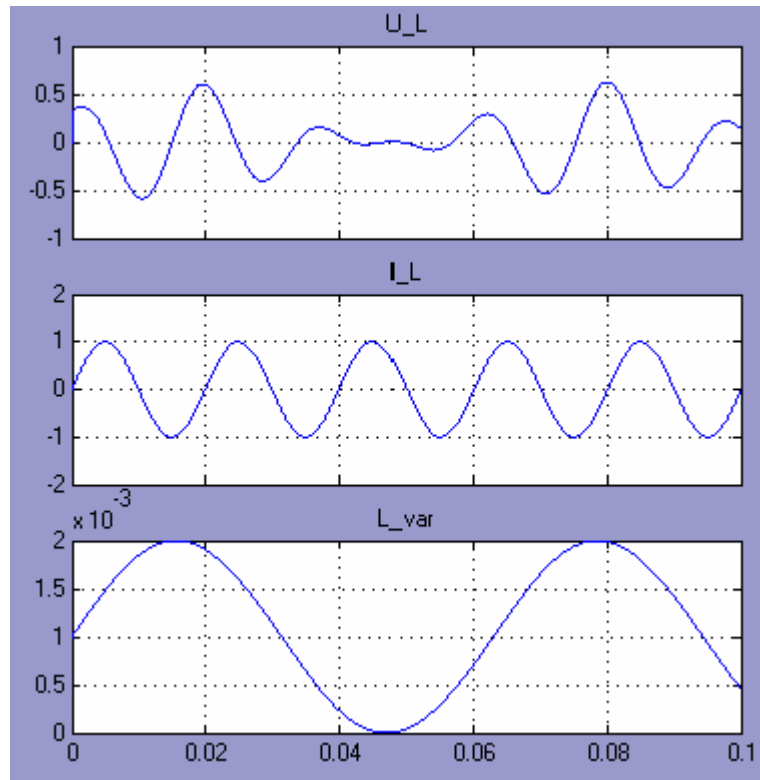
$$U_L(t) = L \frac{d(\sin 100\pi t)}{dt} = 100\pi L \cos 100\pi t,$$

а амплитуда составляет $U_m = 100\pi L = 100\pi \cdot 10^{-3} = 0,314$ [В]. Указанное значение соответствует результату представленному на рис. 13.

Результат тестирования переменной индуктивности при переменном значении индуктивности, изменяющейся по закону

$$L_{var} = 0,001 + 0,001 \sin 100t \text{ [Гн]}, \quad (3)$$

показан на рис. 14. Представленные осциллограммы показывают что напряжение на индуктивности L_{var} подвергается амплитудной модуляции в соответствии с законом изменения индуктивности (3).

Рис. 14. Результат тестирования подсистемы L_var .

Применение управляемых RLC элементов для моделирования параметрических датчиков

Как отмечалось выше, предлагаемые PSB–модели управляемых RLC элементов могут быть использованы в качестве подсистем при моделировании сложных электрических цепей. На практике не сложно и полезно иметь библиотеку соответствующих девяти подсистем, которые моделируют три элемента тремя способами. Рассмотрим конкретный пример моделирования измерительной цепи емкостного датчика рис. 15 и обсудим особенности применения тех или иных моделей управляемых RLC элементов электрических цепей.

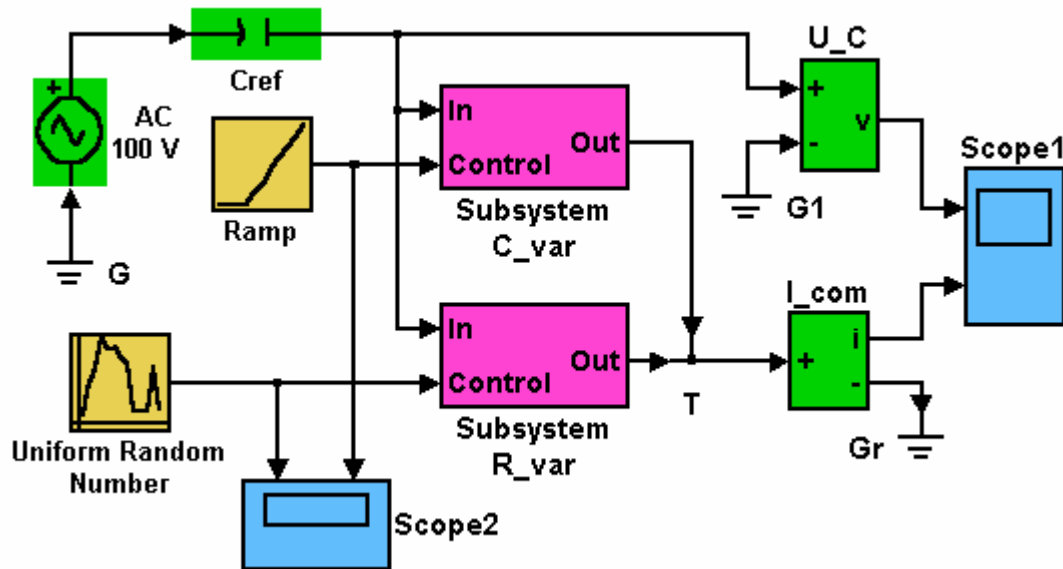


Рис. 15. PSB-модель измерительной цепи емкостного датчика.

Представленная на рис. 15 PSB-модель имитирует работу измерительной цепи датчика включенного по схеме емкостного делителя, где C_{ref} представляет собой опорный конденсатор с емкостью 200 нФ, источник активного воздействия на делитель в виде напряжения AC Voltage Source работает на частоте 50 Гц и имеет амплитуду 20 В. Собственно емкостной датчик моделируется посредством двух параллельно включенных подсистем $Subsystem C_{var}$ и $Subsystem R_{var}$. В данном случае $Subsystem C_{var}$ имитирует выходной информативный параметр датчика, а $Subsystem R_{var}$ учитывает влияние паразитного неинформативного параметра проявляющегося в виде проводимости (сопротивления) потерь в датчике.

Подобного типа датчики находят широкое применение для измерения самых разнообразных физических величин: влажности сыпучих материалов, толщины шпона, содержания сахара в крови и т. п. При этом, оценивая качество работы датчика и соответствующих измерительных систем, важно знать, с одной стороны, динамические характеристики при изменяющихся во времени значениях выходного информативного параметра C_{var} и, с другой стороны, необходимо иметь оценки влияния параметрических помех, которые проявляются через вариации сопротивления потерь.

В рассматриваемой PSB-модели изменения информативного выходного параметра во времени по закону

$$C_{var} = 1000 + 100t \quad [\text{нФ}]$$

задается при помощи блока Ramp, а случайные вариации R_{var} в диапазоне от 1 до 100 кОм задаются генератором случайных чисел Uniform Random Number, который имеет равномерное распределение и работает с частотой дискретизации, равной 5мс.

Результаты работы рассматриваемой PSB-модели при изменениях R_{var} и C_{var} показанных на рис. 16, приведены на рисунке 17. Осциллограммы сигналов представленных на рис. 17 объективно показывают не только динамическую погрешность и влияние параметрических шумов, но и выявляют погрешности возникающие при моделировании. В этой связи следует специально обратить внимание на особенности работы с пакетами Simulink и PSB, которые предлагают пользователю целый ряд методов решения систем дифференциальных уравнений

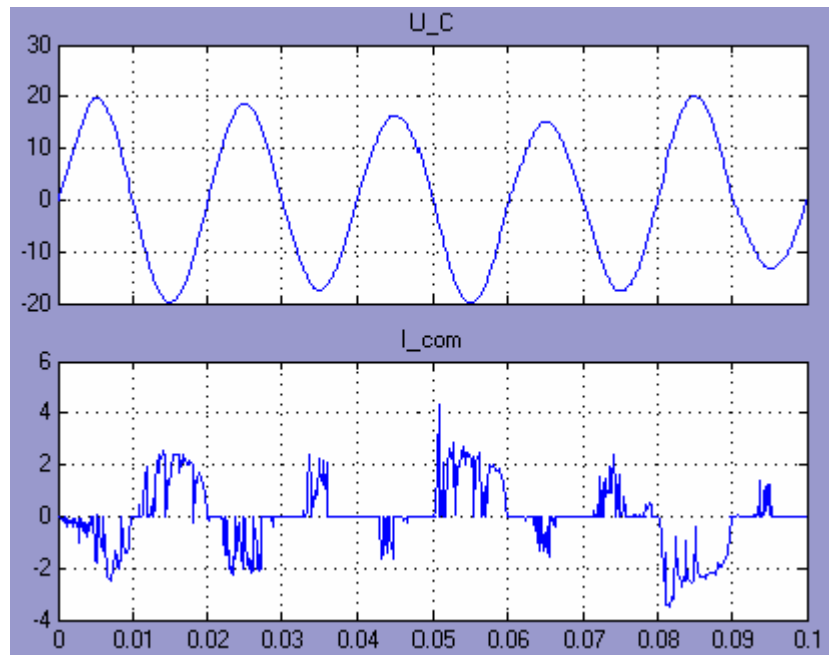


Рис. 16. Результат тестирования PSB-модели измерительной цепи емкостного датчика.

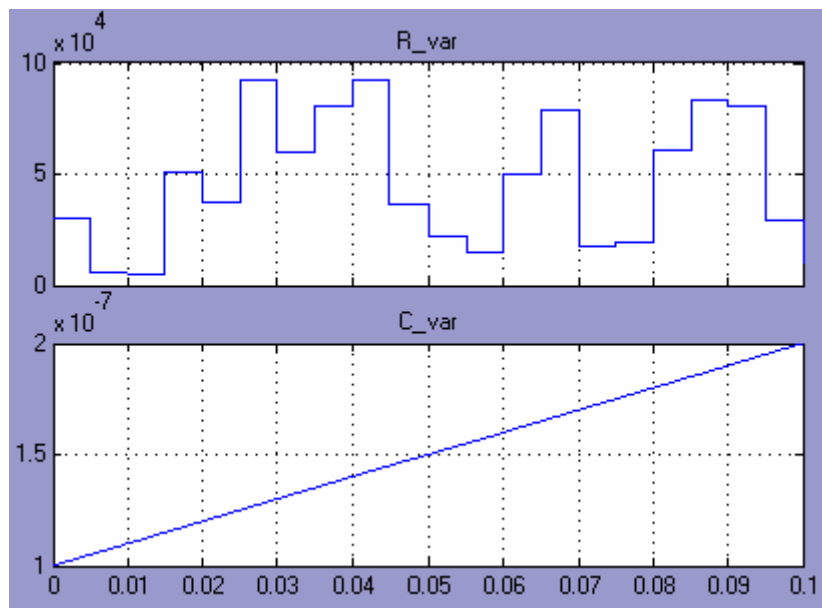


Рисунок 17. Законы изменения параметров емкостного датчика.

Литература

1. *Анохин В. В.* Переменное сопротивление в MATLAB/Simulink // Exponenta Pro. Математика в приложениях.— №1.— 2003.— С.91–92.
2. *Анисимов В. И.* Топологический расчет электронных схем.— Л.: Энергия, 1977.— 240 с.
3. *Медведева С. Н., Михотин В. Д., Пискарев С. П.* Анализ электрических цепей методом сигнальных графов: Учебное пособие / Под. ред. В.И. Чернецова.— Пенза: Изд-во Пенз. гос. ун-та, 2002.— 112 с.
4. *Бессонов Л. А.* Теоретические основы электротехники.— М.: Высшая школа, 1978.— 752 с.

УДК 621.822

МОДЕЛИРОВАНИЕ И ОБРАБОТКА ВИБРАЦИОННЫХ СИГНАЛОВ СИСТЕМЫ «РОТОР–ПОДШИПНИКИ СКОЛЬЖЕНИЯ–ДЕМПФЕРЫ» В СИСТЕМЕ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ MATLAB/SIMULINK

*Комаров М. В, Широков С. В, Соломин О. В, Данчин И. А.
Орловский государственный технический университет, Орел,
e-mail: pmpl@ostu.ru*

Рассматриваемый класс технических объектов представляет собой сложную гидромеханическую систему (рис. 1), в состав которой входят: опоры, в качестве которых широко используются подшипники скольжения, демпферные устройства, вал, радиальные и торцевые уплотнения, турбина, центробежный насос и пр. Повышение производительности турбомашин связано с ростом частот вращения роторов, что порождает комплекс задач, связанных с необходимостью обеспечения устойчивости их движения, снижением уровня вибраций, динамических напряжений и переходом роторов через резонансные частоты. Все вышеуказанные факторы в значительной мере влияют на ресурс и работоспособность турбомашин.

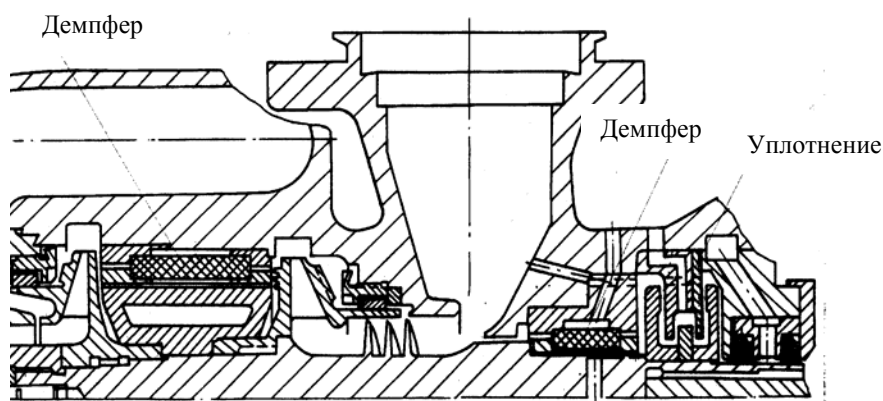


Рис. 1. Типичная компоновка турбомашин.

Одним из эффективных методов повышения виброустойчивости является введение в конструкции опорных узлов демпфирующих устройств. Проектирование опорных узлов, строго говоря, связано с необходимостью решения множества задач таких как: выбор конструкций элементов демпфера и опор скольжения, расчет коэффициентов жесткости, демпфирования и проведение динамического анализ системы в целом, а также ряда за-

дач, связанных с принятием конструктивных решений, исходя из конкретных условий работы турбоагрегата. Решение этих задач позволяет повысить работоспособность и надежность турбомашин в целом.

Решение проблемы видится в построении адекватной математической модели, способной учитывать конструктивные особенности упруго-демпферных опор. Если учесть, что отклонения геометрических параметров, связанных с изготовлением узлов турбомашин, ведет к значительному изменению характеристик всей системы, то необходимо проведение расчета системы в интервалах допусков с применением различных комбинаций опор скольжения и демпферов с последующей оптимизацией параметров всей системы.

Рассматриваемая динамическая система описывается совокупностью нелинейных обыкновенных дифференциальных уравнений (уравнения движения) и нелинейных дифференциальных уравнений в частных производных (уравнения гидродинамики, позволяющие рассчитать силовые факторы в подшипниках скольжения и демпферах). Аналитическое решение такой системы не представляется возможным. Для получения искомого результата и решения задачи моделирования динамики описанной системы авторы используют систему компьютерной математики MATLAB со средой визуально-ориентированного программирования Simulink [1].

Программирование в данной среде основано на композиции отдельных блоков, моделирующих соответствующие элементы рассматриваемой динамической системы [2]. Модель представляется в форме графического описания с организацией связей между компонентами и установлением их параметров. Модель собирается из линейных и нелинейных динамических элементов, представленных своей функциональной блок-схемой. Таким образом, создается модель системы или устройства, т. е. Simulink автоматически составляет и решает сложные алгебраические и дифференциальные уравнения, описывающие заданную функциональную схему. Кроме этого, данный пакет позволяет создавать системные функции с включением их в состав библиотек. Основной особенностью является автоматическое генерирование программного кода в процессе ввода блока, соединение блоков и задание параметров компонентов. Возможность построения сложных иерархических структур и объединение их в единые модели с дальнейшим сохранением созданных модулей в наборах библиотек позволяет проводить моделирование сложных систем на основе ранее созданных моделей.

Одним из примеров моделирования является построение модели двухмассовой динамической системы «ротор — подшипники скольжения — гидродинамические демпферы» (рисунок 2).

Рассмотрим жесткий симметричный ротор на двух одинаковых упругодемпферных опорах, вращающийся с постоянной скоростью. Динамическая модель рассматриваемой системы представляет собой двухмассо-

вый осциллятор с четырьмя степенями свободы, в котором движение ротора происходит в пределах радиального зазора подшипника скольжения под действием возмущающих центробежных сил, веса ротора и реакций смазочного слоя; в свою очередь втулка подшипника скольжения вибрирует под действием возмущающих сил, равных по величине реакциям смазочного слоя и реакциям демпфера.

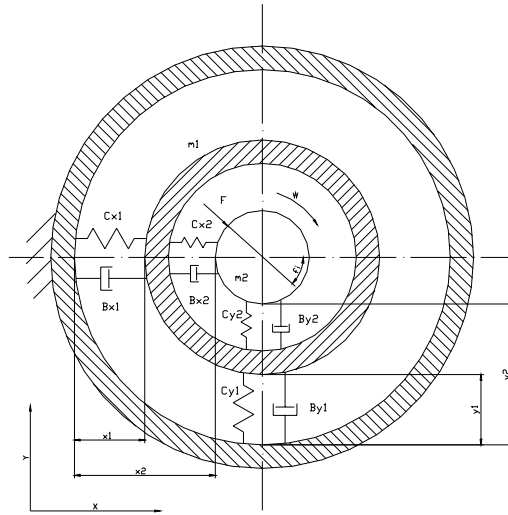


Рис. 2. Расчетная модель упругодемпферной опоры.

Амплитудно-частотные характеристики и вид колебаний ротора определяем путем анализа траекторий движения ротора, которые получаем при решении уравнений движения центра цапфы и центра втулки подшипника скольжения. Уравнения динамики ротора имеют вид:

$$\ddot{a}_{x1} = \frac{-C_{x1}x_1}{m_1} + \frac{C_{x2}(x_2 - x_1)}{m_1} - \frac{B_{x1}\dot{x}_1}{m_1} + \frac{B_{x2}(\dot{x}_2 - \dot{x}_1)}{m_1};$$

$$\ddot{a}_{x2} = \frac{-C_{x2}(x_2 - x_1)}{m_2} + \frac{B_{x2}(\dot{x}_2 - \dot{x}_1)}{m_2} + \Delta\omega^2 \cos(\omega t) + \frac{F \cos(\varphi)}{m_2};$$

$$\ddot{a}_{y1} = \frac{-C_{y1}y_1}{m_1} + \frac{C_{y2}(y_2 - y_1)}{m_1} - \frac{B_{y1}\dot{y}_1}{m_1} + \frac{B_{y2}(\dot{y}_2 - \dot{y}_1)}{m_1} - g;$$

$$\ddot{a}_{y2} = \frac{-C_{y2}(y_2 - y_1)}{m_2} + \frac{B_{y2}(\dot{y}_2 - \dot{y}_1)}{m_2} + \Delta\omega^2 \sin(\omega t) - \frac{F \cos(\varphi)}{m_2} - g;$$

где m_1, m_2 — соответственно массы втулки подшипника скольжения и ротора; Δ — дисбаланс ротора (смещение центра масс); ω — угловая скорость; C — коэффициенты жесткости; B — коэффициенты демпфирования; F — внешняя приложенная сила; φ — угол приложения внешней силы; индексы 1 и 2 относятся соответственно к втулке демпфера и подшипника.

Для численного интегрирования дифференциальных уравнений использовались адаптивные методы реализованные в системе MATLAB, в которых численное решение ищется с переменным шагом интегрирования исходя из заданной точности вычислений и скорости изменения интегрируемой функции.

Для определения реакций подшипника скольжения (ПС) существует два подхода: 1) непосредственное интегрирование уравнений гидродинамики, отражающих распределение давлений по поверхности подшипника и демпфера; 2) линеаризация реакций с помощью взаимосвязанных коэффициентов жесткости и демпфирования. Авторами использовались оба подхода, которые позволяют судить о влиянии различных параметров подшипников и демпферов на динамические характеристики роторной системы в целом. Блок-диаграмма реализованной модели имеет вид (рис. 3).

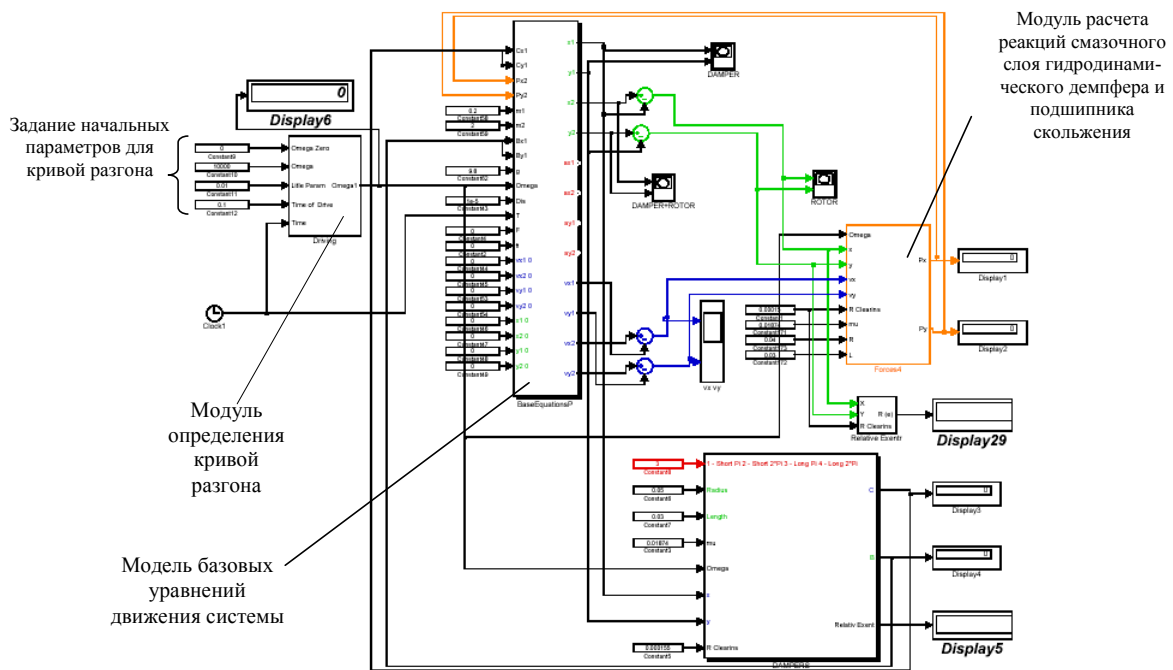


Рис. 3. Обобщенная блок-диаграмма реализованной модели.

Детализация некоторых модулей входящих в диаграмму показана на рис. 4.

Амплитудно-частотные характеристики, развертки и траектории движения ротора, амплитуды колебаний демпфера могут быть представлены графиком как показано на рис. 5.

На рис. 5, а представлены развертки колебаний скорости и ускорения. На рис. 5, б — расчетная траектория движения центра цапфы ротора.

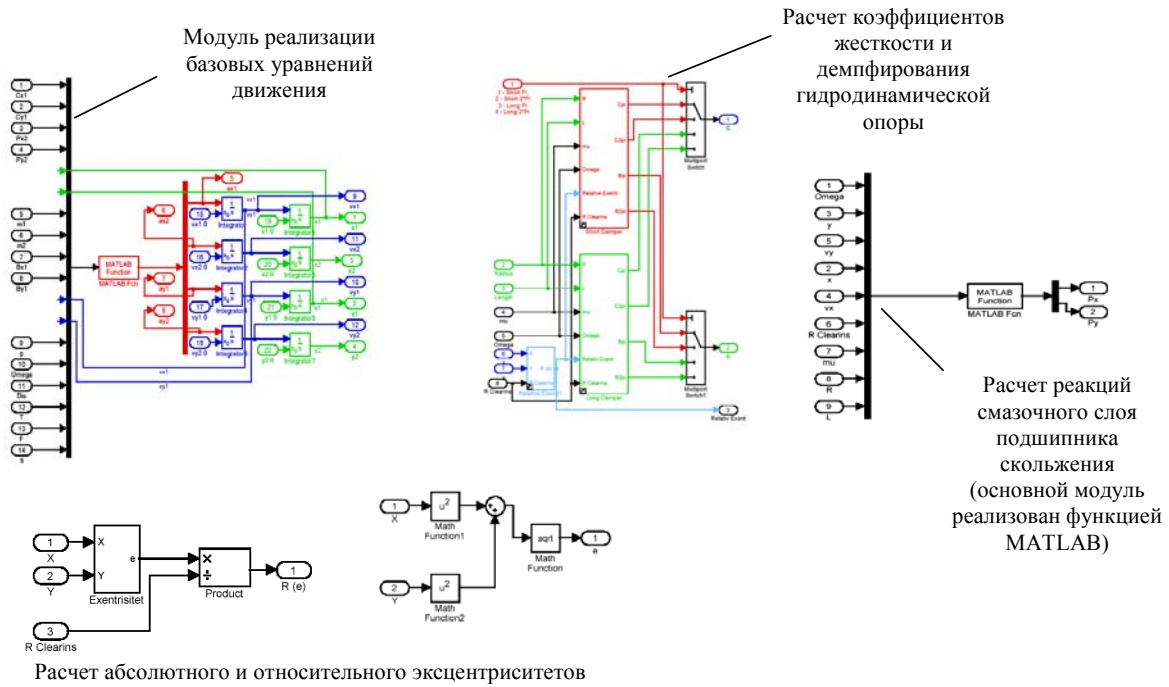


Рис. 4. Детализация расчетного модуля.

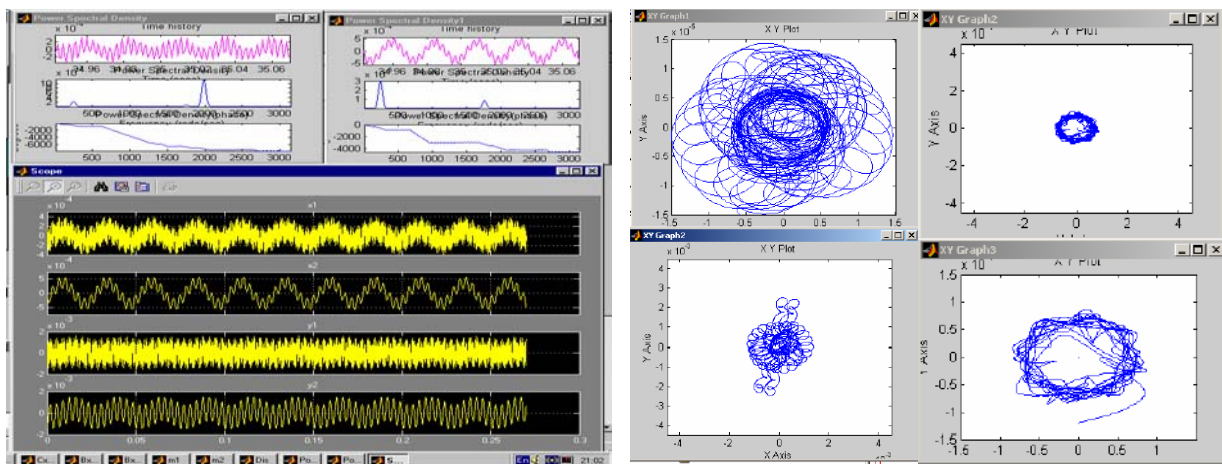


Рисунок 5. — Графическое представление информации.

Разработанная математическая модель и результаты расчетов в Simulink представляют широкие возможности для изучения, проектирования и исследования роторных систем. Такой подход не требует специальной подготовки, позволяет быстро менять рабочие и геометрические параметры исследуемой системы, что повышает качество и снижает трудозатраты при проведении вычислительных экспериментов. Проведенные вычислительные эксперименты показали устойчивую работу разработанных алгоритмов.

Адекватность построенной модели проверялась в ходе модельного физического эксперимента, проведенного на базе экспериментального

стенда и информационно измерительного комплекса, построенного на основе аналого-цифрового преобразователя ЛА2МЗ, интегрируемого в ЭВМ (рисунок 6). При построении данной системы был решен ряд задач, в числе которых:

- выбор преобразователей и модулей согласования для адекватного сбора информации;
- разработка программного обеспечения и выбор стратегии построения алгоритма сбора, обработки и хранения данных;
- анализ результатов проведенных экспериментов.

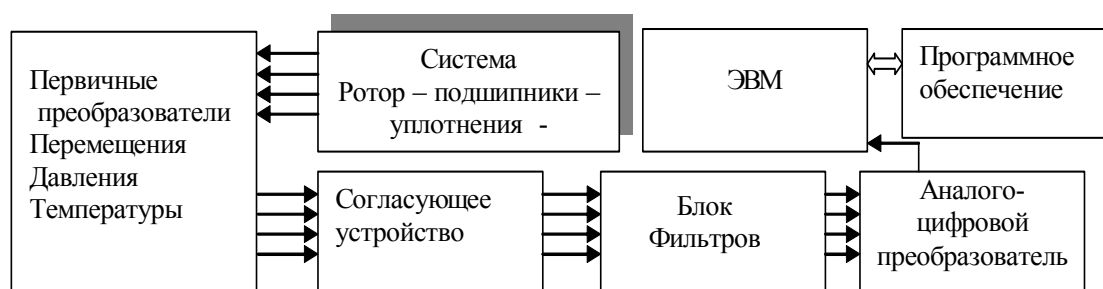


Рис. 6. Структурная схема измерительного комплекса.

Измеряемыми величинами являлись:

- перемещение центра цапфы ротора в радиальном зазоре подшипника скольжения;
- давление по контуру подачи и гидравлических трактах подшипников;
- температура в камерах, на входе и сливе подшипников.

Для удобной обработки полученных экспериментальных данных, в среде MATLAB авторами статьи было разработано специализированное приложение. Актуальность создания такого приложения обусловлена тем, что существует необходимость наглядно и быстро обрабатывать полученные экспериментальные данные по заранее разработанному алгоритму. Существующие средства в системе MATLAB, такие как Signal Processing Toolbox или Wavelet Toolbox, являются достаточно мощными инструментами для обработки сигналов, но плохо подходят для решения данной технической задачи.

Главное интерфейсное окно разработанного приложения выглядит следующим образом (рис. 7).

В данном окне программы отображаются сигналы, характеризующие перемещения центра цапфы по осям X и Y. Справа, используя сигналы X и Y, строится траектория движения цапфы. На рисунке 6 представлены экспериментальные сигналы до их обработки. В правом окне возможно вместо траектории отобразить быстрое Фурье преобразование (БПФ). Для этого необходимо установить переключатель FFT.

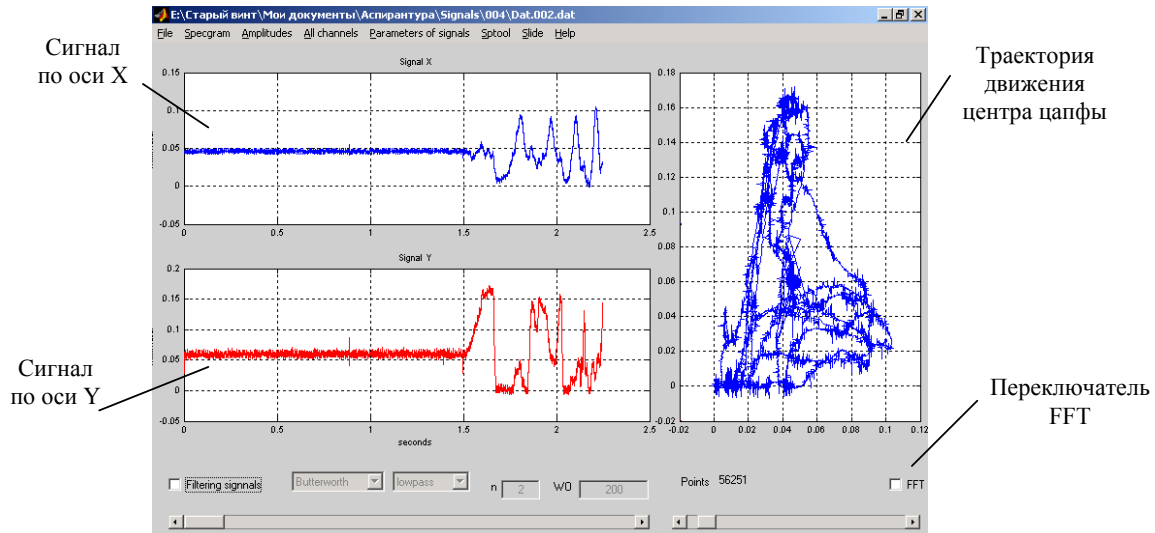


Рис. 7. Главное интерфейсное окно разработанного приложения.

При открытии сигнала в приложении, необходимо установить параметры, предложенные в диалоговом окне на рис. 8. В данном диалоговом окне необходимо установить количество каналов (Quantity channels), частоту дискретизации в герцах (Sampling frequency), коэффициент усиления (Amplification), а также назначить номер канала для сигнала X и сигнала Y.

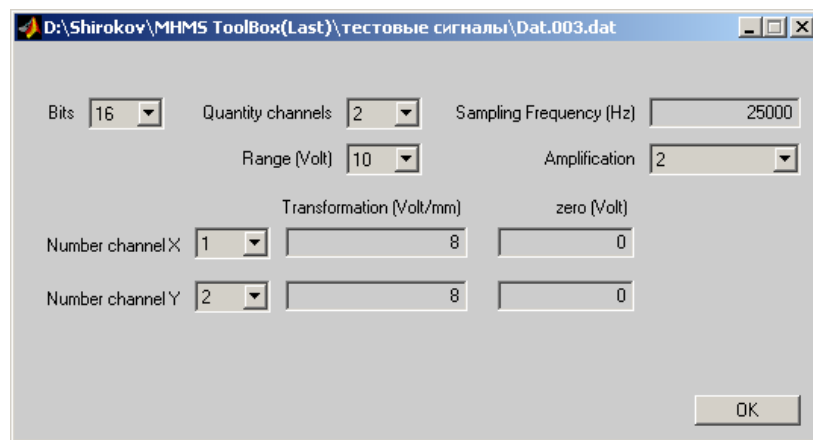


Рис. 8. Диалоговое окно для установки параметров открываемого сигнала.

После загрузки сигнала в приложение существует возможность фильтрации данного сигнала. Для этого необходимо установить переключатель *filtering signals* и выбрать тип фильтра, а также его вид. В настоящий момент в разработанном приложении возможно выбрать фильтр Баттерворта (Butterworth) и фильтры Чебышева первого и второго рода. Фильтрация осуществляется одним из четырех видов фильтра: фильтр нижних частот (lowpass), фильтр высоких частот (highpass), полосовой фильтр (bandpass) и режекторный фильтр (bandstop). Для фильтрации необходимо указать порядок фильтра (параметр *n*) и частоту среза (параметр

W0). Ниже приводится листинг функции, реализующей фильтрацию с помощью фильтра Баттерворта.

```
function output_sig=mybutter(input_sig,n,Fs,W0,type)
if (length(W0)<2) & ((type==3)|(type==4))
    button=questdlg ('Неверная длина вектора! Принять по умолчанию от 20 до 200 Гц?','ДА','no');
if strcmp(button,'ДА') W0=[20 200];end
end
if (length(W0)>1) & ((type==1)|(type==2))
    button=questdlg ('Неверная длина вектора! Принять по умолчанию 500 Гц?','ДА','no');
if strcmp(button,'ДА') W0=500;end
end
switch type
case 1
    [b,a]=butter(n,W0*2/Fs);
case 2
    [b,a]=butter(n,W0*2/Fs,'high');
case 3
    [b,a]=butter(n,W0*2/Fs);
case 4
    [b,a]=butter(n,W0*2/Fs,'stop');
end
output_sig=filtfilt(b,a,input_sig);
```

Входными данными для этой функции являются исходный сигнал (переменная `input_sig`), порядок фильтра (`n`), частота дискретизации (`Fs`), частота среза (`W0`) и вид фильтра (`type`). Следует отметить, что если фильтрация производится полосовым или режекторным фильтром, то значение параметра `W0` должно передаваться в виде вектора, состоящего из двух элементов. Пример отображения фильтрованных данных приведен на рис. 9.

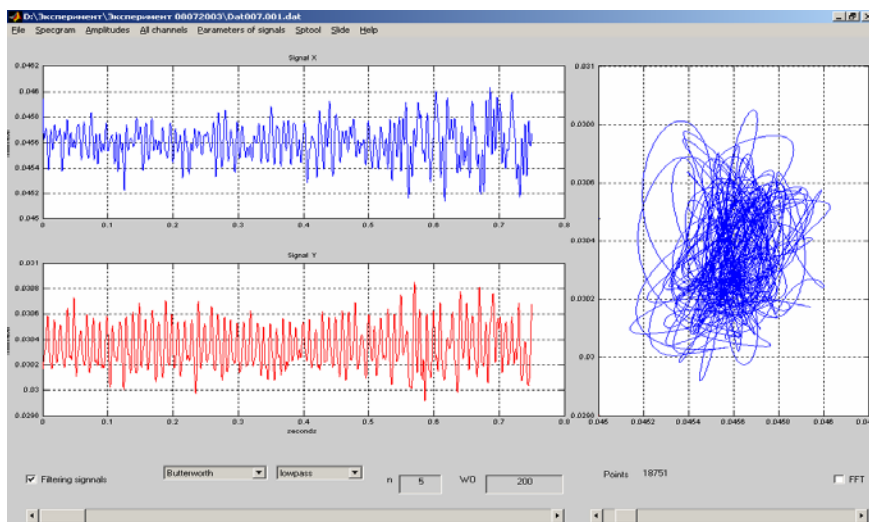


Рис. 9. Экспериментальные данные после фильтрации фильтром Баттерворта 5-го порядка с частотой среза равной 200 Гц.

После загрузки сигнала появляется возможность его прокрутки, так как в окне просмотра сигнал не отображается целиком.

Пункт меню Amplitudes позволяет построить амплитудно-частотную характеристику загруженного сигнала. А меню Specgram строит спектрограмму по сигналу X и по сигналу Y (рис. 10).

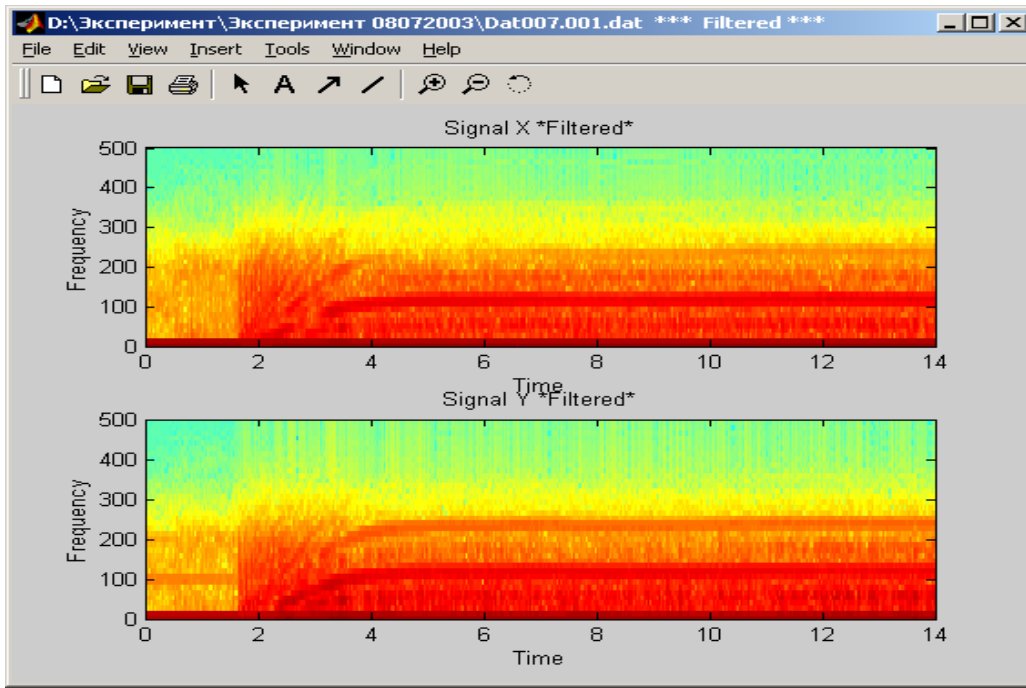


Рис. 10. Спектрограмма сигнала.

В заключение можно отметить, что среда MATLAB является универсальной системой, позволяющей создавать законченные приложения. В данной работе этот комплекс использовался на всех стадиях прикладного исследования динамики роторов на упругодемпферных опорах:

- математическое моделирование с использованием визуально ориентированной среды Simulink;
- анализ и обработка экспериментальных данных с применением специально разработанного модуля.

Разработанные приложения являются достаточно универсальными и с незначительными доработками могут использоваться для решения самых разнообразных проблем динамики роторных систем.

Литература

1. <http://www.mathworks.com>
2. Дьяконов В. П. MATLAB 6/6.1/6/5. Simulink 4/5: основы применения.— М.: 2002.— 768 с.
3. Сергиенко А. Б. Цифровая обработка сигналов.— СПб.: Питер, 2002.— 608 с.

УДК 004.94:004.75

ВОПРОСЫ ПОСТРОЕНИЯ СЕТЕВЫХ КОМПЛЕКСОВ В MATLAB/SIMULINK

Коровкин М. В., Погожев С. В.

*Санкт-Петербургский государственный университет, Санкт-Петербург,
e-mail: pogojev@yrm.apmath.spbu.ru*

При построении модели какого-либо объекта одним из основных критериев ее качества является мера ее адекватности моделируемому объекту. Для объектов, имеющих достаточно сложную структуру (например, морской подвижный объект) особую важность при моделировании имеет учет влияния распределенного размещения различных систем и механизмов, а также различий в их мощности (вычислительной, пропускной и т. п.). Эти особенности влекут за собой возникновение всевозможных задержек прохождения сигнала и обработки сигналов и, соответственно, изменение качества процессов управления, что требует проведения специальных исследований.

Одним из возможных способов создания моделей, максимально приближенных к действительности, является построение сетевых моделирующих систем, т. е. моделей, отдельные компоненты которых размещены на нескольких компьютерах, объединенных в сеть. При этом структура сетевого комплекса должна в максимально возможной степени отражать структуру моделируемой системы.

Например, при моделировании системы управления движением морских подвижных объектов можно выделить несколько функционально обусловленных составных частей:

- собственная модель объекта, описывающая его собственное движение без влияния воздействий внешней среды;
- модели внешних возмущений, действующих на объект (волнение, ветер и т. п.);
- модели алгоритмов управления (навигационная система, автопилот, противоаварийная система и др.).

Реализация подобных сетевых комплексов представляет собой довольно сложную задачу. Это связано с необходимостью обеспечения такого межмашинного обмена, который бы в известном смысле отражал характер обмена в реальной системе, и при этом удовлетворял ряду требований по эффективности и удобству управления процессом моделирования. В частности, особое внимание должно быть уделено такому специфическому вопросу, как синхронизация процессов, моделируемых на отдельных компьютерах.

Разработчикам подобных моделирующих систем MATLAB предлагает:

1. возможность использования технологий ActiveX, реализованных в ядре MATLAB для поддержки эффективного межпрограммного взаимодействия;
2. аппарат MEX-функций, практически безгранично расширяющий возможности среды;
3. инструментальный пакет Real Time Workshop, с помощью которого можно на основе существующих Simulink-моделей получать исполняемый код, соответствующий модели и независимый от системы MATLAB–Simulink.

В работе рассматривается подход, основанный на использовании технологии ActiveX, поскольку он позволяет быстро изменять параметры моделей (не требует компиляции и т. п.), достаточно прост в практическом применении и не требует специальной подготовки в области программирования.

Систему MATLAB можно использовать как в качестве сервера автоматизации (т. е. программы, выполняющей определенные действия по запросу других программ-клиентов), так и в качестве клиента.

Для того, чтобы MATLAB мог выступать в качестве сервера автоматизации, необходимо при его запуске указать ключ «/Automation».

Для использования возможностей клиентской части MATLAB, в первую очередь необходимо создать объект — экземпляр класса `activex`, представляющий интерфейс компонента или приложения — сервера автоматизации, заданный по умолчанию. Для этого в MATLAB предназначена функция `actxserver`. В качестве параметров данной функции выступают идентификатор приложения-сервера (для системы MATLAB — «MATLAB.Application») и DNS имя или IP адрес компьютера, на котором находится сервер автоматизации MATLAB.

Например, для связи среды MATLAB на текущем компьютере с MATLAB на другом компьютере с IP адресом 10.0.115.1 необходимо выполнить команду:

```
hMATLAB=actxserver('MATLAB.Application','10.0.115.1')
```

Следует отметить, что если не указать второй аргумент функции, то автоматически будет загружен экземпляр MATLAB (сервер автоматизации), если же он уже был запущен, то произойдет связь с ним (т. к. на одной машине может быть запущен только один экземпляр сервера автоматизации MATLAB).

- Интерфейс, предоставляемый MATLAB, включает в себя три метода:
- **GetFullMatrix** — для извлечения матриц из рабочего пространства;
 - **PutFullMatrix** — для размещения матриц в рабочем пространстве;
 - **Execute** — для выполнения команд.

Для вызовов указанных методов интерфейса необходимо использовать функцию **invoke**. Например, для размещения матрицы $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ в рабочем пространстве сервера MATLAB необходимо выполнить команду **invoke(hMATLAB, 'PutFullMatrix', 'A', 'base', [1 2; 3 4], zeros(2,2));**

Замечание. подробное описание параметров функции **invoke** приведено в документации к MATLAB [1].

Таким образом, при создании сетевого моделирующего комплекса необходимо «научить» каждую модель осуществлять инициализацию и на каждом шаге передачу или получение информации из другой модели.

Отметим, что единицы времени, используемые при имитационном моделировании в системе MATLAB–Simulink, не привязаны к реальному масштабу времени. Они зависят от вычислительных возможностей используемого компьютера и от сложности производимых вычислений, а потому различны на отдельных машинах сетевого моделирующего комплекса. Следовательно, необходимо принять меры для синхронизации процессов имитационного моделирования на машинах комплекса.

Удобным способом синхронизации является привязка всех процессов моделирования к реальному (машинному) времени. Для ее организации в состав Simulink-модели предлагается добавить блок S-Function, который будет обеспечивать «торможение» процесса имитационного моделирования. В приложении 1 приведен текст S-функции на языке C.

Данную функцию необходимо откомпилировать при помощи команды **mex synchronizer.c**. В случае успешного выполнения команды будет получен файл с именем **synchronizer.dll**, представляющий собой динамическую библиотеку функций, вызываемых из Simulink.

В свойствах добавленного блока S-Function необходимо указать в качестве имени S-функции имя созданной библиотеки (**synchronizer**), а в качестве параметра S-функции — число, которое будет отвечать за соотношение между реальным и имитационным временем. При значениях этого числа, больших единицы, получаем соответствующее ускорение имитационного времени относительно реального, и наоборот, при значениях, меньших единицы — замедление имитационного времени.

Для использования полученной функции в качестве синхронизатора необходимо подобрать такое значение параметра, которое позволит всем компьютерам комплекса справляться с требуемым объемом вычислений.

Пример. Рассмотрим объект управления, который описывается системой уравнений:

$$\begin{aligned}\dot{\varphi} &= \omega, \\ \dot{\omega} &= -0.242\omega - 0.0123\delta, \\ \dot{\delta} &= u - \delta.\end{aligned}\tag{1}$$

Это упрощенная линейная система дифференциальных уравнений, описывающая движение морского подвижного объекта по курсу. Здесь φ — курсовой угол, ω — угловая скорость, δ — отклонение рулей, u — управляющий сигнал. Будем считать, что стабилизация объекта осуществляется регулятором вида

$$u = 35\dot{\varphi} + 10(\varphi - \varphi^*), \tag{2}$$

где φ^* — заданный курсовой угол.

Предположим, что измерения $\varphi, \dot{\varphi}$ в алгоритм управления поступают с периодом 0.1 секунды.

На рис. 1 представлена блок-схема Simulink-модели объекта управления, описываемого уравнениями (1). На рис. 2 приведена блок-схема Simulink-модели алгоритма управления вида (2). Кроме основных блоков, формирующих модели в соответствии с уравнениями (1), (2), в состав данных Simulink-моделей входит блок S-функции `synchronizer`, описанной выше, а также блоки S-функций `get_control`, `save_state`, `from_model`, `to_model`, отвечающих за обмен данными между моделями. Тексты функций приведены в приложении 2.

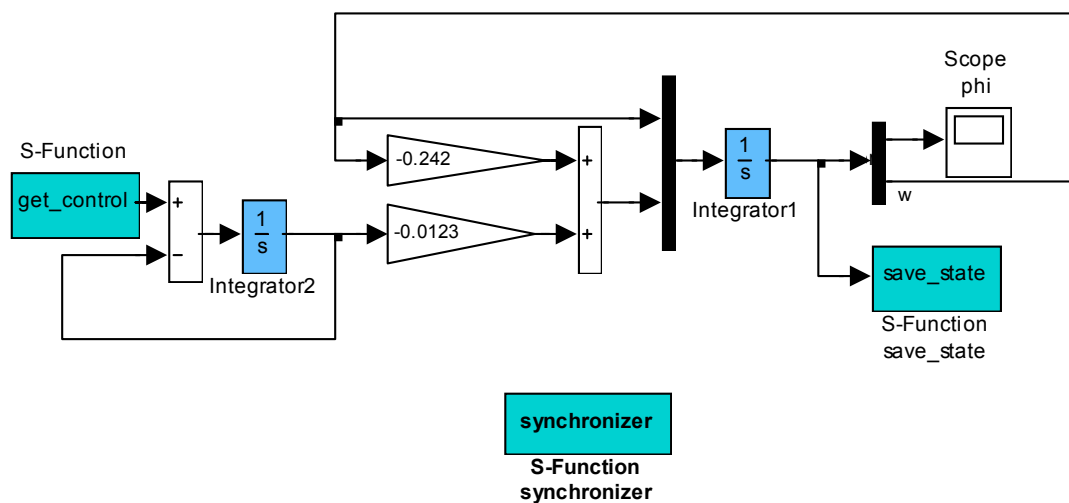


Рис. 1. Блок-схема Simulink-модели объекта управления.

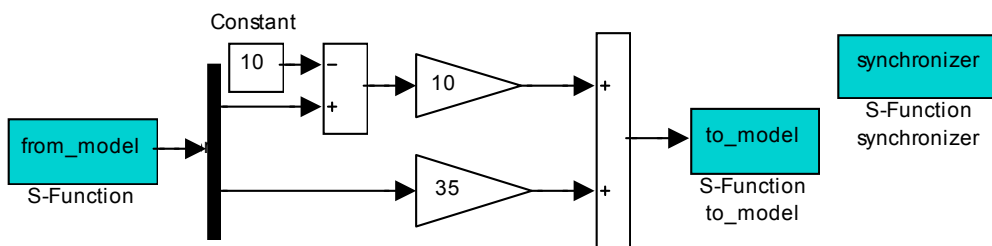


Рис. 2. Блок-схема Simulink-модели алгоритма управления.

Функция **get_control** выполняется раз в 0.1 секунды имитационного времени и получает данные из глобальной переменной Controls рабочего пространства MATLAB. Функция **save_state** сохраняет состояния объекта управления, поступающие на вход блока S-Function в глобальной переменной States рабочего пространства MATLAB.

Функция **from_model** позволяет с периодичностью 0.1 секунды получать данные из переменной States рабочего пространства сервера MATLAB, в котором находится модель объекта управления при помощи вызова функции **invoke**. Функция **to_model** — передает информацию об управлении в глобальную переменную Controls рабочего пространства сервера MATLAB.

Кроме указанных функций для инициализации модели используются функции **control_load**, **control_start**, тексты которых также приведены в Приложении 2. Функция **control_load** должна выполняться при открытии Simulink-модели, соответствующей алгоритму управления, (указывается в свойствах Simulink-модели в разделе Callbacks, свойство «Model pre-load function») или «вручную» из командного окна MATLAB. Данная функция создает сервер автоматизации MATLAB на компьютере с заданным адресом IP и сохраняет результат в глобальной переменной MODEL_PC.

Функция **control_start** должна выполняться в начале имитационного моделирования Simulink-модели алгоритма управления (указывается в свойствах Simulink-модели в разделе Callbacks, свойство «Simulation start function»). Данная функция позволяет начать имитационное моделирование Simulink-модели объекта управления на сервере автоматизации MODEL_PC.

Таким образом, получается две Simulink-модели, которые могут выполняться в двух различных экземплярах среды MATLAB и обмениваться информацией.

Литература

1. External Interfaces.— Natick: The MathWorks, Inc., 2001.— 408 p.

Приложение 1.

Текст функции synchronizer.c

```
#define S_FUNCTION_NAME synchronizer  
#define S_FUNCTION_LEVEL 2
```

```
#include «simstruc.h»  
#include «MATLAB.h»  
#include <time.h>  
#include <windows.h>
```

```
//=====
```

```
static clock_t StartTime;
```

```

#define DISCRETE_STEP 0.1
#define SYNCHR_R_IDX 0
#define SYNCHR_R_PARAM(S) ssGetSFcnParam(S,SYNCHR_R_IDX)
//=====
static void mdlCheckParameters(SimStruct *S)
{ const mxArray *para = (const mxArray *) SYNCHR_R_PARAM(S);
  real_T *N= mxGetPr(para);
  if ((mxGetM(SYNCHR_R_PARAM(S)) > 1) ||
      (mxGetN(SYNCHR_R_PARAM(S)) > 1)) {
    ssSetErrorStatus(S,»Parameter to S-function must be a scalar.»);
    return;
  }
  if (N[0]<=0){ ssSetErrorStatus(S,»Parameter to S-function must be positive!»); return;
  }
}
//=====
static void mdlInitializeSizes(SimStruct *S)
{ ssSetNumSFcnParams(S, 1);
#ifdef(MATLAB_MEX_FILE)
  if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S)) {
    mdlCheckParameters(S);
    if (ssGetErrorStatus(S) != NULL) return;
  } else return;
#endif
  ssSetNumContStates( S, 0);
  ssSetNumDiscStates( S, 0);
  if (!ssSetNumInputPorts(S, 0)) return;
  if (!ssSetNumOutputPorts(S, 0)) return;
  ssSetNumSampleTimes( S, 1);
  ssSetOptions( S, 0);
}
//=====
static void mdlInitializeSampleTimes(SimStruct *S)
{ /* Register one pair for each sample time */
  ssSetSampleTime(S, 0, DISCRETE_STEP);
  ssSetOffsetTime(S, 0, 0.0);
}
//=====
static void mdlStart(SimStruct *S)
{ StartTime=clock();
}
//=====
static void mdlOutputs(SimStruct *S, int_T tid)
{ const mxArray *para = (const mxArray *) SYNCHR_R_PARAM(S);
  double *N= mxGetPr(para);
  clock_t NowTime;
  DWORD diff_M_PC;
  real_T InTime=(real_T)ssGetT(S); //Имитационное время модели
  long Tr_Hold = 1000*DISCRETE_STEP/N[0];
}

```



```
NowTime=clock();
diff_M_PC=NowTime-StartTime;
if ((diff_M_PC>=0) && (diff_M_PC<Tr_Hold))
    Sleep(Tr_Hold-diff_M_PC);
StartTime = NowTime;
}
//=====
static void mdlTerminate(SimStruct *S)
{
//=====
#ifdef MATLAB_MEX_FILE
#include «simulink.c»
#else
#include «cg_sfun.h»
#endif
```

Приложение 2

Текст функции get_control.m

```
function [sys,x0,str,ts] = get_control(t,x,u,flag)
switch flag,
case 0, % Initialization %
    sizes = simsizes;
    sizes.NumContStates = 0; sizes.NumDiscStates = 0;
    sizes.NumOutputs = 1; sizes.NumInputs = 0;
    sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 1;
    sys = simsizes(sizes); str=[];
    ts = [0.1 0];
case 3, % Outputs %
    global Controls;
    sys=Controls(1);
case {1,2,4,9},
    sys=[];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
% end get_control
```

Текст функции save_state.m

```
function [sys,x0,str,ts] = save_state(t,x,u,flag)
switch flag,
case 0,% Initialization %
    sizes = simsizes;
    sizes.NumContStates = 0; sizes.NumDiscStates = 0;
    sizes.NumOutputs = 0; sizes.NumInputs = 2;
    sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 1;
    sys = simsizes(sizes); str=[];
    ts = [0.1 0];
case 3,
    global States;
```

```

    States=u;
    sys=[];
case {1,2,4,9},
    sys=[];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
% end save_state

```

Текст функции from_model.m

```

function [sys,x0,str,ts] = from_model(t,x,u,flag)
switch flag,
case 0, % Initialization %
    sizes = simsizes;
    sizes.NumContStates = 0; sizes.NumDiscStates = 0;
    sizes.NumOutputs = 2; sizes.NumInputs = 0;
    sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 1;
    sys = simsizes(sizes); str=[];
    ts = [0.1 0];
case 3, % Outputs %
    global MODEL_PC;
sys=invoke(MODEL_PC,'GetFullMatrix','States','global',
zeros(2,1),zeros(2,1));
case {1,2,4,9},
    sys=[];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
% end from_model

```

Текст функции to_model.m

```

function [sys,x0,str,ts] = to_model(t,x,u,flag)
switch flag,
case 0,% Initialization %
    sizes = simsizes;
    sizes.NumContStates = 0; sizes.NumDiscStates = 0;
    sizes.NumOutputs = 0; sizes.NumInputs = 1;
    sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 1;
    sys = simsizes(sizes); str=[];
    ts = [0.1 0];
case 3,
    global MODEL_PC;
        invoke(MODEL_PC,'PutFullMatrix','Controls','global',
            [u 0],zeros(1,2));
    sys=[];
case {1,2,4,9},
    sys=[];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

```

% end to_model

Текст функции control_load.m

```
function control_load  
global MODEL_PC States  
MODEL_PC=actxserver('MATLAB.application','127.0.0.1');  
invoke(MODEL_PC,'execute','model');  
invoke(MODEL_PC,'execute','global Controls');  
invoke(MODEL_PC,'execute','Controls=0;');  
States=zeros(2,1);
```

Текст функции control_start.m

```
function control_start  
global MODEL_PC  
invoke(MODEL_PC,'execute','Controls=0;');  
invoke(MODEL_PC,'execute', ...  
'set_param(''model'', 'simulationcommand', 'start');');
```

УДК 629.735.33

РАЗВИТИЕ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ МОДЕЛИРОВАНИЯ И АНАЛИЗА ДИНАМИКИ САМОЛЕТА С СИСТЕМОЙ УПРАВЛЕНИЯ В СРЕДЕ ПРОГРАММИРОВАНИЯ MATLAB/SIMULINK («FLIGHTSIM»)

Кувшинов В. М., Анимица О. В.

Центральный аэрогидродинамический институт (ЦАГИ)

им. проф. Н.Е. Жуковского, Жуковский,

e-mail: vmkouvsh@mail.progtech.ru

На первой конференции авторами был представлен программный комплекс **FlightSim** для моделирования и анализа динамики ЛА с системой управления. За прошедшие два года комплекс получил дальнейшее развитие, в том числе на основе опыта его практического использования в отечественных и зарубежных организациях авиационной промышленности. В данном докладе подробно рассмотрены новые элементы комплекса, при этом основное внимание уделено описанию его элементов, которые могут быть использованы не только в задачах динамики ЛА, но и в других областях возможного применения программной среды MATLAB/Simulink.

В настоящее время комплекс **FlightSim** включает следующие составные части:

Библиотеки элементов для моделирования динамики самолета с системой управления:

- библиотеку уравнений движения самолета с набором диалоговых модулей настройки начальных условий.
- библиотеку вспомогательных элементов для формирования модели динамики ЛА (стандартная атмосфера, измерения датчиков, переходы между различными системами координат и др.)
- библиотеку математических моделей элементов системы управления ЛА таких как нелинейные рулевые приводы (в том числе с учетом влияния шарнирных моментов), датчики, типовые элементы алгоритмов управления и т. п.
- библиотеку вспомогательных элементов для моделирования движения ЛА в реальном времени (сетевые драйверы, драйверы АЦП/ЦАП, драйвер джойстика, модуль синхронизации, функция автоматизированного установления приоритетов).
- библиотеку программ визуализация движения самолета, в том числе с помощью технологии виртуальной реальности, и средств для документирования результатов и создания презентаций.

Пакет программ балансировки моделей, расчета балансировочных характеристик ЛА и характеристик управляемости.

Пакет программ управления базами данных самолетов для использования в динамических моделях системы MATLAB/Simulink.

*Библиотеку программ расчета и анализа динамических характеристик самолета с САУ, заданных моделью системы Simulink и комплекса **FlightSim**:*

- расчет, построение и обработка частотных характеристик;
- расчет, построение и обработка корневых годографов;
- построения и анализ переходных процессов;
- расчет статистических характеристик (спектральных плотностей, среднеквадратических значений, показателей комфорта пассажиров и повреждаемости конструкции) при действии мелкомасштабной турбулентности;
- генерирование реализаций случайного ветра по моделям Кармана и Драйдена;
- расчет статистических характеристик по реализации случайных сигналов.

Примеры моделей динамики гипотетических самолетов с системой управления.

Специализированную программу расчета характеристик взаимодействия систем управления с упругими колебаниями конструкции ЛА и синтеза активных систем управления самолетов.

Библиотеки элементов для моделирования динамики самолета

Библиотека уравнений движения самолета (рис. 1) содержит модули уравнений пространственного движения в виде S-функций на языке С, записанные в различной форме — в углах Эйлера, в направляющих косинусах и в кватернионах. Особенностью этих модулей является наличие прямой коррекции направляющих косинусов или кватернионов и возможность моделирования всех режимов движения самолета, включая режимы взлета, посадки, пробега и руления с учетом модели динамики шасси при скоростях движения $V \approx 0$, что необходимо при профессиональном использовании программ. По сравнению с предыдущей версией дополнительно разработаны блоки пространственного движения с уравнениями сил в скоростной системе координат, что дает возможность рассматривать движение самолета с постоянной скоростью (весьма популярный вариант при анализе динамики самолета). Дополнительно включены также блоки упрощенных уравнений пространственного движения 5-го и 8-го порядков, часто используемых при специальных исследованиях динамики пространственного движения. Библиотека включает также модули изолированного

продольного и бокового движения самолета с возможностью отключения фугоидного и спирального видов движений.

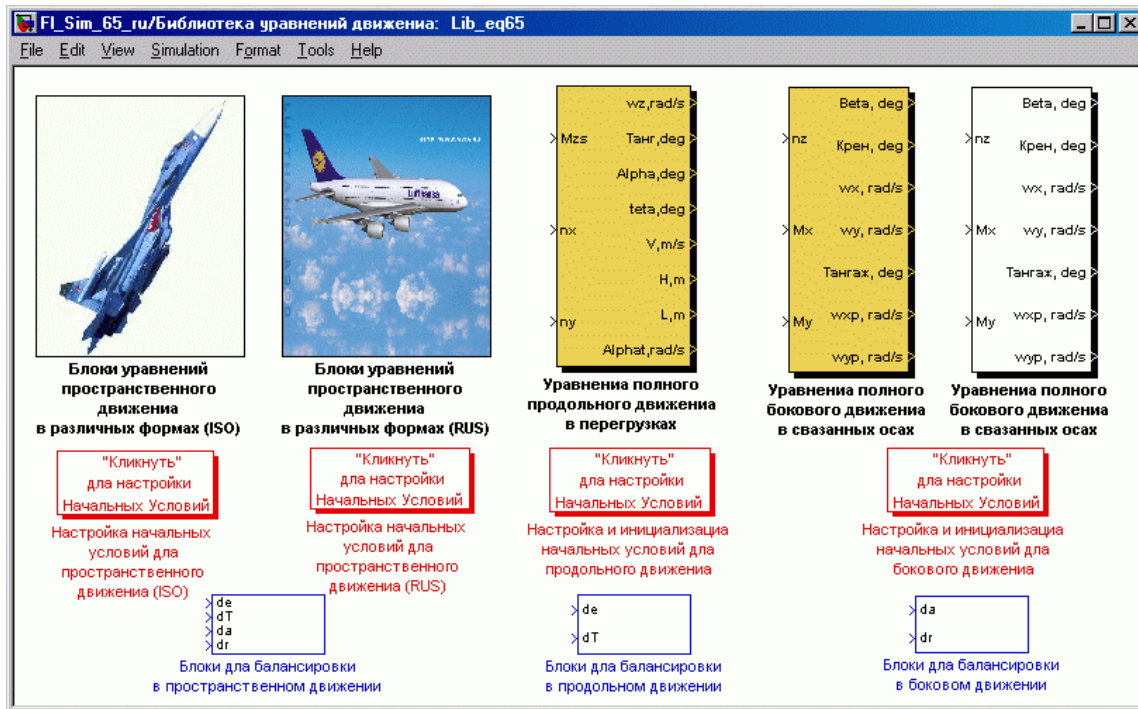


Рис. 1. Библиотека уравнений движения самолета.

Вопрос моделирования движения самолета на пилотажном стенде в реальном времени является одним из основных при исследовании динамики полета. Накопленный в последнее время опыт моделирования показал, что при использовании мощных современных компьютеров ($f \geq 2$ МГц, ≥ 512 Мб) и при использовании операционных систем Windows NT, Windows 2000, Windows XP нет необходимости использовать дорогостоящие пакеты программ Real-Time Workshop и Real-Time Windows Target, имеющие, кроме того, большое количество недостатков и неудобств. Вполне возможна работа непосредственно в Simulink в операционной системе Windows без компиляции модели (или с частичной компиляцией средствами Simulink Performance Tools). Быстродействие современного компьютера вполне достаточно для интегрирования весьма сложной модели динамики современного самолета с системой управления с шагом $\Delta t = 0.001 \div 0.002$ с, а синхронизация модели с реальным временем может быть осуществлена включением в модель специального блока с циклом ожидания соответствующего синхроимпульса с часов операционной системы Windows, либо с других часов. Такой блок в виде специальной S-функции разработан и включен в комплекс **FlightSim**.

Также для моделирования в реальном времени необходима организация связи программы моделирования динамики самолета с устройствами ввода–вывода, в первую очередь платами аналого-цифровых (АЦП) и циф-

ро-аналоговых (ЦАП) преобразователей, для обеспечения связи с аппаратурой реального времени. Наиболее распространенными в нашей стране платами АЦП и ЦАП для персональных компьютеров, имеющими, кроме того, относительно невысокую стоимость, являются платы производства российского предприятия L-CARD. Для плат и внешних блоков («крейтов») данного типа были написаны драйверы в виде S-функций системы Simulink, позволяющие модели динамики самолета осуществлять прием входных сигналов от внешних источников, в том числе, от рычагов управления самолета и выдавать непрерывные выходные сигнал, например, для управления реальными рулевыми приводами при полунатурном моделировании.

Для обеспечения взаимодействия программы моделирования динамики самолета с системами визуализации и управления подвижностью необходима организация межмашинного интерфейса. Для этой цели еще в 1998 г. написаны специальные драйверы сетевой платы в виде S-функций системы Simulink, которые используют специальные функции связи с сетью языка C. Драйверы позволяют осуществлять передачу в сеть и прием до 183 переменных типа **double**. Передача данных производится по протоколу UDP. Аналогичные драйверы появились также в специальном пакете программ xPC Target, начиная с Release 12, однако, этот пакет имеет достаточно высокую стоимость и содержит большое количество ненужных программ.

Для обеспечения связи программы моделирования динамики самолета с устройствами реального времени и другими компьютерами необходимо включить S-функции с разработанными драйверами, в общую модель.

Библиотека программ визуализации движения самолета

За последнее время значительно расширена библиотека программ визуализация движения самолета и средств документирования результатов моделирования и создания презентаций. В первую очередь отметим разработанную подбиблиотеку программ визуализации движения самолета на базе технологии виртуальной реальности. Данная библиотека содержит (рис. 2):

- блоки преобразования углов Эйлера самолета во вращательные координаты **VR Toolbox** (два варианта — через направляющие косинусы и через кватернионы);
- блоки типа **VR Sink**, содержащие сложные примеры виртуальных миров с различными самолетами, ВПП и окружающей средой;
- блок функции для создания видеороликов в формате *.avi из изображений виртуальных миров.

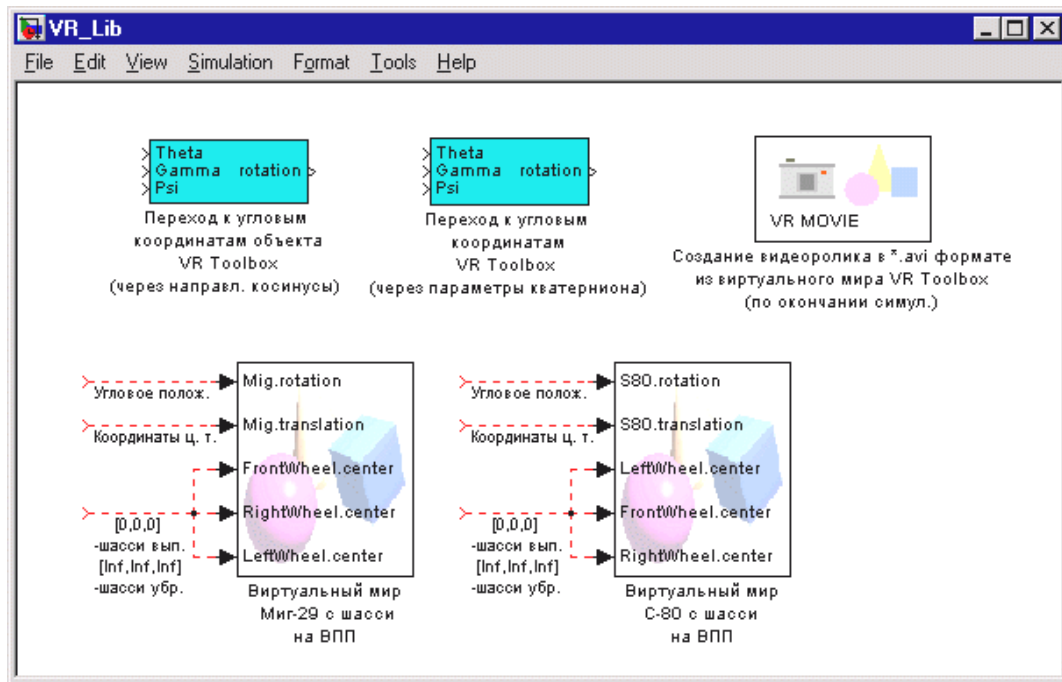


Рис. 2. Блоки библиотеки *VR Aircraft_Lib* для работы с *VR Toolbox*.

Пример виртуального мира с достаточно сложным и детальным изображением самолета МИГ-29 с подвесками и убирающимися шасси показан на рис. 3. Кроме самолета виртуальный мира включает ВПП, башни управления и маяки, несколько групп зданий, автомобили и дороги. Данный виртуальный мир включает также несколько камер, обеспечивающих вид из кабины летчика, вид на самолет с различных сторон и виды на ВПП.

Следует отметить, что, к сожалению, время перерисовки подобного достаточно сложного виртуального мира относительно велико $\tau \approx 0.02 \div 0.04$ с, даже при использовании мощной современной видеокарты. Поэтому при моделировании в реальном времени с использованием средств виртуальной реальности целесообразно проводить моделирование на двух компьютерах, соединенных сетевой связью. На одном компьютере выполняется интегрирование модели динамики самолета с достаточно мелким шагом в реальном времени, на втором выполняется визуализация. При этом в модель на первом компьютере включаются разработанные блоки передачи координат самолета в сеть, синхронизации с реальным временем и приема сигнала с имитатора ручек управления (драйвер АЦП) или драйвер джойстика. Модель на втором компьютере должна включать блоки приема координат самолета из сети, преобразования в координаты виртуального мира и блок самого виртуального мира. Синхронизация этой модели с реальным временем не обязательна, а шаги приема сигнала и перерисовки выбирается так, чтобы обеспечить максимально возможную частоту перерисовки для используемого компьютера и видеокарты.

Большой интерес для создания презентаций представляет разработанная функция записи видеороликов в формате *.avi из изображений виртуальных миров, с возможностью выбора времени и частоты записи, а также способа упаковки видеофайла. Пример в виде последовательности нескольких кадров из созданного видеофайла показан на рис.4.

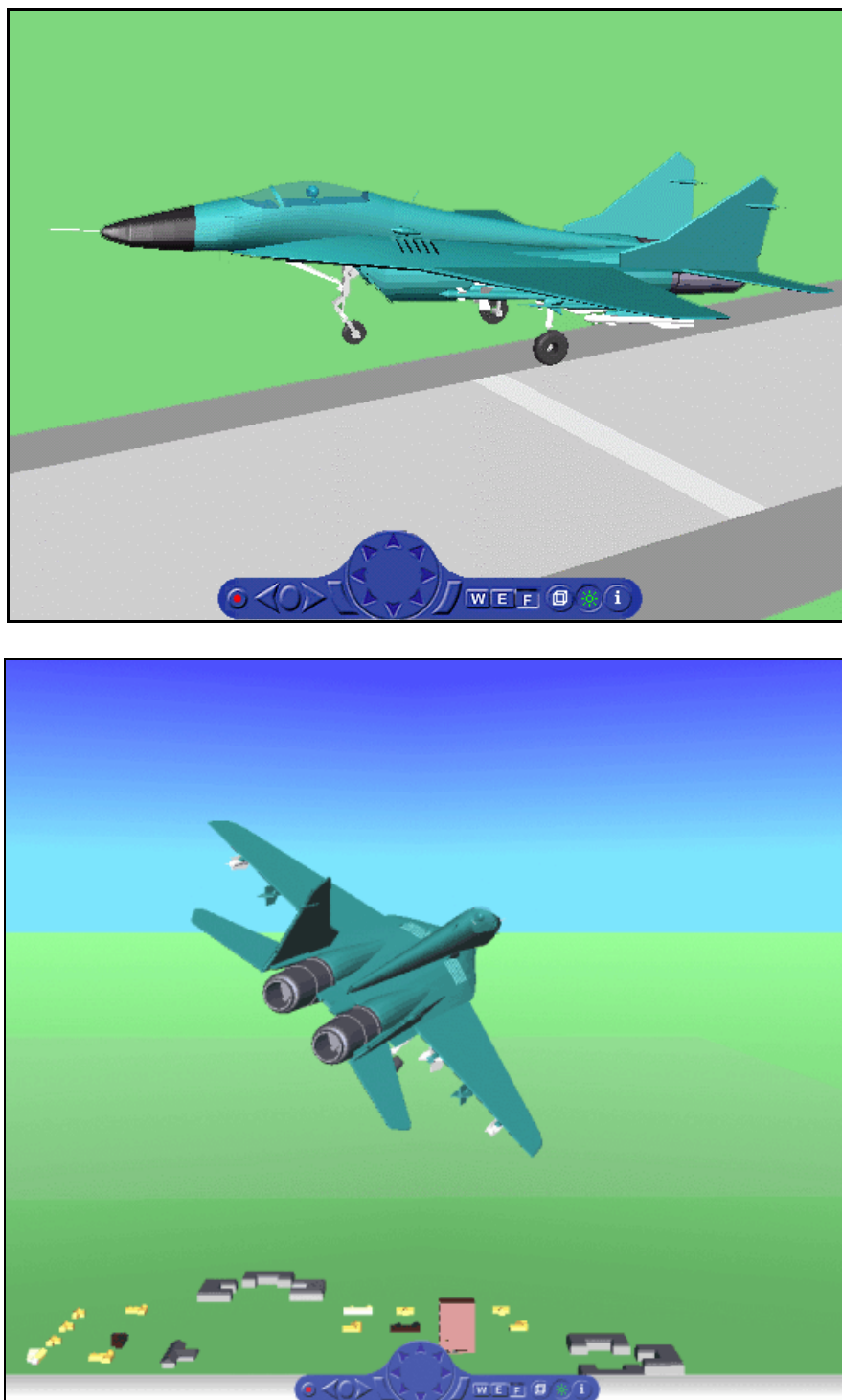


Рис. 3. Пример виртуального мира, содержащего самолет, ВПП и объекты на земле.

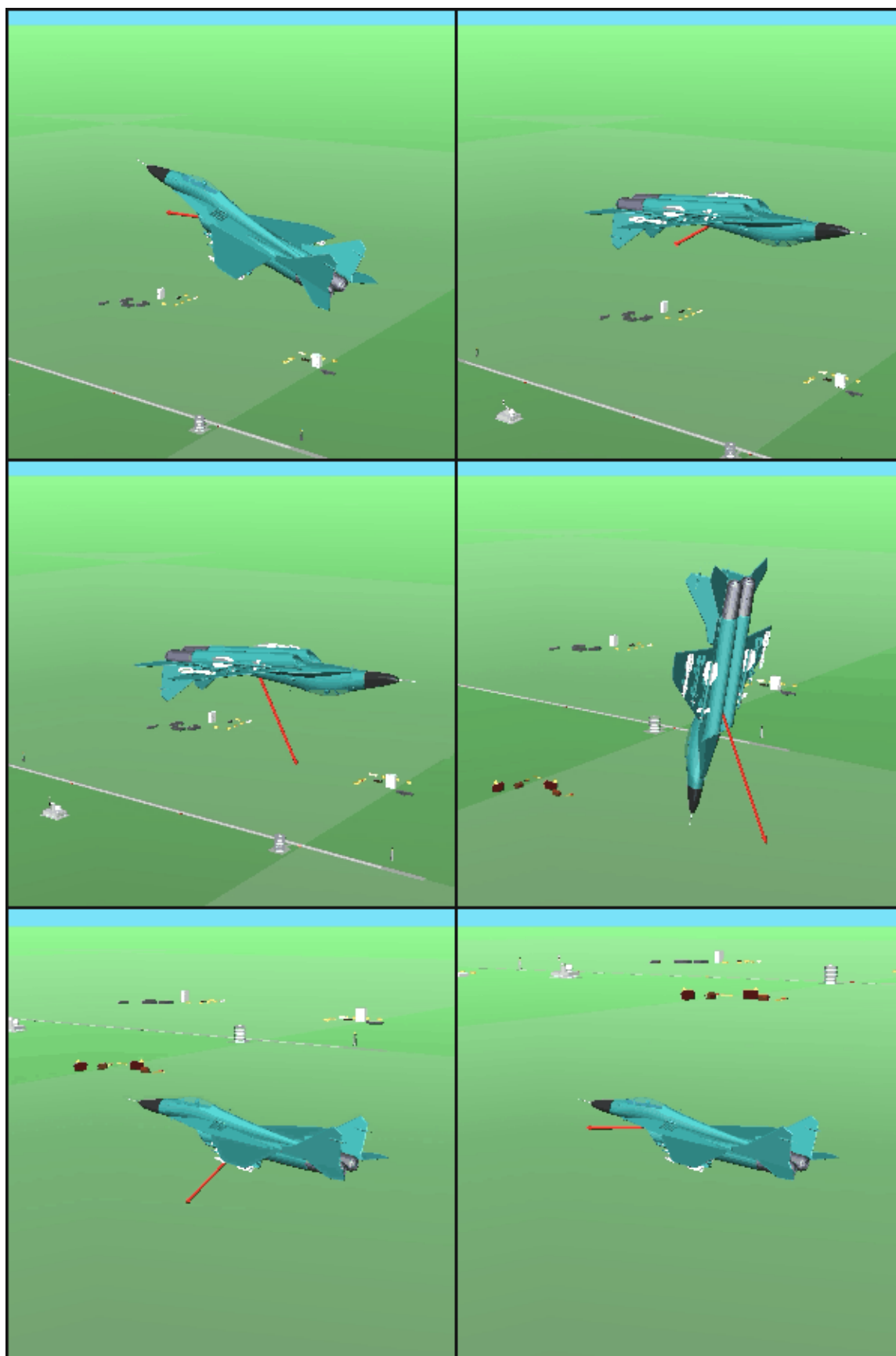


Рис. 4. Пример кадров видеоролика *.avi, записанного из виртуального мира.

Расширена и дополнена также библиотека обычных блоков визуализации и документирования результатов моделирования (рис. 5). Модифицированы блоки, входившие в предыдущую версию комплекса:

- имитаторы приборной доски самолета,
- построение 3-D траектории в процессе и после моделирования,
- визуализацию пространственного положения самолета,
- периодический захват текущих картинок движения самолета и создание видеоклипов в формате .avi.

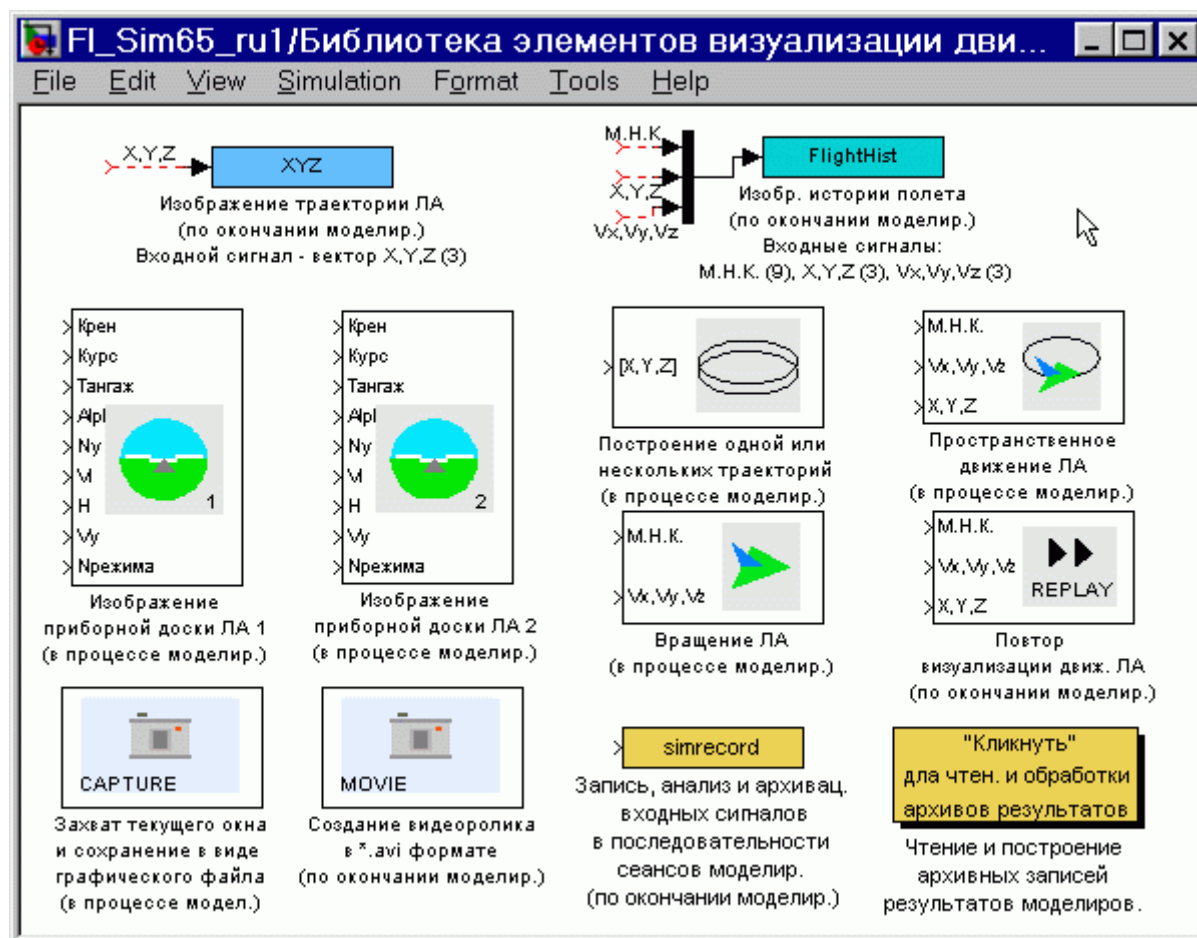


Рис. 5. Библиотека визуализации и документирования результатов моделирования.

Дополнительно разработан блок изображения истории полета, запускаемый после окончания моделирования и рисующий картинку типа показанной на рис. 6.

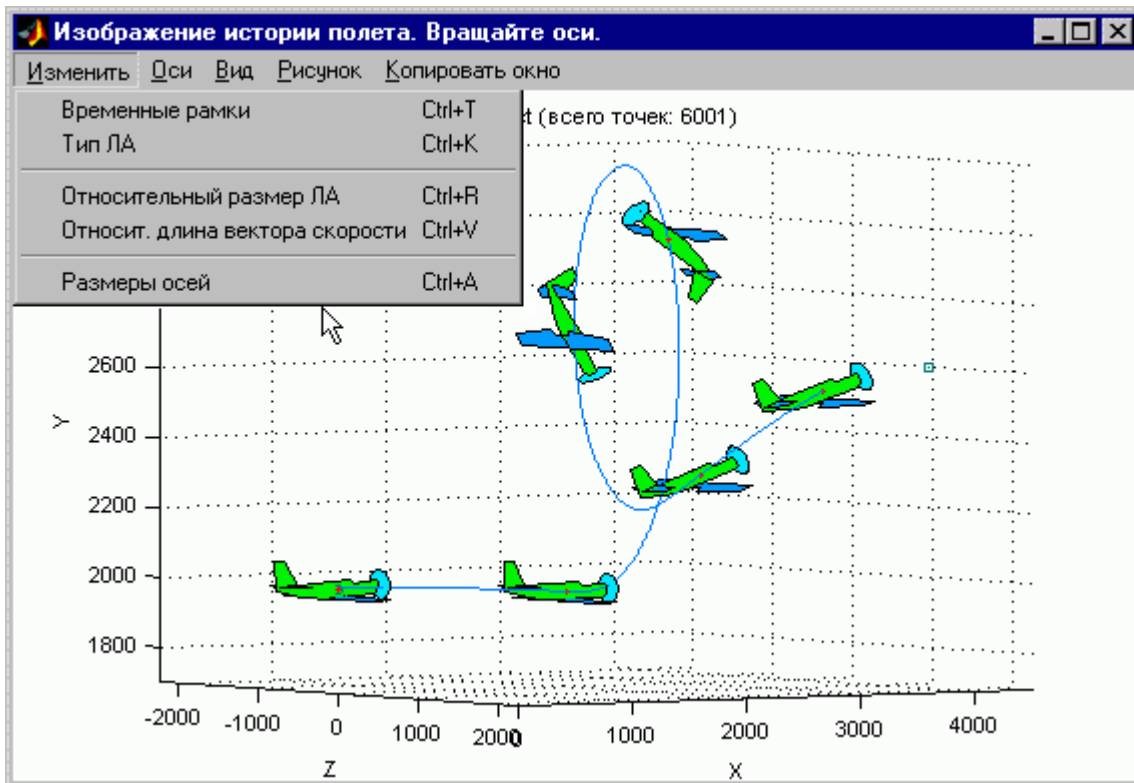


Рис. 6. Пример окна с изображением истории полета ЛА.

Наибольший интерес может представлять универсальный блок записи, анализа и архивации сигналов в последовательности сеансов моделирования. Дело в том, что при разработке систем управления различных объектов, в том числе ЛА, часто необходимо сравнивать переходные процессы при различных значениях параметров системы управления. Чтобы сделать это, необходимо строить переходные процессы, полученные в разных сеансах моделирования (при различных запусках интегрирования модели) на одном рисунке. Simulink не имеет подобных встроенных блоков. Для этих целей разработана специальная функция `sim_record_ru` и связанный с ней блок записи данных в рабочее поле MATLAB.

Для использования данной функции необходимо присоединить связанный блок к рассматриваемому сигналу. Перед началом записи пользователь должен двойным щелчком левой клавиши «мыши» по блоку открыть диалоговое окно с запросом имени переменной для хранения переходных процессов в рабочем поле MATLAB и количества осей (входных сигналов). После выбора имени и количества осей программа откроет специальное окно (пока пустое). Отметим, что можно одновременно использовать несколько блоков для записи различных сигналов, однако имена переменных для записи сигналов должны быть разными.

После окончания каждой сессии моделирования очередной переходной процесс добавляется в каждые из осей к процессам, построенным там ранее. Переходные процессы добавляются до тех пор, пока данное окно не

будет закрыто. Пример переходных процессов, полученных в двух сессиях моделирования, показан на рис.7.

При выборе пункта меню **Анализ** можно быть создано новое окно, с помощью которого пользователь может проводить анализ полученных переходных процессов. Новое окно дает возможность строить асимптоты, наносить на график точки значений заброса, времени срабатывания, другие возможности (рис. 8).

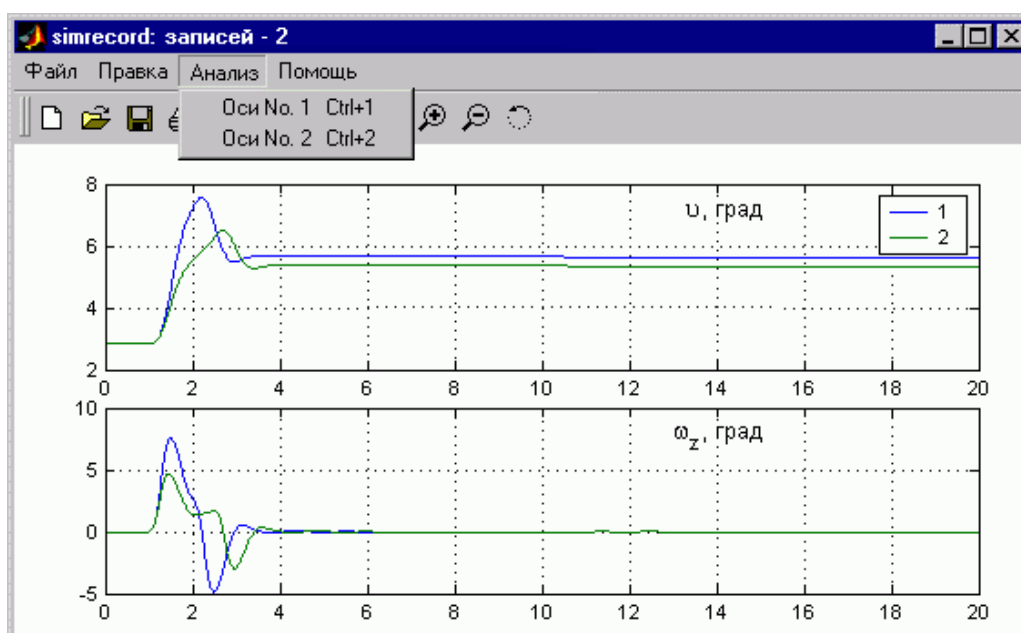


Рис. 7. Окно записи переходных процессов (2 сеанса записи).



Рис. 8. Окно анализа переходных процессов.

Данное окно позволяет также производить запись полученных переходных процессов в архив в текстовом (в ASCII формате) или двоичном виде (в виде .mat-файла) с автоматическим формированием заголовка (комментария) и возможностью его предварительного редактирования.

Чтение, обработка и построение результатов из созданных таким образом архивов осуществляется специально разработанной программой поддержки архивации результатов моделирования.

Библиотека программ балансировки самолета с САУ

Обычно моделирование движения самолета начинают из условий сбалансированного полета на некотором установившемся режиме, чаще всего из установившегося горизонтального полета на некоторой высоте и скорости. Поиск балансировочных значений параметров движения самолета (угла атаки, угла скольжения и т. п.) и балансировочных отклонений ручек управления (ручки летчика, педалей, положения РУД и т. п.) или органов управления (стабилизатора, элеронов, тяги и т. п.) самолета является достаточно сложной задачей, особенно для нелинейных аэродинамических характеристик самолета на больших углах атаки.

Кроме того, одним из важных этапов анализа динамики самолета с системой управления (особенно в ОКБ) является анализ поведения балансировочных характеристик самолета и характеристик управляемости по области режимов полета. Поэтому оказалась остро необходимой разработка и включение в комплекс **FlightSim** программ расчета балансировочных значений переменных состояния самолета с системой управления и автоматической балансировки модели. Как показал опыт использования для балансировки функции *trim* из библиотеки стандартных программ **MATLAB**, эта функция для реальных моделей динамики самолета с системой управления работает крайне неустойчиво и ненадежно и только в редких случаях позволяет рассчитать балансировочные значения переменных. Причины, по нашему мнению, заключаются как в неудачной организации работы управляющей программы расчета балансировки, когда при малейшей трудности в решении задачи программа автоматически освобождает фиксированные переменные — высоту и скорость полета, так и в неудачном выборе алгоритма оптимизации и его конкретной реализации, в частности, в отсутствии возможности введения масштабирования.

Поэтому авторами разработан альтернативный вариант программ балансировки и расчета балансировочных характеристик, основанный на непосредственном решении системы нелинейных алгебраических уравнений равновесия самолета с использованием собственных программ и диалоговом задании условий балансировки.

Для осуществления процесса балансировки необходимо включить в верхний уровень модели динамики ЛА с САУ вспомогательные блоки —

входы модели (стандартные блоки типа Inport) и балансировочные значения (стандартные блоки типа Constant), как это показано на рис.9 на примере продольного движения. Дополнительные вспомогательные блоки необходимы для того, чтобы определить балансировочные значения переменных модели и управляющих сигналов по тангажу и тяге и автоматически установить эти значения как начальные условия и входные сигналы.

Первый вариант программ балансировки осуществляет только полную балансировку, т. е. расчет балансировочных значений, как параметров движения ЛА, так и т.н. «внутренних переменных», т. е. переменных состояния различных элементов системы управления — рулевых приводов, фильтров и т. п. Диалоговое окно такой программы для продольного движения показано на рис.10. После расчета балансировочные значения переменных модели и управляющих сигналов, выполняемого по нажатию кнопки

Начать поиск, по нажатию кнопки **Балансировать модель** может быть автоматически выполнена балансировка модели:

- найденные значения командных сигналов устанавливаются в блоках **de_trim**, **T_trim** вместо предустановленных начальных значений;
- найденные балансировочные значения переменных состояния (как переменные состояния самолета, так и внутренние переменные состояния системы управления) устанавливаются в модели в качестве начальных условий моделирования.

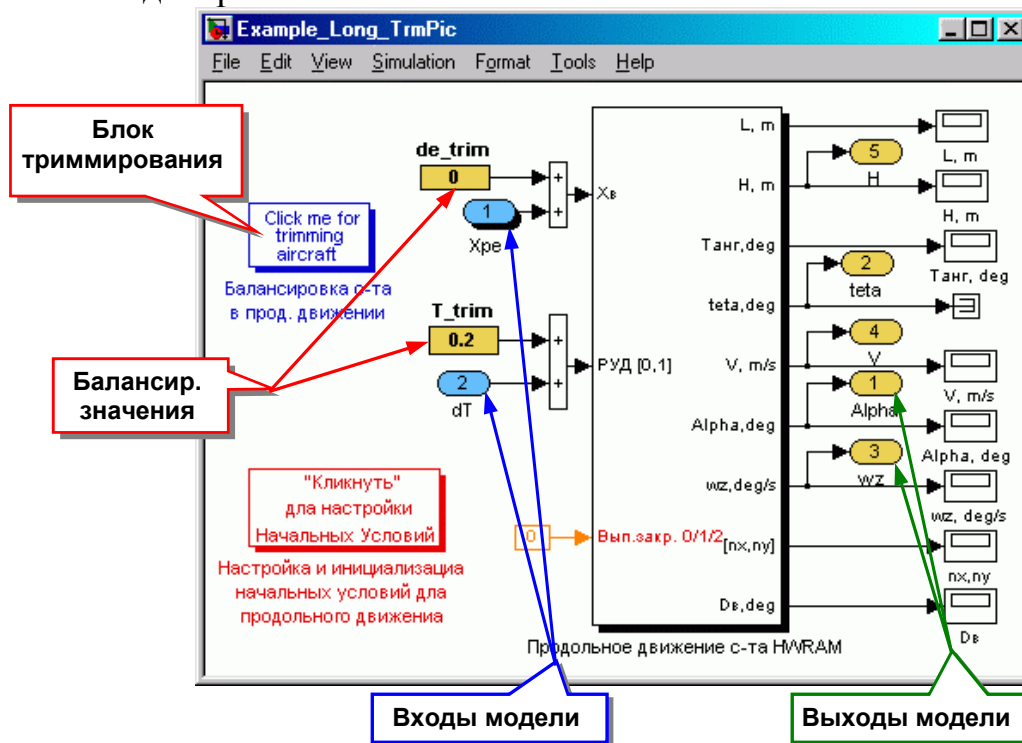


Рис. 9. Схема модели со вспомогательными блоками для поиска балансировки.

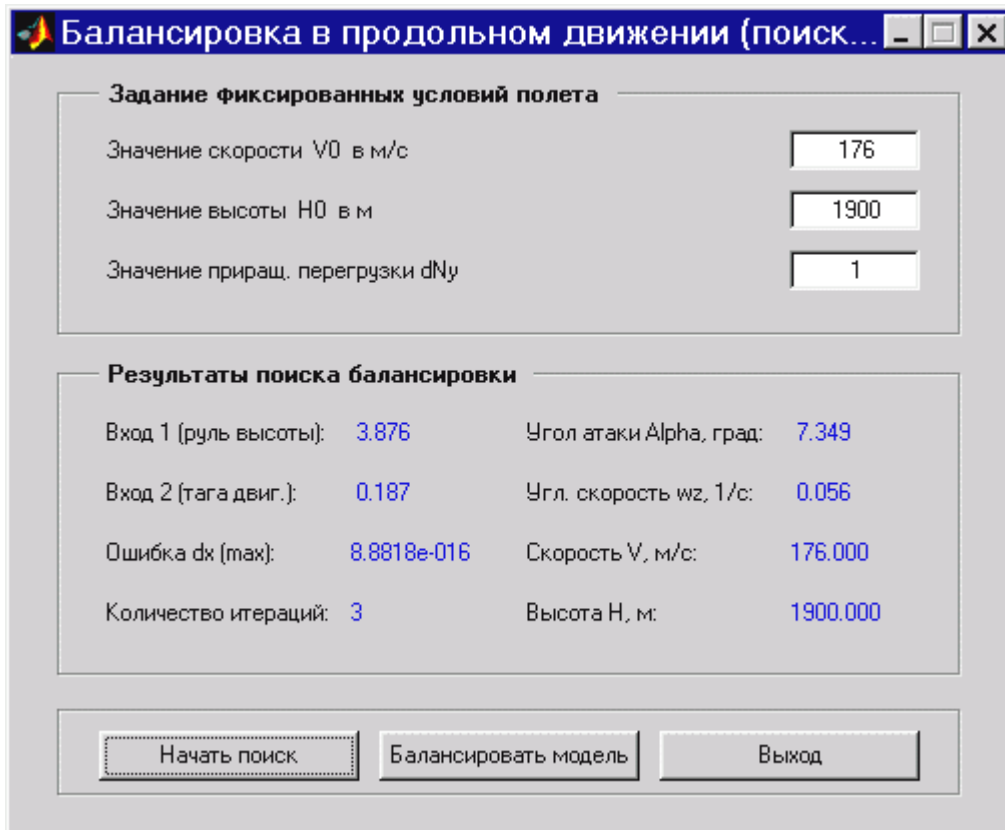


Рис. 10. Диалоговое окно подпрограммы поиска балансировки в продольном канале.

Опыт использования рассмотренных выше программ балансировки показал, что в ряде случаев при рассмотрении систем управления со сложными нелинейными алгоритмами, включающими различные переключения, общая одновременная балансировка самолета с системой управления не удается. Часто эту трудность удается преодолеть, используя прием временного вырождения нелинейностей, но иногда такое простое вырождение невозможно.

В подобных случаях может использоваться частичная или последовательная балансировка замкнутой системы самолет-САУ. При таком подходе вначале балансируют самолет (полностью или частично), а затем, используя полученные балансировки как начальные условия, пытаются сбалансировать систему управления или ее отдельные элементы.

В предыдущих версиях программ балансировки комплекса **FlightSim**, использующих функцию MATLABa **trim**, такой подход был нереализуем. При переходе к непосредственному решению системы нелинейных алгебраических уравнений равновесия самолета в данной версии появилась возможность организовать такую последовательную балансировку. Разработана комплексная программа балансировки самолета с системой управления, реализующая, в том числе, процедуру последовательной балансировки в достаточно удобной диалоговой форме. Программа позволяет также провести расчет балансировочных характеристик при

варьировании заданной перегрузки, а также в зависимости от одного или двух параметров самолета или режима полета (параметров модели). Данная программа может быть названа менеджером балансировки самолета.

Пример диалогового окна данной программы для продольного движения самолета с САУ показан на рис.11. Программа позволяет в диалоговом режиме на любом этапе выбирать набор используемых условий балансировки и варьируемых переменных для удовлетворения этим условиям. Имена переменных формируются автоматически из названий блоков модели при запуске программы. После расчета балансировочных значений варьируемых переменных, которые выводятся в специальных окнах, эти значения могут быть заданы в качестве новой начальной точки (по нажатию кнопки **Обновить нач. значения**) и процедура поиска балансировки может быть повторена при изменении набора условий балансировки. Кроме ручного режима выбора условий балансировки предусмотрен режим последовательной балансировки, когда автоматически выполняются

1. балансировка при выбранном наборе условий и варьируемых переменных,
2. перенос балансировочных значений в начальные значения,
3. установка в качестве условий балансировки, условий исключенных на первом этапе, а в качестве варьируемых переменных, переменных фиксированных на первом этапе;
4. балансировка оставшихся переменных.

Программа менеджер балансировки включает также дополнительные функции расчета балансировочных характеристик при варьировании заданной перегрузки, а также в зависимости от одного или двух параметров самолета или режима полета (параметров модели). Балансировочные характеристики рассчитываются и строятся для всех выходов модели, определяемых стандартными блоками типа **Outport**.

При запуске расчета и вывода на графики балансировочных характеристик при варьировании двух параметров программа выполняет запрос идентификаторов варьируемых параметров и диапазонов и шагов их варьирования (рис.12). Пример рассчитанной зависимости балансировочного положения стабилизатора от скорости полета и положения центра тяжести самолета приведен на рис.13.

Программа создает также общий график балансировочных зависимостей для выбранных выходов на одном рисунке, что предусмотрено по просьбам специалистов авиационных ОКБ, где подобные графики очень распространены. Пример данного графика приведен на рис.14. Для вывода подобных графиков разработана специальная функция **multiplot2.m**, обеспечивающая построение нескольких групп зависимостей с различными масштабами оси ординат для разных групп на одном рисунке. Подобная программа отсутствует в **MATLABe**, и может быть использована и в других случаях независимо от комплекса **FlightSim**.

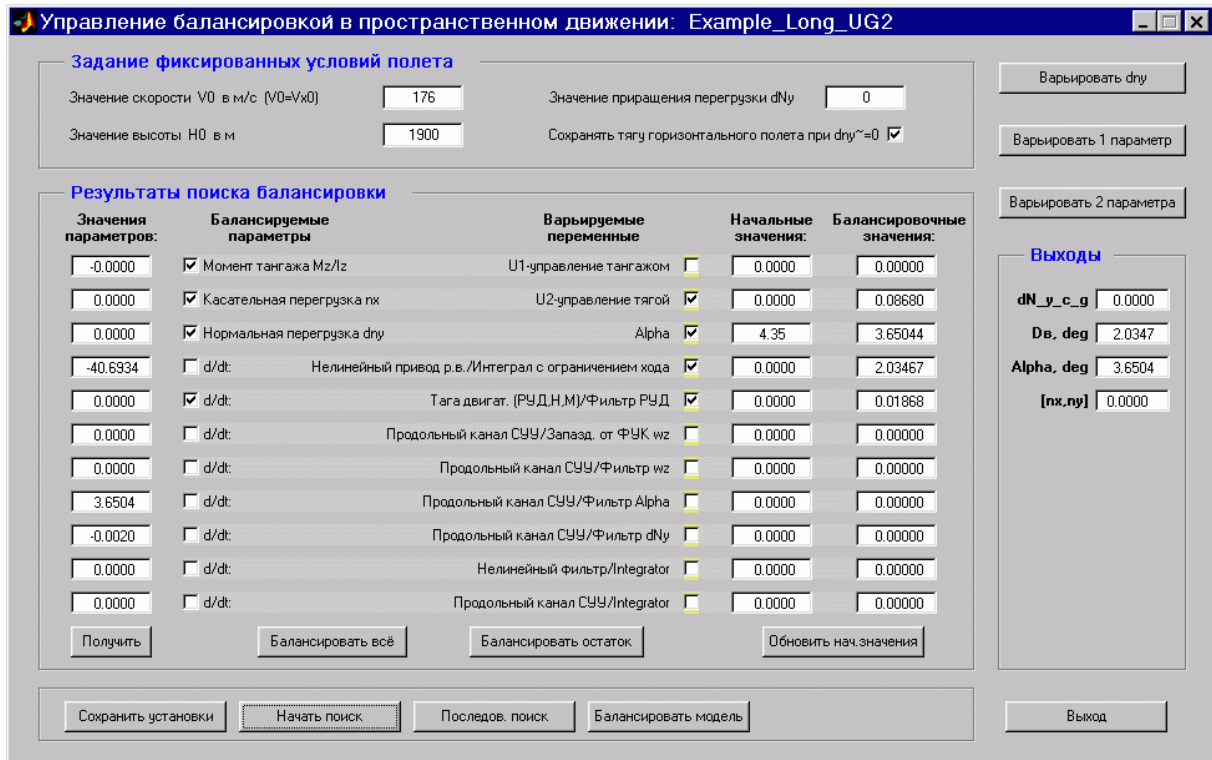


Рис. 11. Диалоговое окно программы менеджера балансировки самолета.

Идея последовательной балансировки и разработанные соответствующие программы — менеджеры балансировки могут быть использованы не только при исследовании динамики ЛА, но и в других областях, где расчет равновесного положения объекта является важной задачей.

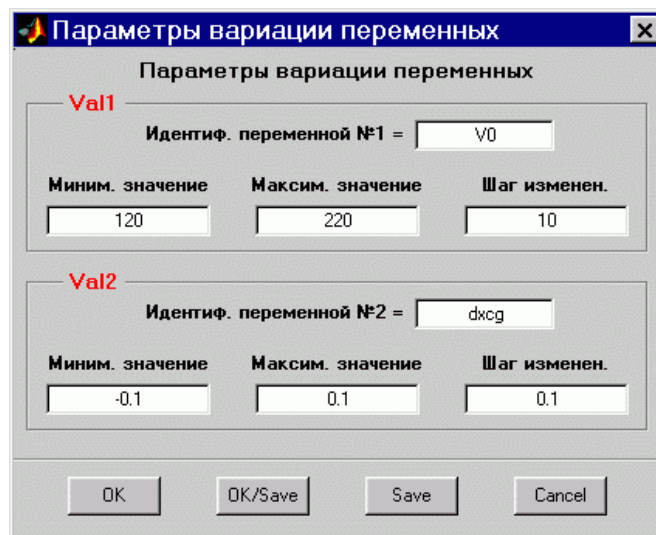


Рис. 12. Диалоговое окно настройки двух варьируемых параметров.

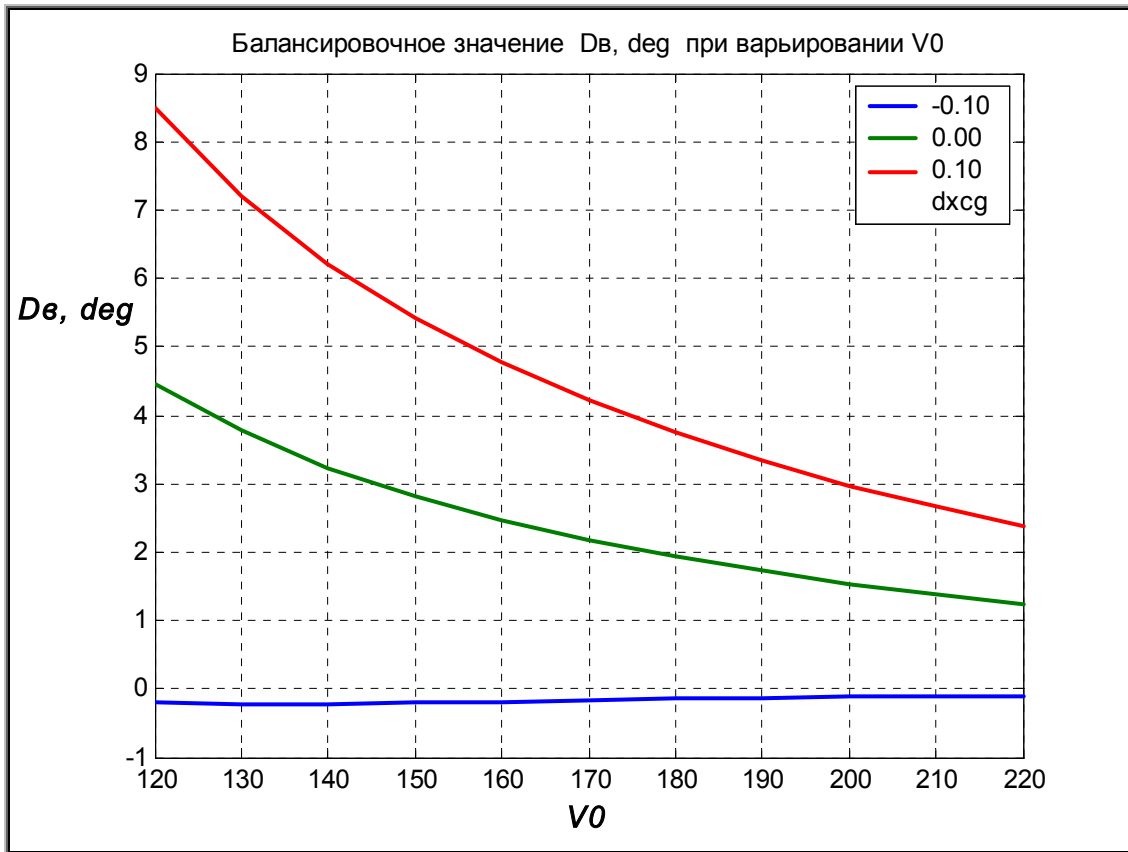


Рис. 13. Зависимость балансирующего положения стабилизатора от скорости полета.

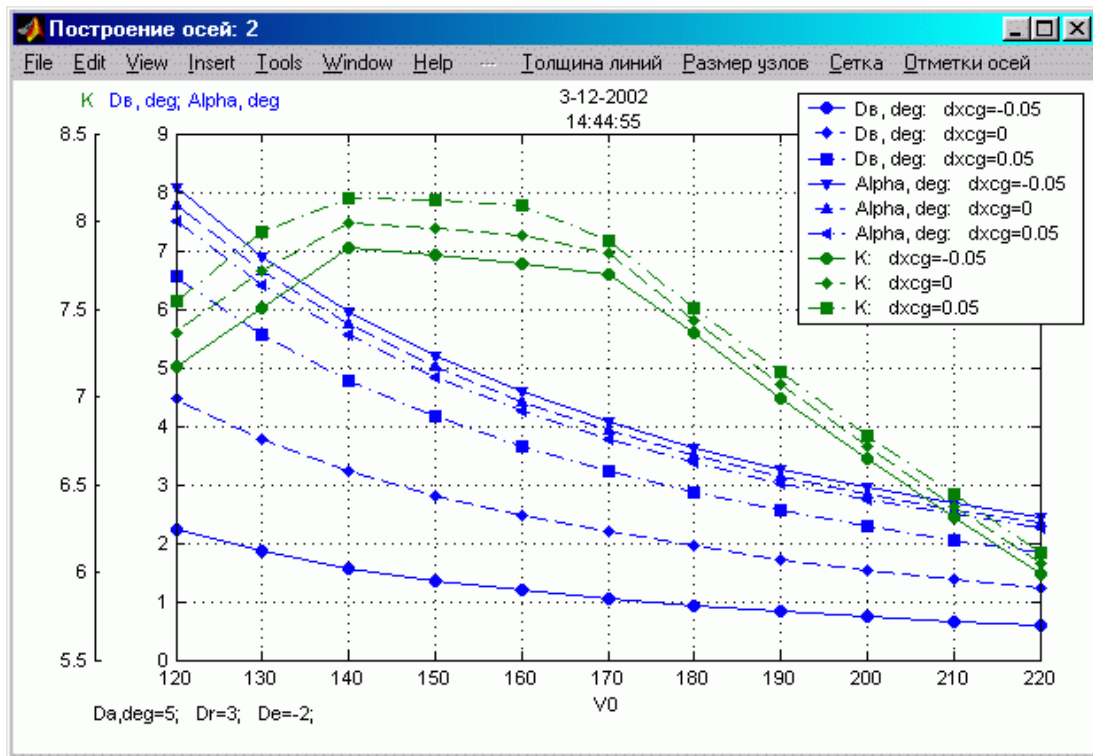


Рис. 14. График балансирующих зависимостей с 2 осями для 3 выходных переменных: отклонения стабилизатора, угла атаки и аэродинамического качества.

Библиотека программ поддержки баз данных самолета с САУ

Модели аэродинамики современных самолетов являются достаточно сложными и содержат большое количество таблиц аэродинамических характеристик по различным аргументам (углам атаки и скольжения, отклонениям органов управления, числу M и др.). При исследовании динамики современного самолета число используемых таблиц данных может достигать 200. В процессе разработки самолета или при его модификациях аэродинамические характеристики и зависимости параметров СУУ по режимам полета уточняются, изменяются, дополняются и т. п. Для выполнения таких операций разработана программа поддержки баз данных характеристик самолетов.

Программа поддержки баз данных характеристик самолетов, написанная как **MATLAB**-приложение, является важной составной частью комплекса **FlightSim**, однако, может использоваться и независимо от него для любых моделей среды **Simulink**. Программа имеет развитый интерфейс стандарта Windows (главное окно программы поддержки баз данных представлено на рис.15) и позволяет выполнять следующие функции:

- создание новых баз данных и дополнение существующих путем «захвата» числовых массивов из различных источников;
- возможность диалогового редактирования состава и содержания базы данных, включая как текстовое, так и графическое (рис.16) редактирование табличных функций;
- автоматизированное формирование элементов моделей системы **Simulink**, содержащих выбранные табличные функции базы данных;
- сервисные функции — открытие новых баз данных, сохранение изменений в базе с записью в старый или новый **mat**-файл, объединение баз данных, вывод графиков функций базы данных, проверка соответствия свойств функции свойствам аргументов, автоматизированное переименование аргументов, открытие моделей системы **Simulink** и др.

Подробно функции программы поддержки баз данных рассматриваются в специальном докладе. Здесь отметим, что по опыту практического использования данная программа является чрезвычайно полезной, удобной и эффективной. Программа также может быть полезна не только в динамике полета ЛА, но и при исследовании динамики других объектов, имеющих большой объем характеристик, задаваемых табличными данными.

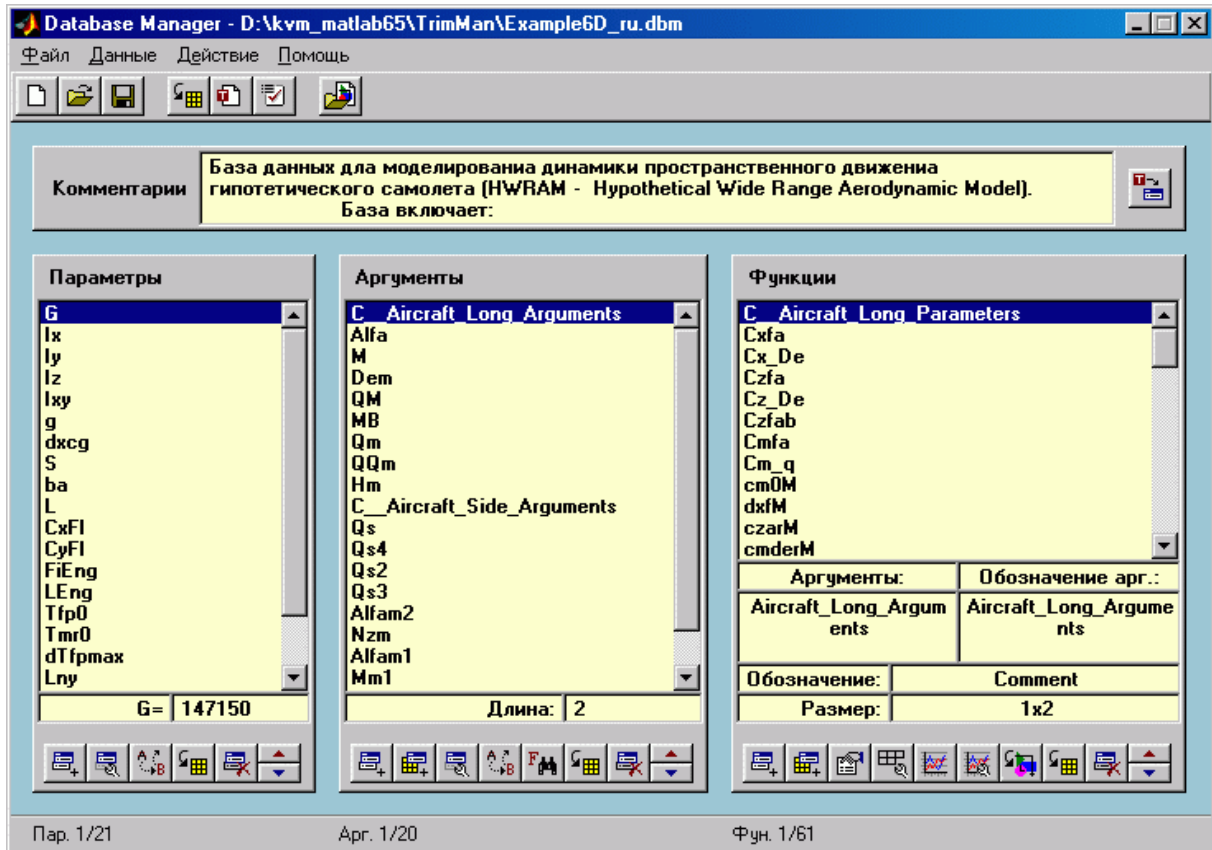


Рис. 15. Главное окно программы управления базами данных характеристик самолета.

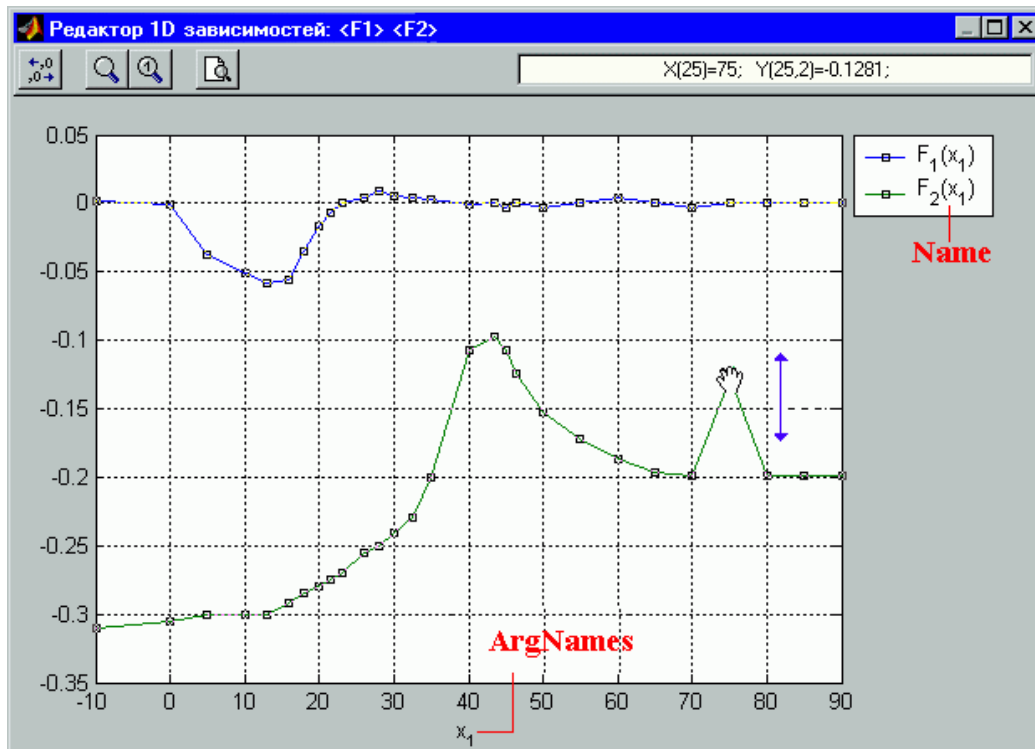


Рис. 16. Окно графического редактора одномерных функций.

Опыт работы с комплексом **FlightSim** показал, что время разработки и отладки математической модели самолета с системой управления значи-

тельно сокращается по сравнению с «ручным» программированием или с использованием САПР предыдущего поколения, использующих командный язык, а вероятность внесения ошибок существенно уменьшается. При этом важную роль играют примеры моделей динамики самолетов, включенные в комплекс. Целесообразно использовать эти примеры в качестве основы (шаблона), сохраняя структуру модели и основные связи между блоками (блок модели сил и моментов, блок уравнений движения, блок модели системы управления, блоки моделей приводов и т. п.), и заменяя или настраивая их конкретное содержание.

Модели, разработанные с использованием рассматриваемого комплекса, были использованы не только для расчетных исследований по анализу характеристик устойчивости, управляемости и синтезу систем управления самолета, но и для моделирования движения самолета в реальном времени на пилотажных стендах [2] и стендах для экспериментальной отработки системы управления и ее элементов.

Литература

1. *Кувшинов В. М., Анимца О. В.* Программный комплекс для моделирования и анализа динамики самолета с системой управления в среде программирования MATLAB/Simulink («FlightSim») // Тр. 1^{ой} Всероссийской науч. конф. «Проектирование научных и инженерных приложений в среде MATLAB» (Москва, 28–29 мая 2002).— М.: ИПУ РАН, 2002.
2. *Кувшинов В. М.* Программный комплекс моделирования динамики ЛА с системой управления в среде MATLAB/Simulink // Труды Международной конф. «Тренажерные технологии и обучение» (Жуковский, 24–25 мая 2001).— Жуковский, 2001.— С.91–99.