



ИЗДАТЕЛЬСТВО

**МОСКОВСКИЙ
АВИАЦИОННЫЙ
ИНСТИТУТ**

В.В. РЫБИН

**РАЗРАБОТКА
И ПРИМЕНЕНИЕ
ПАКЕТА РАСШИРЕНИЯ
СПЕКТР_SM
ПАКЕТА
SIMULINK СКМ MATLAB**

Москва • 2004

УЧЕБНОЕ ПОСОБИЕ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(государственный технический университет)

В.В. РЫБИН

**РАЗРАБОТКА И ПРИМЕНЕНИЕ ПАКЕТА
РАСШИРЕНИЯ СПЕКТР_SM
ПАКЕТА SIMULINK СКМ MATLAB**

Учебное пособие

Утверждено
на заседании редсовета
13 октября 2003 г.

Москва
Издательство МАИ
2004

Рыбки В.В. Разработка и применение пакета расширения Spektr_SM пакета Simulink СКМ Matlab: Учебное пособие. — М.: Изд-во МАИ, 2004. — 92 с.: ил.

В пособии рассмотрена технология разработки пакета расширения Spektr_SM системы визуального моделирования Simulink, который позволяет проводить анализ в спектральной области линейных нестационарных непрерывных, дискретных и непрерывно-дискретных систем управления, находящихся под воздействием детерминированных и случайных сигналов. Даются характеристика и описание пакета. Приводятся примеры автоматизированного ввода постановок задач анализа систем управления (составления S-моделей систем управления) и их расчета с использованием пакета Spektr_SM и Simulink.

Пособие может быть полезно при изучении технологий разработки автоматизированных систем проектирования систем управления и их практического применения при проведении компьютерных занятий и выполнении курсовых и дипломных работ по дисциплинам, которые изучают системы управления и их автоматизированное проектирование.

Предназначено для студентов специальности «Прикладная математика». Оно также может быть полезно студентам и аспирантам других специальностей, изучающих современные методы теории управления и занимающихся вопросами проектирования систем управления.

Рецензенты:

каф. «Автоматизация биотехнических систем» МГУ
прикладной биотехнологии (зав. каф. В.И. Попов), В.Б. Чадов

ПРЕДИСЛОВИЕ

Под расчетом систем управления обычно понимается исследование динамических свойств, формирование, проектирование математической модели системы с желательными динамическими свойствами. Систему, обеспечивающую автоматизацию такого расчета, называют системой математического моделирования систем управления.

В настоящее время для математических расчетов в различных прикладных областях применяются различные системы компьютерной математики (СКМ). Среди них особо выделяется математическая система Matlab [8]. Она ориентирована на операции с матричными объектами. В ее состав входит пакет моделирования блочно заданных динамических систем Simulink [7, 8]. Он предназначен для математического моделирования линейных и нелинейных динамических систем и устройств, представленных своей функциональной блок-схемой, имитируемой S-моделью. При этом возможны различные варианты моделирования: во временной области, в частотной области, на основе спектральных преобразований Фурье и т.д.

Обычно для описания систем управления используют четыре формы математического описания [1—3]: 1) дифференциальными и разностными уравнениями; 2) интегральными уравнениями и их разностными аналогами; 3) интегральными преобразованиями; 4) спектральными преобразованиями.

Первые три формы математического описания используются при моделировании динамических систем, в том или ином виде, в пакете Simulink, а четвертая форма математического описания не используется в нем при моделировании.

Спектральные преобразования применяются для описания линейных нестационарных непрерывных, дискретных и непрерывно-дискретных систем управления [1—4], а алгоритмическое обеспечение спектрального метода расчета систем управления включено в настоящее время в систему прикладных программ [9], которая является подсистемой расчетно-логической системы (РЛС) Спектр. РЛС Спектр позволяет моделировать блочно заданные динамические системы спектральным методом подобно тому, как это делают-

ся в пакете Simulink. РЛС Спектр является проблемно-алгоритмической компонентой компьютерного курса «Спектральная форма математического описания линейных нестационарных систем управления» [4] и используется при проведении лабораторных и курсовых работ по курсам «Теория управления», «Основы построения кибернетических систем» и др. Компьютерный курс «Спектральная форма математического описания линейных нестационарных систем управления» в настоящее время включает в себя пакеты прикладных программ спектрального метода, разработанные для СКМ Mathcad, Maple, Mathematica, Matlab. [14]. Этот компьютерный курс включает в себя пакет расширения Spektr_SM пакета Simulink СКМ Matlab, который позволяет рассчитывать блочно заданные линейные нестационарные непрерывные, дискретные и непрерывно-дискретные системы управления спектральным методом. Краткое описание этого пакета и его применение для анализа нестационарных систем управления в вейвлет-базисе Хаара приводится в учебном пособии [10].

В данном пособии рассмотрена технология разработки пакета расширения Spektr_SM системы Simulink, который позволяет проводить анализ в спектральной области линейных нестационарных непрерывных, дискретных и непрерывно-дискретных систем управления, находящихся под воздействием детерминированных и случайных сигналов. Даются характеристика и описание пакета. Приводятся примеры составления S-моделей систем управления и их расчета с использованием пакета Spektr_SM и Simulink.

1

РАЗРАБОТКА ПАКЕТА РАСШИРЕНИЯ *Spektr_SM*

1.1. Краткая характеристика системы математического визуального моделирования *Simulink*

Пакет расширения *Simulink* [7,8] системы компьютерной математики *Matlab* (последняя версия *Simulink 5*) предназначен для моделирования динамических нелинейных систем, модели которых составляют из отдельных блоков (компонентов), т.е. элементарных стандартных звеньев. В результате такой сборки образуется модель исследуемой системы (*S* — модель).

Формирование *S*-моделей средствами *Simulink* основано на использовании технологии *Drag-and-Drop* (перетащи и оставь). Элементарные стандартные звенья, из которых строятся *S*-модели, хранятся в библиотеке *Simulink*. После переноса в окно формируемой модели они настраиваются по заданным параметрам. Для формирования сложных моделей, имеющих иерархическую структуру, т.е. собранную из *S*-моделей более низкого уровня, в разделе *Ports&Subsystems* библиотеки *Simulink* имеется блок *Subsystem*, позволяющий формировать такие подсистемы. Число уровней сложных иерархических моделей практически неограниченно.

Библиотека блоков *Simulink* представляет собой набор визуальных объектов, при помощи которых можно, соединяя блоки, перенесенные в окно модели, линиями функциональной связи, составлять блок-схему любой системы.

Библиотека блоков *Simulink 5* разделена на 13 разделов, окно браузера которой показано на рис. 1.1.

Рассмотрим некоторые из разделов и выделим в них некоторые компоненты, которые в дальнейшем будем использовать.

Блоки, входящие в раздел *Sources* (источники), предназначены для формирования сигналов, воздействующих на *S*-модель (рис. 1.2).

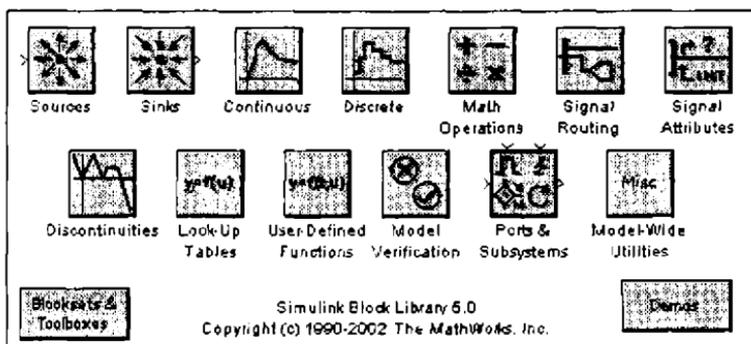


Рис. 1.1

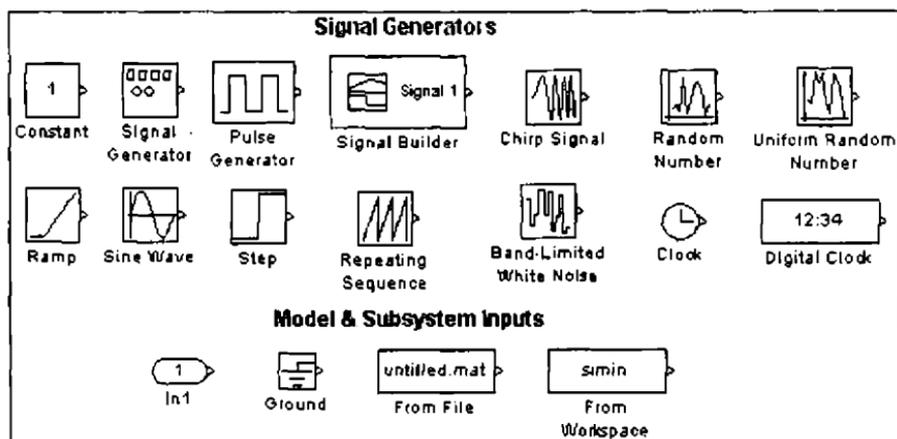


Рис. 1.2

Набор блоков, представленных пиктограммами, содержит часто используемые источники непрерывных детерминированных или случайных сигналов, а также блоки для ввода данных из внешнего файла (From File) или рабочего пространства (From Workspace).

Блоки, входящие в раздел Sinks (Приемники или виртуальные регистраторы), предназначены для визуализации выходных сигналов и их характеристик, а также для управления моделированием и сохранения результатов моделирования во внешнем файле (To File) или рабочем пространстве (To Workspace) (рис. 1.3).

Блоки, входящие в раздел Continuous (непрерывные линейные системы), предназначены для моделирования основных элементарных и типовых операций (рис. 1.4):

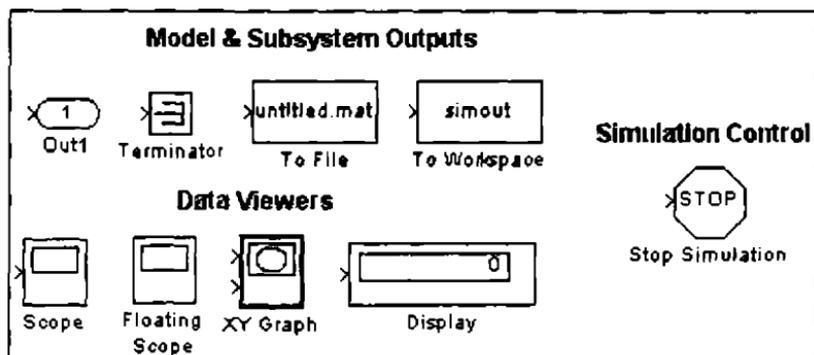


Рис. 1.3

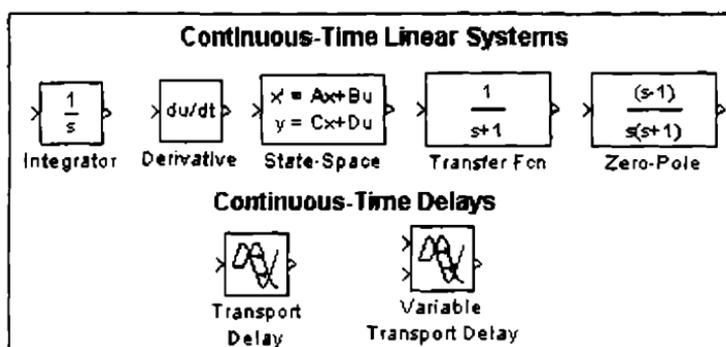


Рис. 1.4

Integrator — интегрирующий блок;

Derivative — дифференцирующий блок;

State-Space — блок задания линеаризованной модели;

Transfer Fcn — блок задания передаточной функции (характеристики) типового звена;

Zero-Pole — блок задания передаточной функции типового звена по заданным полюсам и нулям;

Transport Delay — блок чистого запаздывания (задержки входного сигнала);

Variable Transport Delay — блок управляемой задержки.

Блоки, входящие в раздел Discontinuous (типичные нелинейности), представлены на рис. 1.5.

Блоки, входящие в раздел Math Operations (математических операций), предназначены для моделирования основных арифме-

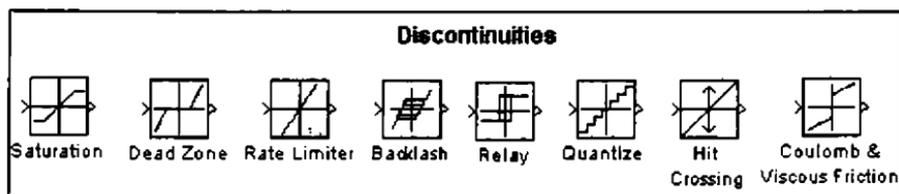


Рис. 1.5

тических, логических и алгебраических операций, а также вычисления элементарных функций. Некоторые блоки раздел Math Operations показаны на рис. 1.6

Блоки, входящие в раздел Signal Routing (обслуживание сигналов), предназначены для приема и передачи сигналов между блоками S-моделей, а также переформирования сигналов. Блоки раздел Signal Routing показаны на рис. 1.7.

Блоки, показанные на рис. 1.8 и входящие в раздел User-Defined Functions (задание функций пользователя), предназначены для задания функций пользователя на языке программирования базовой системы Matlab (блоки Fcn и MATLAB Fcn) или написанные на языках C, Ада, Фортран и Matlab (блоки S-Function, S-Function Builder).

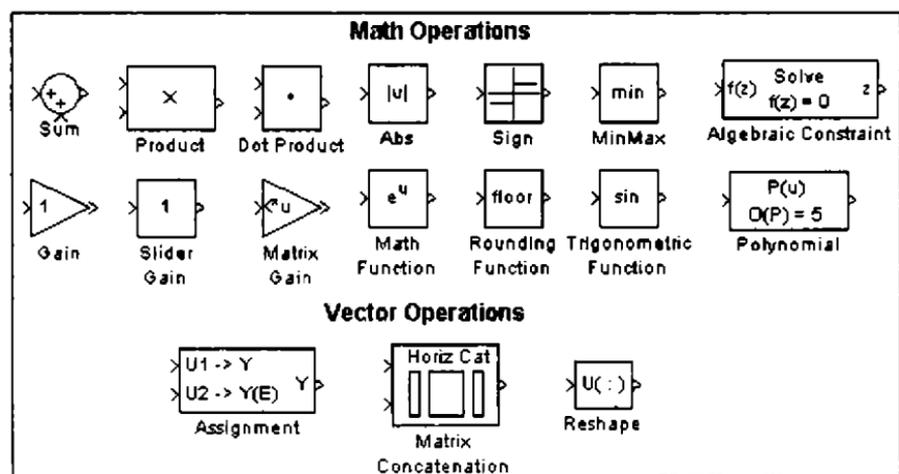


Рис. 1.6

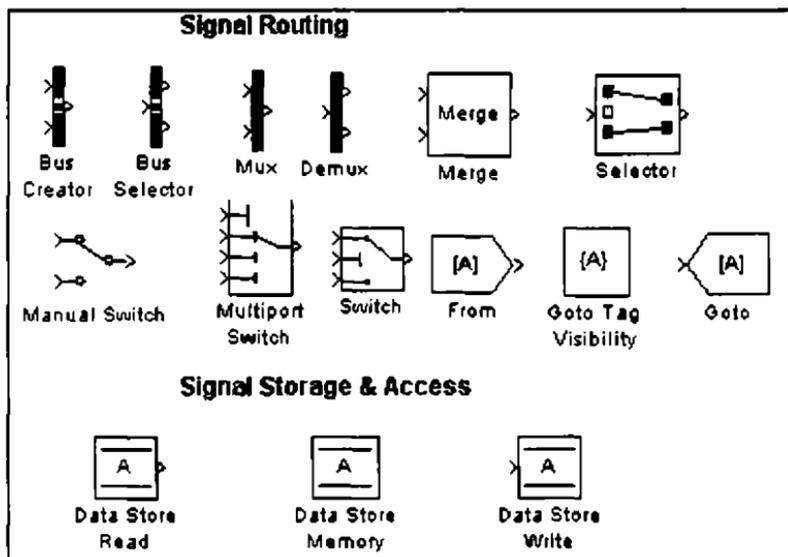


Рис. 1.7

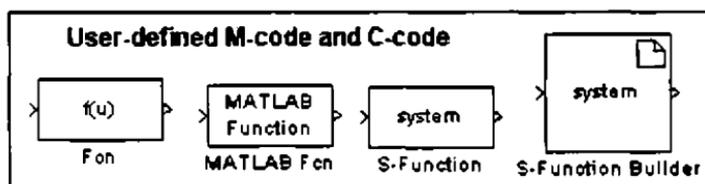


Рис. 1.8

Заметим, что с каждым блоком связано окно настроек, которое открывается при активизации пиктограммы блока в окне компонентов или в окне модели. Для нового блока, помещенного в окно модели, в его окне настроек отображаются значения по умолчанию.

В заключение заметим, что библиотека Simulink сопровождается различными пакетами ее расширения. Среди них отметим только три. Это пакет Simulink Extras, который содержит ряд блоков с более широкими функциями, чем рассмотренные выше. Пакет DSP Blockset, предназначенный для моделирования разномоб-

разных цифровых устройств и содержащий боки матричной алгебры. Пакет событийного моделирования Stateflow, который базируется на теории конечных автоматов и позволяет расширить возможности моделирования сложных систем в системе математического визуального моделирования Simulink.

1.2. Характеристика пакета прикладных программ спектрального метода $MLSY_SM$ в СКМ Matlab

Для анализа и синтеза нестационарных непрерывных и, в общем случае, непрерывно-дискретных систем управления разработан спектральный метод расчета [1–4, 6, 10, 11]. Этот метод удобен для реализации его в системе автоматизации математических расчетов Matlab (Matrix Laboratory — *матричная лаборатория*), так как системные характеристики и характеристики сигналов в спектральной области представляются матрицами и векторами, а алгоритмы анализа и синтеза используют три алгебраические операции: сложение, умножение и обращение матриц. Любая сложная система с помощью метода декомпозиции может быть представлена в виде набора элементарных и типовых звеньев, которые соединяются в систему при помощи трех типов соединений: последовательного, параллельного и с обратной связью.

В спектральной области всем элементарным операциям ставится в соответствие система элементарных алгоритмов. На базе этой системы строится система алгоритмов исследования конкретной системы управления.

1.2.1. Основные характеристики спектральной формы описания непрерывно-дискретных систем

Основным понятием спектрального метода является понятие нестационарной спектральной характеристики (НСХ). Для одномерной функции НСХ (модифицированная НСХ — МНСХ) определяется как скалярное произведение вида

$$S[x] = X(i) = (p\psi(i), x) \left(SM[x] = XM(i) = (\psi(i), x) \right), \quad (1.1)$$

которое раскрывается для непрерывной функции времени как

$$\begin{aligned} S_{\Psi}[x(\tau)] &= X_{\Psi}(i, t) = \int_0^t \rho(t, \tau) \Psi^*(i, t, \tau) x(\tau) d\tau \\ \left(SM_{\Psi}[x(\tau)] &= XM_{\Psi}(i, t) = \int_0^t \Psi^*(i, t, \tau) x(\tau) d\tau \right) \end{aligned} \quad (1.2)$$

и для дискретной функции времени как

$$\begin{aligned} S_{\Psi}[x(l)] &= X_{\Psi}(i, L) = \sum_{l=0}^{L-1} \rho(L, l) \Psi^*(i, L, l) x(l) \\ \left(SM_{\Psi}[x(l)] &= XM_{\Psi}(i, L) = \sum_{l=0}^{L-1} \Psi^*(i, L, l) x(l) \right) \end{aligned} \quad (1.3)$$

Аналогично определяется понятие НСХ функции двух переменных (ДНСХ):

$$S_{qp}[x] = X_{qp}(h, i) = \begin{pmatrix} \rho & \rho q(h) & p^*(i), x \\ q & p & \end{pmatrix}. \quad (1.4)$$

Например, выражение (1.4) для непрерывно-дискретной функции времени $x(\theta, m)$ раскрывается как

$$\begin{aligned} S_{qp}[x(\theta, m)] &= X_{qp}(h, i, t, M) = \\ &= \int_0^t \rho(t, \theta) q^*(h, t, \theta) \sum_{m=0}^{M-1} \rho(M, m) p^*(i, M, m) x(\theta, m) d\theta. \end{aligned} \quad (1.5)$$

Подобными формулами определяются многомерные НСХ функций многих переменных. Заметим, что иногда, по некоторым аргументам, НСХ описываются как МНСХ.

Обращение НСХ, т.е. отыскание искомой функции многих аргументов, производится по формулам обращения. Например, для функций одного и двух аргументов

$$S^{-1} \left[\begin{matrix} X(i) \\ \Psi \end{matrix} \right] = x = \left(\begin{matrix} \psi^*(i), & X(i) \\ \Psi \end{matrix} \right); \quad (1.6)$$

$$S^{-1} \left[\begin{matrix} X_*(h, i) \\ qP \end{matrix} \right] = x = \left(\begin{matrix} q^*(h)p(i), & X_*(h, i) \\ qP \end{matrix} \right). \quad (1.7)$$

Случайные нестационарные сигналы в спектральной области описываются НСХ своих моментных функций: первой нестационарной спектральной плотностью (НСП)

$${}^1 S_x(i) = S_p[m_x], \quad (1.8)$$

где m_x — математическое ожидание случайного сигнала; второй НСП (или просто НСП)

$${}^1 S_x(h, i) = S_{pp}[R_{xx}], \quad (1.9)$$

где R_{xx} — корреляционная функция случайного сигнала.

Аналогично определяются и остальные НСХ моментных функций случайного сигнала.

Системными характеристиками нестационарных непрерывно-дискретных систем являются нестационарные передаточные функции (НПФ): нормальная (ННПФ), сопряженная (СНПФ), двумерная (ДНПФ). Для непрерывно-дискретной системы они классифицируются по типу входа и выхода как Н-Н, Д-Д, Н-Д, Д-Н непрерывно-дискретные системы.

Например, ДНПФ определяется как

$$W_{qp}^*(h, i) = \left(\begin{matrix} p & q(h)p^*(i), & k \\ q \end{matrix} \right). \quad (1.10)$$

т.е. для систем:

— непрерывных и Н-Н непрерывно-дискретных

$$W_{qp}^*(h, i, t, t) = \int_0^t \rho(t, \theta) q^*(h, t, \theta) \int_0^t k(\theta, \tau) p(i, t, \tau) d\tau d\theta; \quad (1.11)$$

— дискретных и Д-Д непрерывно-дискретных

$$W_{qp}^*(h, i, L, M) = \sum_{l=0}^{L-1} \rho(L, l) q^*(h, L, l) \sum_{m=0}^{M-1} k(l, m) p(i, M, m); \quad (1.12)$$

— Д-Н непрерывно-дискретных

$$W_{qp}^*(h, i, L, t) = \sum_{l=0}^{L-1} \rho(L, l) q^*(h, L, l) \int_0^t k(l, \tau) p(i, t, \tau) d\tau; \quad (1.13)$$

— Н-Д непрерывно-дискретных

$$W_{qp}^*(h, i, t, M) = \int_0^t \rho(t, \theta) q^*(h, t, \theta) \sum_{m=0}^{M-1} k(\theta, m) p(i, M, m) d\theta. \quad (1.14)$$

Заметим, что эти характеристики определены как модифицированные НСХ по второму аргументу $\tau(m)$ импульсной переходной функции (ИПФ) k .

НСХ, НСП и ДНПФ удобно представлять в матричной форме. В этой форме ДНПФ непрерывной системы, соответствующая дифференциальному уравнению

$$\sum_{i=0}^n a_i(\theta) \frac{d^i x(\theta)}{d\theta} = \sum_{i=0}^m b_i(\theta) \frac{d^i g(\theta)}{d\theta}, \quad (1.15)$$

выражается соотношением

$$W(t, t) = [A_n(t, t)P^n(t, t) + \dots + A_0(t, t)]^{-1} \times \\ \times [B_m(t, t)P^m(t, t) + \dots + B_0(t, t)]. \quad (1.16)$$

ДНПФ дискретной системы (1.12), соответствующая конечно-разностному уравнению

$$\sum_{i=0}^n a_i(l) \nabla_l^i x(l) = \sum_{i=0}^m b_i(l) \nabla_l^i g(l), \quad (1.17)$$

выражается соотношением, подобным (1.16):

$$W(L, L) = [A_n(L, L)P^n(L, L) + \dots + A_0(L, L)]^{-1} \times \\ \times [B_m(L, L)P^m(L, L) + \dots + B_0(L, L)]. \quad (1.18)$$

ДНПФ (1.16) и (1.18) выражаются через элементарные и типовые звенья непрерывных и дискретных систем.

В качестве элементарных звеньев непрерывно-дискретных систем обычно рассматриваются интегрирующее, суммирующее, дифференцирующее первого и второго рода, разностное, непрерывные и дискретные звенья с переменными коэффициентами передачи, непрерывное и дискретное звено чистого сдвига (запаздывания и упреждения), понижения такта, сдвига тактовых точек, а также собственно непрерывно-дискретные звенья — дискретный элемент и экстраполяторы.

ДНПФ (1.10) этих звеньев имеют вид:

— ДНПФ интегрирующего звена

$$P_{pp}^{-1}(h, i, t, t) = \int_0^t \rho(L, l) p^*(h, t, \theta) \int_0^\theta p(i, t, \tau) d\tau d\theta; \quad (1.19)$$

— ДНПФ суммирующего звена

$$P_{pp}^{-1}(h, i, L, L) = \sum_{l=0}^{L-1} \rho(L, l) p^*(h, L, l) \sum_{m=0}^l p(i, L, m); \quad (1.20)$$

— ДНПФ дифференцирующего звена первого рода

$$P_{pp}(h, i, t, t) = v_{pp}(h, i, t, t) + \mathfrak{Z}_{pp}(h, i, t, t), \quad (1.21)$$

где $v_{pp}(h, i, t, t) = \eta(t, 0) p^*(h, t, 0) p(i, t, 0)$ — ДНПФ начальных значений, в которой регуляризация $\eta(t, \theta)$ весовой функции $\rho(t, \theta)$ отвечает условию $\rho^{-1}(t, \theta) \eta(t, \theta) = 1$,

а $\mathfrak{Z}_{pp}(h, i, t, t) = \int_0^t \rho(t, \theta) p^*(h, t, \theta) \frac{d}{d\theta} p(i, t, \theta) d\theta$ — ДНПФ дифференцирующего звена второго рода;

— ДНПФ разностного звена

$$P_{pp}(h, i, L, L) = \rho(L, 0) p^*(h, L, 0) p^*(i, L, 0) + \sum_{l=0}^{L-1} \rho(L, l) p^*(h, L, l) \nabla_l p(i, L, l); \quad (1.22)$$

— ДНПФ усилительных звеньев:
непрерывного

$$A_{pp}(h, i, t, t) = \int_0^t \rho(t, \theta) a(\theta) p^*(h, t, \theta) p(i, t, \theta) d\theta, \quad (1.23)$$

дискретного

$$A_{pp}(h, i, L, L) = \sum_{l=0}^{L-1} \rho(L, l) a(l) p^*(h, L, l) p(i, L, l); \quad (1.24)$$

— ДНПФ звена чистого запаздывания:

непрерывного ($\theta_0 > 0$)

$$\tau_{pp}^{-\theta_0}(h, i, t, t) = \int_{\theta_0}^t \rho(t, \theta) p^*(h, t, \theta) p(i, t, \theta - \theta_0) d\theta, \quad (1.25)$$

дискретного ($k > 0$)

$$\tau_{pp}^{-k}(h, i, t, t) = \sum_{l=k}^{L-1} \rho(L, l) a(l) \dot{p}^*(h, L, l) \dot{p}^*(i, L, l); \quad (1.26)$$

— ДНПФ звена чистого упрещения:

непрерывного ($\theta_0 < 0$)

$$\tau_{pp}^{-\theta_0}(h, i, t, t) = \int_0^{t+\theta_0} \rho(t, \theta) p^*(h, t, \theta) p(i, t, \theta - \theta_0) d\theta, \quad (1.27)$$

дискретного ($k < 0$)

$$\tau_{pp}^{-k}(h, i, t, t) = \sum_{l=0}^{L+k-1} \rho(L, l) a(l) \dot{p}^*(h, L, l) \dot{p}^*(i, L, l); \quad (1.28)$$

— ДНПФ звена понижения такта ($L < M$)

$$T_{qp}(h, i, L, M) = \sum_{m=0}^{M-1} \rho(L, \tau_m) \dot{q}^*(h, L, \tau_m) \dot{p}^*(i, M, m); \quad (1.29)$$

— ДНПФ дискретного элемента

$$D_{qp}(h, i, L, t) = \sum_{l=0}^{L-1} \rho(L, l) \dot{q}^*(h, L, l) p(i, t, \theta_l); \quad (1.30)$$

— ДНПФ экстраполирующего звена нулевого порядка

$$\mathcal{D}_{q p}^0(h, i, t, M) = \sum_{m=0}^{M-1} p^*(i, M, m) \int_{\tau_m}^{\tau_{m+1}} \rho(t, \theta) q^*(h, t, \theta) d\theta. \quad (1.31)$$

Получение ДНПФ нестационарной непрерывно-дискретной системы связано с определением ДНПФ линейных звеньев и их соединений (параллельного, последовательного и с обратной связью). ДНПФ таких соединений рассчитываются по ДНПФ звеньев их составляющих по формулам:

— для параллельного соединения

$$W = W_1 + W_2; \quad (1.32)$$

— для последовательного соединения

$$W = W_2 \cdot W_1; \quad (1.33)$$

— для соединения с обратной связью

$$W = [E + W_1 W_2]^{-1} W_1 = W_1 [E + W_2 W_1]^{-1}. \quad (1.34)$$

Связи вход-выход по ДНПФ искомой системы и заданным входным НСХ и НСП при нулевых начальных условиях устанавливаются соотношениями:

— для детерминированных сигналов

$$X = W \cdot G; \quad (1.35)$$

— для случайных сигналов:

по математическому ожиданию

$${}^1 S_x = W \cdot {}^1 S_g, \quad (1.36)$$

по корреляционной функции

$$S_x = W \cdot S_g \cdot W^T; \quad S_{xg} = W \cdot S_g. \quad (1.37)$$

Обращение НСХ (1.35) — (1.37) проводится по соотношениям (1.6) и (1.7), а НПФ — по формулам

$$k = Q \cdot N = H \cdot \Lambda_{pp}^{-1} \cdot P^T = Q \cdot W_{qp} \cdot \Lambda_{pp}^{-1} \cdot P^T, \quad (1.38)$$

где Λ_{pp} — двумерная нестационарная характеристика связи (ДНС) между ДНПФ и ДНСХ искомой системы, которая для непрерывных базисных функций имеет вид

$$\Lambda_{pp}(h, i, t, l) = \int_0^l p^*(h, t, \tau) p(i, l, \tau) d\tau, \quad (1.39)$$

а для дискретных

$$\Lambda_{pp}(h, i, L, L) = \sum_{l=0}^{L-1} p^*(h, L, l) p(i, L, l). \quad (1.40)$$

Формулы связи ДНПФ линейной системы с ее одномерными НПФ имеют вид

$$W_{qp} = (Q^+, H); \quad W_{qp} = (N, P); \quad (1.41)$$

$$N = W_{qp} \cdot \Lambda_{pp}^{-1} \cdot P^+; \quad H = Q \cdot W_{qp}. \quad (1.42)$$

В формулах (1.38), (1.41), (1.42) Q и P — матрицы-строки, составленные из систем базисных функций $\{q(i)\}$ и $\{p(i)\}$.

ДНПФ непрерывной и Н-Н непрерывно-дискретной системы представляется бесконечной матрицей, ДНПФ дискретной и Д-Д непрерывно-дискретной системы — конечной прямоугольной матрицей порядка $L \times M$, а ДНПФ Д-Н и Н-Д непрерывно-дискретной системы полубесконечными матрицами.

1.2.2. Пакет прикладных программ спектрального метода *MLSY_SM*

В настоящее время разработано несколько версий пакета прикладных программ анализа и параметрического синтеза систем управления спектральным методом. Одна из них включена в компьютерный курс «Спектральная теория нестационарных систем управления» [4]. Другая версия создана на базе вычислительных сред Mathcad, Maple, Mathematica, Matlab [14]. Эта версия включает в себя все элементарные операции спектрального метода (1.1)–(1.42) и предназначена для моделирования линейных систем управления спектральным методом (*MLSY_SM*). Для спектральных алгоритмов разработана следующая структура имени программного модуля:

<имя программного модуля> := S <идентификатор процедуры (алгоритма), реализованной в данном программном модуле> <идентификатор базисной системы, относительно которой записан этот алгоритм> <идентификатор численной схемы, реализующей исходный алгоритм>

<идентификатор процедуры (алгоритма), реализованной в данном программном модуле> := <латинская буква> <латинская буква>|<цифра>

<идентификатор базисной системы, относительно которой записан этот алгоритм> := <p|t|u|c|f|y|x|r|s|d|k|e|q|v|h> <p|t|u|c|f|y|x|r|s|d|k|e|q|v|h>

<идентификатор численной схемы, реализующей исходный алгоритм> := <цифра>

В этом имени отражены все необходимые признаки, по которым различаются элементарные алгоритмы спектрального метода, представленные программными модулями. Описание идентификаторов имен программных модулей и способов обращения к ним приводится в приложении 1. Состав программных модулей пакета определяет файловую организацию библиотеки программ. Она одинакова для всех вычислительных сред. Для вычислительной среды Matlab файловая организация библиотеки программ спектрального метода может быть представлена в виде

MLSY_SM — библиотека элементарных алгоритмов спектрального метода;

NBF — для непрерывных базисных функций;

DBF — для дискретных базисных функций;

NDBF — для непрерывно-дискретных базисных функций.

Библиотека **NBF** содержит семь разделов: **SM_C**; **SM_F**; **SM_P**; **SM_T**; **SM_U**; **SM_X**; **SM_Y**, где последний символ в названии раздела соответствует идентификатору базисной функции из разд. 1 приложения 1. Каждый раздел содержит программные модули элементарных алгоритмов спектрального метода, имена которых и способы обращения к ним описаны в разд. 3.1 приложения 1.

Библиотека **DBF** содержит восемь разделов: **SM_D**; **SM_E**; **SM_H**; **SM_K**; **SM_Q**; **SM_R**; **SM_S**; **SM_V**, где последний символ в названии раздела соответствует идентификатору базисной функции из разд. 1 приложения 1. Каждый раздел содержит программные модули элементарных алгоритмов спектрального метода, имена которых и способы обращения к ним описаны в разд. 3.2 приложения 1.

Библиотека **NDBF** содержит десять разделов: **SM_CK**; **SM_CQ**; **SM_FE**; **SM_PR**; **SM_TQ**; **SM_TS**; **SM_UD**; **SM_XH**; **SM_YV**, где два последних символа в названии раздела соответствуют идентификатору базисной системы функций, составленной из идентификаторов непрерывных и дискретных функций из разд. 1 приложения 1. Каждый раздел содержит программные модули элементарных алгоритмов спектрального метода, имена которых и способы обращения к ним описаны в разд. 3.3 приложения 1.

*1.2.3. Примеры разработки программных модулей пакета **MLSY_SM***

Пример 1.1. Требуется разработать программный модуль, реализующий вычисление усеченной НСХ аналитически заданной функции $x(\tau) \in [0, 1]$ в базисе полиномов Чебышева первого рода.

Решение задачи. Вычислительные схемы, основанные на квадратурных правилах наивысшей алгебраической степени точности и реализующие вычисление усеченных НСХ в базисах классических ортогональных полиномов, рассмотрены в приложении 2. Для разрабатываемого программного модуля целесообразно выбрать численную схему (П2.29).

1. Составляем программный модуль (файл-функцию) вычисления усеченной НСХ в базе непрерывных полиномов Чебышева первого рода на интервале работы системы управления $[0, t]$ по аналитически заданной функции:

```
function XX=snxtt1(gg,N1,t)
eta=4; L1=2*eta*N1; a1=[]; tau=[]; X=[];
for l=0:L1-1, tau(l+1)=0.5*t*(1+cos((2*L1-2*l-1)*pi*0.5/L1));
    a1(l+1)=feval(gg,tau(l+1)); Q(l+1,1)=sqrt(1/pi);
    for m=1:N1-1, Q(l+1,m+1)=(-1)^m*sqrt(2/pi)*cos(m*pi*(l+0.5)/L1); end end
for h=1:N1, for i=1:N1, sum=0; for l=1:L1,
sum=sum+a1(l)*Q(l,h); end X(h)=pi*sum/L1; end end XX=X;
```

2. Составляем файл-функцию вычисления аналитически заданной функции $g(\tau) = \sin\left(\frac{\pi\tau}{2t}\right)$:

```
function res=g(tau) global t; res=sin(0.5*pi*tau/t);
```

3. Вычисляем усеченную НСХ функции $g(\tau)$ при $L = 8$ и $t = 4$.

```
global t; L=8; t=4; X=snxtt1('g',L,t); X
ans = 1.0674 0.6437 -0.1298 -0.0172 0.0017 0.0001 -0.0000 -0.0000
```

Пример 1.2. Разработать программный модуль, реализующий вычисление усеченной ДНПФ усилительного звена по аналитически заданной функции $a(\tau) \in [0, t]$ в базе полиномов Чебышева первого рода.

Решение задачи. Воспользуемся вычислительной схемой (П2.12). Тогда ДНПФ усилительного звена (1.23) в базе полиномов Чебышева первого рода может быть представлена в виде

$$\begin{aligned} A_{TT^*}(h, i, t, t) &= \frac{\pi}{L} \sum_{k=0}^{L-1} a(\tau_k^t) \hat{T}_h^*(L, k) \hat{T}_i^*(L, k), \\ h, i &= 0, 1, 2, \dots, L-1, \end{aligned} \quad (1.43)$$

где $\tau_k^t = \frac{t}{2} \left(1 + \cos\left(\frac{2L-2k-1}{2L} \pi\right) \right)$, а $\hat{T}_i^*(L, k)$ — дискретные полиномы Хана-Чебышева (П2.21).

Для разрабатываемого программного модуля целесообразно выбрать численную схему (1.43).

1. Составляем программный модуль вычисления усеченной ДНПФ в базисе непрерывных полиномов Чебышева первого рода на интервале работы системы управления $[0, t]$ по аналитически заданной функции:

```
function X=syztt1(g,N1,t)
    eta=4; L1=2*eta*N1; tau=[]; a1=[]; Q=zeros(L1,N1); X=zeros(N1);
    for l=1:L1, tau(l)=0.5*t*(1+cos((2*L1-2*l+1)*pi*0.5/L1));
    a1(l)=feval(g,tau(l));Q(l,1)=sqrt(1/pi);
    for m=2:N1, Q(l,m)=(-1)^(m-1)*sqrt(2/pi)*cos((m-1)*pi*(l-0.5)/L1);
    end end
    for h=1:N1, for i=1:N1, sum=0; for l=1:L1, sum=sum+a1(l)*
    Q(l,h)*Q(l,i); end
    X(h,i)=pi/L1*sum; end end;
```

2. Составляем файл-функцию вычисления аналитически заданной функции $g(\tau) = \sqrt{\tau(t-\tau)}$.

```
function res=g(tau) global t; res=sqrt(tau*(t-tau));
```

3. Составляем Script-файл zsyzt (файл-сценарий) вычисления усеченной ДНПФ усилительного звена с коэффициентом передачи $g(\tau) = \sqrt{\tau(t-\tau)}$ при $L = 5$ и $t = 4$.

```
script global t; t=4; L=5; A=syztt1("g",L,t); A
```

4. Выполняем script-файл zsyzt:

```
A =
    1.2736 -0.0000 -0.5997 -0.0000 -0.1196
    -0.0000  0.8495 -0.0000 -0.5086 -0.0000
    -0.5997 -0.0000  1.1890 -0.0000 -0.4601
    -0.0000 -0.5086 -0.0000  1.2375 -0.0000
    -0.1196 -0.0000 -0.4601 -0.0000  1.2537
```

Пример 1.3. Разработать программный модуль, реализующий вычисление усеченной ДНХС в базисе полиномов Чебышева первого рода.

Решение задачи. Так как ДНХС (1.39) можно рассматривать как ДНПФ усилительного звена (1.23) с коэффициентом передачи $a(\tau) = \sqrt{\tau(t-\tau)}$, то ДНХС (1.39) в базисе полиномов Чебышева пер-

вого рода, с учетом численной схемы (1.43), может быть представлена в виде

$$\frac{\Delta}{TT}(h, i, t, l) = \frac{\pi}{L} \sum_{k=0}^{L-1} \sqrt{\tau_k^L(t - \tau_k^L)} \hat{T}_h^*(L, k) \hat{T}_i^*(L, k), \quad (1.44)$$

$$h, i = 0, 1, 2, \dots, L-1,$$

где $\tau_k^L = \frac{t}{2} \left(1 + \cos \left(\frac{2L-2k-1}{2L} \pi \right) \right)$, а $\hat{T}_i^*(L, k)$ — дискретные полиномы Хана-Чебышева (П2.21). Для разрабатываемого программного модуля целесообразно выбрать численную схему (1.44).

1. Составляем программный модуль вычисления усеченной ДНХС в базе непрерывных полиномов Чебышева первого рода на интервале работы системы управления $[0, t]$:

```
function XC=sxctt1(N1,t)
    eta=4; L1=2*eta*N1; XC=zeros(N1); Q=zeros(L1,N1); tau=[];
    a1=[];
    for l=1:L1, tau(l)=t/2*(1+cos((2*L1-2*l+1)*pi*0.5/L1));
    a1(l)=sqrt(tau(l)*(t-tau(l))); Q(l,1)=sqrt(1/pi);
    for m=2:N1, Q(l,m)=(-1)^(m-1)*sqrt(2/pi)*cos((m-1)*pi/L1*(l-0.5));
    end end
    for h=1:N1, for i=1:N1, sum=0; for l=1:L1, sum=sum+a1(l)*
    Q(l,h)*Q(l,i); end
    XC(h,i)=pi/L1*sum; end end
```

2. Составляем Script-файл zsxctt вычисления усеченной ДНХС при $L = 5$ и $t = 4$.

```
script global t; t=4; L=5; XC=sxctt1('g',L,t); XC
```

3. Выполняем script-файл zsxctt. Результат выполнения программы совпадает с результатом вычисления ДНПФ усилительного звена из примера 1.2.

Пример 1.4. Разработать программный модуль, реализующий вычисление усеченной ДНПФ интегрирующего звена в базе полиномов Чебышева первого рода.

Решение задачи. Вычислим ДНПФ интегрирующего звена (1.19) в базе полиномов Чебышева первого рода в аналитическом виде. Тогда получим

$$P^{-1}(h,i,t) = \begin{cases} t/2 & \text{при } h=i=0; -t\sqrt{2}/8 & \text{при } h=0, i=1; \\ t\sqrt{2}/4 & \text{при } h=1, i=0; -t/(4h) & \text{при } i=h+1, \\ h=1,2,\dots; t/(4h) & \text{при } i=h-1, h=2,3,\dots; \\ 0 & \text{в остальных случаях.} \end{cases} \quad (1.45)$$

Для разрабатываемого программного модуля целесообразно выбрать численную схему (1.45).

1. Составляем программный модуль вычисления усеченной ДНПФ интегрирующего звена в базисе полиномов Чебышева первого рода на интервале работы системы управления $[0, t]$.

```
function C1=SIITT1(t,L)
    C=zeros(L); for h=2:L, C(h,h-1)=1/(4*(h-1));
    if h>2, C(1,h)=(-1)^h/(sqrt(2)*h*(h-2)); C(h-1,h)=-1/(4*(h-2));
end end
```

```
C(1,1)=1/2; C(1,2)=-sqrt(2)/8; C(2,1)=sqrt(2)/4; C1=t*C;
```

2. Составляем Script-файл `zsltt` вычисления усеченной ДНПФ интегрирующего звена при $L = 5$ и $t = 4$.

```
script t=1; L=5; II=zsltt1(L,t); II
```

3. Выполняем script-файл `zsltt`.

```
II =
    0.5000    -0.1768    -0.2357    0.0884    -0.0471
    0.3536     0    -0.250     0     0
     0     0.12500    -0.125     0     0
     0     0     0.0833     0    -0.0833
     0     0     0     0.0625     0
```

Пример 1.5. Разработать программный модуль, реализующий вычисление усеченной ДНПФ дифференцирующего звена в базисе полиномов Чебышева первого рода.

Решение задачи. Найдем численную схему вычисления ДНПФ дифференцирующего звена (1.21) в базисе полиномов Чебышева первого рода в аналитическом виде, используя для этого методику символического обращения матрицы ДНПФ интегрирующего звена (1.45) с помощью пакета Symbolic Math Toolbox:

1. Составляем программный модуль вычисления в символьном виде усеченной ДНПФ интегрирующего звена в базисе полиномов Чебышева первого рода на интервале работы системы управления $[0, t]$.

function C1=SI1TT1S(t,L)

C=sym(zeros(L)); for h=2:L, C(h,h-1)=1/(4*(h-1));

if h>2, C(1,h)=(-1)^h/(sqrt(2)*h*(h-2)); C(h-1,h)=-1/(4*(h-2)); end

end
C(1,1)=1/2; C(1,2)=-sqrt(2)/8; C(2,1)=sqrt(2)/4; C1=t*C;

2. Вычисляем в символьном виде усеченную ДНПФ дифференцирующего звена путем обращения усеченной матрицы ДНПФ интегрирующего звена при $L = 2, 3, 4, \dots$

> t=sym('t'); L=2; PS=SI1TT1S(t,L)^-1; PS

Например, для $L = 2$ PS имеет вид

$$\begin{bmatrix} 0, & 2/t*2^{(1/2)} \\ -4/t*2^{(1/2)}, & 8/t \end{bmatrix}$$

для $L = 3$ PS имеет вид

$$\begin{bmatrix} 6/t, & -4/t*2^{(1/2)}, & 6/t*2^{(1/2)} \\ 0, & 0, & 8/t \\ 6/t*2^{(1/2)}, & -12/t, & 12/t \end{bmatrix}$$

для $L = 4$ PS имеет вид

$$\begin{bmatrix} 0, & 2/t*2^{(1/2)}, & 0, & 6/t*2^{(1/2)} \\ -8/t*2^{(1/2)}, & 16/t, & -8/t, & 16/t \\ 0, & 0, & 0, & 12/t \\ -8/t*2^{(1/2)}, & 16/t, & -16/t, & 16/t \end{bmatrix}$$

и т.д.

3. Анализируя полученные усеченные матрицы, находим численную схему в аналитическом виде для вычисления ДНПФ дифференцирующего звена, по которой составляем программный модуль вычисления усеченной ДНПФ дифференцирующего звена в базисе полиномов Чебышева первого рода на интервале работы системы управления $[0, t]$:

function m=sp1tt1(t,L)

m=zeros(L); ny=zeros(L); for k=0:L-1, if 2*k+1<=L-1,

m(1,2*k+2)=2*sqrt(2)*(2*k+1); end end for h=1:L, for k=0:L,

if (h+2*k+1)<=L-1, m(h+1,h+2*k+2)=4*(h+2*k+1); end end end

m(L,1)=0;

ny(1,1)=1-(-1)^L; for i=2:L, ny(1,i)=(-1)^(i-1)*(1-(-1)^L)*sqrt(2);

ny(i,1)=(-1)^(i-1)*(1-(-1)^(L+i-1))*sqrt(2); end

for h=2:L, for i=2:L, ny(h,i)=(-1)^(h+i)*(1-(-1)^(h-1+L))*2; end

end

ny=L*ny; m=(m+ny)/t;

4. Вычисляем усеченную ДНПФ дифференцирующего звена при $L = 5$ и $t = 4$:

> $t=4$; $L=5$; $P=sp1tt1(t,L)$; P

$P =$

2.5000	-2.8284	3.5355	-1.4142	3.5355
0	0	2.0000	0	4.0000
3.5355	-5.0000	5.0000	-2.0000	5.0000
0	0	0	0	4.0000
3.5355	-5.0000	5.0000	-5.0000	5.0000

Приведенные примеры демонстрируют методику получения численных схем и их программную реализацию в базе полиномов Чебышева первого рода в рамках структуры пакета расширения `MLSY_SM`. Аналогично разрабатываются и другие программные модули как в базе полиномов Чебышева первого рода, так и в других базисах.

1.3. 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 расчета нестационарных непрерывно-дискретных систем управления спектральным методом [9], который входит в компьютерный курс по спектральной теории нестационарных систем управления. Эта САР базируется на ППП спектрального метода [1–4], реализующих 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`, краткая характеристика которого дана в разд. 1.1. 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 непрерыв-

Разработка пакета Spektr_SM

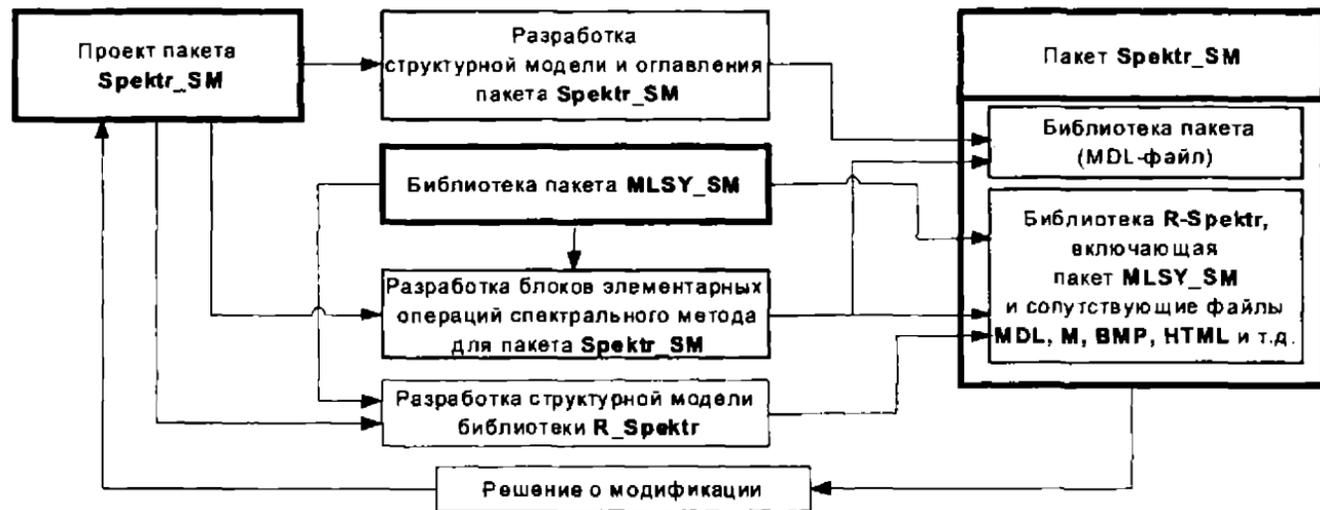


Рис. 1.9

но-дискретных систем управления *MLSY_SM*, рассмотренный в разд. 1.2, 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* [7,8]. На рис. 1.9 показан процесс разработки версии пакета *Spektr_SM*, который предполагает предварительную проработку проекта пакета *Spektr_SM*. 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*.

1.4. Разработка структурной модели и оглавления пакета расширения *Spektr_SM*

1.4.1. Версия структурной модели и оглавления пакета *Spektr_SM*

Механизм разработки структуры библиотеки блоков и оглавления пакета *Spektr_SM* отражает схема, расположенная на рис.1.10. 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 либо системы;
 - в окно библиотеки необходимо перенести блоки *Subsystem* (служащие подразделами будущей библиотеки) с необходимым уровнем вложенности, чтобы разработанная ранее структура библиотеки совпадала со структурой из открытого окна

Разработка структурной модели и оглавления пакета Spektr_SM



Рис. 1.10

Model Browser (изменяющаяся по мере заполнения библиотеки блоками Subsystem);

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

В процессе разработки проекта пакета Spektr_SM была разработана версия структурной модели и оглавления пакета Spektr_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.

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. 1.11 показано специальное окно с разделами основной библиотеки пакета Spektr_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.

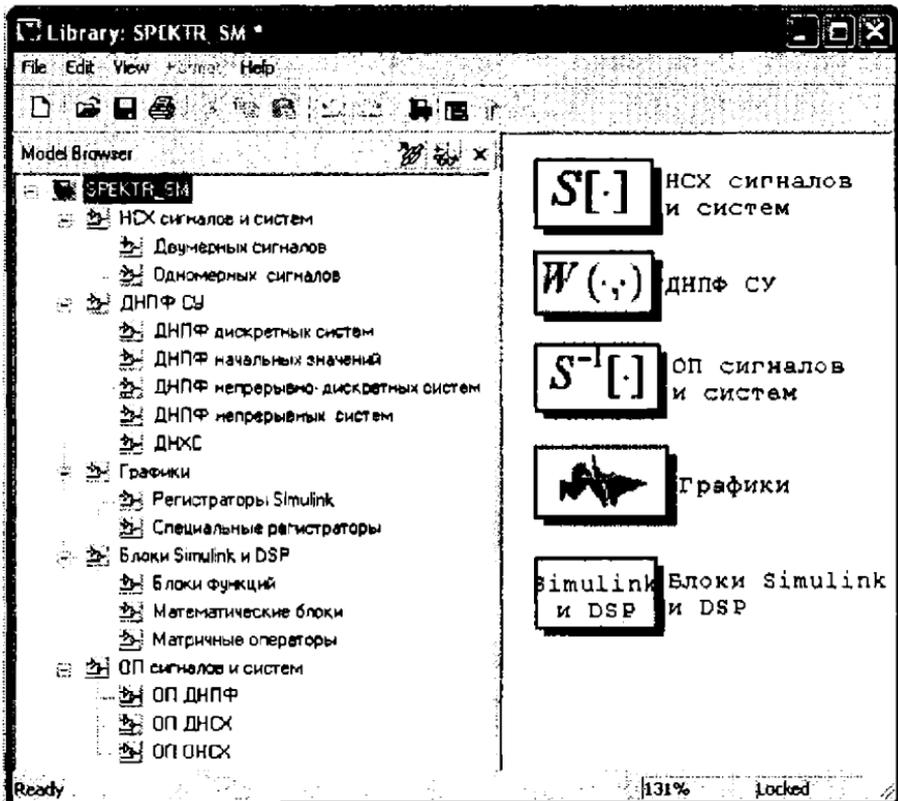


Рис. 1.11

1.4.2. Разработка пиктограмм разделов структурной модели и блоков

У каждого стандартного раздела библиотеки Simulink есть своя индивидуальная пиктограмма. Для создания пиктограмм разделов библиотеки пакета Spektr_SM, показанных на рис.1.11, надо открыть вкладку Icon окна редактора маски (Mask editor), которое показано на рис.1.12.

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:

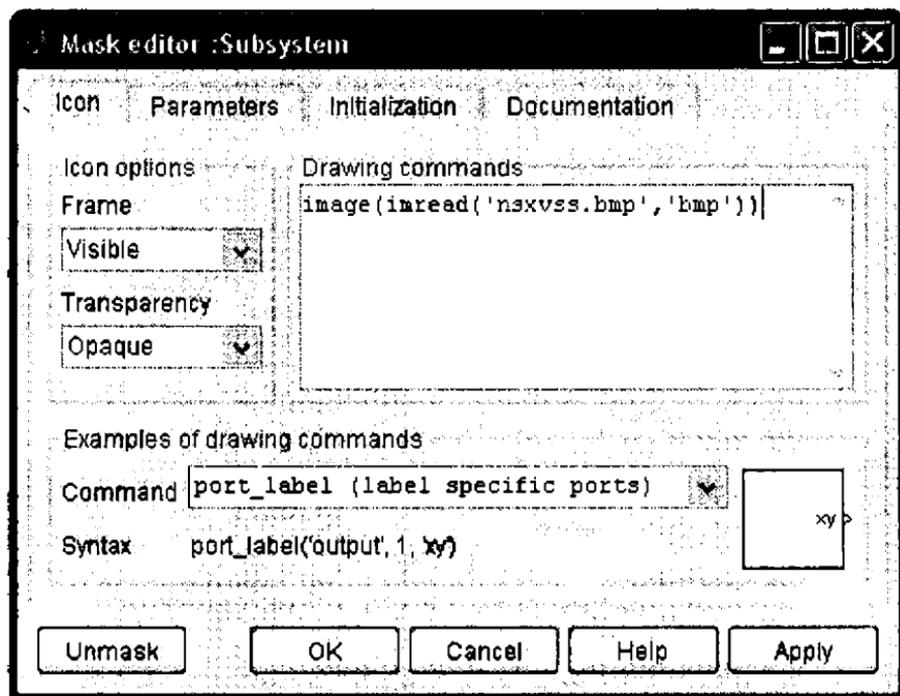


Рис. 1.12

— 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) разрабатывается версия формулы (рисунка);
- 2) данная формула набирается в текстовом редакторе Word при помощи редактора математических формул Microsoft Equation;
- 3) 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. 1.11 и рис. 1.12.

1.5. 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 `MSLY_SM` через связанные с ними файл-функции (M-файлы) и сопутствующими блокам BMP-файлами пиктограмм и HTML-файлами справочной системы. Размещение всех этих файлов требует создания структурированной библиотеки `R_Spektr` пакета `Spektr_SM` (см. рис.1.9), 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 -HCX одномерных сигналов;
OP — файлы обратного преобразования;
OB_DNPF — обращение ДНПФ;
OB_DNSX — обращение ДHCX;
OB_ONSX — обращение ОHCX;
MLSY_SM — файлы библиотеки спектральных алгоритмов
(файловую организацию которой см. разд. 1.2.2);
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 [8].

1.6. Разработка собственных блоков пакета расширения Spektr_SM

1.6.1. 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. 1.13.

Как видно из схемы, в конечном счёте версия блока реализуется маскированием 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 подсистем, m-файла и справочной документации.

Разработка версии подсистемы блока. Simulink обеспечивает создание моделей, внутри которых располагаются подсистемы (суб-модели). Внутри подсистем первого уровня, в свою очередь, могут располагаться подсистемы второго уровня и т.д. Это напоминает ситуацию, когда сложная система набирается из отдельных

Разработка блоков элементарных операций спектрального метода для пакета Spekr_SM

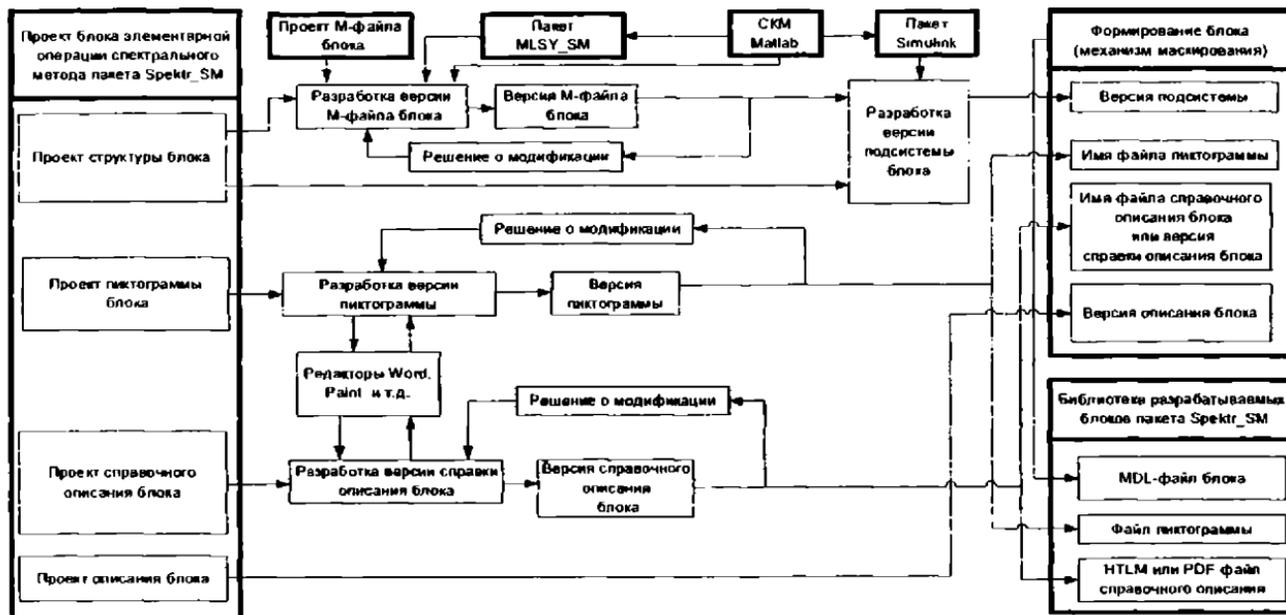


Рис. 1.13

систем — модулей, каждый из которых, в свою очередь, является системой.

Такой принцип конструирования сложных моделей имеет 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.

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 1.6. Надо создать подсистему, вычисляющую усеченную НСХ (1.2) входного непрерывного сигнала, заданного в виде аналитической функции относительно непрерывных базисных функций, идентификаторы которых приведены в приложении 1.1.

Решение задачи. Эта подсистема показана на рис. 1.14, где а — созданная подсистема (ее внутреннее представление), 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.

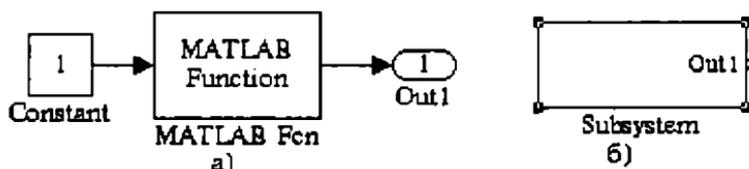


Рис. 1.14

Окно параметров этого блока, показанное на рис. 1.15, 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.

В данном примере в разделе **Parameters MATLAB functions** вводится имя *nsxp1* *m*-файла, который создан и отлажен для нахождения вектора-столбца НСХ. Он содержит функцию **nsxpnp(x1, N1,t,s1)**, которая зависит от параметров: **x1** — аналитической фун-

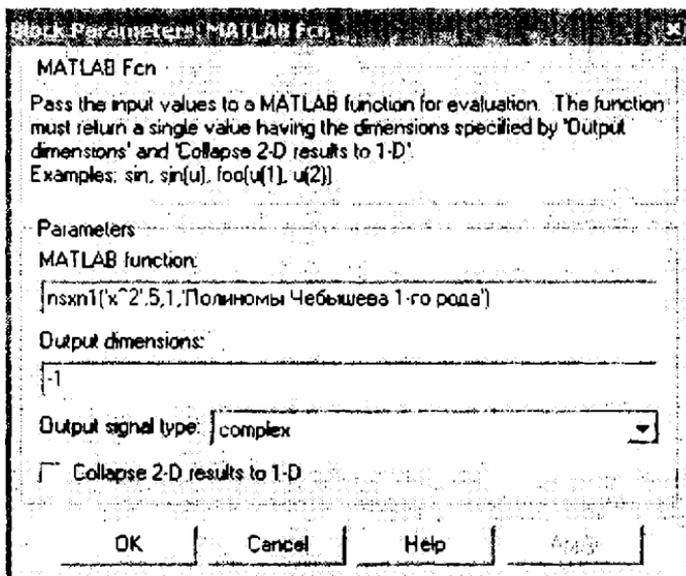


Рис. 1.15

кции, задающей входной сигнал; $N1$ — размера усечения матрицы-столбца HCX ; t — длины интервала работы системы управления; $s1$ — базиса, по которому вычисляется HCX . Эта функция имеет вид

```
function f1 = nsxnpp(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 перемен-

ной, получаемой m -функцией в качестве входного параметра. В случае совпадения с одним из значений выполняется операция вызова соответствующей функции (в нашем случае выполняется программа `snext1(f,N1,t)` — вычисление вектора-столбца HCX в базисе полиномов Чебышева первого рода по численной схеме (П2.29), которая приведена в примере 1.1, и присваивания её значения переменной $f1$.

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 математическим объектам.

Следовательно, в приведенном выше примере в качестве параметра 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; 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 1.7. Надо создать блок (маскированную подсистему) из подсистемы, рассмотренной в примере 1.6.

Решение задачи. Меняем название подсистемы (см. рис. 1.14) **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 которого показан на рис. 1.16). В нашем случае в окно **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 усе-чения НСХ; длина интервала работы системы; базисные функции. Для описания этих параметров открываем вкладку **Parameters** и заполняем окна **Dialog parameters** и **Options for selected parameter**.

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 \checkmark удаляем, так как параметр имеет символьное

значение; 5) в поле Tunable метку \checkmark 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 ввести «N — порядок усечения НСХ:», в поле Variable ввести имя соответствующей переменной «N1», а в поле Evaluate оставить метку \checkmark , так как в M-функцию передаётся параметр, имеющий численное значение. Задание параметра «t — длина 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.

Окно редактора маски с открытой вкладкой Parameters показано на рис. 1.16.

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. 1.17.

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

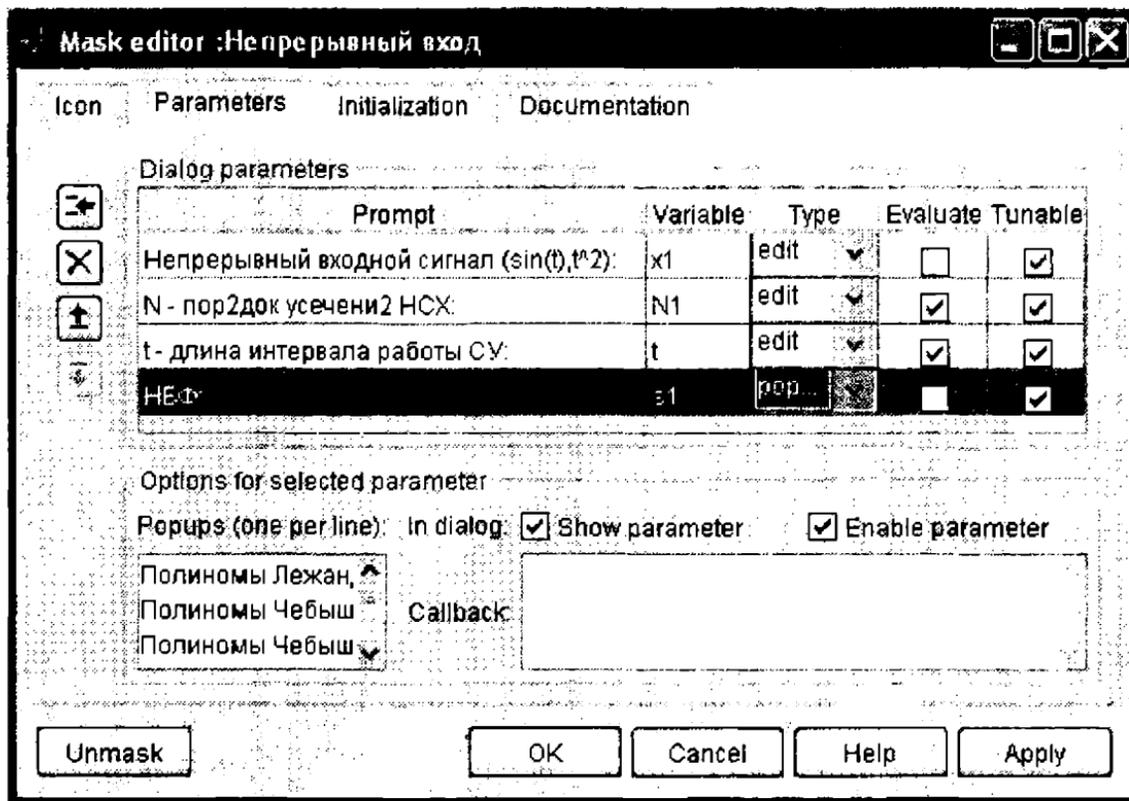


Рис. 1.16

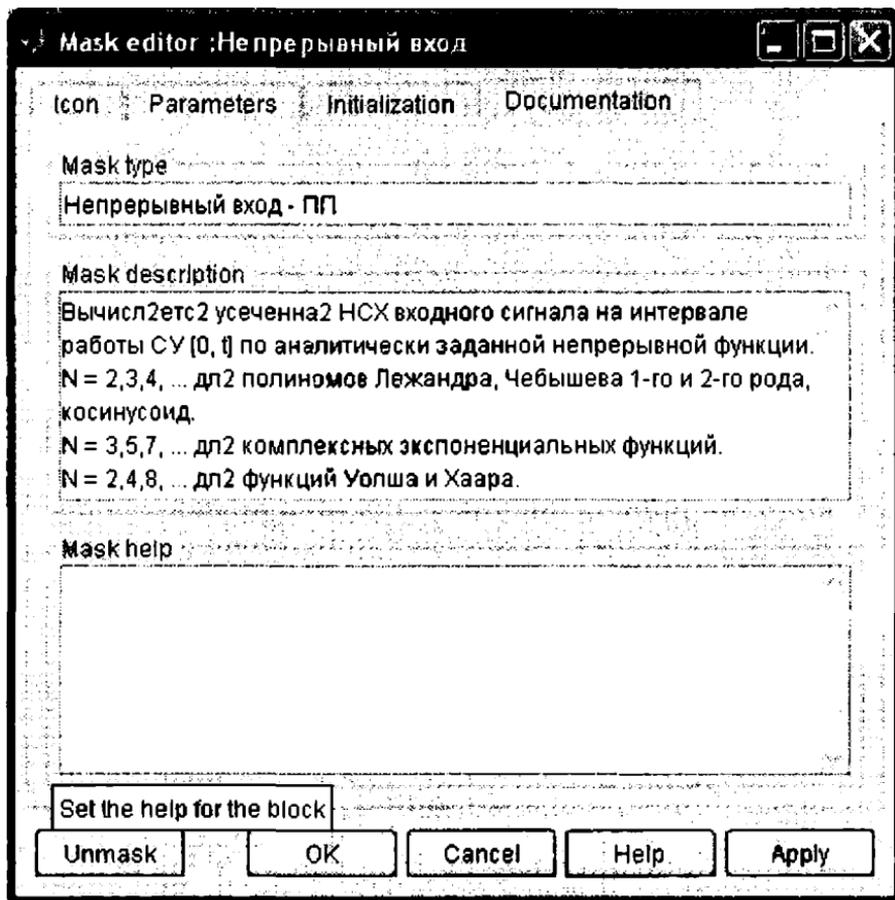


Рис. 1.17

шая 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`).

В результате выполнения всех этих действий получаем блок, внешний вид которого показан на рис. 1.18, а окно его параметров — на рис. 1.19.



Рис. 1.18

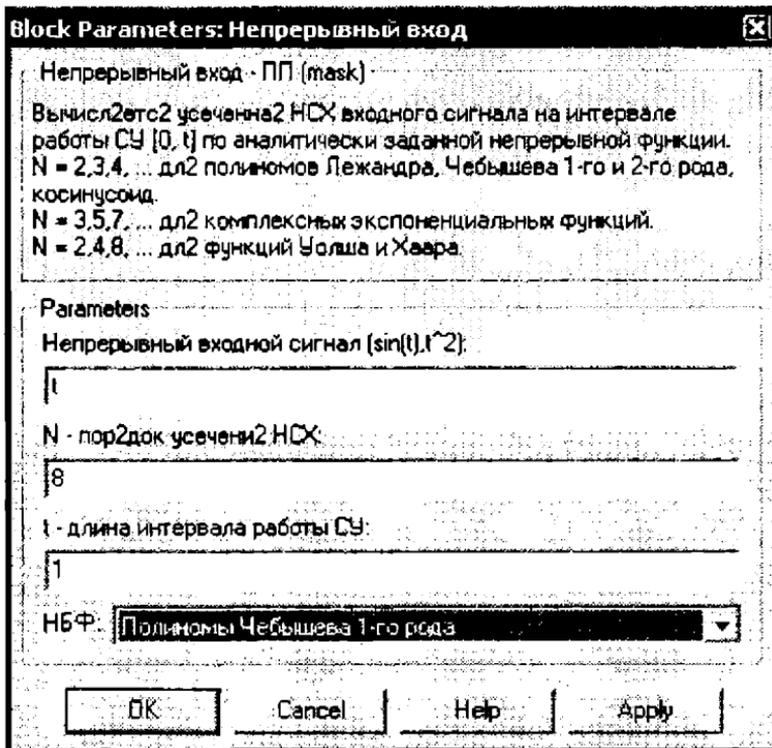


Рис. 1.19

1.6.2. Состав типовых блоков библиотеки *Spektr_SM*

Библиотека *Spektr_SM* разделена на пять разделов, окно браузера которой показано на рис. 1.11. Эти пять 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. 1.20 и рис. 1.21 и предназначены для формирования НСХ непрерывных и дискретных детерминированных сигналов (1.1)–(1.3), а также для формирования НСП непрерывных и дискретных случайных сигналов (1.7)–(1.8).

Блоки, входящие в раздел **ДНПФ СУ**, предназначены для задания ДНПФ непрерывных систем (1.11); ДНПФ дискретных систем (1.12); ДНПФ непрерывно-дискретных систем (1.13), (1.14);

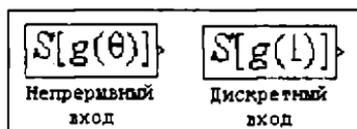


Рис. 1.20

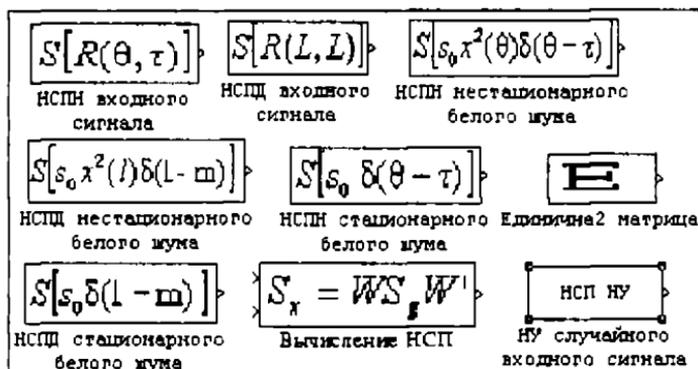


Рис. 1.21

ДНПФ начальных значений (1.21); ДНХС (1.39, 1.40) и показаны на рис. 1.22—1.26.

Блоки, входящие в раздел **ОП сигналов и систем**, предназначены для обращения НСХ одномерных непрерывных и дискретных детерминированных сигналов (1.6), т.е. для вычисления выходных сигналов (рис. 1.27); для обращения НСП (1.7), т.е. вычисления корреляционных функций непрерывных и дискретных случайных выходных сигналов, их дисперсий и средних квадратических значений (рис. 1.28); для обращения ДНПФ непрерывных, дискрет-

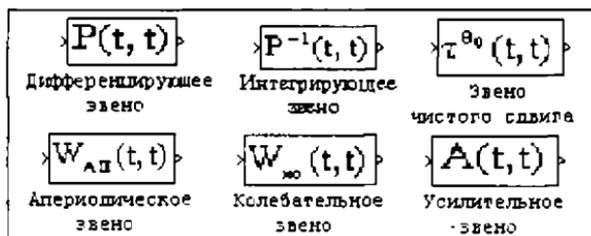


Рис. 1.22



Рис. 1.23

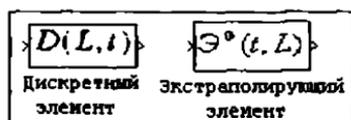


Рис. 1.24

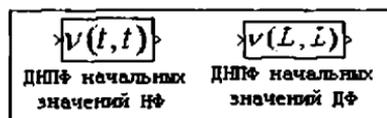


Рис. 1.25

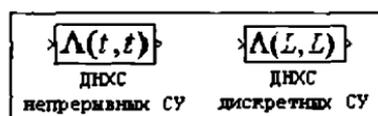


Рис. 1.26

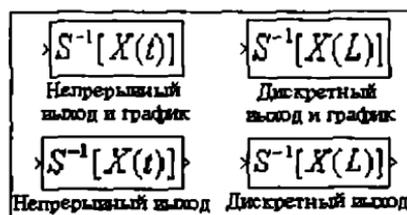


Рис. 1.27

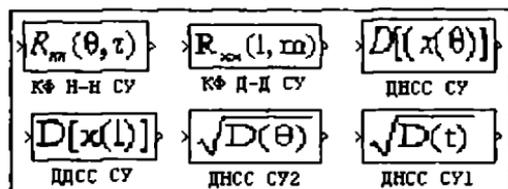


Рис. 1.28

ных и непрерывно-дискретных систем (1.38), т.е. вычисления ИПФ системы управления (рис. 1.29). Многие из этих блоков сразу осуществляют визуализацию сигналов.

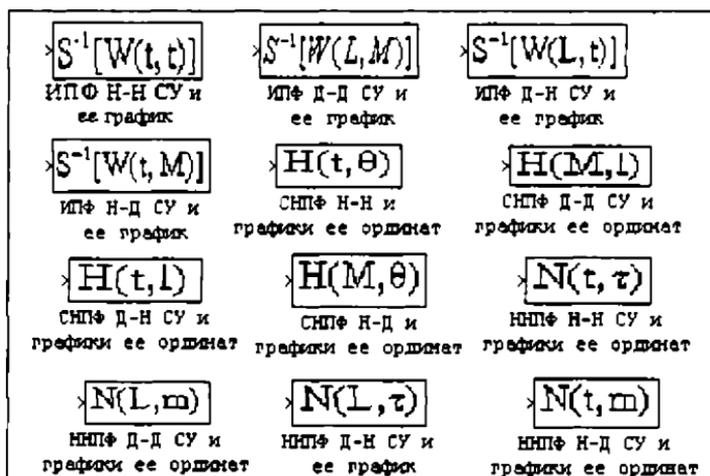


Рис. 1.29

Блоки, входящие в раздел **Графики**, предназначены для визуализации одномерных и двумерных функций (рис. 1.30).

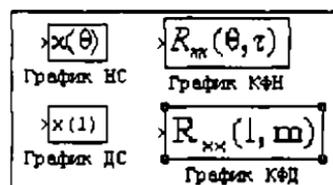


Рис. 1.30

1.6.3. Примеры анализа структуры и содержания типовых блоков пакета *Spektir_SM*

В предыдущих разделах была рассмотрена технология формирования блоков и состав блоков рассмотренной версии пакета *Spektir_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 бло-

ков 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 1.8. Среди блоков пакета *Spektr_SM* найти те блоки, подсистемы которых имеют структурную схему, показанную на рис. 1.31.

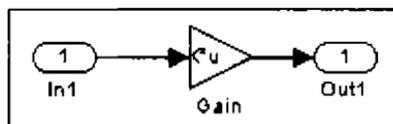


Рис. 1.31

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: SPEKTR_SM/.../ДНПФ непрерывных систем/Дифференцирующее звено**, в котором появится подсистема, изображенная на рис. 1.31. Проведя 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. 1.22–1.26. Кроме этих блоков такой же структурой обладают блоки раздела **ОП ОНСХ**, а именно блоки **Непрерывный выход** и **Дискретный выход** (см. рис. 1.27).

Пример 1.9. Сравнить блоки **Дифференцирующее звено** и **Интегрирующее звено**.

Как видно из примера 1.8, 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 **pfidne** и **pfine**, вызываемые блоком **Gain** подсистем этих блоков.

Для блока Gain Дифференцирующего звена m -функция 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

```
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 1.6). 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 (см. приложение 1), 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.

1.7. 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 находящиеся 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. 1.32.

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 проведенной 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

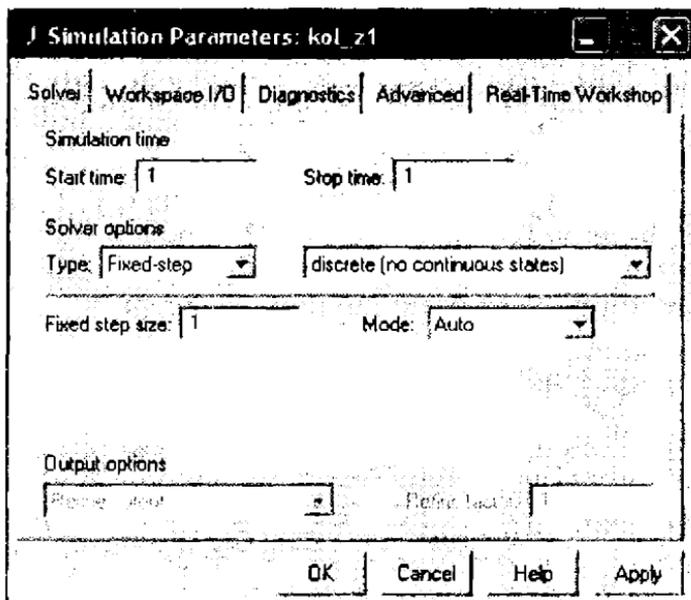


Рис. 1.32

сложные модели многомерные систем управления для анализа их спектральным методом формируются аналогично тому, как они формировались для анализа их спектральным методом при детерминированных воздействиях. Для вычисления корреляционных функций выходных сигналов по корреляционным функциям входных сигналов несколько иначе, чем при детерминированных воздействиях. Это связано с тем, что связи между НСП входных и выходных сигналов определяются выражением [2]

$$S_{x_\nu x_\nu} = \sum_{i=1}^n \sum_{k=1}^n W_{\mu i} S_{g_i g_k} W_{\nu k}^T, \quad \mu, \nu = 1, 2, \dots, m. \quad (1.46)$$

Для вычисления НСП выходных сигналов в случае отсутствия взаимной корреляции между входными сигналами выражение (1.46) преобразуется к виду

$$S_{x_i} = \sum_{k=1}^n W_{ik} S_{g_k} W_{ik}^T, \quad i = 1, 2, \dots, m. \quad (1.47)$$

Поэтому моделируем в Simulink выражение (1.47), которое требует моделирования подсистем, вычисляющих ДНПФ $W_{i,k}$ от k -го входа к i -му выходу и выполнения типовой операции

$$S_x = W \cdot S_g \cdot W^T. \quad (1.48)$$

Блок **Вычисление НСП** раздела **НСХ сигналов и систем** библиотеки пакета **Spektr_SM** реализует операцию (1.48). Структурная схема подсистемы этого блока показана на рис. 1.33.

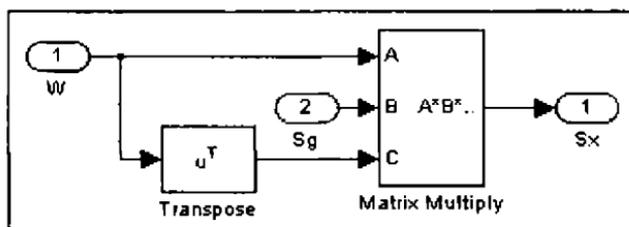


Рис. 1.33

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 (1.48). Таким образом, реализуется связь вход-выход по корреляционной функции при помощи средств Simulink.

ПРИМЕРЫ АНАЛИЗА НЕСТАЦИОНАРНЫХ ЛИНЕЙНЫХ СИСТЕМ УПРАВЛЕНИЯ СПЕКТРАЛЬНЫМ МЕТОДОМ И МЕТОДОМ МОМЕНТОВ В ПАКЕТАХ Spektr_SM и Simulink

2.1. 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 [1–4, 8,9] и методом моментов [5, 6], используя для этих целей пакет расширения системы MATLAB Simulink 4 [7] (или Simulink 5) и пакет его расширения Spektr_SM.

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

Упрощенная расчетная структурная схема системы управления в классе линеаризованных нестационарных систем представлена на рис. 2.1.

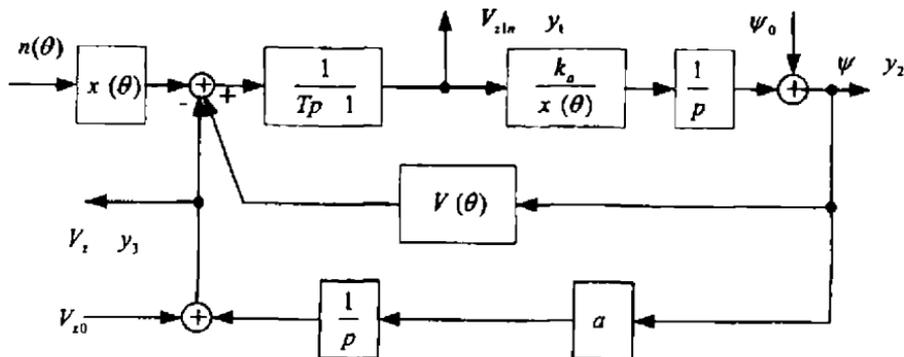


Рис. 2.1

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

$$Y' = AY + Bn, \quad (2.1)$$

где

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}; \quad A = \begin{bmatrix} -1/T & V_* / T & -1/T \\ k_a / x_* & 0 & 0 \\ 0 & a_* & 0 \end{bmatrix}; \quad B = \begin{bmatrix} x_* \\ 0 \\ 0 \end{bmatrix}.$$

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 ве-

роятности, 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: $R_{\eta}(\theta, \tau) = S_0 x_*(\theta) x_*(\tau) \delta(\theta - \tau)$, 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 $D_{\Delta V_z}(t_s)$ при допустимом значении угловой ориентации по углу рыскания $D_{\Delta \psi}(t_s)$, определяемой условием вертикализации посадочного аппарата. Условием эффективности функционирования системы являются 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_s :

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

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

2.1.2. 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. 2.1), используя библиотеку компонентов пакета Sпектр_СМ, окно браузера которой показано на рис. 1.11, составляется спектральная структурная схема перетаскиванием нужных компонент мышью в окно модели ВКМРL_РК (рис. 2.2). Эта модель предназначена для вычисления конечных значений дисперсии $D_{\Delta V_z}(t_s)$ в функции параметра k_a при учете всех воздействий на систему.

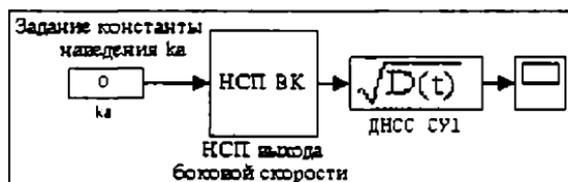


Рис. 2.2

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. 2.3. Задание константы наведения k_a обеспечивается блоком Clock (k_a), а вычисление среднеквадратического значения боковой скорости $\Delta V_z(t_s, k_a)$ в конечный момент вре-

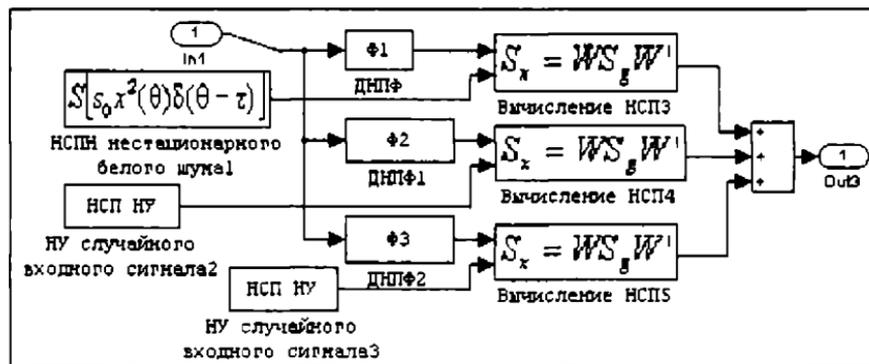


Рис. 2.3

мени 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 $\Delta V_z(t_s, k_a)$ в конечный момент времени в функции константы на-

ведения $k_a = 0.01, 0.2 \dots 10$ систему Simulink нужно настроить через меню редактирования Simulation (см. разд. 1.7). Для этого в окне *Simulation parameters: BMKPL_RK* на вкладке Solver необходимо задать параметр Start time равным 0.01, Stop time равным 10, Fixed step size равным 0.2.

Подсистемы вычисления ДНПФ $\Phi 1, \Phi 2, \Phi 3$ отражают структуру системы управления (см. рис. 2.1) и обеспечивают вычисление ДНПФ от входов $l(\theta), \Delta \psi_0, \Delta V_{z0}$ к выходу ΔV_z . На рис. 2.4 показана подсистема вычисления ДНПФ $\Phi 1$.

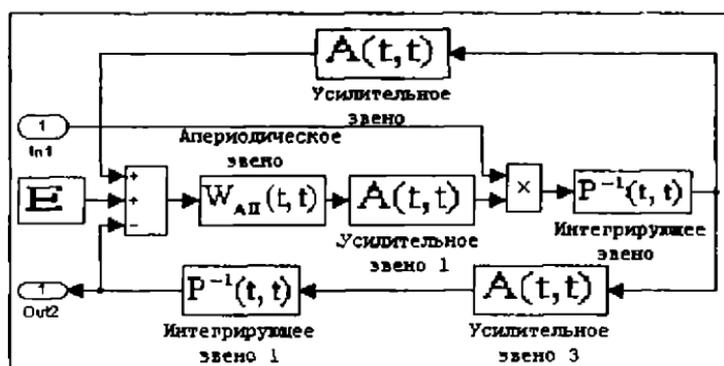


Рис. 2.4

2. Определяются среднеквадратические значения боковой скорости $\Delta V_z(t_s, k_a)$ в конечный момент времени в функции константы наведения k_a при учете всех воздействий на систему. График среднеквадратических значений боковой скорости $\Delta V_z(t_s, k_a)$ в

конечный момент времени в функции константы наведения k_a представлен на рис. 2.5. По этому графику определяется оптимальное значение $k_{a \text{ опт}} = 4$.

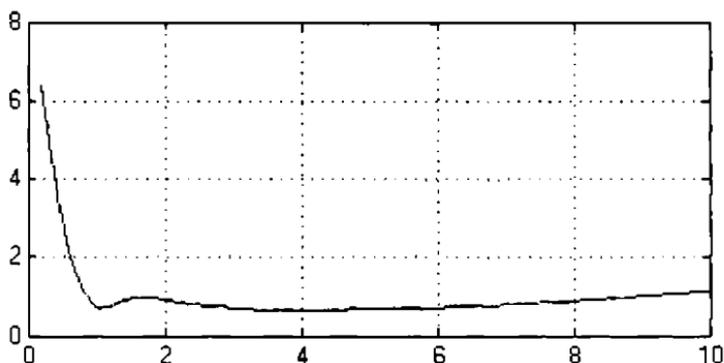


Рис. 2.5

3. По заданной структурной схеме (см. рис. 2.1), используя библиотеку компонентов пакета `SpektR_SM`, составляется спектральная структурная схема перетаскиванием нужных компонент мышью в окно модели `BKMPL_R` (рис. 2.6). Эта модель предназначена для вычисления переходных процессов по $\Delta V_z(\theta)$ и $\Delta \psi(\theta)$ при

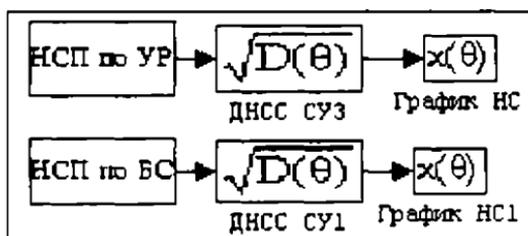


Рис. 2.6

учете всех воздействий на систему. Структурная схема подсистемы **НСП по УР** показана на рис. 2.7. Структурная схема подсистемы **НСП по БС** совпадает с структурной схемой подсистемы **НСП по УР**, но отличается именами пиктограмм трех своих подсистем. Надо заменить имя пиктограммы $\Phi 1$ на $\Phi 4$, $\Phi 2$ на $\Phi 5$, $\Phi 3$ на $\Phi 6$.

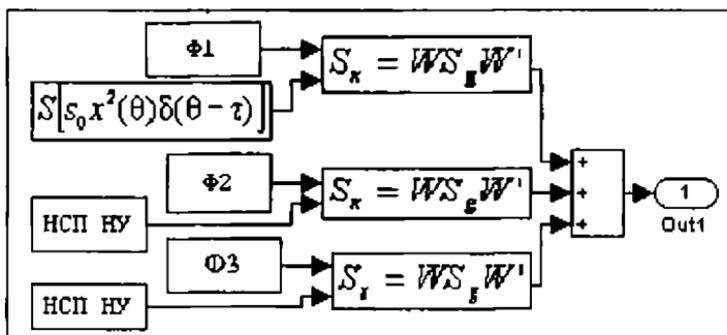


Рис. 2.7

Подсистемы вычисления ДНПФ $\Phi 1$, $\Phi 2$, $\Phi 3$, $\Phi 4$, $\Phi 5$, $\Phi 6$ отражают структуру системы управления (см. рис. 2.1) и обеспечивают вычисление ДНПФ от входов $m(\theta)$, $\Delta\psi_0$, ΔV_{z0} к выходам $\Delta V_z(\theta)$ и $\Delta\psi(\theta)$. На рис. 2.8 показана подсистема вычисления ДНПФ $\Phi 1$.

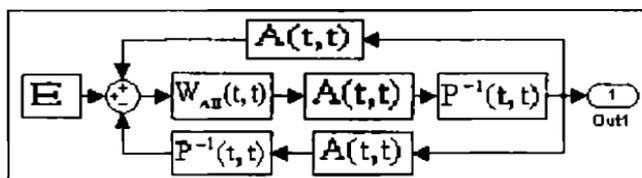


Рис. 2.8

4. Для оптимального значения $k_a = 4$ определяются 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. 2.9,а и рис. 2.9,б.

Исходя из условий эффективности функционирования бокового канала они признаются удовлетворительными.

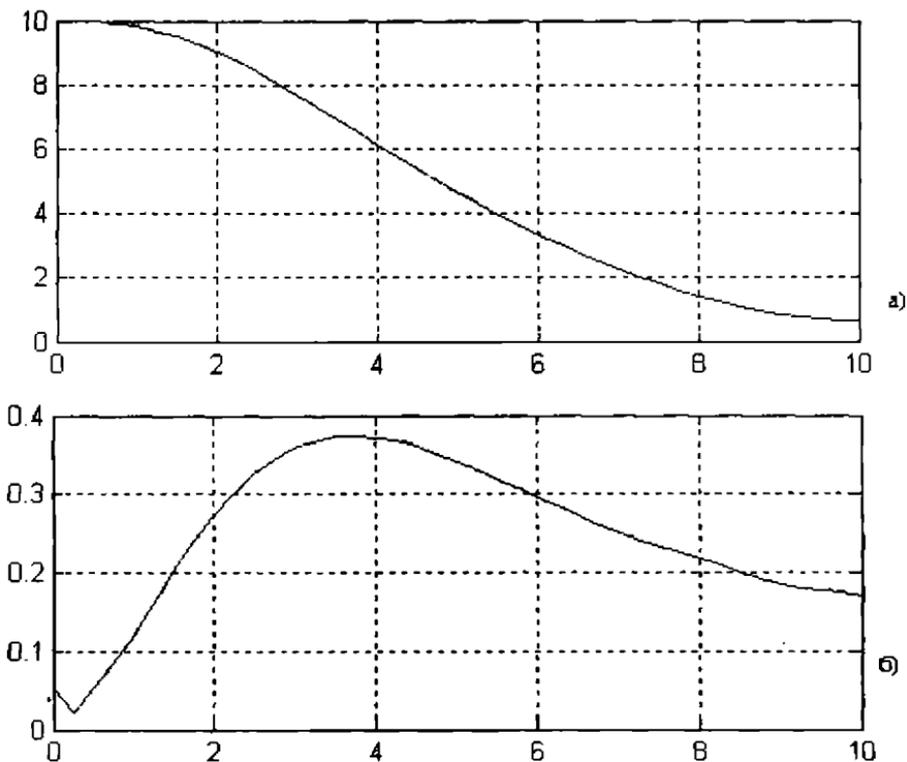


Рис. 2.9

2.1.3. Применение метода моментов для анализа системы управления в системе математического визуального моделирования Simulink

Суть метода моментов [5, 6] заключается в составлении дифференциальных уравнений относительно вероятностных моментов вектора состояния системы и интегрирования их при заданных начальных условиях. Этот метод позволяет получить точные уравнения относительно вероятностных моментов для нестационарных линейных систем. Практически наиболее важно определить первые два вероятностных момента вектора математического ожидания и матрицы корреляционных моментов вектора состояния системы, что для гауссова вектора входного сигнала и линейной сис-

темы полностью решает задачу определения однородной функции распределения.

Если система задана уравнением (2.1), то закон изменения математического ожидания вектора состояния имеет вид

$$\dot{m}_x(\theta) = A(\theta)m_x(\theta) + B(\theta)m_g(\theta), \quad m_x(\theta_0) = m_0. \quad (2.2)$$

Закон изменения ковариационной матрицы вектора состояния $R(\theta) = M\{Y(\theta) \cdot Y^T(\theta)\}$ имеет вид

$$\dot{R}_x(\theta) = A(\theta)R_x(\theta) + R_x(\theta)A^T(\theta) + B(\theta)S_0(\theta)B^T(\theta), \quad R_x(\theta_0) = R_0. \quad (2.3)$$

Для нашей задачи

$$R_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.0002 & 0 \\ 0 & 0 & 100 \end{bmatrix}.$$

Расчет искомой системы управления методом моментов включает в себя следующие этапы:

1. По заданной математической модели (2.1) и системе уравнений моментов (2.3), используя библиотеку компонентов пакета Simulink, составляется расчетная структурная схема перетаскиванием нужных компонентов мышью в окно модели **ВОК_KAN_MM** (рис. 2.10).

Подсистемы этой системы **ФОР**, **ФКМ**, **ФСКЗ** показаны соответственно на рис. 2.11, 2.12, 2.13.

Эта модель предназначена для вычисления средних квадратичных значений боковой скорости $\Delta V_z(\theta)$ и угла рыскания $\Delta\psi(\theta)$ при оптимальном значении $k_a = 4$ (найденного спектральным методом) и учете всех воздействий на систему.

2. Для оптимального значения $k_a = 4$ определяются средние квадратичные значения боковой скорости $\Delta V_z(\theta)$ и угла рыскания

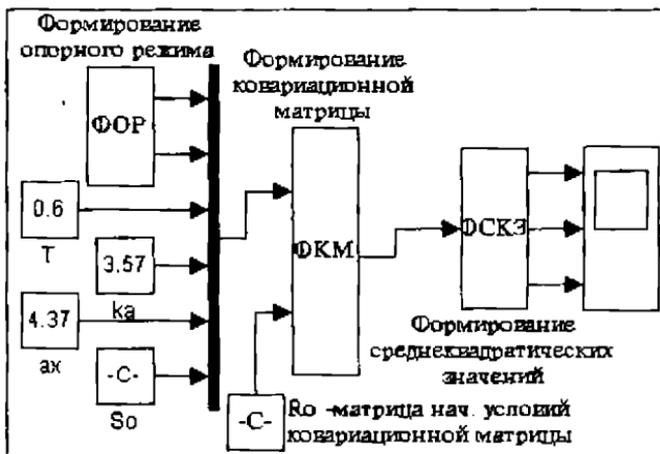


Рис. 2.10

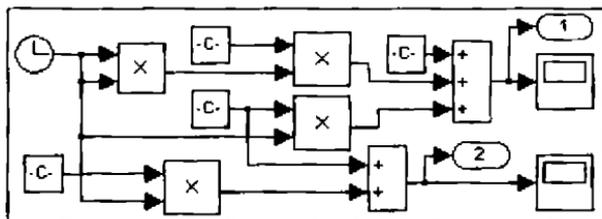


Рис. 2.11

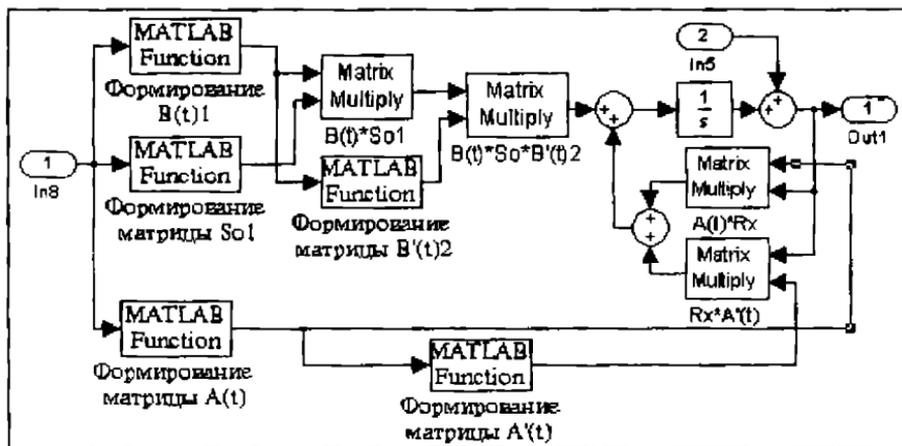


Рис. 2.12

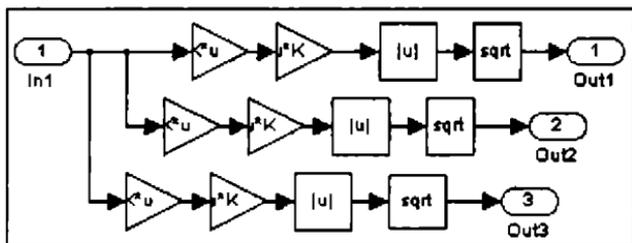


Рис. 2.13

$\Delta\psi(\theta)$ при учете всех воздействий на систему. График средних квадратичных значений боковой скорости $\Delta V_z(\theta)$ и угла рыскания $\Delta\psi(\theta)$ представлен на рис. 2.14,а и рис. 2.14,б.

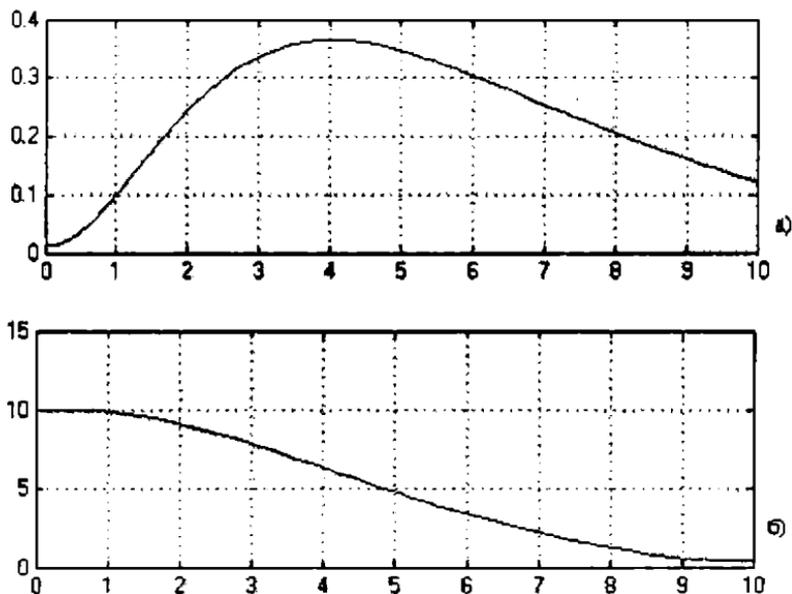


Рис. 2.14

Результаты расчета методом моментов (рис. 2.14) и спектральным методом (см. рис. 2.9) совпадают.

2.2. Пример анализа продольного канала системы управления ЛА посадки на Марс

В учебном пособии [10] рассматривается линеаризованная модель продольного канала система управления мягкой посадки ЛА на Марс.

Здесь проведем анализ продольного канала система управления мягкой посадки ЛА на Марс спектральным методом [1—4, 8,9] и методом моментов [5, 6], используя для этих целей пакет расширения системы MATLAB Simulink 4 (или Simulink 5) и пакет его расширения Spektr_SM.

2.2.1. Математическая модель продольного канала управления и задачи ее анализа

Расчетная структурная схема системы управления в классе линеаризованных нестационарных систем представлена на рис. 2.15.

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

$$Y' = AY + Bu, \quad (2.4)$$

где $Y = [y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7 \ y_8]^T$;

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & V_*(\theta) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x_*(\theta) & 0 & 0 \end{bmatrix}^T;$$

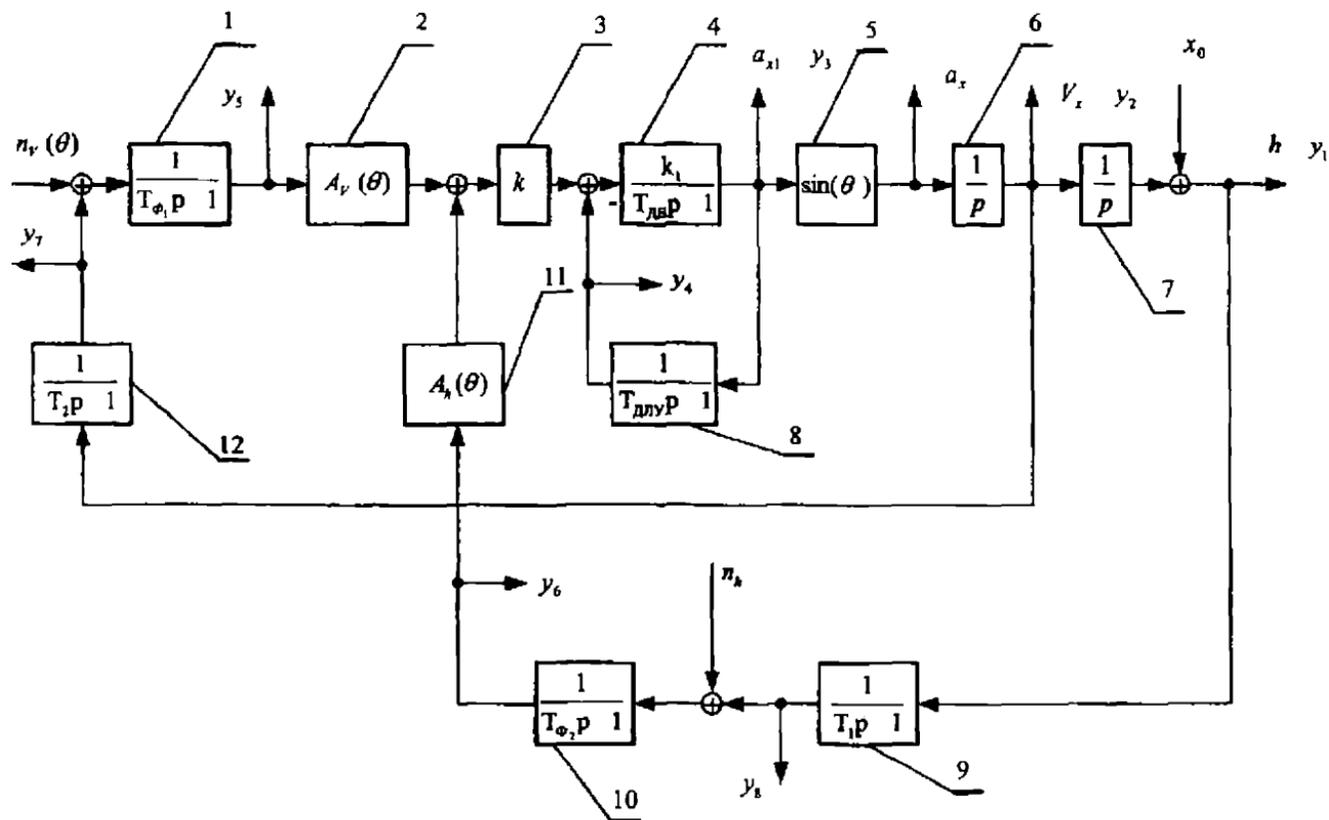


Рис. 2.15

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin(\theta_*) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T_{ДВ}} & -\frac{k_1}{T_{ДВ}} & \frac{k_1 k_{\Pi} A_{V_x}(\theta)}{T_{ДВ}} & \frac{k_1 k_{\Pi} A_h(\theta)}{T_{ДВ}} & 0 & 0 \\ 0 & 0 & \frac{1}{T_{ДЛУ}} & -\frac{1}{T_{ДЛУ}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{T_{\Phi_1}} & 0 & \frac{1}{T_{\Phi_1}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\Phi_2}} & 0 & \frac{1}{T_{\Phi_2}} \\ 0 & \frac{1}{T_2} & 0 & 0 & 0 & 0 & -\frac{1}{T_2} & 0 \\ \frac{1}{T_1} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_1} \end{bmatrix}$$

Система управления реализует метод гравитационного разворота. Управление осуществляется блоками 1, 2, 3, 10, 11. Блоки 1 и 10 реализуют алгоритмы фильтрации по указанным операторным уравнениям. Блоки 2 и 11 реализуют алгоритм управления требуемой скоростью вдоль продольной оси. Поэтому

$$A_h(\theta) = 2g_m(\tau_*^2 - 1)/(V_*(\sin^2(\theta_*) - 2\tau_* \sin(\theta_*) + 1)); \quad (2.5)$$

$$A_{V_x}(\theta) = \frac{(V_*^2 - V_{x1k}^2) \cos^2(\theta_*) (\sin^2(\theta_*) - \tau_*)}{(V_*^2 (\sin^2(\theta_*) - 2\tau_* \sin(\theta_*) + 1))} - 1, \quad (2.6)$$

где τ_* — опорные перегрузки; g_m — ускорение свободного падения; V_* — скорость на программной траектории; V_{x1k} — требуемое конечное значение скорости. Для вертикального спуска выражения (2.5) и (2.6) преобразуются к виду

$$A_h(\theta) = g_m(\tau_* - 1)/V_*; \quad A_{V_x}(\theta) = -1. \quad (2.7)$$

Блок 3 является усилителем сигнала управления (константа навигации), а константа k_n подлежит выбору. Блоки 9 и 12 описывают измерители высоты и скорости. Ошибки измерения учитываются помехами n_{V_x} и n_h , которые задаются как нестационарные белые шумы с равными нулю математическими ожиданиями и с заданными корреляционными функциями:

$$\begin{aligned} R_{n_{V_x} n_{V_x}}(\theta, \tau) &= S_{n_{V_x}} V_*(\theta) V_*(\tau) \delta(\theta - \tau); \\ R_{n_h n_h}(\theta, \tau) &= S_{n_h} x_*(\theta) x_*(\tau) \delta(\theta - \tau), \end{aligned} \quad (2.8)$$

где V_* , x_* — параметры номинальной траектории, определяемые уравнениями опорного режима

$$V_* = V_{*0} + (a_* - g_M)\theta; \quad x_* = x_{*0} + V_{*0}\theta + 0.5(a_* - g_M)\theta^2. \quad (2.9)$$

Блоки 4 и 8 описывают процессы, протекающие в двигательной установке и в датчике линейных ускорений, который обеспечивает стабилизацию продольных перегрузок. Блоки 5, 6 и 7 описывают динамику движения летательного аппарата на траектории посадки. Кроме того, в схеме учтена начальная ошибка прицеливания как случайное начальное условие своим среднеквадратическим значением $\overline{\Delta x_0^2}$.

Выходными координатами системы являются наклонная дальность спускаемого аппарата $\Delta h(\theta)$, скорость ее изменения $\Delta V_x(\theta)$ и ее ускорение $\Delta a_x(\theta)$.

Примем следующие числовые значения параметров системы, начальных условий и помех:

$$\begin{aligned} T_{ДВ} = 0.1 \text{ с}; \quad T_{ДЛУ} = 0.7 \text{ с}; \quad T_1 = T_2 = 1 \text{ с}; \quad T_{\phi_1} = T_{\phi_2} = 1 \text{ с}; \quad k_1 = 1; \\ S_{n_{V_x}} = 0.000025 \text{ 1/с}; \quad S_{n_h} = 0.00003 \text{ 1/с}; \quad \overline{\Delta x_0^2} = 100 \text{ м}^2. \end{aligned} \quad (2.10)$$

Для получения количественных результатов примем следующие числовые значения параметров движения:

$$\begin{aligned} \theta_* &= -\frac{\pi}{2}; g_m = 3.72 \text{ м/с}^2; x_0 = 2827 \text{ м}; V_{x_0} = -229 \text{ м/с}; \\ x_k &= 30 \text{ м}; V_{x_k} = -3 \text{ м/с}; a_* = 13.093 \text{ м/с}^2, \end{aligned} \quad (2.11)$$

которые определяют опорную траекторию гравитационного разворота.

При таких численных данных время управляемого движения получится равным 24 с.

Задачу расчета продольного канала наведения сформулируем следующим образом: проанализировать влияние параметра продольного канала наведения k_n на точность и дисперсию характеристической скорости этого канала управления; выбрать параметр продольного канала наведения k_n из условия минимума среднего квадрата ошибки управления высотой в конечный момент времени

$\overline{\Delta h^2}(t_k) = \min$ при допустимом значении дисперсии вертикальной скорости в конечный момент времени $\overline{\Delta V_x^2}(t_k)$ (равной дисперсии характеристической скорости продольного канала D_{V_x}); проанализировать статистические характеристики переходных процессов в продольном канале по ускорению Δa_{x1} , ошибкам вертикальной скорости ΔV_x и высоты Δh при оптимальном параметре k_n .

2.2.2. 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 (рис. 2.15), используя библиотеку компонентов пакета Spektr_SM, составляется спектральная структурная схема перетаскиванием нужных компонентов мы-

шью в окно модели APKYPLASM_NN_K (рис. 2.16). Каждый компонент (блок) настраивается по заданным параметрам исходной задачи. Эта модель предназначена для вычисления среднеквадратического значения суммарной ошибки по высоте в функции параметра k_{II} при учете всех воздействий на систему.

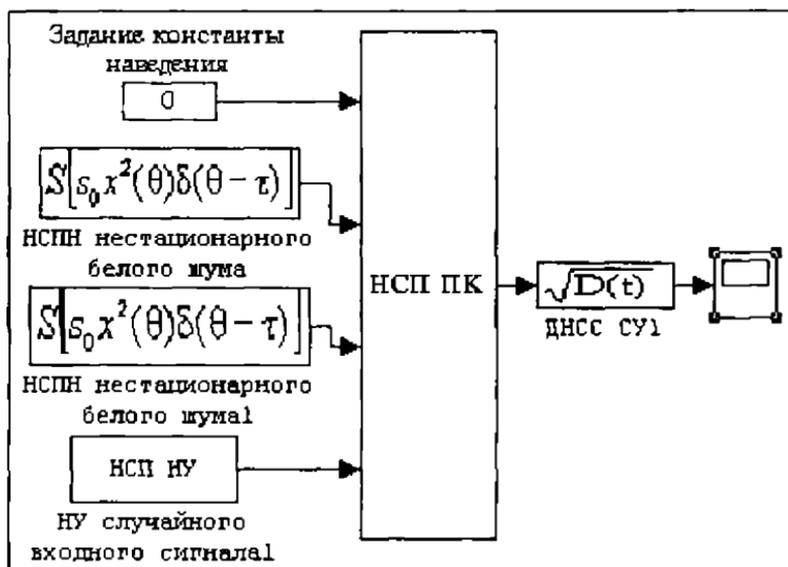


Рис. 2.16

Подсистема НСП ПК (блок Subsystem из раздела Ports&Subsystems библиотеки Simulink) обеспечивает вычисление нестационарной спектральной плотности (НСП) выходного сигнала Δh и имеет структуру, показанную на рис. 2.17. Задание константы наведения k_{II} обеспечивается блоком Clock, а вычисление среднеквадратического значения $\Delta h(t_s, k_{II})$ в конечный момент времени в функции константы наведения k_{II} типовым блоком ДНСС СУ пакета расширения Spectr_SM. Для визуализации вычисленных данных используется гиповой блок осциллографа Scope (источник текущего времени). Так как все спектральные характеристики вы-

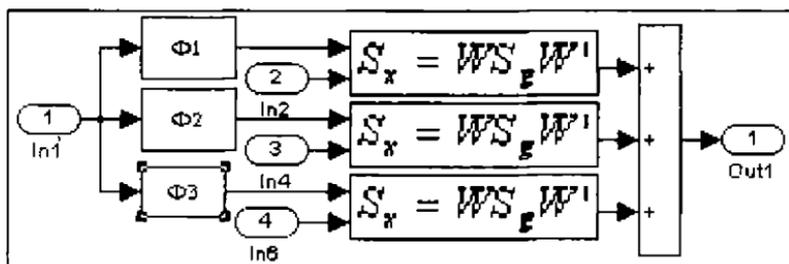


Рис. 2.17

числяются в пакете расширения *Spektr_SM* за один такт, то для вычисления среднеквадратического значения $\Delta h(t_s, k_{II})$ в конечный момент времени в функции константы наведения $k_{II} = 0.0001, 0.01 \dots 1$ систему Simulink нужно настроить через меню редактирования *Simulation* (см. разд. 1.7). В окне *Simulation parameters: APKYPLASM_NN_K* на вкладке *Solver* необходимо задать параметры *Start time* равным 0.0001, *Stop time* равным 1, *Fixed step size* равным 0.01.

Подсистемы вычисления ДНПФ $\Phi 1, \Phi 2, \Phi 3$ отражают структуру системы управления (см. рис. 2.15) и обеспечивают вычисление ДНПФ от входов $n_{V_x}(\theta), n_h, \overline{\Delta x_0^2}$ к выходу.

2. Определяем средние квадратичные значения промаха $\Delta h(t_k, k_{II})$ в конечный момент времени в функции константы наведения k_{II} при учете всех воздействий на систему. График средних квадратичных значений промаха $\Delta h(t_k, k_{II})$ в конечный момент времени в функции константы наведения k_{II} представлен на рис. 2.18. По этому графику определяется оптимальное значение $k_{II \text{ опт}} = 0.12$.

3. По заданной структурной схеме (см. рис. 2.15), используя библиотеку компонентов пакета *Spektr_SM*, составляется спектральная структурная схема перегаскиванием нужных компонентов

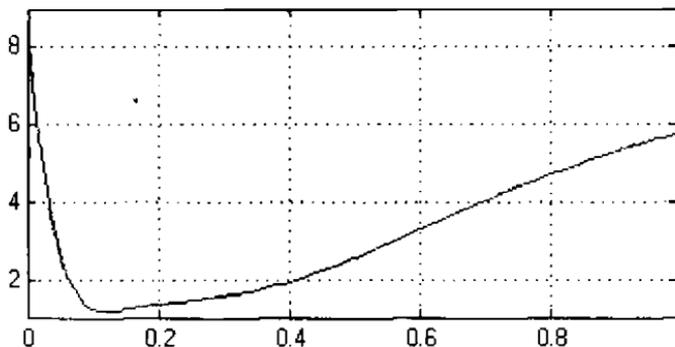


Рис. 2.18

мышью в окно модели `APKYLASM_NN_3` (рис. 2.19). Эта модель предназначена для вычисления переходных процессов по $\Delta h(\theta)$, $\Delta V_x(\theta)$ и $\Delta a_x(\theta)$ при учете всех воздействий на систему. Подсистема этой системы **НСН по В** показана на рис. 2.20.

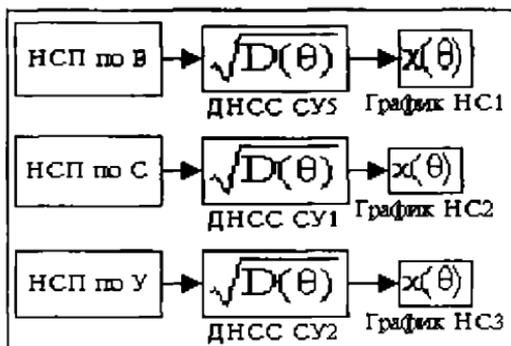


Рис. 2.19

Структурные схемы подсистем **НСП по С** и **НСП по У** совпадают со структурной схемой подсистемы **НСП по В**, но отличаются именами пиктограмм трех своих подсистем. Для их получения надо заменить имя пиктограммы $\Phi 1$ на $\Phi 4$ или $\Phi 7$, $\Phi 2$ на $\Phi 5$ или $\Phi 8$, $\Phi 3$ на $\Phi 6$ или $\Phi 9$.

Подсистемы вычисления ДНЛФ $\Phi 1 - \Phi 9$ (расчетная схема для $\Phi 3$ показана на рис. 2.21) отражают структуру системы управле-

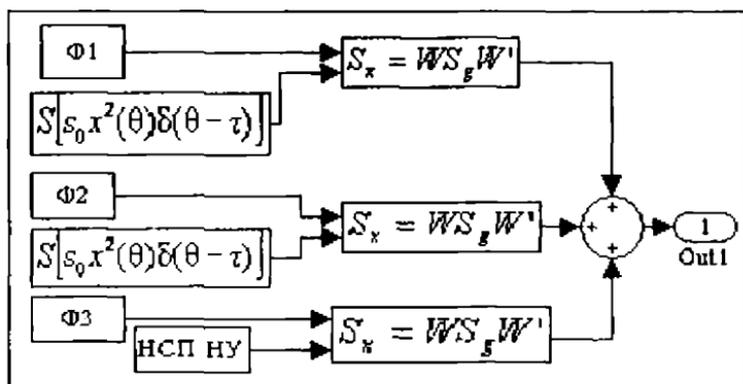


Рис. 2.20

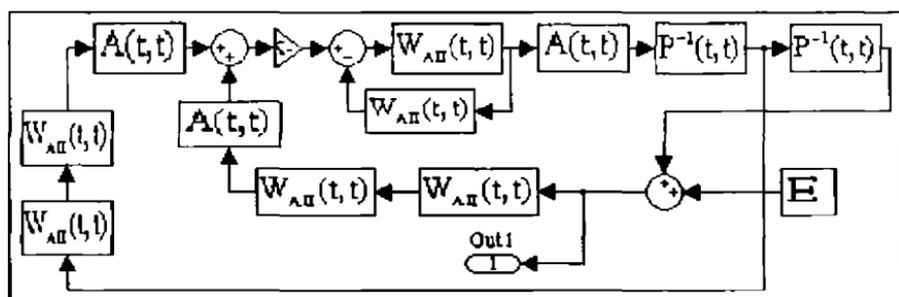


Рис. 2.21

ния (см. рис. 2.15) и обеспечивают вычисление ДНПФ от входов $n_{\psi}(\theta)$, $n_h(\theta)$, Δx_0 к выходам $\Delta h(\theta)$, $\Delta V_x(\theta)$ и $\Delta a_x(\theta)$.

4. Для оптимального значения $k_{II} = 0.12$ определяем средние квадратичные значения $\Delta h(\theta)$, $\Delta V_x(\theta)$ и $\Delta a_x(\theta)$ при учете всех воздействий на систему. Графики средних квадратичных значений $\Delta h(\theta)$, $\Delta V_x(\theta)$ и $\Delta a_x(\theta)$ представлены на рис. 2.22,а, рис. 2.22,б и рис. 2.22,в.

Исходя из условий эффективности функционирования бокового канала они признаются удовлетворительными.

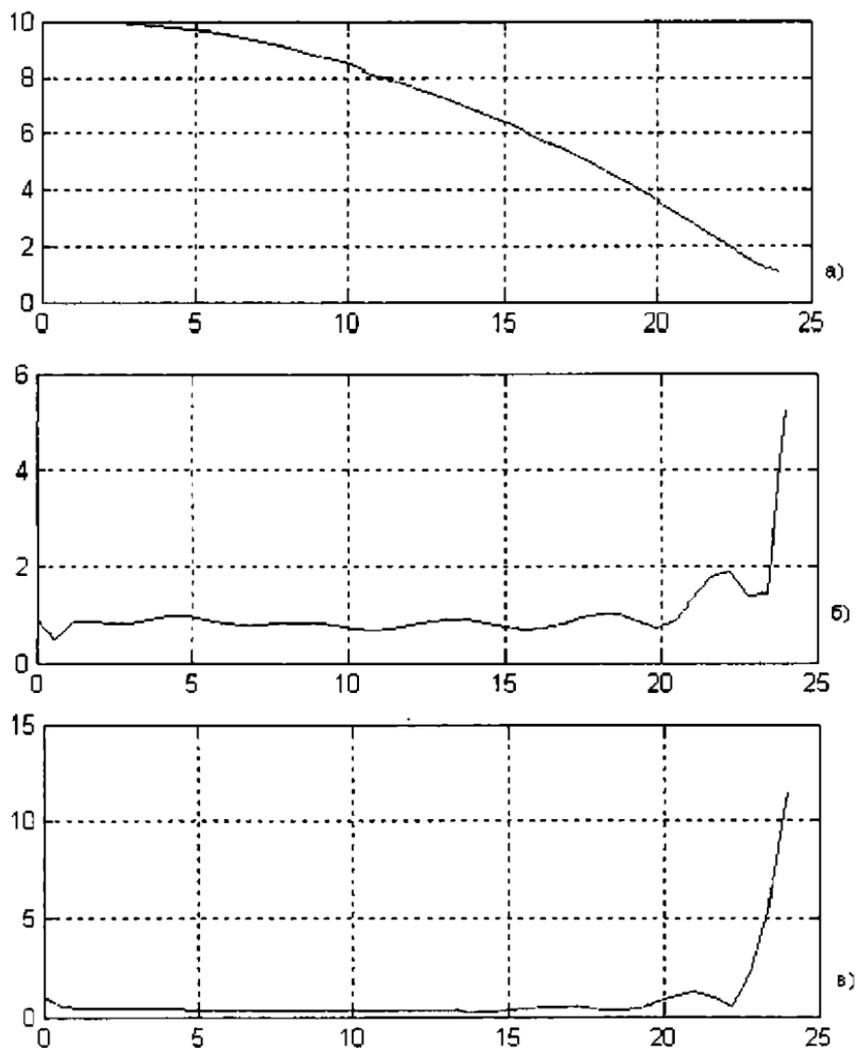


Рис. 2.22

2.2.3. Применение метода моментов для анализа системы управления в системе математического визуального моделирования Simulink

Система управления мягкой посадки ЛА на Марс описывается уравнением (2.4), а закон изменения математического ожидания и

ковариационной матрицы вектора состояния уравнениями (2.2) и (2.3). Для нашей задачи в уравнении (2.3) надо положить

$$R_0 = 100 \|c_{i,j}\|, \text{ где } c_{i,j} = \begin{cases} 1 & \text{при } i=j=0; \\ 0 & \text{в остальных случаях.} \end{cases}$$

Расчет искомой системы управления методом моментов включает в себя следующие этапы:

1. По заданной математической модели (2.4) и системе уравнений моментов (2.3), используя библиотеку компонентов пакета Simulink, составляется расчетная структурная схема перетаскиванием нужных компонентов мышью в окно модели ARKYPLASM_NN_MM (рис. 2.23).

Подсистемы этой системы ФОР, ФКМ, ФСКЗ показаны соответственно на рис. 2.11—2.13.

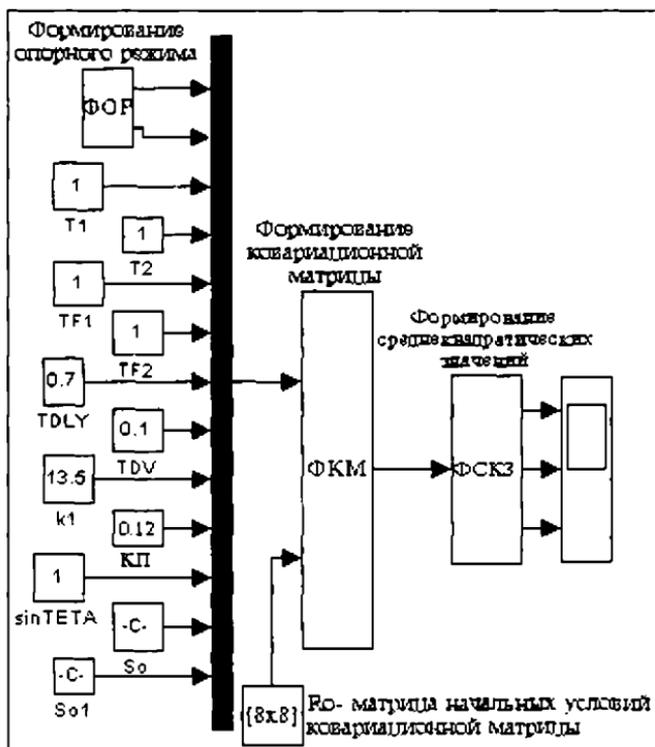


Рис. 2.23

Эта модель предназначена для вычисления средних квадратичных значений от $\Delta h(\theta)$, $\Delta V_x(\theta)$ и $\Delta a_x(\theta)$ при оптимальном значении $k_{\Pi} = 0.12$ (найденного спектральным методом) и учете всех воздействий на систему.

2. Для оптимального значения $k_{\Pi} = 0.12$ определяем средние квадратичные значения $\Delta h(\theta)$, $\Delta V_x(\theta)$ и $\Delta a_x(\theta)$ при учете всех воздействий на систему. Графики средних квадратичных значений $\Delta h(\theta)$, $\Delta V_x(\theta)$ и $\Delta a_x(\theta)$ представлены на рис. 2.24,а, 2.24,б, 2.24,в соответственно.

Результаты расчета методом моментов (рис. 2.24) и спектральным методом (см. рис. 2.22) совпадают.

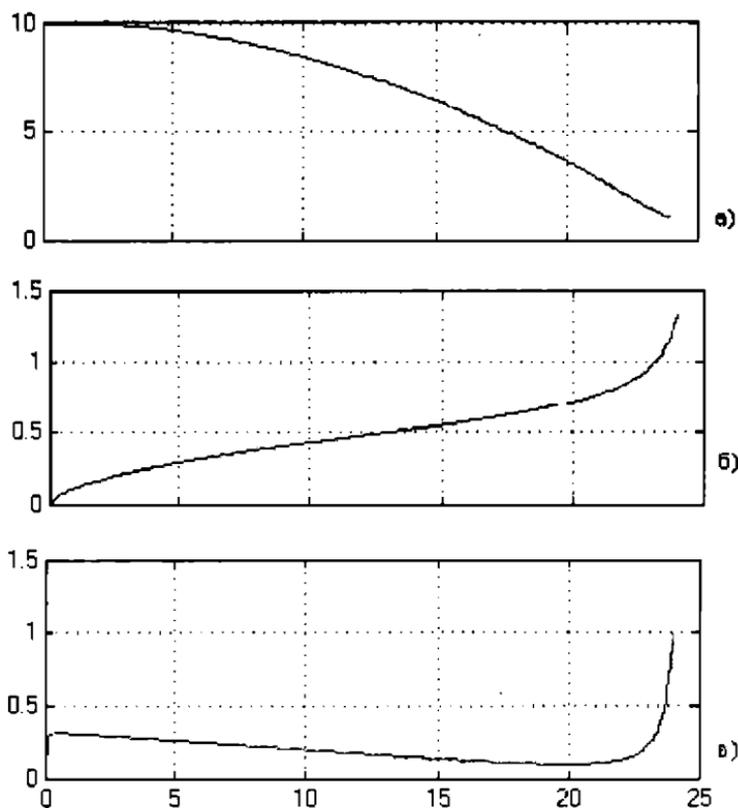


Рис. 2.24

Приложение 1

ОПИСАНИЕ ИДЕНТИФИКАТОРОВ И ПРОЦЕДУР ЭЛЕМЕНТАРНЫХ ОПЕРАЦИЙ СПЕКТРАЛЬНОГО МЕТОДА

1. Идентификаторы базисных функций (БФ)

1.1. Непрерывные БФ:

p — полиномы Лежандра.

t — полиномы Чебышева первого рода.

u — полиномы Чебышева второго рода.

c — косинусоиды.

f — комплексные экспоненциальные функции.

y — диадно-упорядоченные функции Уолша.

x — функции Хаара.

1.2. Дискретные БФ:

r — полиномы Чебышева.

s — полиномы Хана-Чебышева первого рода.

d — полиномы Хана-Чебышева второго рода.

k — полиномы Кравчука.

e — комплексные экспоненциальные функции.

q — косинусоиды.

v — диадно-упорядоченные функции Уолша.

h — функции Хаара.

2. Идентификаторы элементарных операций

К ним относятся:

NB — вычисление непрерывных БФ.

DB — вычисление дискретных БФ.

ND — вычисление непрерывных и дискретных БФ.

NX — вычисление НСХ непрерывной одномерной функции.

DX — вычисление НСХ дискретной одномерной функции.

NC — вычисление НСП непрерывной корреляционной функции.
DC — вычисление НСП дискретной корреляционной функции.
D1(R1) — вычисление ДНПФ дифференцирующего (разностного) звена.

I1(C1) — вычисление ДНПФ интегрирующего (суммирующего) звена.

YZ — вычисление ДНПФ усилительного звена.

ZM — вычисление ДНПФ множительного звена.

CD — вычисление ДНПФ звена чистого сдвига.

E0 — вычисление ДНПФ экстраполирующего звена нулевого порядка.

DE — вычисление ДНПФ дискретного элемента с бесконечно малым временем замыкания ключа.

PT — вычисление ДНПФ звена понижения такта.

AP — вычисление ДНПФ аperiodического звена.

AK — вычисление ДНПФ кратного аperiodического звена.

KO — вычисление ДНПФ колебательного звена.

DW — вычисление ДНПФ дискретного экстраполятора нулевого порядка.

OX — вычисление функции времени по НСХ (операция обращения).

OP — вычисление дисперсии случайного сигнала по НСП (операция обращения).

KP — вычисление корреляционной функции случайного сигнала по НСП (операция обращения).

3. Описание процедур (элементарных операций спектрального метода) и их формальных параметров

3.1. Описание процедур непрерывных элементарных операций спектрального метода

1. $SNB_{11}(L1, L, t)$ — вычисляется матрица-строка L непрерывных БФ на отрезке $[0, t]$ на системе тактовых точек $(l-1)t/L1$, где $l=1, \dots, L1+1$. Результат представляется матрицей порядка $L1 \times L$.

2. $SNX_{11}(f, N1, t)$ — вычисляется усеченная НСХ порядка $N1$ на отрезке $[0, t]$ по аналитически заданной функции $f(x)$.

3. $SNC_{11}(R, N1, t)$ — вычисляется усеченная матрица НСП порядка $N1 \times N1$ на отрезке $[0, t]$ по аналитически заданной корреляционной функции $R(x, y)$.

4. $SI_{11}(t, L)$ — вычисляется усеченная матрица ДНПФ интегрирующего звена порядка $L \times L$ на отрезке $[0, t]$.

5. $SP1_{11}(t, L)$ — вычисляется усеченная матрица ДНПФ дифференцирующего звена порядка $L \times L$ на отрезке $[0, t]$.

6. $SM1_{11}(t, L)$ — вычисляется усеченная матрица ДНПФ звена начальных значений порядка $L \times L$ на отрезке $[0, t]$.

7. $SAP_{11}(N1, T, k, t)$ — вычисляется усеченная матрица ДНПФ апериодического звена порядка $N1 \times N1$ на отрезке $[0, t]$; T — постоянная времени апериодического звена; k — коэффициент усиления апериодического звена.

8. $SKO_{11}(N1, T, k1, k, t)$ — вычисляется усеченная матрица ДНПФ апериодического звена порядка $N1 \times N1$ на отрезке $[0, t]$; T — постоянная времени колебательного звена; k — коэффициент усиления колебательного звена; $k1$ — коэффициент демпфирования колебательного звена.

9. $SCD_{11}(N1, T1, t)$ — вычисляется усеченная матрица ДНПФ звена чистого сдвига порядка $N1 \times N1$ на отрезке $[0, t]$; $T1$ — величина чистого сдвига: если $T1 > 0$, то $T1$ — величина запаздывания, если $T1 < 0$, то $T1$ — величина упреждения.

10. $SYZ_{11}(f, N1, t)$ — вычисляется усеченная матрица ДНПФ усилительного звена порядка $N1 \times N1$ на отрезке $[0, t]$ по аналитически заданной функции $f(x)$.

Заметим, что идентификатор $\langle \rangle$ в имени процедуры должен быть заменен комбинацией имен базисных систем функций, т.е. $\langle \rangle = \langle pp|uu|cc|ff|vv|xx \rangle$.

3.2. Описание процедур дискретных элементарных операций спектрального метода

1. $SDB_{11}(L1, L)$ — вычисляется матрица-строка L дискретных БФ на отрезке $[0, t]$ на системе тактовых точек $0, 1, \dots, L1 - 1$. Результат представляется матрицей порядка $L1 \times L$.

2. $SNX_{l}(f, N1, L1)$ — вычисляется усеченная НСХ порядка $N1$ на отрезке $[0, t]$ по аналитически заданной дискретной функции f на системе равноотстоящих тактовых точек $lt/L1$, где $l=0,1,2,\dots,L1-1$.

3. $SNC_{l}(KF, N1, L1)$ — вычисляется усеченная матрица НСП порядка $N1 \times N1$ на системе равноотстоящих тактовых точек $lt/L1$, где $l=0,1,2,\dots,L1-1$, на отрезке $[0, t]$ по аналитически заданной дискретной корреляционной функции $KF(l, m)$ ($L1 \geq N1$).

4. $SC_{l}(L1, L)$ — вычисляется усеченная матрица ДНПФ суммирующего звена порядка $L \times L$ на отрезке $[0, t]$; $L1$ — число тактовых точек на интервале работы системы управления ($L1 \geq L$).

5. $SRI_{l}(L1, L)$ — вычисляется усеченная матрица ДНПФ разностного звена порядка $L \times L$ на отрезке $[0, t]$; $L1$ — число тактовых точек на интервале работы системы управления ($L1 \geq L$).

6. $SM_{l}(L1, L)$ — вычисляется усеченная матрица ДНПФ звена начальных значений порядка $L \times L$ на отрезке $[0, t]$; $L1$ — число тактовых точек на интервале работы системы управления ($L1 \geq L$).

7. $SAP_{l}(N1, T, k, L1, t)$ — вычисляется усеченная матрица ДНПФ апериодического звена порядка $N1 \times N1$ на отрезке $[0, t]$; T — постоянная времени апериодического звена; k — коэффициент усиления апериодического звена; $L1$ — число тактовых точек на интервале работы системы управления ($L1 \geq N1$).

8. $SCD_{l}(N1, k, t)$ — вычисляется усеченная матрица ДНПФ звена чистого сдвига порядка $N1 \times N1$ на отрезке $[0, t]$; k — величина чистого сдвига: если $k > 0$, то k — величина запаздывания, если $k < 0$, то k — величина упреждения; $L1$ — число тактовых точек на интервале работы системы управления ($L1 \geq N1$).

9. $SYZ_{l}(f, N1, t)$ — вычисляется усеченная матрица ДНПФ усилительного звена порядка $N1 \times N1$ на отрезке $[0, t]$ по аналитически заданной функции $f(x)$.

10. $SPT??1(N1, N2, L1, L2)$ — вычисляется усеченная матрица ДНПФ звена понижения такта порядка $N2 \times N1$ на отрезке $[0, t]$; $L1$ — число тактовых моментов на входе звена понижения такта; $L2$ — число тактовых моментов на выходе звена понижения такта ($L2$ кратно $L1$ и $L2 < L1, L1 \geq N1; L2 \geq N2$).

Заметим, что идентификатор $\langle ?? \rangle$ в имени процедуры должен быть заменены комбинацией имен базисных систем функций, т.е. $\langle ?? \rangle = \langle rr | ss | dd | kk | ee | qq | vv | hh \rangle$.

3.3. Описание процедур непрерывно-дискретных элементарных операций спектрального метода

1. $SDE?#1(N1, N2, L1, t0, t)$ — вычисляется усеченная матрица ДНПФ дискретного элемента порядка $N2 \times N1$ на отрезке $[0, t]$; $L1$ — число тактовых точек на интервале работы системы управления; $t0$ — начало отсчета тактовых точек ($L1 \geq N2; 0 \leq t0 \leq t / L1$).

2. $SE0#?1(N1, N2, L1, t)$ — вычисляется усеченная матрица ДНПФ экстраполирующего звена нулевого порядка на отрезке $[0, t]$. Порядок матрицы $N1 \times N2$; $L1$ — число тактовых точек на интервале работы системы управления ($L1 \geq N2$).

Заметим, что идентификаторы $\langle ?# \rangle$ и $\langle \#? \rangle$ в имени процедуры должны быть заменены комбинацией имен базисных систем функций, т.е.

$$\langle ?# \rangle = \langle pr | ck | cq | ud | tq | fe | ts | vy | xh \rangle ,$$

$$\langle \#? \rangle = \langle rp | kc | qc | du | qt | ef | st | yv | hx \rangle .$$

Приложение 2

ВЫЧИСЛЕНИЕ НСХ В БАЗИСЕ КЛАССИЧЕСКИХ ОРТОГОНАЛЬНЫХ ПОЛИНОМОВ

1. Квадратурные правила наивысшей алгебраической степени точности и их частные случаи

Численное интегрирование функции $x(\tau)$ с весовой функцией $\rho(\tau)$ на конечном или бесконечном отрезке можно осуществить заменой интеграла квадратурной суммой, т.е. оно сводится к правилу вычисления следующего вида:

$$\int_a^b \rho(\tau)x(\tau)d\tau = \sum_{k=0}^{L-1} \omega_k^L x(\tau_k^L), \quad (1)$$

где ω_k^L — квадратурные коэффициенты и τ_k^L — квадратурные узлы; L, ω_k^L, τ_k^L — параметры правила (1), и способ их выбора приводит к различным квадратурным формулам.

Правило (1) при фиксированном L содержит $2L$ параметров τ_k^L , ω_k^L , и если их выбрать так, чтобы (1) выполнялось точно для всех алгебраических полиномов степени не выше $(2L-1)$, то получим квадратурное правило наивысшей алгебраической точности. В этом случае параметры ω_k^L можно представить в следующем виде:

$$\omega_k^L = \frac{\mu_L}{\mu_{L-1} \hat{\psi}'_L(\tau_k^L) \hat{\psi}_{L-1}(\tau_k^L)}. \quad (2)$$

Рассмотрим некоторые частные случаи.

Пусть теперь интеграл из формулы (1) приведен к виду

$$\int_0^t \tau^{G-1/2}(t-\tau)^{G-1/2} x(\tau) d\tau. \quad (3)$$

Системой полиномов, ортогональных на $[0, t]$ по весу

$$\rho(t, \tau) = \tau^{G-1/2}(t-\tau)^{G-1/2}, \quad (4)$$

является система полиномов Гегенбауэра

$$\tilde{y}_i(t, \tau) = \frac{1}{\Gamma(G+1/2)} \sqrt{\frac{2(G+i)\Gamma(2G+i)}{i!t^{2G}}} \sum_{k=0}^i l_{i,k} \frac{\tau^k}{t^k}, \quad (5)$$

где

$$l_{i,k} = (-1)^{i-k} \frac{\Gamma\left(G + \frac{1}{2}\right) \Gamma(2G+i+k) C_i^{j-k}}{\Gamma(2G+i) \Gamma(G+k+1/2)}, \quad (6)$$

а поэтому правило (1) принимает вид

$$\int_0^t \tau^{G-1/2}(t-\tau)^{G-1/2} x(\tau) d\tau = \sum_{k=0}^{L-1} \omega_k^L x(\tau_k^L), \quad (7)$$

в котором узлы τ_k^L должны располагаться в корнях полинома Гегенбауэра степени L $\gamma_L(t, \tau_k^L) = 0$ ($k=0, 1, \dots, L-1$). Тогда коэффициент ω_k^L , вычисленный по формуле (2), может быть приведен к виду

$$\omega_k^L = \frac{t^{2G} \Gamma^2(G+1/2)(L!)(1-\gamma_1^2(t, \tau_k^L))}{\Gamma(2G+L)L\gamma_{L-1}^2(t, \tau_k^L)}. \quad (8)$$

Квадратурная формула (7) содержит произвольный параметр G и является источником многих полезных случаев. Например, при $G = 1/2$ полиномы Гегенбауэра совпадают с полиномами Лежандра и получается квадратурное правило Гаусса [13], нули которого

совпадают с нулями полинома $p_L(t, \tau_k^L)$, а при $G = 0$ полиномы Генбауэра (5) совпадают с полиномами Чебышева первого рода

$$\bar{T}_i(t, \tau) = \begin{cases} \frac{1}{\sqrt{\pi}}, & i = 0; \\ \sqrt{\frac{2}{\pi}} \cos\left(i \arccos\left(\frac{2\tau}{t} - 1\right)\right), & i = 1, 2, \dots, \end{cases} \quad (9)$$

которые ортогональны по весу

$$\rho(t, \tau) = 1 / \sqrt{\tau(t - \tau)}. \quad (10)$$

Узлы квадратурного правила (7) в данном случае должны совпадать с нулями полинома $T_L(t, \tau)$, т.е. располагаться в точках

$$\tau_k^L = \frac{t}{2} \left(1 - \cos \frac{(2k+1)\pi}{2L} \right) \quad (k = 0, 1, \dots, L-1). \quad (11)$$

Коэффициент $\omega_k^L = \frac{\pi}{L}$ находится при помощи формулы (8). Таким образом, квадратурное правило наивысшей степени точности с весом (10) имеет вид

$$\int_0^t \frac{x(\tau)}{\sqrt{t(\tau-t)}} d\tau = \frac{\pi}{L} \sum_{k=0}^{L-1} x \left(\frac{t}{2} \left(1 - \cos \left(\frac{2}{2} \frac{k+1}{L} \pi \right) \right) \right) \quad (12)$$

Аналогично находятся квадратурные правила и при других G .

2. Ортонормированные полиномы дискретной переменной, построенные на сетке, определенной в корнях классических ортогональных полиномов

Квадратурная формула (1) является точной для функции $x(\tau) = \hat{\psi}_i(\tau)$ ($i = 0, 1, \dots, 2L-1$). Покажем, что с помощью формул (1) легко указать систему ортонормированных дискретных полиномов по весу (2).

Положим в формуле (1) $x(\tau) = \hat{\psi}_i(\tau)\hat{\psi}_h(\tau)$; $h, i = 0, 1, \dots, L-1$. Тогда получим

$$\delta_{h,i} = \sum_{k=0}^{L-1} \rho(L,k) \hat{\psi}_h(\tau_k^L) \hat{\psi}_i(\tau_k^L), \quad (13)$$

где $\rho(L,k) = \omega_k^L$. Это тождество является условием ортонормированности системы дискретных полиномов $\{\hat{\psi}_i(\tau_k^L)\}$, построенных на сетке $\{\tau_k^L\}$ с помощью уравнения $\hat{\Psi}_L(\tau_k^L) = 0$.

Соотношение (13) перепишем в виде

$$\sum_{k=0}^{L-1} C_{h,k} C_{i,k} = \delta_{h,i}, \quad (14)$$

где $c_{h,k} = \sqrt{\rho(L,k)} \hat{\psi}_h(\tau_k^L)$.

Из формулы (14) вытекает, что матрица C с элементами $c_{h,k}$ ($h, k = 0, 1, \dots, L-1$) является унитарной, и поэтому для матрицы C справедливо также другое соотношение ортогональности:

$$\sum_{i=0}^{L-1} C_{i,k} C_{i,j} = \delta_{k,j} \quad (k, j = 0, 1, \dots, L-1). \quad (15)$$

Соотношение (15) можно записать в виде

$$\sum_{i=0}^{L-1} \hat{\psi}_i(\tau_k^L) \hat{\psi}_i(\tau_j^L) = \rho(L,k) \delta_{k,j}. \quad (16)$$

Примерами дискретных ортонормированных систем базисных функций являются:

1. Дискретные ортонормированные тригонометрические функции, определенные на отрезке $[0, t]$, задаются формулами

$$\hat{C}_i(L, l) = \begin{cases} \sqrt{\frac{1}{L}}, & i = 0; \\ \sqrt{\frac{2}{L}} \cos \frac{i\pi}{L} \left(l + \frac{1}{2} \right), & i = 1, 2, \dots, L-1; \\ i = 0, 1, 2, \dots, L-1; & L = 2, 3, 4, \dots; \end{cases} \quad (17)$$

$$\hat{\Psi}_i(L, l) = \sqrt{\frac{2}{L+1}} \sin \frac{(i+1)\pi}{L+1} (l+1), \quad (18)$$

$i, l = 0, 1, \dots, L-1; L = 2, 3, 4, \dots;$

$$\hat{\alpha}_i(L, l) = \frac{2}{\sqrt{2L+1}} \cos \frac{(2i+1)\pi}{2L+1} \left(l + \frac{1}{2} \right), \quad (19)$$

$i, l = 0, 1, \dots, L-1; L = 2, 3, 4, \dots;$

$$\hat{\beta}_i(L, l) = \frac{2}{\sqrt{2L+1}} \sin \frac{(2i+1)\pi}{2L+1} (l+1), \quad (20)$$

$i, l = 0, 1, \dots, L-1; L = 2, 3, 4, \dots$

Для точечного аппроксимирования удобно считать, что дискретные косинусоиды (17) и (19) заданы на системе точек $\{x_0 = h/2, x_1, x_2, \dots, x_{L-1}\}$ отрезка $[0, t]$ с постоянным шагом $h = x_{i+1} - x_i$, а дискретные синусоиды (18) и (20) заданы на системе точек $\{x_0 = h, x_1, x_2, \dots, x_{L-1}\}$ отрезка $[0, t]$ с постоянным шагом $h = x_{i+1} - x_i$. Дискретные косинусоиды и синусоиды ортонормированны на указанных системах точек с весом $\rho(L, l) = 1$.

2. Дискретные ортонормированные полиномы Хана-Чебышева первого рода, определенные на отрезке $[0, t]$, задаются формулами

$$\hat{T}_i^*(L, t) = \begin{cases} \sqrt{\frac{1}{\pi}}, & i = 0; \\ (-1)^{-i} \sqrt{\frac{2}{\pi}} \cos \frac{i\pi}{L} \left(t + \frac{1}{2} \right); & i = 1, 2, \dots, L-1; \\ t = 0, 1, \dots, L-1; & L = 2, 3, 4, \dots \end{cases} \quad (21)$$

Для точечного аппроксимирования, удобно считать, что дискретные полиномы Хана-Чебышева первого рода (17) заданы на системе точек $\{x_0 = h/2, x_1, x_2, \dots, x_{L-1}\}$ отрезка $[0, t]$ с постоянным шагом $h = x_{i+1} - x_i$. Дискретные полиномы Хана-Чебышева ортонормированны на указанных системах точек с весом $\rho(L, t) = \pi/L$.

3. Дискретные ортонормированные полиномы Хана-Чебышева второго рода, определенные на отрезке $[0, t]$, задаются формулами

$$\hat{\Psi}_i^*(L, t) = (-1)^i \frac{2}{t} \sqrt{\frac{2}{\pi}} \sin \frac{(i+1)\pi}{L+1} (t+1) / \sin \frac{(i+1)\pi}{L+1}, \quad (22)$$

$$i = 0, 1, \dots, L-1; \quad t = 0, 1, \dots, L-1; \quad L = 2, 3, 4, \dots$$

Для точечного аппроксимирования, удобно считать, что дискретные полиномы Хана-Чебышева второго рода (18) заданы на системе точек $\{x_0 = h/2, x_1, x_2, \dots, x_{L-1}\}$ отрезка $[0, t]$ с постоянным шагом $h = x_{i+1} - x_i$. Дискретные полиномы Хана-Чебышева второго рода ортонормированны на указанных системах точек с весом

$\rho(L, t) = \frac{t^2 \pi}{4(L+1)} \sin^2 \left(\frac{(t+1)\pi}{L+1} \right)$. Эти базисные системы тесно связаны между собой, а именно

$$\hat{T}_i^*(L, t) = (-1)^i \sqrt{\frac{L}{\pi}} \hat{C}_i^*(L, t); \quad (23)$$

$$\hat{U}_i(L, l) = \frac{2(-1)^l \sqrt{L+1}}{l \sqrt{\pi} \sin \frac{(l+1)\pi}{L+1}} \hat{\psi}_i(L, l). \quad (24)$$

Заметим, что существуют и другие дискретные базисные системы функций, которые находятся по классическим ортогональным полиномам Якоби и выражаются через функции тригонометрических базисных систем (19), (20).

3. Алгоритмы вычисления НСХ в базисах классических ортонормированных полиномов

Рассмотренные квадратурные формулы можно использовать как численные схемы вычисления НСХ. Используя квадратурную формулу (1) для вычисления L ординат НСХ функции $x(\tau)$, получаем

$$X_{\Psi}^L(i, l) \equiv \sum_{k=0}^{L-1} \omega_k^l \hat{\psi}(i, l, \tau_k^l) x(\tau_k^l). \quad (25)$$

Правая часть приближенного равенства представляет собой НСХ дискретной функции $x(\tau_k^l)$ в базисе ортонормированных дискретных полиномов

$\left\{ \hat{\psi}(i, L, \tau_k^l) = \hat{\psi}(i, l, \tau_k^l) \right\}$ с весом $\rho(L, k) = \omega_k^l$, т.е.

$$X_{\Psi}^L(i, l) \equiv X_{\hat{\Psi}}(i, L). \quad (26)$$

Соотношение (26) устанавливает способ вычисления НСХ в базисе классических ортогональных полиномов по их дискретным аналогам, построенным на сетке $\{\tau_k^l\}$ с помощью уравнения $\psi_L(\tau_k^l) = 0$.

Рассмотрим некоторые частные случаи. Положим $\hat{\psi}(i, l, \tau) = \tilde{\gamma}(i, l, \tau)$. Тогда из формул (5), (8), (25) и (26) находим, что

$$X_{\gamma}(i, t) \equiv \frac{\Gamma(G + \frac{1}{2})L!}{L^2 \Gamma(2G + L)} \sqrt{\frac{t^{2G} 2(G+i) \Gamma(2G+i)}{i!}} \times \\ \times \sum_{k=0}^{L-1} \frac{1 - \gamma_*^2(1, L, k)}{\gamma_*^2(L-1, L, k)} \gamma_*(i, L, k) x(\tau_k^L). \quad (27)$$

Несложные вычисления позволяют найти алгоритмы вычисления НСХ для различных значений G :

— для $G = 1/2$ (полином Лежандра, τ_k^L — нули L -го полинома Лежандра)

$$X_p(i, t) \equiv \frac{1}{L^2} \sqrt{(2i+1)t} \sum_{k=0}^{L-1} \frac{1 - P_*^2(i, L, k)}{P_*^2(L-1, L, k)} P_*(i, L, k) x(\tau_k^L); \quad (28)$$

— для $G = 0$ (полиномы Чебышева 1-го рода)

$$X_T(i, t) \equiv \begin{cases} \frac{\sqrt{\pi}}{L} \sum_{k=0}^{L-1} x(\tau_k^L) & \text{при } i=0; \\ \frac{\sqrt{2 \cdot \pi}}{L} \sum_{k=0}^{L-1} T(i, L, k) x(\tau_k^L) & \text{при } i=1, 2, \dots, L-1; \\ \tau_k = \frac{t}{2} \left(1 + \cos\left(\frac{2L-2k-1}{2L} \pi\right) \right); \end{cases} \quad (29)$$

— для $G = 1$ (полиномы Чебышева 2-го рода)

$$X_U(i, t) \equiv \begin{cases} \frac{t(i+1)\sqrt{2\pi}}{L+1} \sum_{k=0}^{L-1} \sin^2\left(\frac{(k+1)\pi}{L+1}\right) x(\tau_k^L) U_*(i, L, k); \\ \tau_k = \frac{t}{2} \left(\cos\left(\frac{(k+1)\pi}{L+1}\right) + 1 \right). \end{cases} \quad (30)$$

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Солодовников В.В.* и др. Расчет систем управления на ЦВМ. — М.: Машиностроение, 1979.
2. *Солодовников В.В., Семенов В.В.* Спектральная теория нестационарных систем управления. — М.: Наука, 1974.
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.
4. Изучение математических дисциплин в компьютерной среде: Учебное пособие / Т.Б. Волкова, В.В. Рыбин и др.; Под ред. В.В. Семенова. — М.: Изд-во МАИ, 1996.
5. *Солодов А.В., Петров Ф.С.* Линейные автоматические системы с переменными параметрами. — М.: Физматгиз, 1971.
6. *Семенов В.В.* и др. Математическая теория управления в примерах и задачах: Учебное пособие. — М.: Изд-во МАИ, 1997.
7. *Дьяконов В.П.* Simulink 4. Специальный справочник. — СПб.: Питер, 2002.
8. *Дьяконов В.П.* MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. Полное руководство пользователя. — М.: СОЛОН-Пресс, 2002.
9. Алгоритмическое и математическое обеспечение автоматизации проектирования систем управления / Под ред. В.В. Семенова. — М.: МАИ, 1982.
10. *Рыбин В.В.* Описание сигналов и линейных нестационарных систем управления в базисах вейвлетов и их анализ в вычислительных средах: Учебное пособие. — М.: Изд-во МАИ, 2003.
11. Системное и прикладное обеспечение автоматизации проектирования систем управления: Учебное пособие / А.В. Пантелеев, В.В. Рыбин и др.; Под ред. В.В. Семенова. — М.: МАИ, 1983.
12. *Никифоров А.Ф., Сулов С.К., Уваров В.Б.* Классические ортогональные полиномы дискретной переменной. — М.: Наука, 1985.
13. *Рыбин В.В.* Приближение функций. Компьютерная практика в системе компьютерной математики MATHCAD: Учебное пособие. — М.: Изд-во МАИ, 2004.
14. *Рыбин В.В., Харченко Т.В.* Компьютерный курс «Спектральная теория систем управления» для дистанционного обучения // Труды VI Международного совещания-семинара «Информационные технологии в учебном процессе кафедр физики и математики». — Ульяновск: УлГТУ, 2002.

ОГЛАВЛЕНИЕ

Предисловие	3
1. Разработка пакета расширения Spekr_SM	5
1.1. Краткая характеристика системы математического визуального моделирования Simulink	5
1.2. Характеристика пакета прикладных программ спектрального метода MLSY_SM в СКМ Matlab	10
1.2.1. Основные характеристики спектральной формы описания непрерывно-дискретных систем	10
1.2.2. Пакет прикладных программ спектрального метода MLSY_SM	19
1.2.3. Примеры разработки программных модулей пакета MLSY_SM	20
1.3. Проект и технологические особенности разработки пакета Spekr_SM	26
1.4. Разработка структурной модели и оглавления пакета расширения Spekr_SM	28
1.4.1. Версия структурной модели и оглавления пакета Spekr_SM	28
1.4.2. Разработка пиктограмм разделов структурной модели и блоков	31
1.5. Разработка структурной модели библиотеки R_Spekr	33
1.6. Разработка собственных блоков пакета расширения Spekr_SM	34
1.6.1. Технология формирования блока и сопутствующих программ	34
1.6.2. Состав типовых блоков библиотеки Spekr_SM	44
1.6.3. Примеры анализа структуры и содержания типовых блоков пакета Spekr_SM	47
1.7. Особенности применения пакета Spekr_SM в системе Simulink для анализа систем управления при детерминированных и случайных воздействиях	50
2. Примеры анализа нестационарных линейных систем управления спектральным методом и методом моментов в пакетах Spekr_SM и Simulink	53

2.1. Пример анализа бокового канала системы управления ЛА посадки на Луну	53
2.1.1. Математическая модель канала управления по рысканию и задачи ее анализа	53
2.1.2. Спектральный расчет системы управления в системе математического визуального моделирования Simulink с применением пакета расширения Spekr_SM.	55
2.1.3. Применение метода моментов для анализа системы управления в системе математического визуального моделирования Simulink	60
2.2. Пример анализа продольного канала системы управления ЛА посадки на Марс	64
2.2.1. Математическая модель продольного канала управления и задачи ее анализа	64
2.2.2. Спектральный расчет системы управления в системе математического визуального моделирования Simulink с применением пакета расширения Spekr_SM	68
2.2.3. Применение метода моментов для анализа системы управления в системе математического визуального моделирования Simulink	73
Приложение 1. Описание идентификаторов и процедур элементарных операций спектрального метода	76
Приложение 2. Вычисление НСХ в базисе классических ортогональных полиномов	81
Библиографический список	89