

Разработка и применение пакета расширения Spektr_SM пакета Simulink СКМ Matlab

В.В. Рыбин

В настоящее время для изучения спектральной формы математического описания систем управления в процессе обучения применяются различные системы компьютерной математики (СКМ) [8]. Среди них особо выделяется математическая система Matlab [10]. В ее состав входит пакет моделирования блочно заданных динамических систем – Simulink.

В данной статье рассмотрена технология разработки пакета расширения Spektr_SM [9] пакета Simulink. Пакет расширения Spektr_SM базируется на пакете расширения MLSY_SM [8] СКМ Matlab, а ввод постановки задачи осуществляется с использованием функциональной блок-схемы системы управления. Пакет расширения Spektr_SM позволяет проводить анализ в спектральной области линейных нестационарных непрерывных, дискретных и непрерывно-дискретных систем управления, находящихся под воздействием детерминированных и случайных сигналов. Дается характеристика и описание пакета. Приводятся примеры составления S-моделей системы управления и ее расчета с использованием пакета Spektr_SM и Simulink.

1. Проект и технологические особенности разработки пакета Spektr_SM.

Одной из концепций систем автоматизированного расчета (САР) является САР с диалоговым формирователем программ (ДФП). Эта концепция реализована в ППП расчете нестационарных непрерывно-дискретных систем управления спектральным методом [6], который входит в компьютерный курс по спектральной теории нестационарных систем управления [4]. Эта САР базируется на ППП спектрального метода [1-4], реализующего его элементарные операции. Математическая постановка задачи на базе структурной схемы вводится в систему автоматизированного расчета с использованием ДФП. ДФП позволяет интегрировано решать задачи анализа нестационарных систем управления со многими входами и выходами.

Можно считать, что эта концепция реализована в пакете Simulink СКМ Matlab [10]. Пакет Simulink это открытая система, т.е. система, позволяющая создавать библиотеки собственных блоков, обладающих свойствами стандартных библиотечных блоков пакета Simulink. ППП спектрального метода расчета линейных нестационарных непрерывно-дискретных систем управления MLSY_SM [8] является основой для создания пакета расширения Spektr_SM системы Simulink. Технологические особенности процесса разработки версии пакета Spektr_SM диктуются составом блочных компонентов, редактором блок-схем и специализированным механизмом создания собственных блоков и библиотек на базе маскирования подсистем пакета Simulink [10].

На рис.1 показан процесс разработки версии пакета Spektr_SM, который предполагает предварительную проработку проекта пакета Spektr_SM.



Рис.1.

Проект должен содержать:

- 1) проект структурной модели и оглавления пакета Spektr_SM;
- 2) проект структурной модели библиотеки R_Spektr;
- 3) проект системы блоков элементарных операций спектрального метода.

Рассмотрим реализацию этих проектов средствами пакета Simulink.

2. Разработка структурной модели и оглавления пакета расширения Spektr_SM

2.1. Версия структурной модели и оглавления пакета Spektr_SM

Механизм разработки структуры библиотеки блоков и оглавления пакета Spektr_SM отражает схема, расположенная на рис. 2.

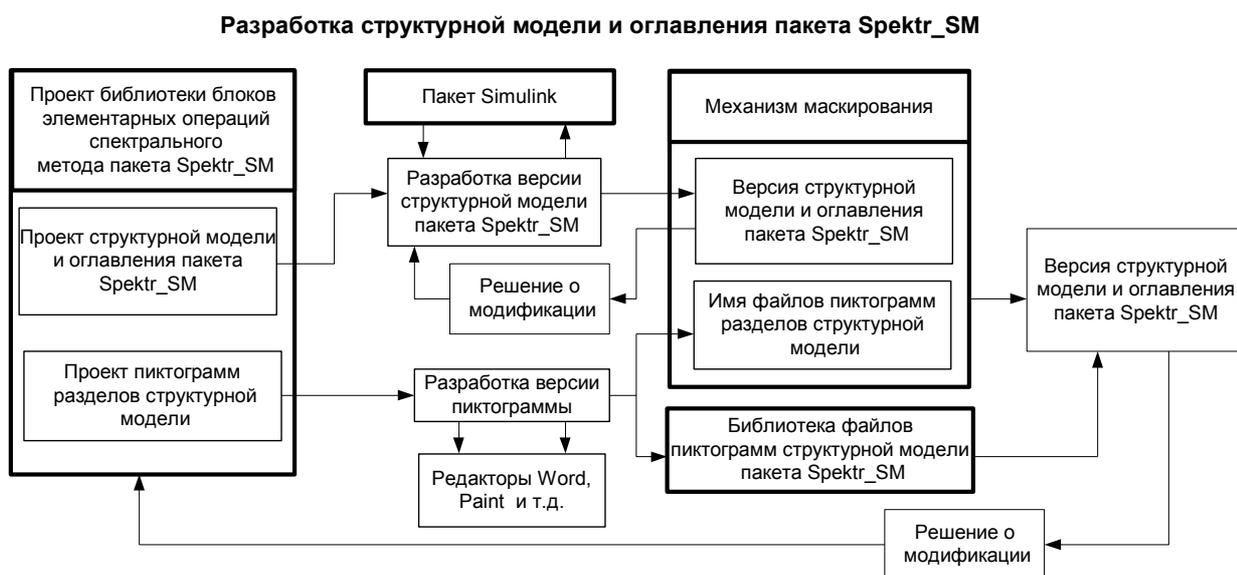


Рис. 2

Для создания новой библиотеки необходимо:

1) В любом окне Simulink (блок-диаграммы или одного из существующих разделов библиотеки) выбрать в разделе File команду New->Library.

2) В открывшемся пустом окне необходимо задать структуру (ранее разработанную) будущей библиотеки:

- в окне библиотеки необходимо нажать на кнопку Toggle model browser, для открытия окна Model Browser, служащего для просмотра структуры модели любой системы;

- в окне библиотеки необходимо перенести блоки Subsystem (служащие подразделами будущей библиотеки), с необходимым уровнем вложенности, чтобы разработанная ранее структура библиотеки совпадала со структурой из открытого окна Model Browser (изменяющаяся по мере заполнения библиотеки блоками Subsystem).

3) Для созданных разделов библиотеки (Subsystem) необходимо задать их названия (путём изменения подписи под блоком Subsystem на требуемую подпись) и разработать для них пиктограммы.

В процессе разработки проекта пакета Spekr_SM была разработана версия структурной модели и оглавления пакета Spekr_SM следующего вида:

1. НСХ сигналов и систем.
 - 1.1 Одномерных сигналов.
 - 1.2 Двумерных сигналов.
2. ДНПФ систем управления.
 - 2.1 ДНПФ непрерывных систем.
 - 2.2 ДНПФ дискретных систем.
 - 2.3 ДНПФ непрерывно-дискретных систем.
 - 2.4 ДНХС.
 - 2.5 ДНПФ начальных значений.
3. ОП сигналов и систем.
 - 3.1 ОП ОНСХ.
 - 3.2 ОП ДНСХ.
 - 3.3 ОП ДНПФ.
4. Графики.
 - 4.1 Регистраторы Simulink.
 - 4.2 Специальные регистраторы.
5. Блоки Simulink и DSP.
 - 5.1 Блоки функций.
 - 5.2 Матричные операторы.
 - 5.3 Математические блоки.

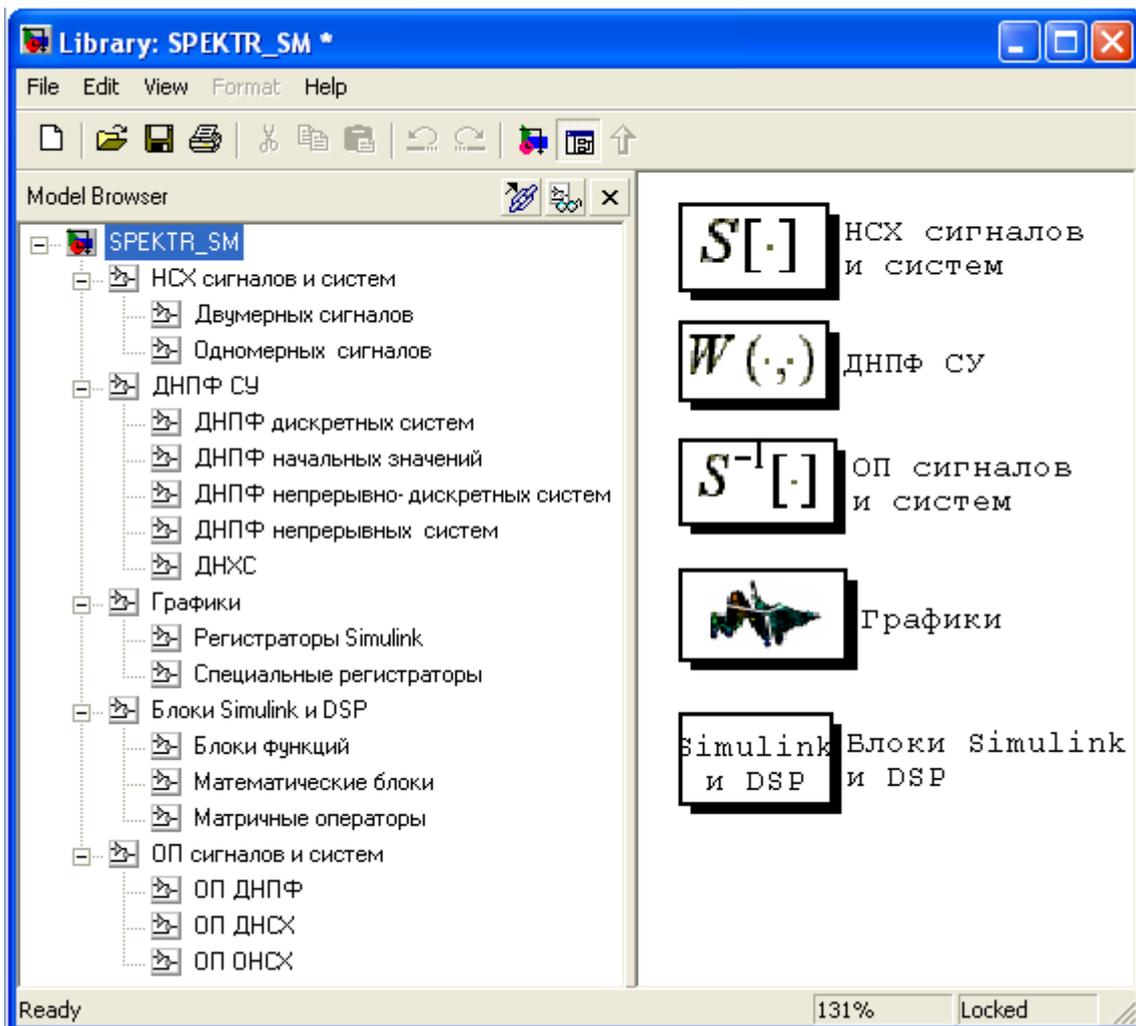


Рис. 3

Для создания данной библиотеки выберем в разделе File команду New->Library, далее, открыв браузер библиотеки (Model Browser), создадим данную структуру на основе блока Subsystem и описанных выше действий. На рис. 3 показано специальное окно с разделами основной библиотеки пакета Spekr_SM. В левой части этого окна показана полная структура библиотеки пакета, а в правой части – пиктограммы основных разделов библиотеки.

Созданную пустую форму библиотеки далее необходимо заполнить разработанными блоками пользователя. Сохранить созданную библиотеку на диске можно с помощью команды Save as... Как и S-модели, библиотеки сохраняются в файлах с расширением .mdl.

2.2. Разработка пиктограмм разделов структурной модели и блоков

У каждого стандартного раздела библиотеки Simulink есть своя индивидуальная пиктограмма. Для создания пиктограмм разделов библиотеки пакета Spekr_SM, показанных на рис. 3, надо открыть вкладку Icon окна редактора маски (Mask editor), которое показано на рис.4.

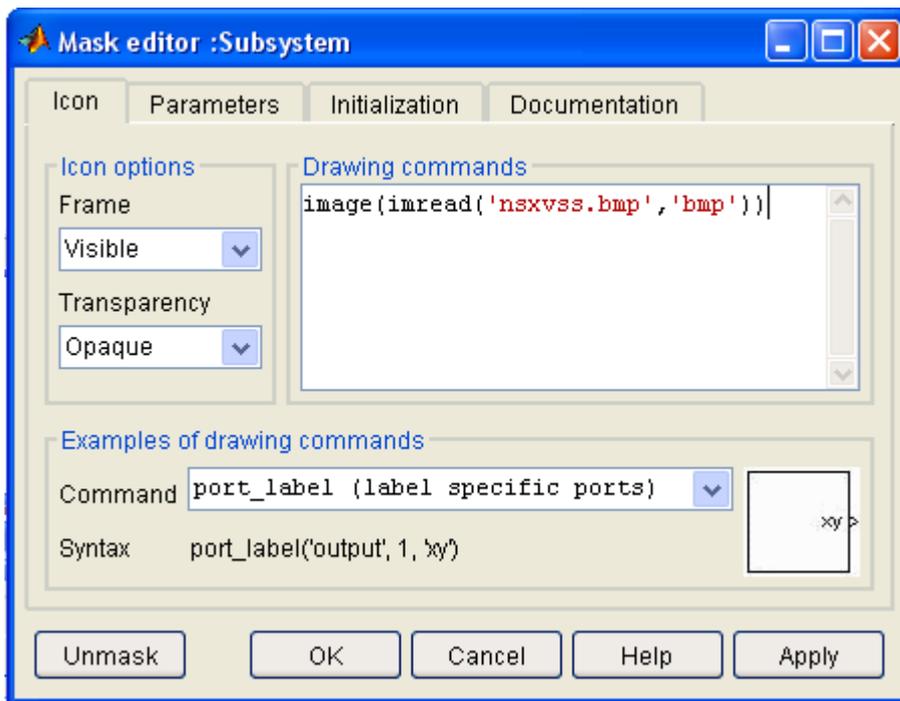


Рис. 4

Вкладка Icon содержит поле Drawing commands, в котором задаются текстовые и графические команды создания пиктограммы. Они задаются по правилам языка программирования Matlab.

На вкладке Icon расположены два элемента – раскрывающиеся списки:

- Icon frame – тип отображения рамки пиктограммы: Visible – рамка видна; Invisible – рамка не видна.

- Icon transparency – задание прозрачности пиктограммы: Opaque – пиктограмма непрозрачна; Transparent – пиктограмма прозрачна.

Для создания пиктограммы в виде текстовой надписи используется команда - **disp('Текст')**. Пиктограмму в Simulink можно создать, используя практически любой графический редактор или файл графического формата, который поддерживается системой Matlab. В число таких файлов входят файлы хорошо известных форматов: PCX, JPG, TIF, BMP. Для загрузки такого файла служит графическая команда - **image(imread(имя_файла, тип_файла))**. Но технология создания пиктограммы включает также и технологию создания графического файла. Поэтому рассмотрим технологию создания пиктограммы для любого раздела библиотеки или любого типового блока, реализующего элементарную операцию пакета Spektr_SM:

- 1) Сначала разрабатывается версия формулы (рисунок).
- 2) Затем данная формула набирается в текстовом редакторе Word при помощи редактора математических формул Microsoft Equation.
- 3) Далее формула копируется через буфер в графический редактор Paint и преобразуется в файл графического формата (например, формата BMP) и запоминается.

- 4) Для загрузки созданного файла применяется команда, которая набирается в поле Drawing commands: `image(imread('имя файла', 'тип файла'))`

Результат применения этой технологии показан на рис. 3 и рис. 4.

3. Разработка структурной модели библиотеки *R_Spektr*

Разработка блоков элементарных операций спектрального метода сопровождается созданием версий этих блоков (MDL-файлы), моделей их отладки (MDL-файлы). Эти блоки используют программные модули элементарных операций спектрального метода пакета *MLSY_SM* [8], через связанные с ними файл-функции (М-файлы) и сопутствующими блокам BMP-файлами пиктограмм и HTML-файлами справочной системы. Размещение всех этих файлов требует создание структурированной библиотеки *R_Spektr* пакета *Spektr_SM* (см. рис. 1), файловая организация которой содержит следующие поддиректории:

- DNPF – файлы, относящиеся к определению ДНПФ звеньев.
 - DNPF_NDS –ДНПФ Н-Д звеньев.
 - DNPF_NS –ДНПФ Н-Н звеньев.
 - DNPF_NZ –ДНПФ звеньев с ненулевыми начальными значениями.
 - DNPF_DS –ДНПФ Д-Д звеньев.
 - DNXS –ДНХС звеньев.
- NSX – файлы, относящиеся к определению НСХ входных сигналов.
 - NSX_DS –НСХ двумерных сигналов.
 - NSX_OS –НСХ одномерных сигналов.
- OP – файлы обратного преобразования.
 - OB_DNPF – обращение ДНПФ.
 - OB_DNSX – обращение ДНСХ.
 - OB_ONSX – обращение ОНСХ.
- MLSY_SM – файлы библиотеки спектральных алгоритмов (файловую организацию которой смотри в [8]).
- PRIMER – файлы, относящиеся к разработанным файлам примеров.
- VSS – файлы, относящиеся к визуализации сигналов.
 - SP_R – файлы специальных регистраторов
 - SR_SIM – файлы стандартных регистраторов Simulink

Разработанные версии блоков и сопутствующие им файлы размещаются в соответствующих разделах, а пути доступа к ним заранее устанавливаются по типовой схеме [10].

4. Разработка собственных блоков пакета расширения *Spektr_SM*

4.1. Технология формирования блока и сопутствующих программ

Механизм разработки блоков элементарных операций спектрального метода в системе визуального моделирования Simulink, на базе пакета *MLSY_SM* демонстрирует схема, расположенная на рис. 5.

Как видно из схемы, в конечном счёте, версия блока реализуется механизмом маскирования разработанной ранее подсистемы, при помощи разработанной версии сопутствующего m-файла элементарной операции спектрального метода, версии пиктограммы, версии описания блока и справочной документации по блоку. Вследствие этого необходимо рассмотреть механизм создания подсистемы, m-файла и справочной документации.

Разработка версии подсистемы блока. Simulink обеспечивает создание моделей, внутри которых располагаются подсистемы (субмодели). Внутри подсистем первого уровня, в свою очередь, могут располагаться подсистемы второго уровня и т.д. Это напоминает ситуацию, когда сложная система набирается из отдельных систем – модулей, каждый из которых, в свою очередь, является системой.

Такой принцип конструирования сложных моделей даёт ряд важных достоинств:

- имеется возможность разбивки решаемой задачи на ряд более мелких задач, решаемых подсистемами;
- каждая подсистема может отлаживаться отдельно и использоваться в полной системе уже после отладки;
- существенно упрощается вид основной модели за счёт исключения из неё второстепенных блоков;
- облегчается модификация полной модели за счёт модификации её более простых подсистем.

Разработка блоков элементарных операций спектрального метода для пакета Spektr_SM

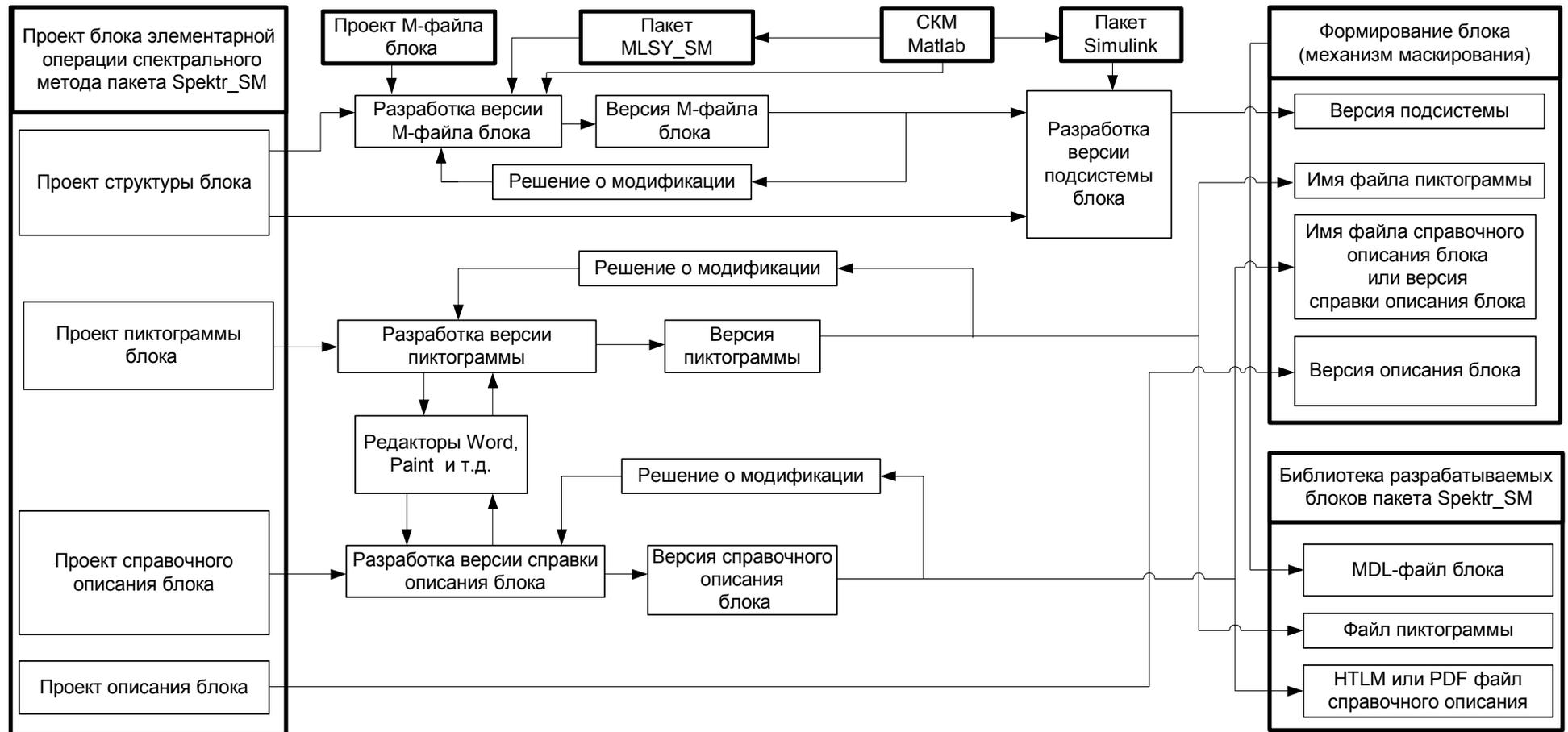


Рис. 5

Для создания подсистемы в Simulink предназначен блок SubSystem из раздела Ports&Subsystems. Его следует разместить в открытом окне основной модели. Затем, после открытия этого блока (двойным нажатием левой кнопкой мыши), открывается поле пустой подсистемы. Подсистема набирается из стандартных блоков библиотеки Simulink. Основное отличие подсистемы заключается в том, что в подсистеме должны присутствовать следующие блоки – порт ввода In1 и порт вывода Out1. Порты изображаются овалом с номером внутри и подписями.

Благодаря этим портам подсистема включается в состав основной модели. Если выделенный в подсистему блок содержит несколько портов ввода и вывода (они обозначаются как In1, In2, In3, ... и Out1, Out2, Out3, ...), то у блока подсистемы появится соответствующее количество входов и выходов. Порты ввода и вывода являются вполне полноценными блоками и имеют свои окна параметров.

Окно порта ввода **Inport** позволяет задавать следующие параметры: **Port number** – номер порта; **Port dimension** – размерность порта (-1 – при динамической установке размерности); **Sample time** – эталонное время; **Data type** – тип данных (выбор из раскрывающегося списка); **Signal type** – тип сигнала (выбор из раскрывающегося списка);

Окно порта вывода **Outport** задаёт параметры: **Port number** – номер выходного порта; **Output when disable** – выход при пассивности системы; **Initial output** – инициализация выхода.

Пример 1. Надо создать подсистему, вычисляющую усеченную НСХ [8] входного непрерывного сигнала, заданного в виде аналитической функции относительно непрерывных базисных функций, идентификаторы которых приведены в приложении 1.1. работы [8].

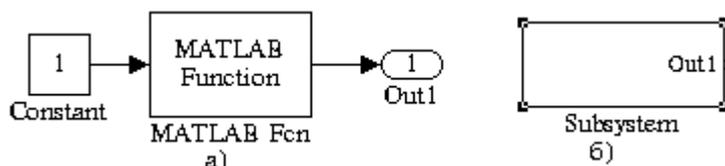


Рис. 6

Решение задачи. Эта подсистема показана на рис. 6. На рис. 6 а) показана созданная подсистема (ее внутреннее представление), а на рис. 6 б) ее внешнее представление. На вход этой подсистемы сигнал не поступает. Это значит, что блок **In** в подсистеме должен отсутствовать, в качестве “заглушки” используется блок **Constant**. Сама НСХ вычисляется при помощи блока **MATLAB Function**, служащего для задания функций одной переменной или ряда переменных $u(i)$ по правилам, принятым для языка программирования базовой системы Matlab. Это означает, что в теле функции могут встречаться как встроенные функции Matlab, так и любые процедуры и функции, реализованные в виде m-файлов.

Окно параметров этого блока, расположенное на рис. 7, содержит описание блока и раздел параметров Parameters. В разделе Parameters MATLAB functions: задаётся выражение для функции,

в разделе Parameters Output dimensions длина вектора выхода (если она должна совпадать с длиной входного сигнала, то вводится значение -1), в раскрывающемся списке Parameters Output signal type можно выбрать тип выходного сигнала в виде вещественного числа (real), комплексного числа (complex) или задать автоматический выбор (auto), в разделе Parameters Collapse 2-D results to 1-D выставляется метка \checkmark , если надо выходную матрицу преобразовать в одномерный выход.

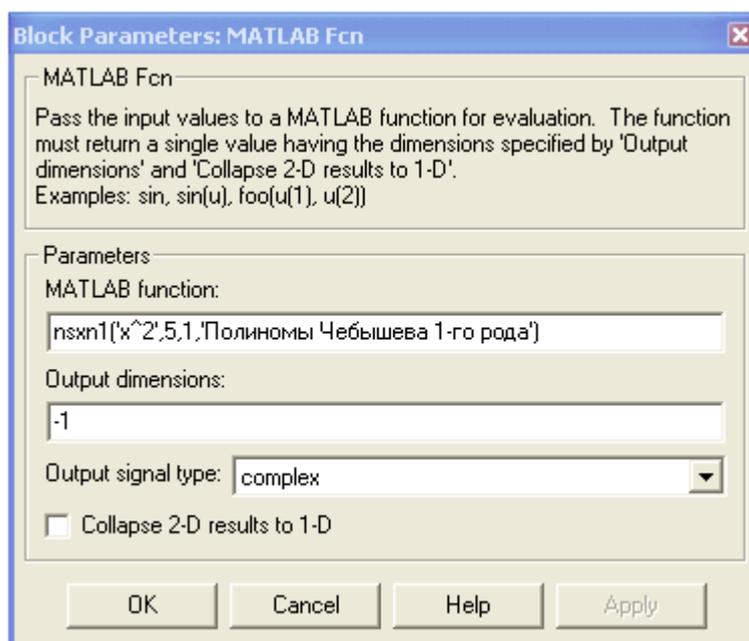


Рис. 7

В данном примере в разделе Parameters MATLAB functions вводится имя **nsxn1** m-файла, который создан и отлажен для нахождения вектора-столбца НСХ. Он содержит функцию **nsxnnn(x1,N1,t,s1)**, которая зависит от параметров: **x1** - аналитической функции задающей входной сигнал; **N1** - размера усечения матрицы-столбца НСХ; **t** - длины интервала работы системы управления; **s1** - базиса, по которому вычисляется НСХ. Эта функция имеет вид:

```
function f1 = nsxnnn(x1,N1,t,s1)
f=inline(x1); switch s1
case 'Полиномы Лежандра' f1=snxpp1(f,N1,t)
case 'Полиномы Чебышева 1-го рода' f1=snxtt1(f,N1,t)
case 'Полиномы Чебышева 2-го рода' f1=snxuu1(f,N1,t)
case 'Косинусоиды' f1=snxcc1(f,N1,t)
case 'Комплексные экспоненциальные функции' f1=snxff1(f,N1,t)
case 'Функции Уолша' f1=SNXYY1(f,N1,t)
case 'Функции Хаара' f1=SNXXX1(f,N1,t)
end.
```

При написании этой функции использован оператор переключения, который использует ключевые слова **switch** (переключить), **case** (случай), **otherwise** (иначе). В программе осуществляется переключение по параметру **s1**, который является строковой переменной, получаемой m-функцией в качестве входного параметра. В случае совпадения с одним из значений выполняется операция вызова соответствующей функции (в нашем случае выполняется программа

$\text{snxtt1}(\mathbf{f}, \mathbf{N1}, \mathbf{t})$ - вычисление вектора-столбца НСХ в базисе полиномов Чебышева первого рода) и присваивания её значения переменной $\mathbf{f1}$.

Обратим внимание и на использованный здесь оператор **inline(x1)**. В рамках системы Matlab определён класс объектов **Inline**. Он предназначен для описания функций в естественной форме, т.е. соответствующей их математическому описанию. Объекты этого класса – это функции, заданные в символьном виде, что позволяет обращаться к ним как к привычным для нас математическим объектам.

Следовательно, в приведенном выше примере в качестве параметра \mathbf{f} в соответствующую функцию передаётся **inline**-функция.

Формирование блока. Нужно отметить, что подсистемы не обладают свойствами стандартных библиотечных блоков пакета Simulink. Главное отличие подсистем от блоков в том, что подсистемы не имеют ни своей уникальной пиктограммы, ни окна параметров и не связаны с разделом библиотеки. Для построения пользовательских блоков Simulink предлагает специальный механизм маскирования подсистем. Маскированные подсистемы – это такие подсистемы, которые имеют специальный признак (маску), скрывающий их внутреннюю структуру. В результате такая подсистема в деталях не видна и воспринимается как библиотечный модуль.

Маскированные подсистемы обладают рядом важных достоинств: 1) они имеют свои пиктограммы с уникальными изображениями; 2) их можно использовать как библиотечные блоки; 3) у них есть своё окно установки параметров; 4) есть возможность в любой момент сбросить маску и наблюдать структуру блока; 5) применение масок расширяет возможности построения сложных моделей; 6) имеется возможность отредактировать подсистему, превращённую в маскированную подсистему; 7) повышается наглядность моделей; 8) повышается защищённость подсистемы от модификации.

Для создания маскированных подсистем надо выполнить следующие операции: 1) заменить название подсистемы **Subsystem** на название создаваемого блока; 2) с помощью специального редактора масок создать окно установок параметров, документацию под маскированную подсистему и её справочную систему.

В заключение заметим, что рабочая область маскированной подсистемы **Mask Workspace** – это локальная рабочая область исполняемой подсистемы. Она создаётся Simulink в следующих случаях: 1) если маска подсистемы содержит команды инициализации; 2) если при описании маски заданы параметры настройки подсистемы. Переменные, содержащиеся в рабочей области маскированной подсистемы, доступны всем блокам, входящим в её состав. То есть при создании маски можно определять параметры, которые должны иметь одинаковое значение для всех блоков, входящих в состав маскированной подсистемы.

Пример 2. Надо создать блок (маскированную подсистему) из подсистемы, рассмотренной в примере 1.

Решение задачи. Меняем название подсистемы (рис. 6) **Subsystem** на **Непрерывный вход**. Для создания маски достаточно выделить подсистему, после чего выбрать команду Mask Subsystem... меню Edit. Данная команда запускает редактор маски – появляется его окно с открытой незаполненной вкладкой Icon (вариант заполнения которой показан на рис. 4). В нашем случае в окне Drawing commands вводится команда `image(imread('nsxons.bmp','bmp'))`, которая вызывает пиктограмму, разработанную для этого блока.

Следующим шагом в создании маски является описание параметров настройки подсистемы. Таких параметров в данном примере четыре: аналитически заданный входной сигнал; порядок усечения НСХ; длина интервала работы системы; базисные функции. Для описания этих параметров открываем вкладку Parameters и заполняем окна Dialog parameters и Options for selected parameter (см. рис. 8).

Для задания первого из них необходимо: 1) В поле Prompt окна Dialog parameters ввести строку “Непрерывный входной сигнал ($\sin(t), t^2$):” – пояснение к параметру. 2) В поле Variable ввести имя переменной “**x1**”, в которой будет храниться значение данного параметра и с помощью которой происходит его передача в m-функцию (`nsxn1.m`). 3) В поле Type оставить значение Edit. 4) В поле Evaluate метку удаляем, так как параметр имеет символьное значение. 5) В поле Tunable метку оставляем. Аналогично заполняются и другие строки.

Для задания второго параметра блока (порядок усечения НСХ) необходимо повторить действия 1. – 5. с той разницей, что в поле Prompt нужно ввести “N - порядок усечения НСХ:”, в поле Variable нужно ввести имя соответствующей переменной “N1”, а в поле Evaluate оставить метку , так как в M-функцию передаётся параметр, имеющий численное значение. Задание параметра “t - длины интервала работы СУ” аналогично заданию второго параметра.

Задание четвёртого параметра отличается тем, что тип параметра является выпадающим списком, элементы которого строки символов задающие базисные функции для передачи в m-файл. В поле Prompt введём “НБФ:”, в поле Type выберем из выпадающего списка: Popup (выпадающий список). И в ставшее доступным поле Popup (one per line) необходимо ввести элементы выпадающего списка (базисные функции) через команду Enter.

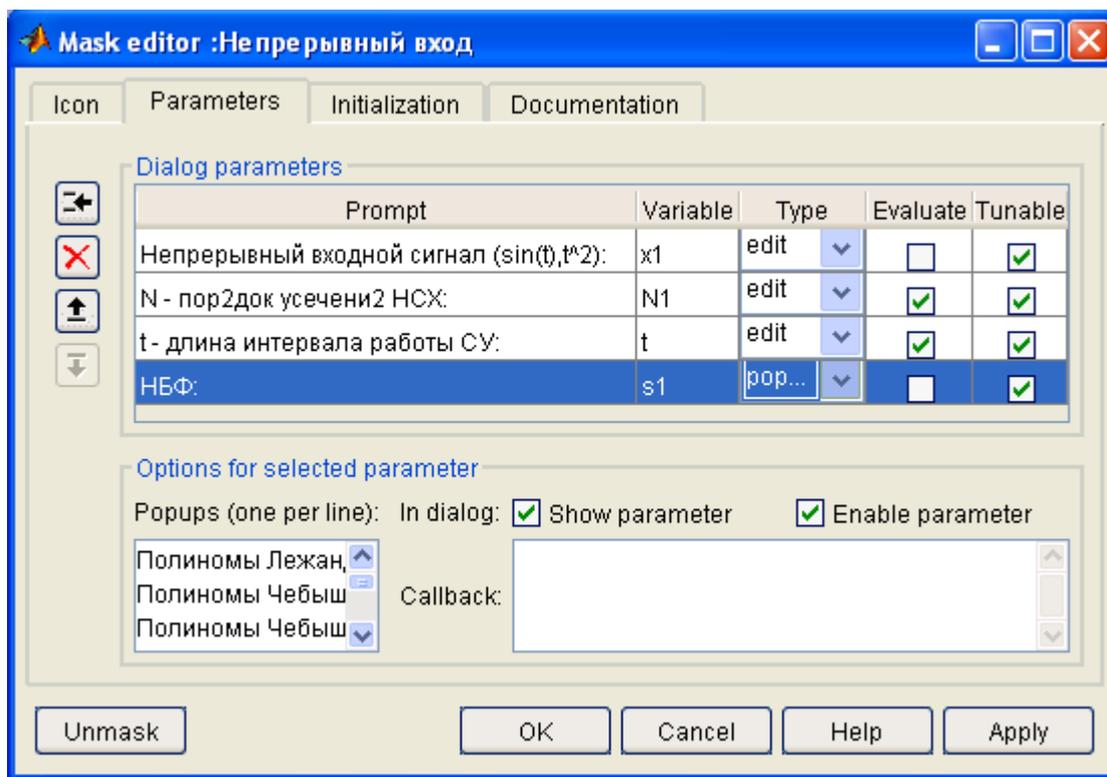


Рис. 8

После применения к маскированной подсистеме этих опций, необходимо подготовить описание этого блока. Вкладка Documentation содержит три окна Mask type (имя подсистемы), Mask description (описание блока), Mask help (справочное описание блока).

Окно редактора маски с открытой заполненной вкладкой Documentation для данного примера показано на рис. 9.

Поле Mask Help (Справка по блоку) здесь пусто, но оно может служить для ввода текстовой части справки, которая будет размещена в справочной системе Simulink. Эта информация впоследствии будет доступна при нажатии кнопки Help в окне настройки параметров замаскированного блока; эта информация сохраняется в формате HTML, включается Simulink в состав его справочной системы и может быть просмотрена с помощью интернет-браузера (Internet Explorer, Netscape Navigator и т.п.). При необходимости в поле Mask Help вместо статичного текста могут содержаться другие средства помощи пользователю: 1) адрес WEB-узла, на котором можно найти дополнительные сведения, адрес электронной почты или другие URL-спецификации; 2) команда **web**, запускающая web-браузер, установленный на компьютере (например, команда **web(['file:/// which('foo.html')])**) приводит к запуску web-браузера, поиску и открытию файла **foo.html**. 3) команда **eval**, которая обеспечивает интерпретацию строк как исполняемых команд Matlab (например, команда **eval('!Word Book.doc')**, приводит к запуску редактора MS Word и открытию файла Book.doc.

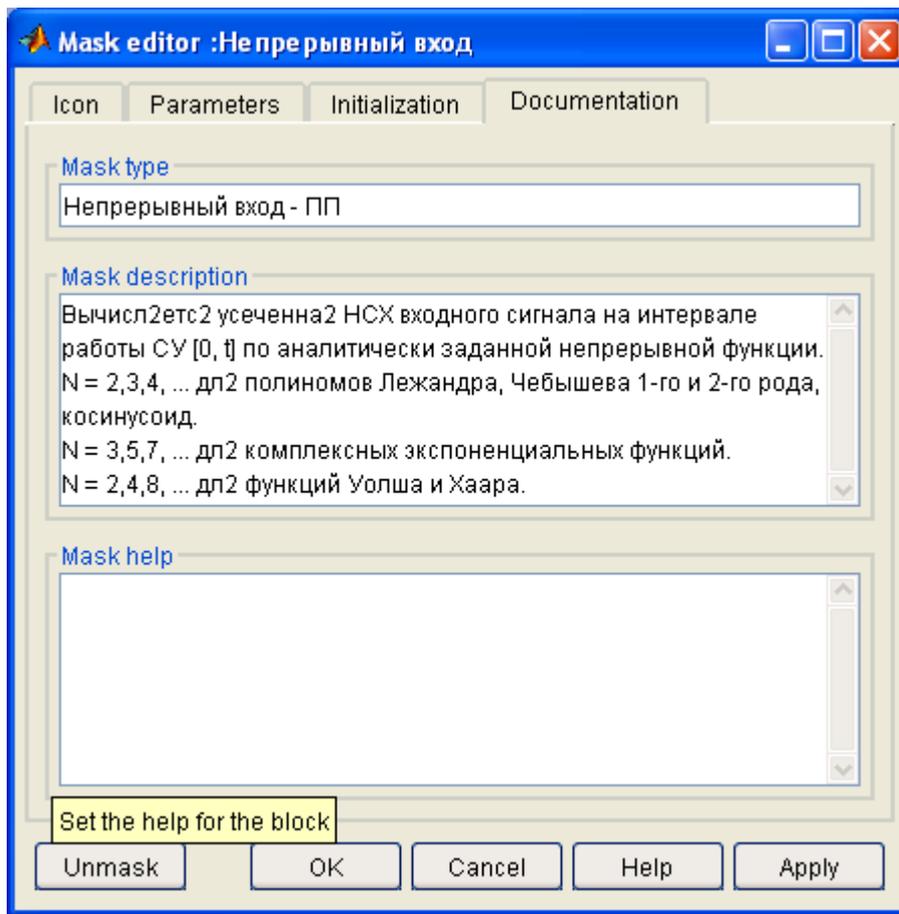


Рис. 9

В результате выполнения всех этих действий получаем блок, внешний вид которого показан на рис. 10, а окно его параметров на рис. 11.



Рис.10

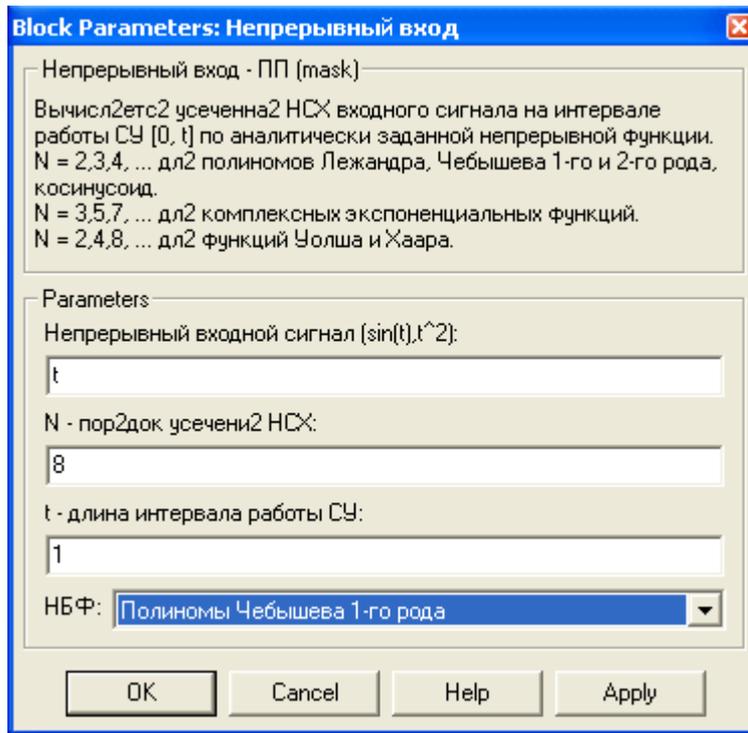


Рис.11

4.2. Состав типовых блоков библиотеки *Spektr_SM*

Библиотека *Spektr_SM* разделена на 5 разделов, окно браузера которой показано на рис. 3. Эти 5 разделов, в общем виде, содержат 15 подразделов. Рассмотрим состав блоков входящих в эти разделы.

Блоки, входящие в раздел **НСХ сигналов и систем**, показаны на рис. 12 и рис. 13 и предназначены для формирования НСХ непрерывных и дискретных детерминированных сигналов, а также для формирования НСП случайных непрерывных и дискретных сигналов.

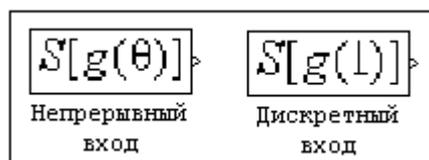


Рис. 12

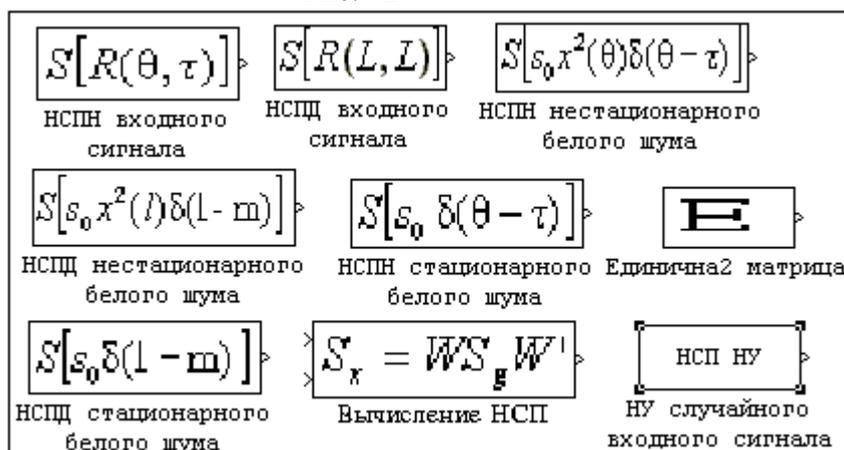


Рис. 13

Блоки, входящие в раздел **ДНПФ СУ**, предназначены для задания: ДНПФ непрерывных систем; ДНПФ дискретных систем; ДНПФ непрерывно-дискретных систем; ДНПФ начальных значений; ДНХС. Они показаны на рис. 14– 18.

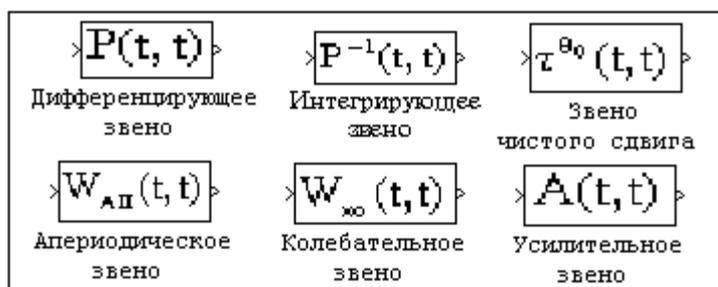


Рис. 14

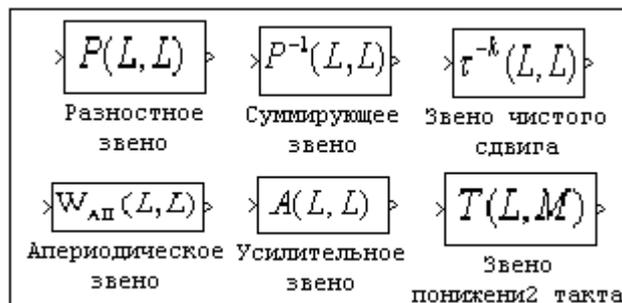


Рис. 15

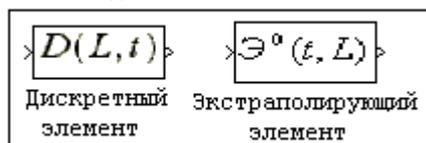


Рис.16

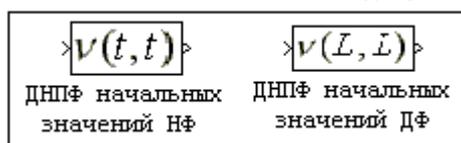


Рис.17

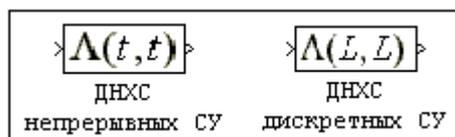


Рис. 18

Блоки, входящие в раздел **ОП сигналов и систем**, предназначены: для обращения НСХ одномерных непрерывных и дискретных детерминированных сигналов, т.е. для вычисления выходных сигналов (см. рис. 19); для обращения НСП, т.е. вычисления корреляционных функций случайных непрерывных и дискретных выходных сигналов, их дисперсий и средних квадратических значений (см. рис. 20); для обращения ДНПФ непрерывных, дискретных и непрерывно-дискретных систем, т.е. вычисления ИПФ системы управления (см. рис. 21). Многие из этих блоков сразу осуществляют визуализацию сигналов.



Рис. 19

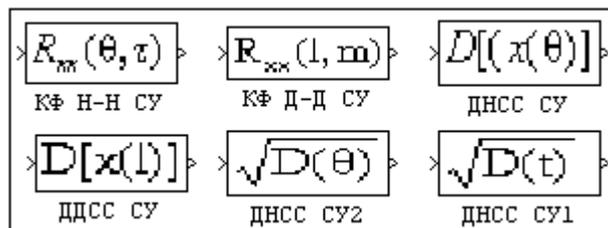


Рис. 20

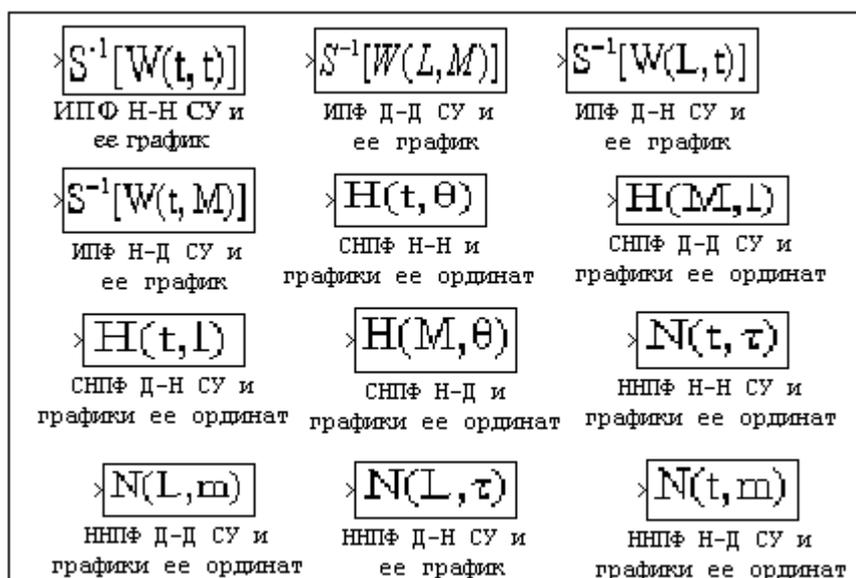


Рис. 21

Блоки, входящие в раздел **Графики**, предназначены для визуализации одномерных и двумерных функций.

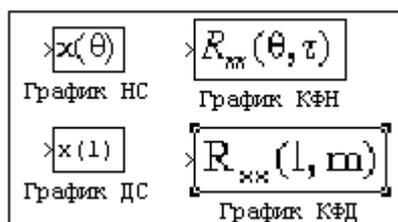


Рис. 22

4.3. Примеры анализа структуры и содержания типовых блоков пакета *Spektr_SM*

В предыдущих разделах была рассмотрена технология формирования блоков и состав блоков, рассмотренной версии пакета *Spektr_SM*. Среди этих блоков можно выделить группы блоков, структурные схемы которых совпадают. Однако m -функции передаваемые в эти блоки отличаются друг от друга, так как они выполняют разные элементарные операции спектрального метода. Среди m -функций можно выделить такие m -функции, которые также обладают общей структурой. Такое внутреннее различие блоков порождает и внешнее их отличие друг от друга не только по пиктограмме и названию, но и по виду окна передаваемых параметров и описания блока. Рассмотрим примеры блоков с общей структурной схемой.

Пример 3. Среди блоков пакета *Spektr_SM* найти те блоки, подсистемы которых имеют структурную схему, показанную на рис. 23.

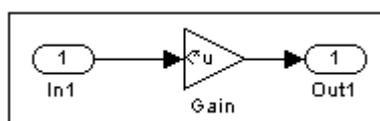


Рис. 23

Откроем окно интегрированного браузера библиотеки пакета Spektr_SM. Для просмотра того или иного раздела библиотеки выделим его мышью. Пусть выделен раздел **ДНПФ непрерывных систем**. Выделим мышью блок **Дифференцирующее звено** и выполним команду **Look under mask** меню **Edit**. Откроется окно **Library: СПЕКТР_SM/.../ДНПФ непрерывных систем/Дифференцирующее звено**, в котором появится подсистема, изображенная на рис. 23. Прделав эту операцию с каждым из блоков пакета Spektr_SM, найдем все блоки такой структуры. К ним относятся все блоки раздела библиотеки **ДНПФ СУ**, показанные на рис. 14 - 18. Кроме этих блоков такой же структурой обладают блоки раздела **ОП ОНСХ**, а именно блоки **Непрерывный выход** и **Дискретный выход** (см. рис. 19).

Пример 4. Сравнить блоки Дифференцирующее звено и Интегрирующее звено.

Как видно из примера 3 структурные схемы этих блоков совпадают. Однако их окна параметров имеют незначительные отличия. Названия блоков и их описания отличаются только одним словом, а именно надо заменить слово **дифференцирующее** на **интегрирующее** или наоборот. Сравним теперь m-файлы **pdfine** и **pfinne**, вызываемые блоком **Gain** подсистем этих блоков. m-функция для блока **Gain Дифференцирующего звена** имеет вид:

```
function fp1 = fp1(L,t,s1)  switch s1
    case 'Полиномы Лежандра'          fp1=sp1pp1(t,L)
    case 'Полиномы Чебышева 1-го рода'  fp1=sp1tt1(t,L)
    case 'Полиномы Чебышева 2-го рода'  fp1=sp1uu1(t,L)
    case 'Косинусоиды'                fp1=sp1cc1(t,L)
    case 'Комплексные экспоненциальные функции'  fp1=sp1ff1(t,L)
    case 'Функции Уолша'              fp1=sp1yy1(t,L)
    case 'Функции Хаара'              fp1=sp1xx1(t,L)
end
```

Для блока **Gain Интегрирующего звена** вид:

```
function fi1 = fi1(t,L,s1) switch s1
    case 'Полиномы Лежандра'          fi1=si1pp1(t,L)
    case 'Полиномы Чебышева 1-го рода'  fi1=si1tt1(t,L)
    case 'Полиномы Чебышева 2-го рода'  fi1=si1uu1(t,L)
    case 'Косинусоиды'                fi1=si1cc1(t,L)
    case 'Комплексные экспоненциальные функции'  fi1=si1ff1(t,L)
    case 'Функции Уолша'              fi1=si1yy1(t,L)
    case 'Функции Хаара'              fi1=si1xx1(t,L)
end
```

При написании этих функций опять использован оператор переключения (см. пример 1). В программе осуществляется переключение по параметру **s1**, который является строковой переменной, получаемой m-функцией в качестве входного параметра. В случае совпадения с одним из значений выполняется операция вызова соответствующей функции. По своей структуре программы полностью совпадают, а отличаются только идентификаторами элементарных операций в описании процедур [8] вызываемых из раздела **MLSY_SM** библиотеки **R_Spektr** пакета **Spektr_SM**.

Аналогично можно изучить любой блок пакета Spektr_SM и при необходимости провести его модификацию.

5. Особенности применения пакета Spektr_SM в системе Simulink для анализа систем управления при детерминированных и случайных воздействиях.

В предыдущем разделе были рассмотрены основные виды блоков пакета Spektr_SM, используемые для анализа систем управления в спектральной области и находящихся под воздействием как детерминированных, так и случайных сигналов.

Сложные модели многомерных систем управления, для анализа их спектральным методом при детерминированных воздействиях, формируются как многомерные модели из блоков библиотеки пакета Spektr_SM, которые входят в разделы ДНПФ СУ и Блоки Simulink и DSP. Входные сигналы такой модели формируются из блоков, вычисляющих НСХ детерминированных одномерных сигналов, которые находятся в разделе НСХ сигналов и систем. Множество выходных сигналов, от каждого входа к каждому выходу, вычисляется при помощи блоков, входящих в разделы ОП сигналов и систем, Блоки Simulink и DSP и Графики. Готовая и отлаженная модель запускается на выполнение. Но запуску системы Simulink на выполнение должен предшествовать этап ее настройки. Это связано с тем, что моделирование системы управления в спектральной области проводится за один такт ее работы.

Для настройки параметров модели: в меню Simulation окна модели необходимо выбрать пункт Simulation parameters – откроется окно *Simulation parameters: Имя модели* по умолчанию с открытой вкладкой Solver. В этом окне можно установить параметры, как показано на рис. 24.

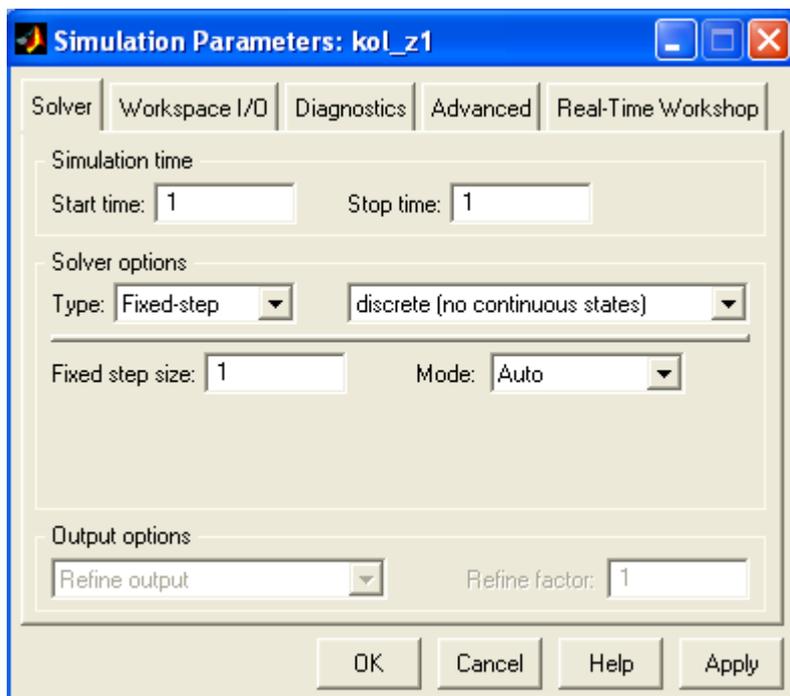


Рис. 24

Данная настройка производится во всех случаях анализа систем управления спектральным методом, а при параметрическом синтезе необходимо в Start time и Stop time установить соответствующие значения, а фиксированному шагу присвоить нужное значение (например, для параметра меняющегося от 1 до 10, Start time = 1, а Stop time = 10, при Fixed step size = 1).

После проведённой настройки необходимо нажать кнопку Apply или ОК, затем можно запускать процесс моделирования.

Если требуется вычислить математические ожидания выходных сигналов по математическим ожиданиям входных сигналов, то сложные модели многомерных систем управления формируются аналогично тому, как они формировались для анализа их спектральным методом при детерминированных воздействиях. Вычисления корреляционных функций выходных сигналов по корреляционным функциям входных сигналов производится несколько иначе, чем при детерминированных воздействиях. Это связано с тем, что связи между НСП входных и выходных сигналов определяются выражением [2]:

$$S_{x_\mu x_\nu} = \sum_{i=1}^n \sum_{k=1}^n W_{\mu i} S_{g_i g_k} W_{\nu k}^T, \quad \mu, \nu = 1, 2, \dots, m, \quad (1)$$

Для вычисления НСП выходных сигналов, в случае отсутствия взаимной корреляции между входными сигналами, выражение (1) преобразуется к виду

$$S_{x_i} = \sum_{k=1}^n W_{ik} S_{g_k} W_{ik}^T, \quad i = 1, 2, \dots, m. \quad (2)$$

Поэтому моделируем в Simulink выражение (2), которое требует моделирования подсистем, вычисляющих ДНПФ $W_{i,k}$ от k -го входа к i -му выходу и выполнения типовой операции

$$S_x = W \cdot S_g \cdot W^T. \quad (3)$$

Блок **Вычисление НСП** раздела **НСХ сигналов и систем** библиотеки пакета Spektr_SM реализует операцию (3). Структурная схема подсистемы этого блока показана на рис. 25.

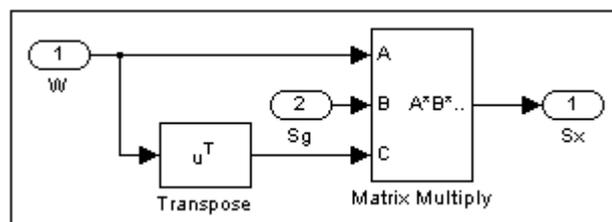


Рис. 25

На вход W подаётся ДНПФ одномерной системы управления, на вход S_g НСП входного сигнала. Стандартные блоки пакета DSP Blockset **Transpose** и **Matrix Multiply** реализуют соответственно операции транспонирования матрицы ДНПФ и перемножения матриц по формуле (3). Таким образом, реализуется связь вход-выход по корреляционной функции при помощи средств Simulink.

6. Пример анализа бокового канала системы управления ЛА посадки на Луну

В работе [2] рассматривается система управления мягкой посадки на Луну аппарата типа “Сервейер”. Проводится линеаризация уравнений движения системы управления относительно опорного режима, за который принимается вертикальный спуск по номинальной траектории посадки на плоскую горизонтальную поверхность при заданном значении начальной вертикальной скорости. При этом полагалось, что все помехи, возмущения и погрешности направления тяг двигателей отсутствуют. Приводится и изучается линеаризованная структурная схема, на которой выделены два боковых канала и продольный канал. Боковые каналы в линеаризованном варианте независимы друг от друга и от продольного канала наведения. Сама линеаризованная система управления является существенно нестационарной и стохастической. Ее полный динамический расчет проводится спектральным методом.

Здесь рассмотрим пример анализа бокового канала управления по рысканию спектральным методом [1-3], используя для этих целей пакет расширения системы MATLAB – Simulink 4 [10] (или Simulink 5) и пакет его расширения – Spektr_SM.

6.1. Математическая модель канала управления по рысканию и задачи ее анализа

Упрощенная структурная расчетная схема системы управления в классе линеаризованных нестационарных систем представлена на рис. 26.

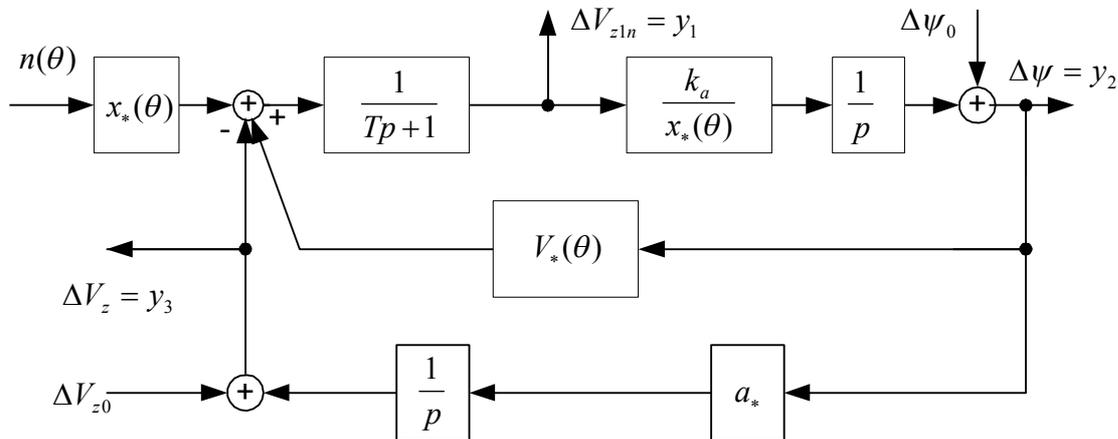


Рис. 26

Этому структурному представлению системы управления соответствует система дифференциальных уравнений, векторно-матричное уравнение которой имеет вид:

$$Y' = AY + Bn, \quad (4)$$

где



Система управления реализует метод гравитационного разворота. Блок управления представлен на структурной схеме нестационарным коэффициентом передачи $k_a / x_*(\theta)$, где

$x_*(\theta) = 197.5 - 30.5\theta + 1.375\theta^2$ - номинальное значение высоты спускаемого аппарата, определяемое

выбранным опорным режимом при линеаризации. При этом номинальное значение скорости $V_* = -30.5 + 2.75\theta$, а ускорения $a_* = 4.37 \text{ м/с}^2$. Входными воздействиями являются случайные начальные условия боковой скорости ΔV_{z0} и угла рыскания $\Delta \psi_0$, которые рассматриваются как случайные величины с нормальным законом распределения плотности вероятности, имеющие нулевые математические ожидания и заданные дисперсии $\overline{\Delta V_{z0}^2} = 100 \text{ м}^2 / \text{с}^2$, $\overline{\Delta \psi_0^2} = 2 \cdot 10^{-4} \text{ рад}^2$. Начальные условия будем считать некоррелированными между собой. Возмущением является нестационарный белый шум доплеровского измерителя скорости, определяемого аperiодическим звеном с постоянной времени $T = 1 \text{ с}$. Корреляционная функция шума определяется следующим образом: $R_{nn}(\theta, \tau) = S_0 x_*(\theta) x_*(\tau) \delta(\theta - \tau)$, где $S_0 = 6.8 \cdot 10^{-4} \text{ м}^2 / \text{с}^2$. Выходными координатами системы являются боковая скорость спускаемого аппарата $\Delta V_z(\theta)$ и угол рыскания $\Delta \psi(\theta)$. Расчет позволяет определить оптимальную константу наведения k_a блока выработки команд управления при оптимальном значении конечной дисперсии боковой скорости $D_{\Delta V_z}(t_k)$ при допустимом значении угловой ориентации по углу рыскания $D_{\Delta \psi}(t_k)$, определяемой условием вертикализации посадочного аппарата. Условием эффективности функционирования системы является монотонный характер переходных процессов выходных координат и ограничение угла рыскания по модулю, определяемое условиями работы доплеровской аппаратуры.

Для получения количественных результатов примем следующие числовые значения параметров движения в конечный момент времени t_k :

$$x_k = 30 \text{ м}; \quad V_{xk} = -3 \text{ м/с}.$$

При таких числовых данных время управляемого движения получается равным 10 с.

6.2. Спектральный расчет системы управления в системе математического визуального моделирования Simulink с применением пакета расширения Spektr_SM.

Спектральный расчет искомой системы управления включает в себя следующие этапы:

- 1) По заданной структурной схеме (см. рис. 26), используя библиотеку компонентов пакета Spektr_SM, окно браузера которой показано на рис. 3, составляется структурная спектральная схема перетаскиванием нужных компонент мышью в окно модели ВКМPL_RK (см. рис. 27). Эта модель предназначена для вычисления конечных значений дисперсии $D_{\Delta V_z}(t_k)$ в функции параметра k_a при учете всех воздействий на систему.

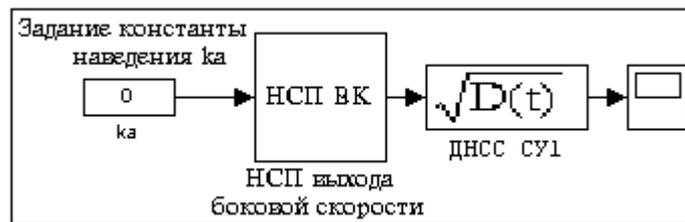


Рис. 27

Подсистема НСП БК (блок Subsystem из раздела Ports&Subsystems библиотеки Simulink) обеспечивает вычисление НСП выходного сигнала боковой скорости ΔV_z и имеет структуру, показанную на рис. 28. Задание константы наведения k_a обеспечивается блоком Clock (k_a), а вычисление среднего квадратичного значения боковой скорости $\Delta V_z(t_k, k_a)$ в конечный момент времени в функции константы наведения k_a типовым блоком ДНСС СУ пакета расширения Spektr_SM. Для визуализации вычисленных данных используется типовой блок осциллографа Scope (источник текущего времени). Так как все спектральные характеристики вычисляются в пакете расширения Spektr_SM за один такт, то для вычисления среднеквадратического значения боковой скорости $\Delta V_z(t_k, k_a)$ в конечный момент времени в функции константы наведения $k_a = 0.01, 0.2 \dots 10$ систему Simulink нужно настроить через меню редактирования Simulation (см. раздел 5). Для этого в окне *Simulation parameters: ВКМPL_RK* на вкладке Solver необходимо задать параметры Start time равным 0.01, Stop time равным 10, Fixed step size равным 0.2.

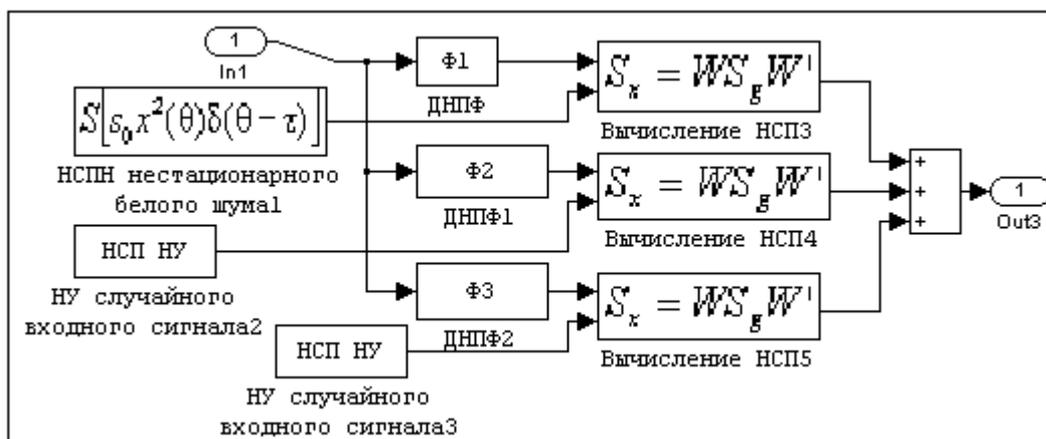


Рис. 28

Подсистемы вычисления ДНПФ Φ_1, Φ_2, Φ_3 отражают структуру системы управления (рис. 26) и обеспечивают вычисление ДНПФ от входов $n(\theta), \Delta\psi_0, \Delta V_{z0}$ к выходу ΔV_z . На рис. 29 показана подсистема вычисления ДНПФ Φ_1 .

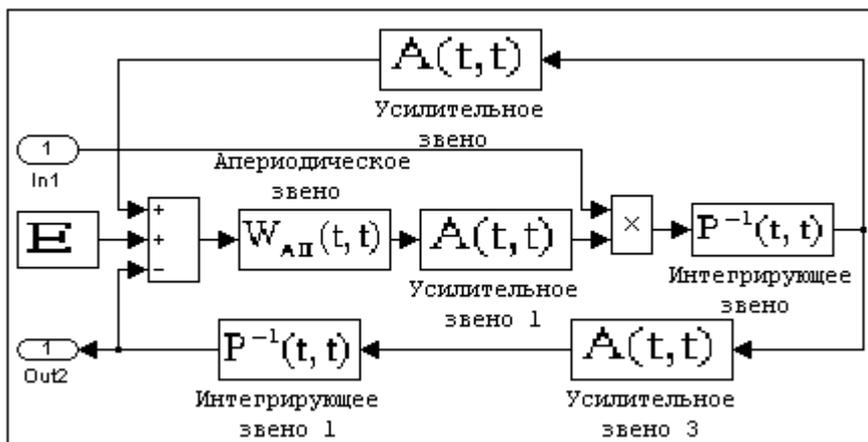


Рис. 29

- 2) Определяем среднеквадратические значения боковой скорости $\Delta V_z(t_k, k_a)$ в конечный момент времени в функции константы наведения k_a при учете всех воздействий на систему. График средних квадратичных значений боковой скорости $\Delta V_z(t_k, k_a)$ в конечный момент времени в функции константы наведения k_a представлен на рис. 30. По этому графику определяется оптимальное значение $k_{a\text{opt}} = 4$.

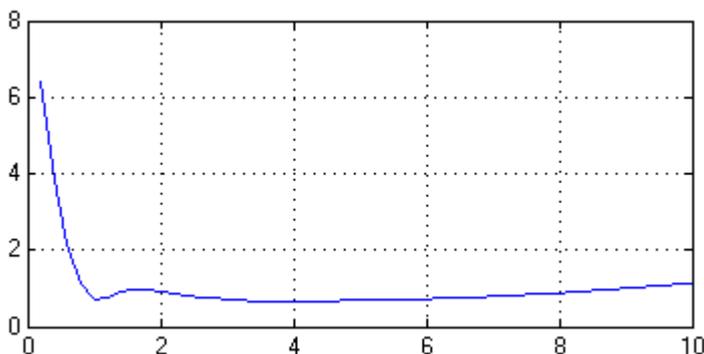


Рис. 30

- 3) По заданной структурной схеме (см. рис. 26), используя библиотеку компонентов пакета Spektr_SM, составляется структурная спектральная схема перетаскиванием нужных компонент мышью в окно модели ВКМРЛ_Р (рис. 31). Эта модель предназначена для вычисления переходных процессов по $\Delta V_z(\theta)$ и $\Delta\psi(\theta)$ при учете всех воздействий на систему. Структурная схема подсистемы **НСП по УР** показана на рис. 32. Структурная схема подсистемы **НСП по БС** совпадает со структурной схемой подсистемы **НСП по УР**, но отличается именами пиктограмм трех своих подсистем. Надо заменить имя пиктограммы Φ_1 на Φ_4 , Φ_2 на Φ_5 , Φ_3 на Φ_6 .

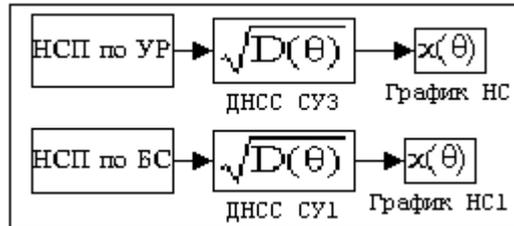


Рис. 31

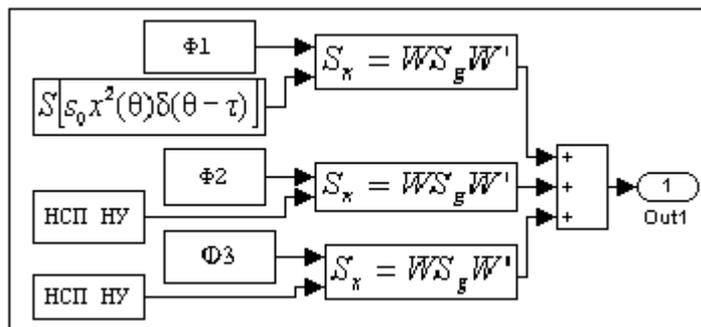


Рис. 32

Подсистемы вычисления ДНПФ $\Phi 1, \Phi 2, \Phi 3, \Phi 4, \Phi 5, \Phi 6$ отражают структуру системы управления (Рис. 26) и обеспечивают вычисление ДНПФ от входов $n(\theta), \Delta \psi_0, \Delta V_{z0}$ к выходам $\Delta V_z(\theta)$ и $\Delta \psi(\theta)$. На рис. 33 показана подсистема вычисления ДНПФ $\Phi 1$.

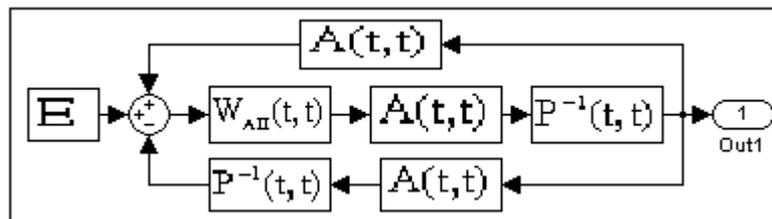


Рис. 33

- 4) Для оптимального значения $k_a = 4$ определяем средние квадратичные значения боковой скорости $\Delta V_z(\theta)$ и угла рыскания $\Delta \psi(\theta)$ при учете всех воздействий на систему. Графики средних квадратичных значений боковой скорости $\Delta V_z(\theta)$ и угла рыскания $\Delta \psi(\theta)$ представлены на рис. 34 а) и рис. 34 б).

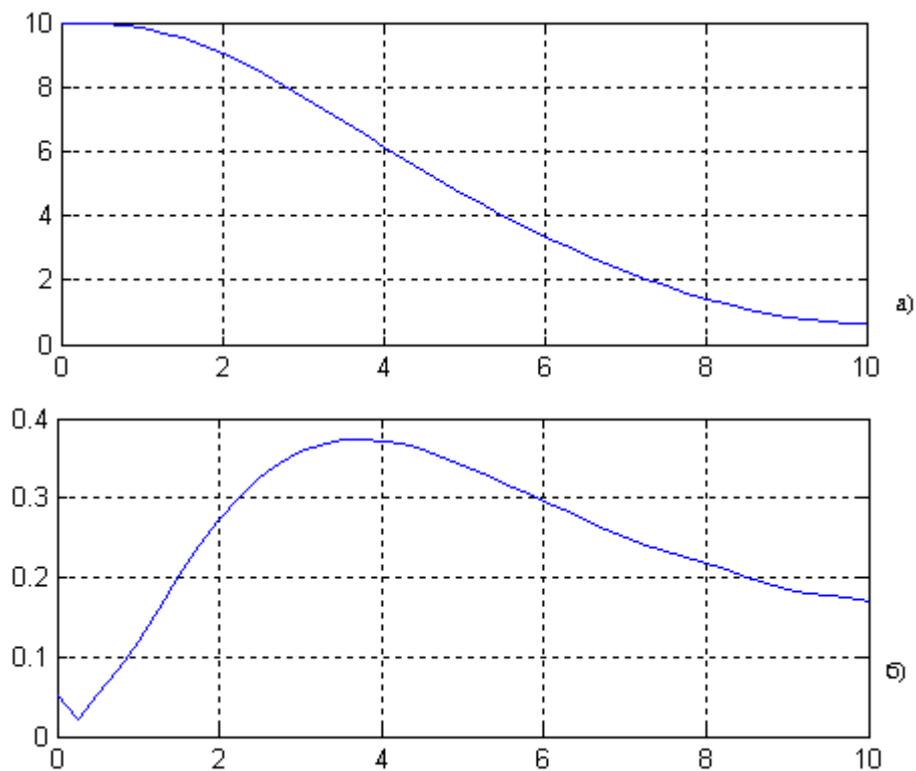


Рис. 34

Исходя из условий эффективности функционирования бокового канала, переходные процессы признаются удовлетворительными.

Список литературы

1. Солодовников В.В. и др. Расчет систем управления на ЦВМ. – М.: Машиностроение, 1979. – 664 с.
2. Солодовников В. В., Семенов В. В. Спектральная теория нестационарных систем управления. - М.: Наука, 1974. – 335 с.
3. Семенов В.В., Рыбин В.В. Алгоритмическое и программное обеспечение расчета нестационарных непрерывно-дискретных систем управления ЛА спектральным методом: Учебное пособие. - М.: МАИ, 1984. – 84 с.
4. Т.Б. Волкова, В.В. Рыбин и др. Изучение математических дисциплин в компьютерной среде: Учебное пособие / Под ред. В.В. Семенова. – М.: МАИ, 1996. – 104 с.
5. Семенов В.В. и др. Математическая теория управления в примерах и задачах: Учебное пособие.- М.: МАИ, 1997. – 264 с.
6. Семенов В.В., Рыбин В.В и др. Алгоритмическое и математическое обеспечение автоматизации проектирования систем управления: Учебное пособие / Под ред. В.В. Семенова. - М.: МАИ, 1982. – 51 с.

7. Рыбин В.В. Описание сигналов и линейных нестационарных непрерывных систем управления в базисах вейвлетов. // Электронный журнал “Труды МАИ”, № 10. – <http://www.mai.ru> (23.01.03).
8. Рыбин В.В. Разработка и применение пакетов расширения MLSY_SM СКМ Mathcad, Maple, Mathematica, Matlab.// Электронный журнал “Труды МАИ”- 2003, № 13.
9. Рыбин В.В. Описание сигналов и линейных нестационарных непрерывных систем управления в базисах вейвлетов и их анализ в вычислительных средах: Учебное пособие. – М.: МАИ, 2003. – 96 с.
10. Дьяконов В.П. MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. Полное руководство пользователя. - М.: СОЛОН-Пресс, 2002. – 768 с.

*Рыбин Владимир Васильевич, доцент кафедры математической кибернетики Московского авиационного института (государственного технического университета), к.т.н.
контактный телефон: 158-48-11
E-mail: dep805@mai.ru*