

Федеральное агентство по образованию РФ  
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
(ГОУВПО «АмГУ»)

УТВЕРЖДАЮ  
Зав. кафедрой ИиУС  
\_\_\_\_\_ А.В. Бушманов  
«\_\_\_» \_\_\_\_\_ 2006 г.

**Учебно-методический комплекс дисциплины**  
**МЕТОДЫ АНАЛИЗА ДИНАМИЧЕСКИХ СИСТЕМ**

для специальности  
230102 – автоматизированные системы обработки  
информации и управления

Составитель: Ерёмин Е.Л.

2006 г.

*Печатается по решению  
редакционно-издательского совета  
факультета математики  
и информатики  
Амурского государственного  
университета*

Методы анализа динамических систем для специальности 230102 «Автоматизированные системы обработки информации и управления»: учебно-методический комплекс дисциплины. / Еремин Е.Л. – Благовещенск. Изд-во Амурского гос. ун-та, 2006. 89 с.

©Амурский государственный университет, 2006

©Кафедра информационных и управляющих систем, 2006

## ОГЛАВЛЕНИЕ

1.	Рабочая программа	4
2.	График самостоятельной работы студентов	13
3.	Методические рекомендации по проведению самостоятельной работы студентов	14
4.	Перечень учебников, учебных пособий	26
5.	План-конспект лекций	27
6.	Методические рекомендации и задания к выполнению курсовой работы	29
7.	Методические рекомендации и комплект заданий к выполнению лабораторных работ	56
8.	Перечень программных продуктов, используемых в практике выпускников и учебно-методическое пособие	78
9.	Методические указания по применению современных ИТ для преподавания учебной дисциплины	79
10.	Методические указания по организации межсессионного и экзаменационного контроля знаний студентов	80
11.	Фонд тестовых и контрольных заданий для оценки качества знаний по дисциплине	81
12.	Комплект экзаменационных билетов	83
13.	Карта кадровой обеспеченности дисциплины	89

## 1. РАБОЧАЯ ПРОГРАММА

по дисциплине " Методы анализа динамических систем "  
для специальности 230102 «Автоматизированные системы обработки информации и управления»

курс 4 семестр 8

Лекции 30 Экзамен 8 семестр

Практические (семинарские) занятия \_\_ (час.)

Лабораторные занятия 30 (час.) Зачет

Самостоятельная работа 60/30 (к.р.) (час.)

Всего часов 120/30 (к.р.) (час)

1. Цели и задачи дисциплины, ее место в учебном процессе.

1.1. Цель преподавания дисциплины.

Изучение методов анализа различных типов динамических систем, описываемых линейными и нелинейными, детерминированными и стохастическими, дифференциальными и разностными, а также иными математическими моделями.

1.2. Задачи изучения дисциплины.

По завершению курса "Математические методы анализа динамических систем", студент должен получить устойчивые навыки и умения по владению основными математическими методами исследования динамических систем, включая этап их имитационного моделирования на ПЭВМ.

1.3. Перечень разделов (тем) необходимых дисциплин.

1.3.1. Математика: математический анализ; вычислительная математика; теория вероятностей, математическая статистика и случайные процессы.

1.3.2. Моделирование систем.

1.3.3. Основы теории управления

1.3.4. Теоретические основы автоматизированного управления.

## 2. Содержание дисциплины

### 2.1. Федеральный компонент

Дисциплина специализации

ГОС ВПО: 500 ДС – 1.

### 2.2. Лекционные занятия

2.2.1. Формы представления и преобразования моделей линейных динамических систем в пространстве состояний (4 часа): прямое, последовательное и параллельное программирование; векторно-матричная форма представления типовых форм записи линейных моделей, описываемых обыкновенными дифференциальными уравнениями.

2.2.2. Математические модели линейных динамических систем при случайных воздействиях и анализ их свойств (8 часов): основные понятия и определения случайных процессов и их характеристик (корреляционные функции, спектральные плотности и т.п.); прохождение случайных сигналов через динамическую систему; анализ погрешностей, вносимых случайными процессами в САУ; расчет САУ с учетом случайных воздействий.

2.2.3. Приближенные модели и методы анализа нелинейных систем управления (6 часов): гармоническая линеаризация нелинейностей; алгебраические способы определения автоколебаний в нелинейных системах 1<sup>ого</sup> типа; частотный метод определения автоколебаний; статические и скоростные ошибки автоколебательных систем; оценка качества в нелинейных системах регулирования.

2.2.4. Вариационные методы оптимального управления (6 часов): решение задач оптимального управления с использованием метода множителей Лагранжа (задачи с закрепленным и подвижными концами и т.д.); принцип максимума Понтрягина, условие нормальности.

2.2.5. Прямой метод Ляпунова в адаптивном управлении (4 часа): синтез адаптивного управления с неявной эталонной моделью (НЭМ); разработка адаптивного управления с НЭМ для объектов с запаздыванием по состоянию.

2.2.6. Робастное управление с явной эталонной моделью (ЯЭМ) (2 часа): синтез разработка робастного закона управления на основе прямого метода Ляпунова и критерия гиперустойчивости.

## 2.3. Лабораторные работы

- 2.3.1. Лабораторная работа №1 (4-х часовая). Тема: использование типовых математических моделей и их преобразований при исследовании свойств линейных динамических систем.
- 2.3.2. Лабораторная работа №2 (6-х часовая). Тема: применение теории случайных функций для анализа линейных систем управления.
- 2.3.3. Лабораторная работа №3 (4-х часовая). Тема: анализ систем управления с типовыми нелинейностями.
- 2.3.4. Лабораторная работа №4 (4-х часовая). Тема: Программирование в SIMULINK пользовательских функций.
- 2.3.5. Лабораторная работа №5 (4-х часовая). Тема: Разработка S-функций непрерывных систем управления.
- 2.3.6. Лабораторная работа №6 (4-х часовая). Разработка S-функций гибридных систем управления.
- 2.3.7. Лабораторная работа №7 (4-х часовая). Разработка S-функций дискретных систем управления.

## 2.4. Курсовая работа.

- 2.4.1. Создание MATLAB-приложений для исследования адаптивных систем управления.

### 2.4.2. Задание:

- 2.4.2.1. Построить SIMULINK-модель (S-модель) гибридной системы адаптивного управления с помощью пакета MATLAB (согласно заданному варианту, используя структурную схему, математическое описание и исходные данные, приведены в табл.).
- 2.4.2.2. Обеспечить возможность наблюдения следующих функций времени S-модели:  $x(t) \in R^n$ ;  $y(t) \in R^l$ ;  $e_k \in R^n$ ;  $u_k \in R^m$ ;  $f(t) \in R^n$ ;  $c_k$ ;  $r_k \in R^m$ . S-модель представить в виде файла с именем – N\_НомерВарианта\_НомерСхемы\_1.mdl.
- 2.4.2.3. Сформировать блок S-функции из заданного состава элементов S-модели (см. соответствующую структурную схему) и повторно выполнить все задания п.1. S-модель совместно с S-функцией представить в файле – N\_НомерВарианта\_НомерСхемы\_2.mdl.

- 2.4.2.4. Объединить S-модели, построенные в п.1 и 2, представив их в виде схемы рассогласования для выходов систем управления. S-модель представить в файле N\_НомерВарианта\_НомерСхемы\_3.mdl.
- 2.4.2.5. Разработать графический интерфейс для адаптивной системы управления в виде S-модели (см. результаты п.2), используя возможности подсистемы Guide пакета MATLAB.
- 2.4.2.6. Интерфейс должен позволять: просматривать результаты моделирования и справочную информацию; изменять время моделирования, параметры объекта управления и эталонной модели, значения коэффициентов контура адаптации, а также шаг дискретизации.

Рекомендуемая литература:

1. Еремин Е.Л., Еремина В.В., Семичевская Н.П., Шевко Д.Г. Алгоритмы и S-модели гибридных систем адаптивного управления (практикум в среде SIMULINK). Благовещенск: Амурский гос. ун-т, 2005. 205 с.

## 2.5. Самостоятельная работа студентов

2.5.1. Тема: гибридные системы адаптивного управления.

- 2.5.1.1. Гибридные системы со скалярным управлением – 15 час.
- 2.5.1.2. Гибридные системы с векторным управлением – 15 час.

Рекомендуемая литература:

1. Статьи в периодических журналах: Автоматика и телемеханика; Теория и системы управления, Известия РАН; Информатика и системы управления.
2. Еремин Е.Л., Еремина В.В., Семичевская Н.П., Шевко Д.Г. Алгоритмы и S-модели гибридных систем адаптивного управления (практикум в среде SIMULINK). Благовещенск: Амурский гос. ун-т, 2005. 205 с.

## 2.6. Вопросы к зачету.

- 2.6.1. Охарактеризуйте назначение m-файлов: conv(a, b),

- `tf2ss(num,den), ctrb(a,b), obsv(a,c).`
- 2.6.2. Поясните суть методов прямого, последовательного и параллельного программирования.
  - 2.6.3. Сформулируйте критерии управляемости и наблюдаемости Калмана.
  - 2.6.4. Охарактеризуйте назначение m-файлов `urprime(t,y)`, `ode45('имя файла',t0,tfinal,y0,tol,trace).`
  - 2.6.5. Дайте определение основных временных и частотных характеристик динамической системы.
  - 2.6.6. Приведите в общем виде уравнения частотных характеристик: `mag`, `phase`, `re` и `im`.
  - 2.6.7. Охарактеризуйте назначение m-файлов `bode(a,b,c,d,1,w)`, `nyquist(a,b,c,d,1,w)`.
  - 2.6.8. Охарактеризуйте назначение m-файлов: `cloop(num0,den0,sign)`, `trapz(x,y)`.
  - 2.6.9. Дайте определения и охарактеризуйте назначение корреляционной функции и спектральной плотности стационарных случайных процессов.
  - 2.6.10. В чем суть гипотезы эргодичности для стационарных случайных процессов?
  - 2.6.11. Почему знание корреляционной функции и спектральной плотности позволяют полностью охарактеризовать свойства стационарных случайных процессов?

## 2.7. Вопросы к экзамену

- 2.7.1. Прямое, последовательное и параллельное программирование.
- 2.7.2. Вероятностные характеристики дискретных и непрерывных случайных величин.
- 2.7.3. Случайные процессы и корреляционные функции.
- 2.7.4. Стационарные случайные процессы и их характеристики.
- 2.7.5. Спектральная плотность случайных процессов.
- 2.7.6. Прохождение случайных сигналов через линейную систему.
- 2.7.7. Расчет установившихся ошибок при случайных стационарных воздействиях.
- 2.7.8. Расчет по минимуму среднеквадратической ошибки.
- 2.7.9. Приближенные методы анализа нелинейных систем первого типа на основе гармонической линеаризации.
- 2.7.10. Коэффициенты гармонической линеаризации релейных элементов без гистерезиса.

- 2.7.11. Коэффициенты гармонической линейзации релейных элементов с гистерезисом.
- 2.7.12. Методы анализа периодических режимов (основной алгебраический метод и с использованием графиков коэффициентных гармонической линейзации).
- 2.7.13. Методы анализа периодических режимов (использование годографа Михайлова и метод коэффициентных соотношений).
- 2.7.14. Частотный метод определения автоколебаний
- 2.7.15. Статические и скоростные ошибки автоколебательных систем.
- 2.7.16. Вариационные методы оптимального управления (вывод уравнения Эйлера).
- 2.7.17. Уравнение Эйлера-Лагранжа и метод множителей Лагранжа в задаче оптимального управления с закрепленными концами и фиксированным временем.
- 2.7.18. Пример решения задачи оптимального управления с закрепленными концами и фиксированным временем.
- 2.7.19. Вывод условий трансверсальности и запись уравнений Эйлера-Лагранжа в задаче с подвижными концами и фиксированным временем.
- 2.7.20. Пример решения задачи оптимального управления с подвижными концами и фиксированным временем.
- 2.7.21. Прямой метод Ляпунова в задаче адаптивного управления.
- 2.7.22. Прямой метод Ляпунова и критерий гиперустойчивости в задаче робастного управления.

## 2.8.

### Оценочные критерии

- 2.8.1. Студент получает зачет по изучаемой дисциплине в случае, если он свободно владеет основными теоретическими понятиями и определениями, а также умеет правильно использовать рассмотренные практические методы.
- 2.8.2. При оценке знаний на экзамене учитывается: правильность и осознанность изложения содержания ответа на вопросы, полнота раскрытия понятий и закономерностей, точность употребления и трактовки общенаучных и специальных терминов; степень сформированности интеллектуальных и научных способностей экзаменуемого; самостоятельность ответа; речевая грамотность и логическая последовательность ответа. Критерии оценок:  
 -отлично – полно раскрыто содержание вопросов в объеме программы и рекомендованной литературы; четко и правильно даны определения и раскрыто содержание концептуальных по-

определения и раскрыто содержание концептуальных понятий, закономерностей, корректно использованы научные термины; для доказательства использованы различные теоретические знания, выводы из наблюдений и опытов; ответ самостоятельный, исчерпывающий, без наводящих дополнительных вопросов, с опорой на знания, приобретенные в процессе специализации по выбранному направлению информатики.

- хорошо – раскрыто основное содержание вопросов; в основном правильно даны определения понятий и использованы научные термины; ответ самостоятельный; определения понятий неполные, допущены нарушения последовательности изложения, небольшие неточности при использовании научных терминов или в выводах и обобщениях, исправляемые по дополнительным вопросам экзаменаторов.

- удовлетворительно – усвоено основное содержание учебного материала, но изложено фрагментарно, не всегда последовательно; определение понятий недостаточно четкое; не использованы в качестве доказательства выводы из наблюдений и опытов или допущены ошибки при их изложении; допущены ошибки и неточности в использовании научной терминологии, определении понятий;

- неудовлетворительно – ответ неправильный, не раскрыто основное содержание программного материала; не даны ответы на вспомогательные вопросы экзаменаторов; допущены грубые ошибки в определении понятий и использовании терминологии.

### 3. Учебно-методические материалы по дисциплине

#### 3.1. Используемая и рекомендуемая литература

Основная:

- 3.1.1. Еремин Е.Л., Еремина В.В., Семичевская Н.П., Шевко Д.Г. Алгоритмы и S-модели гибридных систем адаптивного управления (практикум в среде SIMULINK). Благовещенск: Амурский гос. ун-т, 2005. 205 с.
- 3.1.2. Еремин Е.Л., Еремин И.Е. Методы анализа динамических систем. Лабораторный практикум на ПЭВМ. / Благовещенск, Амурский гос. у-нт - 2005. <http://intra.amursu.ru/amursu>.
- 3.1.3. Афанасьев В.Н., Колмановский В.Б. Носов В.Р. Математическая теория конструирования систем управления. М.: Высш. Шк., 1989, 447 с.

3.1.4. Зимницкий В.А., Устинов С.М. Методы анализа математических моделей динамических систем: Учебное пособие. - Л.: Изд-во ЛГУ, 1991. - 81 с.

Дополнительная:

3.1.5. Андриевский Б.Р., Фрадков А.Л. Избранные главы теории автоматического управления с примерами на языке MATLAB. – СПб.: Наука 1999. – 467.

3.1.6. Мирошник И.В., Никифоров В.О. Фрадков А.Л. Нелинейное и адаптивное управление сложными объектами. – СПб.: Наука 2000. – 549.

Учебные пособия:

3.1.7. Еремин Е.Л. Теоретические основы автоматизированного управления. Курс лекций для специальности 220200. 2005. - <http://intra.amursu.ru/amursu>.

3.1.8. Еремин Е.Л. Лабораторно-курсовой практикум по ТОАУ с применением Matlab for Windows: Учебное пособие. - Благовещенск: Изд-во АмГУ, 1998. - 114 с.

3.1.9. Еремин Е.Л. Практикум по ТОАУ с применением Matlab for Windows. - <http://www.amursu.ru>.

4. Учебно-методическая (технологическая) карта дисциплины<sup>1</sup>

Номер недели	Номер темы	Вопросы, изучаемые на лекции	Занятия		Используемые наглядные и методические пособия	Самостоятельная работа студентов		Форма контроля				
			Практические	Лабораторные		Содержание	Часы					
1	2	3	4	5	6	7	8	9				
1	1	2.2.1	-	2.3.1	3.1.8, 3.1.10	2.5	30	злр				
2			-									
3			-									
4	2	2.2.2	-	2.3.2	3.1.8, 3.1.10			2.5	30	злр		
5			-									
6			-									
7	3	2.2.3	-	2.3.3	3.1.8, 3.1.10					2.5	30	злр, сб
8			-									
9			-									
10	4	2.2.4	-	2.3.5	3.1.8, 3.1.10	2.4	30					злр, зкр
11			-									
12			-									
13	5	2.2.5	-	2.3.5	3.1.8, 3.1.10			2.4	30			злр
14			-									
15			-									
15	6	2.2.6	-	2.3.7	3.1.8, 3.1.10							злр, з

<sup>1</sup> Принятые сокращения:

защита отчета о выполнении лабораторной работы – злр;

собеседование по результатам самостоятельной работы студентов – сб;

защита курсовой работы – зкр;

зачет по изучаемой дисциплине – з.

## 2. ГРАФИК САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Содержание	Объем в часах	Сроки и форма контроля
3.1. Тема: гибридные системы адаптивного управления.		
3.1.1. Гибридные системы со скалярным управлением.	15 час.	Собеседование (6 неделя)
3.1.2. Гибридные системы с векторным управлением.	15 час.	Собеседование (7 неделя)

### 3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

#### ГИБРИДНЫЕ СИСТЕМЫ АДАПТИВНОГО УПРАВЛЕНИЯ

ГСПАУ с явной моделью составляют большой класс адаптивных систем управления, в которых желаемое движение задается конкретным физически реализованным устройством, построенным с использованием традиционных методов синтеза адаптивных систем автоматического управления.

За основу работы контура адаптации ГСПАУ принимается вектор рассогласования  $e(t)$ . Поскольку желаемое качество процесса в основном контуре ГСПАУ определяется динамикой ЭМ, то при разработке адаптивной системы управления, а также ее технической реализации не требуется каких-либо дополнительных измерителей качества функционирования основного контура ГСПАУ, что придает системе относительную простоту, делая ее доступной и удобной для практического применения.

В параграфе 3.1 приведено решение задачи синтеза гибридных систем со скалярным управлением для случая как полного, так и неполного измерения вектора пространства состояний, опирающееся на использование нелинейного преобразования фазовых координат и применение критерия гиперустойчивости. Эта же задача рассматривается в параграфе 3.2, но уже для систем с векторным управлением. Параграф 3.3 посвящен обсуждению результатов моделирования адаптивных систем управления динамическими объектами, рассмотренными в настоящей главе.

##### 3.1. Гибридные системы со скалярным управлением

Постановка задачи синтеза систем с явным эталоном. Рассмотрим объект управления, описываемый уравнением

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + f(t), \quad y(t) = L^T x(t), \quad (3.1)$$

и дискретный адаптивный регулятор со следующей структурой:

$$u_k = c_{1,k} r_k + c_{2,k} y_k, \quad y_k = y(t_k), \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (3.2)$$

где  $x(t) \in R^n$  – вектор состояния объекта;  $y(t) \in R^l$  – вектор выхода объекта;  $u(t) \in R^m$  – вектор управляющих воздействий;  $c_{1,k}$  и  $c_{2,k}$  – матрицы настраиваемых коэффициентов регулятора;  $r_k \in R^m$  – вектор задающих воздействий;  $t_k = kt$  – дискретный аналог времени;  $t = const > 0$  – шаг дискретизации;  $k = 0, 1, 2, \dots$  – номер шага;  $A$ ,  $B$  и  $L$  – матрицы заданного размера соответственно состояния, управления и выхода;  $f(t) \in R^n$  – вектор возмущений или помех, который может быть как затухающим и удовлетворять неравенству

$$\int_0^{\infty} \|f(t)\|^2 dt < \infty, \quad (3.3)$$

так и ограниченным по норме

$$\|f(t)\| \leq f_0 = const. \quad (3.4)$$

Относительно функционирования объекта (3.1) предполагается, что уровень априорной неопределенности задан условиями

$$A = A(x), \quad B = B(x), \quad f(t) = f_x(t), \quad x \in \Xi, \quad (3.5)$$

где  $x$  – набор всех неизвестных параметров;  $\Xi$  – известное множество возможных значений  $X$ .

Желаемое поведение ОУ (3.1) задается с помощью эталонной модели, описываемой

уравнениями:

$$\frac{d\bar{x}(t)}{dt} = A_M \bar{x}(t) + B_M r(t), \quad \bar{y}(t) = L^T \bar{x}(t), \quad (3.6)$$

где  $\bar{x}(t) \in R^n$  – вектор состояния ЭМ;  $\bar{y}(t) \in R^l$  – вектор выхода ЭМ;  $A_M$  и  $B_M$  – постоянные матрицы соответствующих размеров, причем  $A_M$  – гурвицева;  $r(t) = r_k$  при  $t_k \leq t < t_{k+1}$ .

Как обычно при адаптивном подходе, осуществляется настройка коэффициентов адаптивного регулятора по некоторым алгоритмам, вид которых подлежит определению, исходя из выполнения целевых условий.

Пусть требуется решить две задачи.

**Задача 1.** Если вектор возмущений  $f(t)$  удовлетворяет соотношению (5.3), то при любых начальных условиях и любом  $x \in \Xi$  синтезировать систему, обладающую свойствами

$$\lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} (\bar{x}(t) - x(t)) = 0, \quad (3.7)$$

$$\lim_{k \rightarrow \infty} c_{1,k} = c_{1,*} = const, \quad \lim_{k \rightarrow \infty} c_{2,k} = c_{2,*} = const. \quad (3.8)$$

**Задача 2.** Если вектор помех удовлетворяет ограничению (3.4), но противоречит условию (3.3), то при любых начальных условиях и любом  $x \in \Xi$  синтезировать систему со свойствами

$$\lim_{t \rightarrow \infty} \|e(t)\| = \lim_{t \rightarrow \infty} \|\bar{x}(t) - x(t)\| \leq S = const, \quad (3.9)$$

$$\lim_{k \rightarrow \infty} c_{1,k} \leq c_{1,*} = const, \quad \lim_{k \rightarrow \infty} c_{2,k} \leq c_{2,*} = const. \quad (3.10)$$

Решение задач синтеза будем осуществлять, выделяя соответствующие этапы синтеза адаптивных систем управления, основываясь на методике построения ГСПАУ, суть которой изложена в главе 1.

Сначала рассмотрим решение задачи синтеза алгоритмов настройки для системы со скалярным управлением, т.е. случай, когда она описывается уравнениями

$$\frac{dx(t)}{dt} = Ax(t) + bu(t) + f(t), \quad y(t) = L^T x(t), \quad (3.11)$$

$$u_k = c_{1,k} r_k + c_{2,k}^T y_k, \quad y_k = y(t_k), \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (3.12)$$

$$\frac{d\bar{x}(t)}{dt} = A_M \bar{x}(t) + b_M r(t), \quad \bar{y}(t) = L^T \bar{x}(t). \quad (3.13)$$

**Первый этап.** В предположении малости шага дискретизации  $t$  и используя обозначение

$$e(t) = \bar{x}(t) - x(t), \quad (3.14)$$

а также учитывая соотношение (3.12) и условия структурного согласования

$$A_M - A = bc_{2,*}^T L^T, \quad b_M = bc_{1,*},$$

можно в ходе преобразований результата вычитания первого уравнения (3.11) из первого уравнения (3.13) получить следующее эквивалентное математическое описание исследуемой системы:

$$\frac{de(t)}{dt} = A_M e(t) + bm(t), \quad n(t) = g^T L^T e(t), \quad (3.15)$$

$$m_k = (c_{1,*} - c_{1,k}) r_k + (c_{2,*} - c_{2,k}^T)^T y_k, \quad (3.16)$$

$$m(t) = m_k \text{ при } t_k \leq t < t_{k+1}, \quad (3.17)$$

где  $n(t) \in R^1$  – обобщенный выход эквивалентной системы;  $g$  – постоянный вектор, элементы которого подлежат выбору.

**Второй этап.** Определение условий положительности ЛСЧ-системы. Проведение

синтеза на этой стадии разработки ГСПАУ состоит в разрешении проблемы положительности относительно линейной части исходной системы управления с эквивалентным математическим описанием вида (3.15), (3.16), (3.17). Стандартный подход к решению такой задачи – обеспечение свойств вещественности и положительности передаточной функции линейной стационарной части системы:

$$W(I) = g^T L^T (IE - A_M)^{-1} b = \frac{g^T L^T (IE - A_M)^+ b}{\det(IE - A_M)}, \quad (3.18)$$

где  $E$  – единичная матрица;  $(IE - A_M)^+$  – матрица, присоединенная к матрице  $(IE - A_M)$ . Известно, что для получения  $W(I)$  с указанными свойствами необходимо и достаточно вектор  $g$  выбрать таким образом, чтобы в условиях априорной неопределенности (3.5) полином  $g^T L^T (IE - A_M)^+ b$  был бы гурвицевым степени  $(n - 1)$  с положительными коэффициентами.

Третий этап. Для ННЧ исследуемой системы необходимо показать справедливость следующего неравенства:

$$h(0, k_1) = -\sum_{k=0}^{k_1} m_k n_k \geq -g_0^2 = const, \quad \forall k_1 \geq 0, \quad (3.19)$$

где  $n_k = n(t_k)$ .

При решении проблемы положительности ННЧ исходной системы (3.15), (3.16), (3.17) воспользуемся результатами нелинейного преобразования и рассмотрим вместо неравенства (3.19) неравенство, записанное относительно нелинейно преобразованной системы:

$$h(0, k_1) = -\sum_{k=0}^{k_1} m_k z_k \geq -g_0^2 = const, \quad \forall k_1 \geq 0, \quad (3.20)$$

где  $z_k = z(t_k)$ ,  $z(t) = g^T L^T e(t) \| e(t) \|^q$ .

Используя уравнение (3.16), получим:

$$\sum_{k=0}^{k_1} ((c_{1,k} - c_{1,*}) r_k + (c_{2,k} - c_{2,*})^T y_k) z_k \geq -g_0^2. \quad (3.21)$$

Теперь положим:

$$c_{1,k} = c_{1,k-1} + j(z_k), \quad (3.22)$$

$$c_{2,k} = c_{2,k-1} + f(z_k), \quad (3.23)$$

или

$$c_{1,k} = \sum_{i=0}^k j(z_i) + c_{1,-1}, \quad (3.24)$$

$$c_{2,k} = \sum_{i=0}^k f(z_i) + c_{2,-1}, \quad (3.25)$$

тогда получим неравенство:

$$\begin{aligned} h(0, k_1) &= \sum_{k=0}^{k_1} z_k \left( \sum_{i=0}^k j(z_i) + c_{1,-1} - c_{1,*} \right) r_k + \\ &+ \sum_{k=0}^{k_1} z_k \left( \sum_{i=0}^k f(z_i) + c_{2,-1} - c_{2,*} \right)^T y_k \geq -g_0^2, \end{aligned} \quad (3.26)$$

которое будет выполняться, если оба члена левой части удовлетворяют неравенству того же типа.

Для определения явного вида функций  $j$  и  $f$ , удовлетворяющих неравенствам, воспользуемся следующим соотношением:

$$\sum_{k=0}^{k_1} F_k \left( \sum_{i=0}^k F_i + C \right) = \frac{1}{2} \left( \left( \sum_{k=0}^{k_1} F_k + C \right)^2 + \sum_{k=0}^{k_1} F_k^2 - C^2 \right) \geq -\frac{1}{2} C^2, \quad (3.27)$$

где  $C = const$ . Используя (3.27), получим функции  $j$  и  $f$  в виде

$$j(z_k) = h_1 z_k r_k, \quad h_1 = const > 0, \quad (3.28)$$

$$f(z_k) = H_2 z_k y_k, \quad H_2 = diag\{h_{2i}\}, \quad h_{2i} = const > 0, \quad i = \overline{1, l}, \quad (3.29)$$

алгоритмы адаптации коэффициентов регулятора

$$c_{1,k} = c_{1,k-1} + h_1 z_k r_k, \quad (3.30)$$

$$c_{2,k} = c_{2,k-1} + H_2 z_k y_k. \quad (3.31)$$

Рассматривая вопрос технической реализуемости алгоритмов (3.30), (3.31), необходимо указать, что для их реализации требуется полностью измерять вектор состояния объекта (3.11).

В тех случаях, когда вектор состояния ОУ измеряется не полностью, алгоритмы адаптации (3.30), (3.31) должны быть модифицированы. Для этой цели перепишем неравенство (3.20) следующим образом:

$$h(0, k_1) = -\sum_{k=0}^{k_1} m_k z_k \Phi_k \geq -g_0^2 = const, \quad \forall k_1 \geq 0, \quad (3.32)$$

где введена функция  $\Phi_k \geq 0$ , которая явно описывается уравнением

$$\Phi_k = z_k^{-1} n_k \|n_k\|^q, \quad q = 0, 1, 2, \dots \quad (3.33)$$

Как показано в приложении 4, если разрешимо неравенство (3.32), то из этого следует и разрешимость (3.20). Следовательно, выполняя синтез адаптивных алгоритмов по приведенной выше схеме, но используя вместо выражения (3.20) соотношения (3.32), (3.33), находим, что алгоритмы (3.30), (3.31) получают следующую модифицированную форму:

$$c_{1,k} = c_{1,k-1} + h_1 n_k \|n_k\|^q r_k, \quad (3.34)$$

$$c_{2,k} = c_{2,k-1} + H_2 n_k \|n_k\|^q y_k. \quad (3.35)$$

**Четвертый этап.** Достижимость поставленных целей управления и адаптации. В силу решения в системе управления (3.15), (3.16), (3.17) проблем положительности ЛСЧ и ННЧ, причем для любых начальных условий, и при наличии априорной неопределенности (3.5) эту систему, согласно критерию гиперустойчивости, следует считать асимптотически гиперустойчивой.

Таким образом, благодаря выполнению предельного соотношения

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (3.36)$$

цель управления вида (3.7) также имеет место.

При этом с учетом явного вида алгоритмов самонастройки коэффициентов регулятора, очевидно, будут выполнены предельные соотношения

$$\lim_{k \rightarrow \infty} c_{1,k} = const, \quad \lim_{k \rightarrow \infty} c_{2,k} = const, \quad (3.37)$$

отвечающие требованиям соответствующих целевых условий (3.8).

Если же вернуться от математического описания ГСПАУ, представленного в эквивалентном виде, к исходному описанию (уравнениям объекта управления, эталонной модели и адаптивного регулятора), то синтезированная ГСПАУ с алгоритмами (3.30), (3.31) математически будет выглядеть следующим образом:

$$\frac{dx(t)}{dt} = Ax(t) + bu(t) + f(t), \quad y(t) = x(t), \quad (3.38)$$

$$\frac{d\bar{x}(t)}{dt} = A_M \bar{x}(t) + b_M r(t), \quad \bar{y}(t) = \bar{x}(t), \quad (3.39)$$

$$e(t) = \bar{x}(t) - x(t), \quad z(t) = g^T e(t) \| e(t) \|^q, \quad (3.40)$$

$$u_k = c_{1,k} r_k + c_{2,k}^T y_k, \quad y_k = y(t_k), \quad (3.41)$$

$$c_{1,k} = c_{1,k-1} + h_1 z_k r_k, \quad h_1 = \text{const} > 0, \quad z_k = z(t_k), \quad (3.42)$$

$$c_{2,k} = c_{2,k-1} + H_2 z_k y_k, \quad (3.43)$$

$$H_2 = \text{diag}\{h_{2i}\}, \quad h_{2i} = \text{const} > 0, \quad i = \overline{1, l},$$

$$u(t) = u_k \text{ при } t_k \leq t < t_{k+1}. \quad (3.44)$$

При этом система управления с алгоритмами (3.34), (3.35) описывается уравнениями:

$$\frac{dx(t)}{dt} = Ax(t) + bu(t) + f(t), \quad y(t) = L^T x(t), \quad (3.45)$$

$$\frac{d\bar{x}(t)}{dt} = A_M \bar{x}(t) + b_M r(t), \quad \bar{y}(t) = L^T \bar{x}(t), \quad (3.46)$$

$$n(t) = g^T (\bar{y}(t) - y(t)), \quad (3.47)$$

$$u_k = c_{1,k} r_k + c_{2,k}^T y_k, \quad y_k = y(t_k), \quad (3.48)$$

$$c_{1,k} = c_{1,k-1} + h_1 n_k \| n_k \|^q r_k, \quad h_1 = \text{const} > 0, \quad (3.49)$$

$$c_{2,k} = c_{2,k-1} + H_2 n_k \| n_k \|^q y_k, \quad n_k = n(t_k), \quad (3.50)$$

$$H_2 = \text{diag}\{h_{2i}\}, \quad h_{2i} = \text{const} > 0, \quad i = \overline{1, l},$$

$$u(t) = u_k \text{ при } t_k \leq t < t_{k+1}. \quad (3.51)$$

**Пятый этап.** Следует отметить, что ГСПАУ вида (3.38) – (3.44) имеет сложную техническую реализацию, если нелинейные преобразования осуществлять на аналоговой технике (следуя рассмотренным выше процедурам синтеза управляющего контура системы адаптации), что весьма затруднительно – в силу ограниченности функциональных возможностей аналоговых вычислительных устройств. "Аналоговый" вариант технической реализации ГСПАУ, представленный как S-модель системы (3.38) – (3.44), показан на рис. 3.1, а явный вид S-моделей цифрового и аналогового вычислительных устройств изображен соответственно на рис. 3.2 и 3.3.

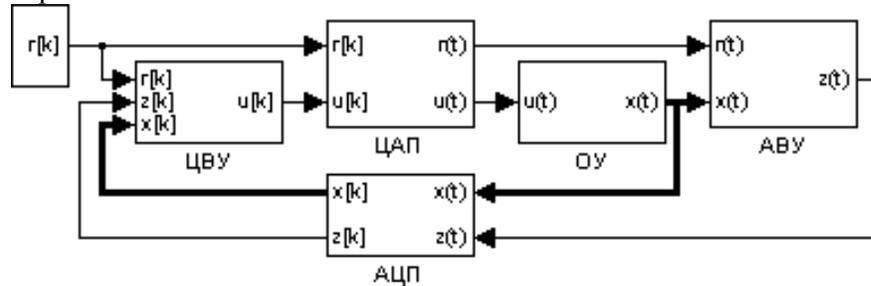


Рис. 3.1. S-модель ГСПАУ в виде системы (3.38) – (3.44).

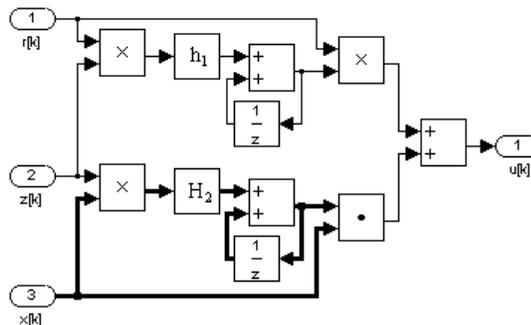


Рис. 3.2. ЦВУ системы управления (3.38) – (3.44).



Имитационное моделирование проводилось при нулевых начальных условиях и следующих исходных данных:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -23 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad A_M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -15 & -23 & -9 \end{pmatrix} \quad (3.59)$$

$$b_M^T = (0 \ 0 \ 1), \quad g^T = (8 \ 6 \ 1), \quad r_k = 0.1 \sin(kt) + 1,$$

представлены на рис. 3.6, 3.7, где  $\Delta_1$  и  $\Delta_2$  – рассогласования соответственно значений вектора состояний и значений управляющих воздействий в системах (3.38) – (3.44) и (3.52) – (3.58), где шаг дискретизации и коэффициенты  $h_1$ ,  $H_2$  были выбраны со значениями:

$$t = 8.8 \text{ с}, \quad h_1 = 18.47, \quad H_2 = \text{diag}\{195160; 40000; 8000000\}. \quad (3.60)$$

На рис. 3.8 показаны результаты моделирования рассогласования в системах (3.38) – (3.44) и (3.52) – (3.58) с шагом  $t = 1.6 \text{ с}$ , при этом коэффициенты контура адаптации были заданы со значениями:

$$h_1 = 280, \quad H_2 = \text{diag}\{110000; 10000; 200000\}. \quad (3.61)$$

На рис. 3.9 представлены результаты вычислительного эксперимента в системах (5.38) – (5.44), (5.52) – (5.58), но уже для другого объекта управления и следующих данных:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & -3 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad A_M = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix}, \quad b_M = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad t = 1, \quad (3.62)$$

$$g = \begin{pmatrix} 1.5 \\ 1 \end{pmatrix}, \quad r_k = 0.4 \sin(kT) + 1, \quad h_1 = 2.5, \quad H_2 = \text{diag}\{15, 9\}.$$

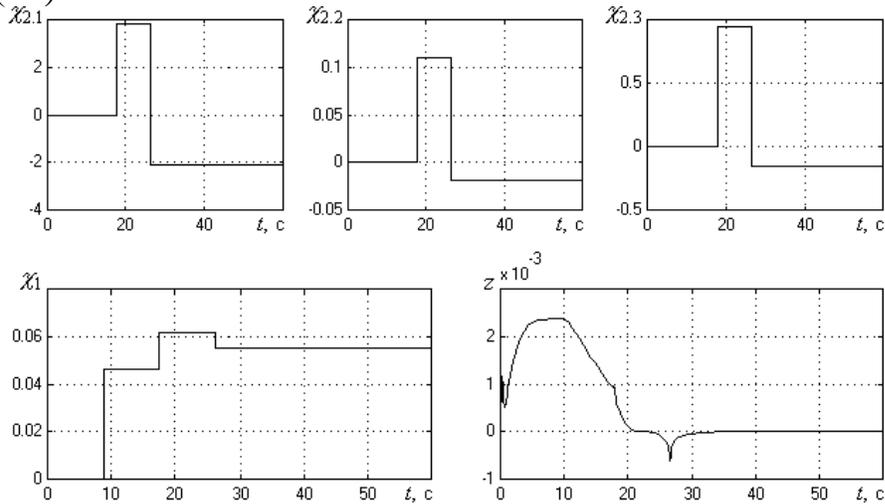


Рис. 3.6. Временные характеристики настроек контура адаптации и динамика обобщенного выхода системы управления (3.52) – (3.58).

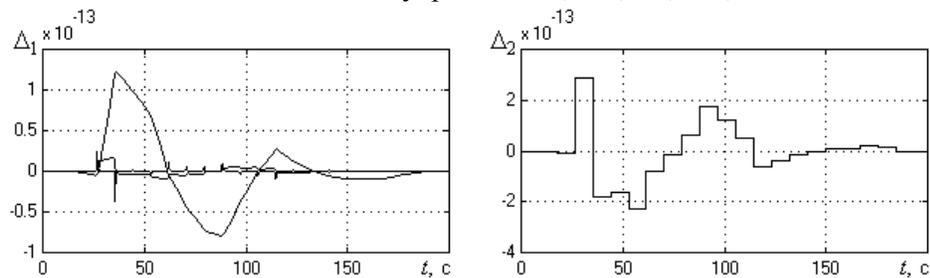


Рис. 3.7. Динамика рассогласований вектора состояний и управляющих воздействий в системах (3.38) – (3.44), (3.52) – (3.58), (3.60), при  $\tau = 8.8 \text{ с}$ .

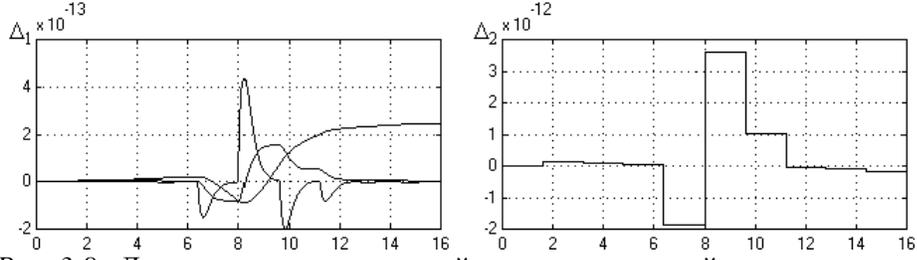


Рис. 3.8. Динамика рассогласований вектора состояний и управляющих воздействий в системах (3.38) – (3.44), (3.52) – (3.58), (3.61), при  $t = 1.6$  с.

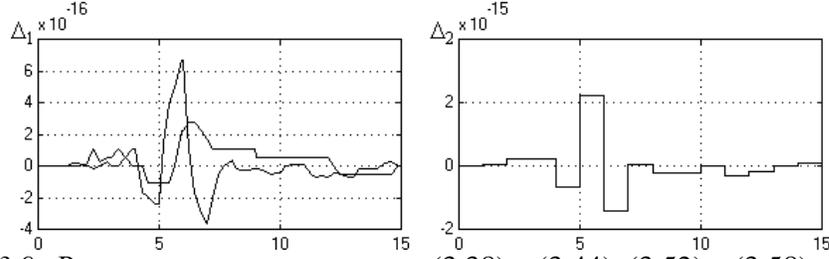


Рис. 3.9. Рассогласования в системах (3.38) – (3.44), (3.52) – (3.58), (3.62).

Результаты этих и других экспериментов показывают, что две ГСПАУ – (3.38) – (3.44) и (3.52) – (3.58) – технически эквивалентны.

Упрощение системы управления (3.45) – (3.51) имеет вид:

$$\frac{dx(t)}{dt} = Ax(t) + bu(t) + f(t), \quad y(t) = L^T x(t), \quad (3.63)$$

$$\bar{x}_{k+1} = P_M \bar{x}_k + d_M r_k, \quad \bar{y}_k = L^T \bar{x}_k, \quad (3.64)$$

$$\mathbf{n}_k = g^T (\bar{y}_k - y_k), \quad y_k = y(t_k), \quad (3.65)$$

$$u_k = c_{1,k} r_k + c_{2,k}^T y_k, \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (3.66)$$

$$c_{1,k} = c_{1,k-1} + h_1 \mathbf{n}_k \|\mathbf{n}_k\|^q r_k, \quad c_{2,k} = c_{2,k-1} + H_2 \mathbf{n}_k \|\mathbf{n}_k\|^q y_k, \quad (3.67)$$

$$h_1 = \text{const} > 0, \quad H_2 = \text{diag}\{h_{2i}\}, \quad h_{2i} = \text{const} > 0, \quad i = \overline{1, l} \quad (3.68)$$

S-модель ГСПАУ (3.63) – (3.68) показана на рис. 3.10.

### 3.2. Гибридные системы с векторным управлением

ГСПАУ с векторным управлением описываются уравнениями

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + f(t), \quad (3.69)$$

$$y(t) = L^T x(t), \quad (3.70)$$

$$u_k = c_{1,k} r_k + c_{2,k} y_k, \quad y_k = y(t_k), \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (3.71)$$

$$\frac{d\bar{x}(t)}{dt} = A_M \bar{x}(t) + B_M r(t), \quad \bar{y}(t) = L^T \bar{x}(t). \quad (3.72)$$

Синтез контура адаптации в системе (3.69) – (3.72) выполним, следуя методике, изложенной в предыдущем разделе.

Первый этап. Используя обозначение (3.14) с учетом соотношений (3.69) – (3.72), запишем эквивалентное математическое описание исследуемой системы в виде:

$$\frac{de(t)}{dt} = A_M e(t) + Bm(t), \quad \mathbf{n}(t) = G^T L^T e(t), \quad (3.73)$$

$$\mathbf{m}_k = (c_{1,*} - c_{1,k}) r_k + (c_{2,*} - c_{2,k}) y_k, \quad (3.74)$$

$$\mathbf{m}(t) = \mathbf{m}_k \text{ при } t_k \leq t < t_{k+1}, \quad (3.75)$$

где  $n(t) \in R^m$  – обобщенный выход эквивалентной системы;  $G$  – постоянная матрица, элементы которой подлежат выбору.

Второй этап. Для передаточной матрицы ЛСЧ системы (3.73) – (3.75), имеющей вид

$$W(l) = G^T L^T (lE - A_M)^{-1} B, \quad (3.76)$$

необходимо обеспечить выполнение условий ее строгой положительности и вещественности. Поскольку передаточные матрицы (3.76) и (1.35) полностью идентичны, выбор матрицы  $G$  может осуществляться аналогично тому, как это описано в разделе 1.4 при выборе матрицы  $G$  в уравнении (1.35). Будем полагать, что матрица  $G$  выбрана таким образом, что передаточная матрица (3.76) удовлетворяет требуемым условиям положительности и вещественности.

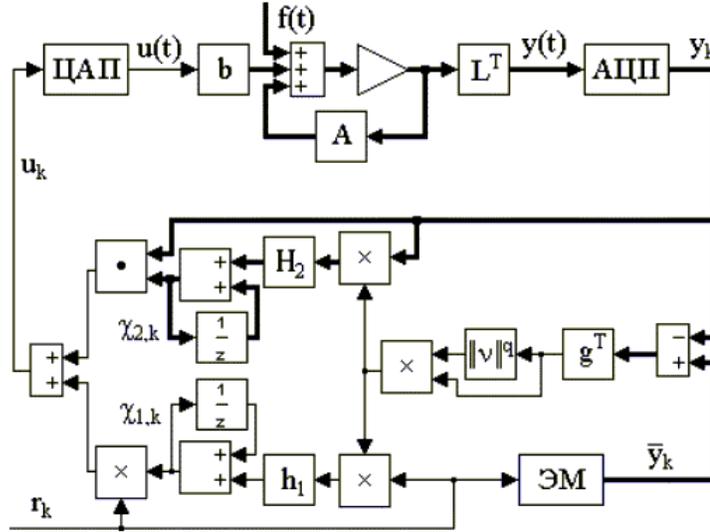


Рис. 3.10. S-модель ГСПАУ в виде системы (3.63) – (3.68).

Третий этап. Рассмотрим решение неравенства Попова, которое запишем так:

$$h(0, k_1) = -\sum_{k=0}^{k_1} m_k^T z_k \geq -g_0^2 = const, \quad \forall k_1 \geq 0, \quad (3.77)$$

где  $z_k = z(t_k)$ ;  $z(t) = G^T L^T e(t) \|e(t)\|^q$ .

Представляя неравенство (3.77) следующим образом:

$$h(0, k_1) = -\sum_{j=1}^m \sum_{k=0}^{k_1} m_{j,k} z_{j,k} \geq -g_0^2 = const, \quad \forall k_1 \geq 0 \quad (3.78)$$

и синтезируя алгоритмы адаптации в виде

$$c_{1ji,k} = c_{1ji,k-1} + h_{1ji} r_{i,k} z_{j,k}, \quad h_{1ji} > 0, \quad i, j = \overline{1, m}, \quad (3.79)$$

$$c_{2ji,k} = c_{2ji,k-1} + h_{2ji} y_{i,k} z_{j,k}, \quad h_{2ji} > 0, \quad i = \overline{1, l}, \quad j = \overline{1, m}, \quad (3.80)$$

можно показать, аналогично предыдущему, что неравенство (3.77) будет справедливо.

Для случая неполного измерения вектора состояния объекта управления представим неравенство Попова следующим образом:

$$h(0, k_1) = -\sum_{k=0}^{k_1} m_k^T n_k \|n_k\|^q \geq -g_0^2 = const, \quad \forall k_1 \geq 0, \quad n_k = n(t_k), \quad (3.81)$$

тогда алгоритмы адаптации получают вид

$$c_{1ji,k} = c_{1ji,k-1} + h_{1ji} r_{i,k} n_{j,k} \|n_k\|^q, \quad (3.82)$$

$$h_{1ji} = const > 0, \quad i, j = \overline{1, m},$$

$$\begin{aligned} c_{2ji,k} &= c_{2ji,k-1} + h_{2ji} y_{i,k} n_{j,k} \|n_k\|^q, \\ h_{2ji} &= \text{const} > 0, \quad i = \overline{1, l}, \quad j = \overline{1, m}. \end{aligned} \quad (3.83)$$

**Четвертый этап.** Проведение синтеза на заключительной стадии разработки ГСПАУ вида (3.73) – (3.75) или (3.69) – (3.72) с алгоритмами (3.79), (3.80) или (3.82), (3.83) не имеет принципиальных отличий от стандартной методики, применение которой гарантирует достижение в системе сформулированных целей управления в заданном классе адаптивности. Таким образом, аналитическая часть проектирования ГСПАУ завершена. Однако ряд числовых значений, в частности значения положительных констант, входящих в алгоритмы контура адаптации (3.79), (3.80) или (3.82), (3.83), требуют своего задания, что чаще всего осуществляется на этапе имитационного моделирования ГСПАУ вида (3.69) – (3.72), (3.79), (3.80) или (3.69) – (3.72), (3.82), (3.83).

**Пятый этап.** С целью упростить техническую реализацию синтезированной системы управления осуществим дискретизацию непрерывной части ее контура адаптации, тогда ГСПАУ (3.69) – (3.72) с алгоритмами (3.79), (3.80) получит вид:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + f(t), \quad y(t) = x(t), \quad (3.84)$$

$$\bar{x}_{k+1} = P_M \bar{x}_k + D_M r_k, \quad \bar{y}_k = \bar{x}_k, \quad (3.85)$$

$$e_k = \bar{x}_k - x_k, \quad x_k = x(t_k), \quad z_k = G^T e_k \|e_k\|^q, \quad (3.86)$$

$$u_k = c_{1,k} r_k + c_{2,k} y_k, \quad y_k = y(t_k), \quad (3.87)$$

$$c_{1ji,k} = c_{1ji,k-1} + h_{1ji} r_{i,k} z_{j,k}, \quad h_{1ji} > 0, \quad i, j = \overline{1, m}, \quad (3.88)$$

$$c_{2ji,k} = c_{2ji,k-1} + h_{2ji} y_{i,k} z_{j,k}, \quad h_{2ji} > 0, \quad i = \overline{1, l}, \quad j = \overline{1, m}, \quad (3.89)$$

$$u(t) = u_k \quad \text{при} \quad t_k \leq t < t_{k+1}, \quad (3.90)$$

где  $P_M = \exp(A_M t)$ ;  $D_M = A_M^{-1}(P_M - E)B_M$ .

S-модель ГСПАУ (3.84) – (3.90) изображена на рис. 3.11.

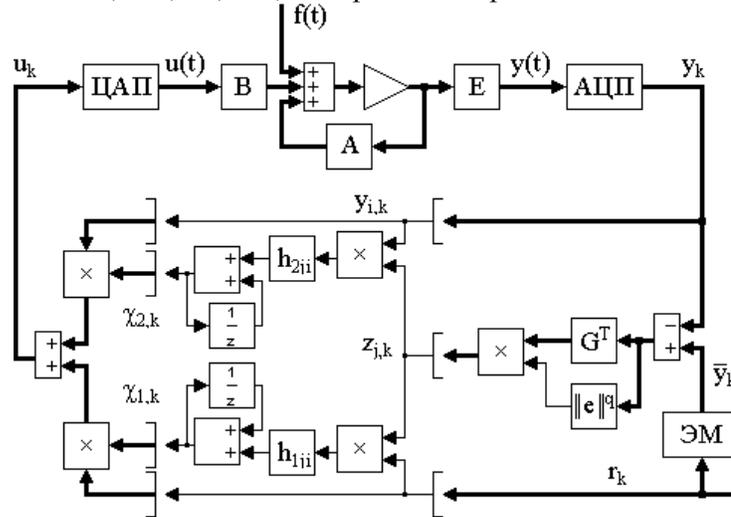


Рис. 3.11. S-модель ГСПАУ (3.84) – (3.90).

В свою очередь система с алгоритмами (3.82), (3.83) может быть описана в виде:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + f(t), \quad y(t) = L^T x(t), \quad (3.91)$$

$$\bar{x}_{k+1} = P_M \bar{x}_k + D_M r_k, \quad \bar{y}_k = L^T \bar{x}_k, \quad (3.92)$$

$$n_k = G^T (\bar{y}_k - y_k), \quad y_k = y(t_k), \quad (3.93)$$

$$u_k = c_{1,k} r_k + c_{2,k} y_k, \quad (3.94)$$

$$c_{1ji,k} = c_{1ji,k-1} + h_{1ji} r_{i,k} n_{j,k} \|n_k\|^q, \quad (3.95)$$

$$h_{1ji} = \text{const} > 0, \quad i, j = \overline{1, m},$$

$$c_{2ji,k} = c_{2ji,k-1} + h_{2ji} y_{i,k} n_{j,k} \|n_k\|^q, \quad (3.96)$$

$$h_{2ji} = \text{const} > 0, \quad i = \overline{1, l}, \quad j = \overline{1, m},$$

$$u(t) = u_k \quad \text{при} \quad t_k \leq t < t_{k+1}. \quad (3.97)$$

S-модель системы управления (3.91) – (3.97) представлена на рис. 3.12.

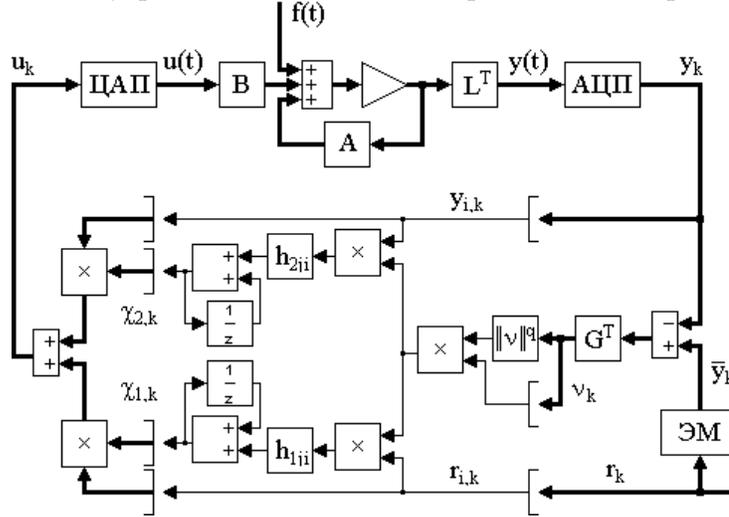


Рис. 3.12. S-модель ГСПАУ (3.91) – (3.97).

Адаптивные алгоритмы всех рассмотренных ГСПАУ разрабатывались в предположении, что возмущение  $f(t) \equiv 0$ , поэтому выясним их работоспособность при  $f(t) \neq 0$ .

При затухающих возмущениях вида (3.3) синтезированные системы сохраняют свою работоспособность. Действительно, из выполнения соотношения (3.3) следует справедливость соотношения  $\lim_{t \rightarrow \infty} f(t) = 0$ , с учетом которого выполнение целевого условия (3.7) не будет нарушено, т.к. действие возмущения является исчезающим во времени. В свою очередь при незатухающих, но ограниченных по норме помехах указанные алгоритмы адаптации оказываются неработоспособными. Для построения контура самонастройки, работоспособного в условиях постоянного действия помех, обычно проводят регуляризацию (огрубление) алгоритмов. Наиболее часто регуляризация алгоритмов достигается за счет введения в контур адаптации обратных связей.

Алгоритмы самонастройки, имеющие вид (3.42), (3.43), (3.49), (3.50), (3.88), (3.89), (3.95), (3.96), после регуляризации могут быть представлены следующими робастными алгоритмами:

для системы (3.38) – (3.41), (3.44) в виде уравнений

$$c_{1,k} = c_{1,k-1} + h_1 z_k r_k - a c_{1,k-1}, \quad (3.98)$$

$$h_1 = \text{const} > 0, \quad a = \text{const} \geq 0,$$

$$c_{2,k} = c_{2,k-1} + H_2 z_k y_k - b c_{2,k-1}, \quad (3.99)$$

$$H_2 = \text{diag}\{h_{2i}\}, \quad h_{2i} = \text{const} > 0, \quad b = \text{diag}\{b_i\}, \quad b_i = \text{const} \geq 0, \quad i = \overline{1, l};$$

для системы (3.45) – (3.48), (3.41) в виде соотношений

$$c_{1,k} = c_{1,k-1} + h_1 n_k \|n_k\|^q r_k - a c_{1,k-1}, \quad (3.100)$$

$$h_1 = \text{const} > 0, \quad a = \text{const} \geq 0,$$

$$c_{2,k} = c_{2,k-1} + H_2 n_k \|n_k\|^q y_k - b c_{2,k-1}, \quad (3.101)$$

$$H_2 = \text{diag}\{h_{2i}\}, h_{2i} = \text{const} > 0, \mathbf{b} = \text{diag}\{b_i\}, b_i = \text{const} \geq 0, i = \overline{1, l};$$

для системы (3.84) – (3.87), (3.90) в виде уравнений

$$\begin{aligned} c_{1ji,k} &= c_{1ji,k-1} + h_{1ji} r_{i,k} z_{j,k} - a_{ji} c_{1ji,k-1}, \\ h_{1ji} &= \text{const} > 0, a_{ji} = \text{const} \geq 0, i, j = \overline{1, m}, \end{aligned} \quad (3.102)$$

$$\begin{aligned} c_{2ji,k} &= c_{2ji,k-1} + h_{2ji} y_{i,k} z_{j,k} - b_{ji} c_{2ji,k-1}, \\ h_{2ji} &= \text{const} > 0, b_{ji} = \text{const} \geq 0, i = \overline{1, l}, j = \overline{1, m}; \end{aligned} \quad (3.103)$$

для системы (3.91) – (3.94), (3.97) в виде соотношений

$$\begin{aligned} c_{1ji,k} &= c_{1ji,k-1} + h_{1ji} r_{i,k} \mathbf{n}_{j,k} \|\mathbf{n}_k\|^q - a_{ji} c_{1ji,k-1}, \\ h_{1ji} &= \text{const} > 0, a_{ji} = \text{const} \geq 0, i, j = \overline{1, m}, \end{aligned} \quad (3.104)$$

$$\begin{aligned} c_{2ji,k} &= c_{2ji,k-1} + h_{2ji} y_{i,k} \mathbf{n}_{j,k} \|\mathbf{n}_k\|^q - b_{ji} c_{2ji,k-1}, \\ h_{2ji} &= \text{const} > 0, b_{ji} = \text{const} \geq 0, i = \overline{1, l}, j = \overline{1, m}. \end{aligned} \quad (3.105)$$

Применение алгоритмов, регуляризованных с помощью введения отрицательных обратных связей, имеет достаточно широкое распространение. Однако недостатком этого способа регуляризации алгоритмов является существенное ухудшение процесса управления при значительном уменьшении модуля настраиваемых параметров, что можно преодолеть за счет включения и отключения обратных связей. Рассмотренный способ построения робастных алгоритмов адаптации не единственно возможный, еще один способ регуляризации алгоритмов самонастройки основан на введении в алгоритмы адаптации зоны нечувствительности.

Применение робастных алгоритмов позволяет обеспечить работоспособность синтезированных ГСПАУ в условиях стохастических возмущений ограниченной интенсивности, а также в условиях нестационарности объекта управления.

#### 4. ПЕРЕЧЕНЬ УЧЕБНИКОВ, УЧЕБНЫХ ПОСОБИЙ

1. Еремин Е.Л., Еремина В.В., Семичевская Н.П., Шевко Д.Г. Алгоритмы и S-модели гибридных систем адаптивного управления (практикум в среде SIMULINK). Благовещенск: Амурский гос. ун-т, 2005. 205 с.
2. Еремин Е.Л., Еремин И.Е. Методы анализа динамических систем. Лабораторный практикум на ПЭВМ. / Благовещенск, Амурский гос. ун-т - 2005. [http://libserv.amursu.ru/DiBooks/HTML\\_electronic\\_textbook](http://libserv.amursu.ru/DiBooks/HTML_electronic_textbook)
3. Афанасьев В.Н., Колмановский В.Б. Носов В.Р. Математическая теория конструирования систем управления. М.: Высш. Шк., 1989, 447 с.
4. Зимницкий В.А., Устинов С.М. Методы анализа математических моделей динамических систем: Учебное пособие. - Л.: Изд-во ЛГУ, 1991. - 81 с.
5. Андриевский Б.Р., Фрадков А.Л. Избранные главы теории автоматического управления с примерами на языке МАТЛАВ. – СПб.: Наука 1999. – 467.
6. Мирошник И.В., Никифоров В.О. Фрадков А.Л. Нелинейное и адаптивное управление сложными объектами. – СПб.: Наука 2000. – 549.
7. Учебные пособия:
8. Еремин Е.Л. Теоретические основы автоматизированного управления. Курс лекций для специальности 220200. 2005. - [http://libserv.amursu.ru/DiBooks/HTML\\_electronic\\_textbook](http://libserv.amursu.ru/DiBooks/HTML_electronic_textbook)
9. Еремин Е.Л. Лабораторно-курсовой практикум по ТОАУ с применением Matlab for Windows: Учебное пособие. - Благовещенск: Изд-во АмГУ, 1998. - 114 с.
10. Еремин Е.Л. Практикум по ТОАУ с применением Matlab for Windows. - [http://libserv.amursu.ru/DiBooks/HTML\\_electronic\\_textbook](http://libserv.amursu.ru/DiBooks/HTML_electronic_textbook)

## 5. ПЛАН-КОНСПЕКТ ЛЕКЦИЙ

- Лекция 1.** Формы представления и преобразования моделей линейных динамических систем в пространстве состояний.  
Прямое, последовательное и параллельное программирование.  
Эквивалентное прямое программирование.
- Лекция 2.** Векторно-матричная форма представления типовых форм записи линейных моделей, описываемых обыкновенными дифференциальными уравнениями.
- Лекция 3.** Математические модели линейных динамических систем при случайных воздействиях и анализ их свойств.  
Дискретные случайные величины и их характеристики.  
Функция и плотность распределения непрерывной случайной величины.  
Векторные случайные величины.
- Лекция 4.** Основные понятия и характеристики непрерывных случайных процессов (корреляционные функции, спектральные плотности).  
Корреляционные функции и спектральные плотности случайных стационарных процессов.
- Лекция 5.** Прохождение случайных сигналов через динамическую систему.  
Анализ погрешностей, вносимых случайными процессами в САУ.
- Лекция 6.** Расчеты по минимуму среднеквадратической ошибки.  
Вывод уравнения Винера-Хопфа.
- Лекция 7.** Гармоническая линеаризация нелинейностей.  
Коэффициенты гармонической линеаризации релейных элементов.  
Алгебраические способы определения автоколебаний в нелинейных системах 1<sup>ого</sup> типа.  
Устойчивость периодических решений.
- Лекция 8.** Частотный метод определения автоколебаний.
- Лекция 9.** Статические и скоростные ошибки автоколебательных систем.  
Оценка качества в нелинейных системах регулирования.
- Лекция 10.** Решение задач оптимального управления с использованием метода множителей Лагранжа (классический метод вариационного исчисления).  
Уравнения Эйлера и Эйлера-Лагранжа.  
Задача оптимального управления с закрепленными концами и фиксированным временем.

- Лекция 11.** Стандартная форма записи уравнений Эйлера-Лагранжа.  
Правило множителей Лагранжа.  
Пример оптимального управления двойным интегратором.  
Задача оптимального управления с закрепленным и подвижными концами.  
Условия трансверсальности.
- Лекция 12.** Задача оптимального управления с подвижными концами и нефиксированным временем.  
Принцип максимума Понтрягина, условие нормальности.  
Теорема об  $n$ -интервалах. Задачи с несколькими ограничениями.
- Лекция 13.** Прямой метод Ляпунова в адаптивном управлении.  
Синтез систем адаптивного управления с неявной эталонной моделью.
- Лекция 14.** Разработка систем адаптивного управления с неявной эталонной моделью для объектов с запаздыванием по состоянию.
- Лекция 15.** Алгоритмы робастного управления с явной эталонной моделью.  
Синтез робастного закона управления на основе прямого метода Ляпунова  
Построение алгоритмов нелинейного робастного управления на основе критерия гиперустойчивости.

## 6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ И ЗАДАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

### Тема курсовой работы – «Создание matlab приложений»

Приложения MATLAB являются графическими окнами, содержащими элементы управления (кнопки, списки, переключатели, флаги, полосы скроллинга, области ввода, меню), а также оси и текстовые области для вывода результатов работы. Создание приложений включает следующие основные этапы – расположение нужных элементов интерфейса в пределах графического окна и определение действий (команд MATLAB), которые выполняются при обращении пользователя к данным объектам, – например, при нажатии кнопки. Процесс работы над приложением допускает постепенное добавление элементов в графическое окно, запуск и тестирование приложения и возврат в режим редактирования. Конечным результатом является программа с графическим интерфейсом пользователя (GUI), содержащаяся в нескольких файлах, запуск ее производится указанием ее имени в командной строке MATLAB или в другом приложении.

#### 6.1. Графический редактор в среде GUIDE

Перейдите в среду GUIDE, для этого введите в командной строке MATLAB команду `guide`. Появляется редактор окна приложения (рис. 5.1), заголовок которого `untitled.fig`<sup>1</sup> означает, что в нем открыт новый файл.

Редактор приложения содержит:

строку меню;

панель инструментов управления приложением;

заготовку окна приложения с нанесенной сеткой;

вертикальную и горизонтальную линейки;

панель инструментов для добавления элементов интерфейса на окно приложения.

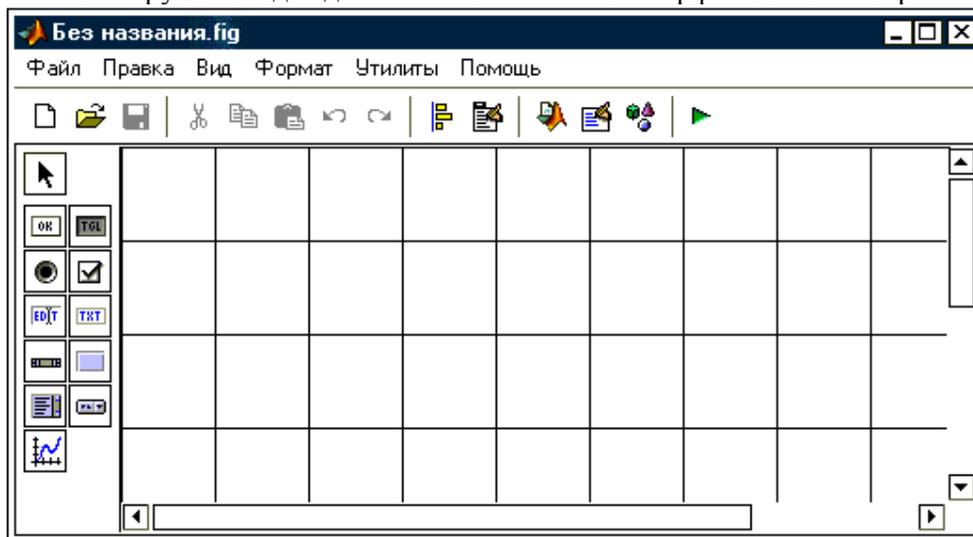


Рис. 5.1. Редактор приложения в версии 6.5.0.

Редактор приложения MATLAB 6.x позволяет разместить различные элементы интерфейса. Для этого требуется нажать соответствующую кнопку на панели инструментов и поместить выбранный объект щелчком мыши в требуемое место заготовки окна приложе-

---

<sup>1</sup>Здесь и далее используется руссифицированная версия MATLAB 6.5.0.

ния. Другой способ состоит в задании прямоугольной области объекта перемещением мыши по области заготовки окна с удержанием левой кнопки. Размер и положение добавленных объектов изменяются при помощи мыши. Перед изменением размера следует выбрать режим выделения объектов и сделать его текущим, щелкнув по нему клавишей мыши.

Любой объект можно удалить из окна при помощи <Delete>, предварительно выделив его. Запуск приложения производится кнопкой Activate Figure. Появляется диалоговое окно GUIDE, которое сообщает о необходимости сохранить приложение. Нажмите Yes и сохраните приложение в файле с расширением fig.

Приложение запускается в отдельном окне с заголовком Untitled. Пользователь может нажимать на кнопки, устанавливать флаги, переключатели, обращаться к спискам. Разумеется, при этом ничего полезного не происходит. Более того, при нажатии на кнопки в командное окно MATLAB выводится сообщение следующего содержания:

```
PushButton1 Callback not implemented yet.
```

Данное сообщение говорит о том, что обработка события, которое происходит при нажатии на кнопку, не запрограммировано. Недостаточно разместить элементы интерфейса в окне приложения, следует позаботиться о том, чтобы каждый элемент выполнял нужные функции при обращении к нему пользователя. Например, при нажатии на кнопку производятся вычисления и строятся графики полученных результатов, переключатели позволяют установить цвет линий, полоса скроллинга изменяет толщину линии, в области ввода пользователь указывает некоторые параметры, управляющие ходом вычислений.

Программирование событий в версии 6.x. Рассмотрим принцип программирования событий в версии 6.x. Приложение MATLAB 6.x хранится в двух файлах с расширением fig и m, первый из них содержит информацию о размещенных в окне приложения объектах, а второй является m-файлом с основной функцией и подфункциями. Добавление элемента интерфейса из редактора приложения приводит к автоматическому созданию соответствующей подфункции. Данную подфункцию следует наполнить содержимым – операторами, которые выполняют обработку события, возникающего при обращении пользователя к элементу интерфейса.

Начните с создания простого приложения, окно которого содержит оси и две кнопки, предназначенные для построения графика и очистки осей.

Перейдите в среду приложения командой guide. Выберите в меню Tools редактора приложений пункт Application Options, появится диалоговое окно GUIDE Application Options. Данное окно позволяет устанавливать некоторые общие свойства создаваемого приложения. Выберите в раскрывающемся списке Command-line accessibility строку On (recommended for plotting windows) и нажмите ОК. Приложение, выводящее графики на оси, которые расположены в пределах окна приложения, требуют установки этой опции для корректной работы.

Расположите на форме оси кнопку. На кнопке автоматически размещается надпись Push Button. Следующий этап очень важный. Кнопка является элементом интерфейса, ей следует дать имя, которое уникальным образом идентифицировало бы ее среди всех объектов приложения.

Выделите кнопку Push Button и вызовите при помощи панели инструментов управления приложением редактор свойств Property Inspector. Появляется окно редактора свойств, в котором содержится таблица названий свойств кнопки и их значений. Занесите в свойство Tag значение btnPlot, щелкните мышью по строке справа от названия свойства, наберите требуемое значение и нажмите <Enter>. В дальнейшем будет говориться, что некоторому объекту или элементу управления следует дать имя. Например, btnPlot теперь является именем кнопки Push Button. Удобно задавать имена, часть которых определяет тип элемента управления (btn соответствует button – кнопке). Аналогичным образом дайте осям имя axMain.

Выберите в меню File редактора приложения пункт Save as, создайте папку My-

FirstGui и сохраните приложение в файле mygui.fig. Обратите внимание, что открылся редактор М-файлов, содержащий файл mygui.m.

Приложение mygui содержит одну кнопку Press Button. Когда пользователь нажимает на Press Button в работающем приложении, то происходит событие Callback данного элемента управления. Вызывается подфункция btnPlot\_Callback. Сейчас она содержит оператор вывода в командное окно текста с предупреждением о том, что событие Callback пока не определено. Имя подфункции образовано названием кнопки и события. Очень важно задавать имена объектам в свойстве Tag сразу после их добавления в окно приложения в редакторе приложений, иначе генерируемая подфункция получит имя, которое сохранится при последующем изменении значения Tag и повлечет ошибки при выполнении приложения.

Завершающий этап состоит в программировании действий, которые выполняются при нажатии пользователем на кнопку Press Button. Измените функцию обработки события нажатием на Press Button в соответствии с листингом 1.

Листинг 1. Обработка события Callback кнопки с именем btnPlot.

```
function varargout = btnPlot_Callback (h, eventdata, handles, varargin)
x = [-2:0.2:2];
y = exp(-x.^2);
plot(x,y)
```

Сохраните файл mygui.m в редакторе М-файлов и запустите приложение из редактора приложений, нажав кнопку Activate Figure. Нажатие на Press Button в запущенном приложении приводит к отображению графика функции на осях. Закройте окно приложения при помощи кнопки с крестиком в правом верхнем углу и продолжите работу над mygui в редакторе приложений.

Добавьте кнопку, задайте ей имя btnClear в редакторе свойств. Быстрый доступ к свойствам выделенного объекта в редакторе приложений производится из пункта Inspect Properties всплывающего меню при нажатии правой кнопки мыши на объекте. Перейдите в подфункции обработки события Callback добавленной кнопки, для чего следует выбрать пункт Edit Callback всплывающего меню. Выбор данного пункта делает активным редактор М-файлов и выделяет заголовок соответствующей подфункции btnClear\_Callback. Разместите единственный оператор очистки cla в подфункции (листинг 2).

Листинг 2. Обработка события кнопки с именем btnClear.

```
function varargout = btnClear_Callback (h, eventdata, handles, varargin)
cla
```

Запустите приложение и убедитесь, что нажатие на левую кнопку приводит к отображению графика функции, а правая служит для очистки осей.

Следует позаботиться о том, чтобы интерфейс приложения был очевиден для пользователя.

### Конструирование интерфейса в версии 6.x.

*Управления свойствами объектов.* Разработка приложения сопряжена с изменением свойств объектов, которые они получают по умолчанию при размещении их на заголовке окна. Некоторые из свойств, например, надпись на кнопке или ее размер, устанавливаются при создании объекта в режиме редактирования. Другие свойства могут изменяться программно в работающем приложении.

*Установка свойств при редактировании.* Продолжите работу над приложением mygui. Очевидно, что следует написать кнопки, – например, Построить и Очистить. Кнопки являются графическими объектами с определенными свойствами, среди которых имеется свойство, отвечающее за надпись на кнопке. Сделайте левую кнопку приложения mygui текущей и вызовите редактор свойств Property Inspector. Установите свойство String левой кнопки в значение Построить. Важно понять, что значение свойства String соответствует надписи на кнопке, а Tag – имени или тегу кнопки как объекта. Имена объектов используются для изменения их свойств в ходе работы приложения при выполнении блоков обра-

ботки событий от других элементов интерфейса. Перейдите теперь к свойствам правой кнопки и установите String в Очистить.

Следует отметить, что доступ к редактору свойств выделенного объекта производится либо из панели инструментов управления приложением, либо из меню Tools редактора приложений, либо при помощи пункта Inspect Properties всплывающего меню.

Значение свойства String сразу отображается на кнопке приложения, находящегося в режиме редактирования. Запустите приложение, кнопки теперь снабжены надписями, которые определяются соответствующими значениями свойств String. Аналогичным образом устанавливаются и другие свойства объектов.

*Программное изменение свойств.* Большинство свойств объектов можно устанавливать программно прямо в ходе работы приложения для обеспечения согласованного поведения элементов управления. Усовершенствуйте приложение `mugui` следующим образом. Пусть при запуске доступной является только кнопка Построить. При нажатии на кнопку Построить выводится график, и она становится недоступной, зато пользователь может нажать кнопку Очистить для очистки осей, и наоборот. Итак, всегда доступна только одна из кнопок, в зависимости от состояния осей.

Решение поставленной задачи требует привлечения свойства Enable. Это свойство объекта отвечает за возможность доступа к нему пользователя, значение `on` разрешает доступ, а `off`, соответственно, запрещает. Установка значений свойствам объектов в программе производится при помощи функции `set`.

Функция `set` вызывается с тремя входными аргументами – указателем на объект, названием свойства и его значением, последние два аргумента заключаются в апострофы. Обратите внимание, что свойства одного объекта должны изменяться в блоке операторов обработки события `Callback` другого объекта. Следовательно, должна иметься возможность доступа к указателю на любой существующий объект. Аргументы `h` и `handles` подфункций обрабатывают события элементов управления, содержат требуемые указатели. В `h` хранится указатель на тот объект, событие которого обрабатывается в данный момент, а `handles` является структурой указателей. Поля структуры совпадают со значениями свойств `Tag` существующих элементов интерфейса. Например, `handles.btnPlot` является указателем на кнопку Построить с именем `btnPlot`.

Доступ к Очистить должен быть запрещен в начале работы приложения, пока пользователь не нажмет Построить. Установите в редакторе свойств для кнопки Очистить `Enable` в `Off`, используйте кнопку со стрелкой в строке со значением свойства. Остальные изменения значения `Enable` кнопок должны происходить в ходе работы приложения. Для разрешения и запрещения доступа к кнопкам нужно внести дополнения в обработку их событий `Callback`.

В подфункцию обработки события `Callback` кнопки Построить добавьте при помощи редактора вызовов:

установку свойства `Enable` кнопки Очистить в значение `on` (после вывода графика следует разрешить доступ к Очистить);

установку свойства `Enable` кнопки Построить в значение `off` (после вывода графика следует запретить доступ к Построить).

Аналогичные изменения произведите в обработке события `Callback` кнопки Очистить, а именно:

установку свойства `Enable` кнопки Построить в значение `on` (после вывода графика следует разрешить доступ к Построить);

установку свойства `Enable` кнопки Очистить в значение `off` (после вывода графика следует запретить доступ к кнопке).

Подфункции `btnPlot_Callback` и `btnClear_Callback` должны быть запрограммированы так, как показано на листинге:

```

function varargout = btnPlot_Callback (h, eventdata, handles, varargin)
% Построение графика функции
x = [-2:0.2:2];
y = exp(-x.^2);
plot (x,y)
% Кнопка Построить должна стать недоступной после вывода
% графика
set (h, 'Enable', 'off')
% Кнопка Очистить должна быть доступной
set (handles.btnClear, 'Enable', 'on')
function varargout = btnClear_Callback (h, eventdata, handles, varargin)
cla % очистка осей
% Кнопка Очистить должна стать недоступной после очистки осей
set (h, 'Enable', 'off')
Кнопка Построить должна быть доступной
set (handles.btnPlot, 'Enable', 'on')

```

Сохраните изменения в редакторе m-файлов. Запустите приложение `mygui` и убедитесь, что всегда доступной является только одна из кнопок – Построить или Очистить, что является хорошей подсказкой для пользователя о возможных действиях. Закройте окно приложения и редактор приложений. Следующий раздел посвящен запуску приложения из командной строки и переходу в режим редактирования.

*Работа над приложением.* Запуск приложения осуществляется не только из редактора приложений. Возникают вопросы: как работать с уже созданным приложением с графическим интерфейсом и вносить в него требуемые изменения? Для запуска приложения достаточно в качестве команды задать его имя в командной строке.

Появляется окно приложения, обращение к элементам интерфейса окна приводит к соответствующим действиям.

Заметим, что каталог с приложением должен содержаться в путях поиска MATLAB или являться текущим.

Очень часто сразу не удается написать законченное приложение, необходимое усовершенствование появляется только в ходе работы с приложением. В любой момент можно продолжить редактирование двумя способами:

задать `guide` в командной строке, что приводит к отображению редактора приложений с заготовкой для нового приложения `untitled.fig`, и открыть соответствующий файл с расширением `fig`, выбрав пункт `Open...` в меню `File` редактора (закройте ненужный редактор приложений с `untitled.fig`);

указать имя приложения через пробел после команды `guide`, – например, `guide mygui` появляется редактор приложений, в котором открыто `mygui`.

Перейдите в режим редактирования приложения `mygui` любым из перечисленных способов и продолжите работу над ним. Измените название окна приложения на «Визуализация функций».

Заголовок окна задается свойством `Name` графического окна. Сделайте графическое окно текущим, щелкнув мышью по области заготовки окна в редакторе приложений. Установите в редакторе свойств `Name` в «Визуализация функций». Запустите `mygui`, появляется окно `GUIDE` с предупреждением, что продолжение повлечет сохранение приложения. Установите флаг `Don't tell me again` для подавления данного предупреждения при дальнейшей работе и нажмите `OK`. Убедитесь, что приложение имеет нужный заголовок.

Обобщая изложенные выше сведения, можно сказать, что процесс программирования приложения с графическим интерфейсом включает этапы:

размещение элементов интерфейса в окне приложения;

программирование событий данных элементов в подфункциях m-файла приложения;

сохранение приложения;

запуск и работа над приложением;

переход в режим редактирования для дальнейшего усовершенствования приложения.

Желательно располагать элементы интерфейса в порядке, обеспечивающем удобную работу пользователя с приложением. Приложение хорошо выглядит, если однотипные элементы, – например, кнопки, флаги и т.д., определенным образом выровнены в окне приложения. MatLab 6.x предоставляет разработчику приложений простые способы выравнивания добавляемых объектов – сетку и линейку.

*Оформление интерфейса.* Редактор приложений содержит заготовку окна приложения с нанесенной сеткой, а также и вертикальную и горизонтальную линейки. Линейка позволяет размещать элементы интерфейса в позиции с любыми координатами в пикселях, отсчитываемыми от левого нижнего угла заготовки окна приложения. Следует выделить объект щелчком мыши и перемещать его по окну, следя за указателями его положения на линейке. Происходит непрерывное движение объекта, что не всегда удобно.

Часто требуется, чтобы небольшое перемещение мыши вызывало изменение положения объекта на некоторый фиксированный шаг. Сетка редактора приложений позволяет осуществить такое дискретное движение. Выбор пункта Grid and Rules меню Layout приводит к появлению диалогового окна Grid and Rules.

Флаги Show rulers и Show grid соответствуют отображению линеек и сетки в редакторе приложений, а раскрывающийся список Grid Size позволяет выбрать размер ячеек сетки. Минимально допустимый размер десять пикселей позволяет достаточно точно располагать элементы управления в окне приложений. Привязка перемещения к линиям сетки происходит при установленном флаге Snap to grid. Привязка разрешает разместить объект и измерить его размеры только при условии прохождения границы объекта по линиям сетки. Выбор мелкого шага сетки в сочетании с привязкой предоставляет разработчику возможность быстро оформить приложение. Плавно изменять положение выделенного объекта можно при помощи клавиш со стрелками. Одновременное удержание <Ctrl> приводит к перемещению с учетом привязки к сетке.

Размещение объектов в ряд по вертикали или горизонтали требует предварительного определения некоторой вспомогательной линии. Можно, конечно, использовать линии сетки, но проще добавить горизонтальные или вертикальные линии выравнивания. Следует навести курсор мыши на соответствующую линейку (курсор меняет форму на двустороннюю стрелку), и движение объектов к линии выравнивания вызывает автоматическое расположение их границ на данной линии. Сами линии выравнивания можно убирать с окна приложения, перетаскивая их мышью обратно на линейку. Snap Show guides окна Grid and Rules отвечает за отображение линий выравнивания в области окна приложения.

Отметим, что привязка действует даже при выключенных флагах Show rulers, Show guides и Show grid.

Размер окна приложения изменяется при помощи перетаскивания мышью квадратика, расположенного в правом нижнем углу заготовки окна приложения.

Отредактируйте вид приложения тугui, используя описанные возможности. Следующие разделы посвящены размещению и программированию основных элементов интерфейса.

#### Программирование элементов интерфейса.

*Флаги и рамки.* Флаги позволяют произвести одну или несколько установок, определяющих ход работы приложения. Продолжите работу над тугui, предоставьте пользователю возможность наносить линии сетки на график. Окно приложения должно содержать два флага с названиями Сетка по x и Сетка по y. Если пользователь нажимает кнопку Построить, то на оси наносится сетка по выданным координатам. Нажатие на Очистить должно приводить не только к исчезновению графика функции, но и скрытию сетки.

Обычно несколько элементов управления со схожими назначениями группируются и

помещаются внутри рамки. Измените размеры осей, освободив справа место для рамки. Нанесите рамку на окно приложения при помощи соответствующей кнопки. В рамку добавьте два флага.

Разместите поясняющие подписи рядом с флагами и дайте им имена. Задайте свойству Tag верхнего флага значение `chbxGridX`, а свойству String, отвечающему за подпись флага, значение Сетка по x. Аналогичным образом определите свойства нижнего флага, установите свойство Tag в `chbxGridY`, String – в Сетка по y. Если текст не помещается рядом с флагом, увеличьте ширину области флага при помощи мыши, удерживая нажатой левую кнопку. Сохраните приложение в редакторе приложений для автоматического создания в редакторе m-файлов заготовок для подфункций обработки событий добавленных объектов.

Осталось сделать так, чтобы при нажатии пользователем кнопки Построить происходило отображение линий сетки в зависимости от установленных флагов, а нажатие на Очистить приводило к скрытию сетки. Блок обработки события Callback кнопки Построить следует дополнить проверкой состояния флагов. Свойство флага Value принимает значение логической единицы при включении флага пользователем и, соответственно, равно нулю, если флаг выключен. Указатели на флаги содержатся на полях `chbxGridX` и `chbxGridY` структуры `handles`.

Состояние флагов определяет значение свойств `Xgrid` и `Ygrid` осей.

Произведите необходимые изменения в подфункции обработки события Callback кнопки Построить с именем `btnPlot`.

Листинг. Обработка события кнопки `btnPlot` с учетом состояния флагов.

```
function varargout = btnPlot_Callback (h, eventdata, handles, varargin)
% Построение графика функции
x = [-2:0.2:2];
y = exp(-x.^2);
plot (x,y)
% проверка флага Сетка по x
if get (handles.chbxGridX, 'Value')
% Флаг включен, следует добавить линии по сетки
set (gca, 'Xgrid', 'on')
else
% Флаг выключен, следует убрать линии сетки
set (gca, 'Xgrid', 'off')
% проверка флага Сетка по y
if get (handles.chbxGridY, 'Value')
% Флаг включен, следует добавить линии по сетки
set (gca, 'Ygrid', 'on')
else
%Флаг выключен, следует убрать линии сетки
set (gca, 'Ygrid', 'off')
end
% Кнопка Построить должна стать недоступной после вывода графика
set (h, 'Enable', 'off')
% Кнопка Очистить должна стать доступной
set (handles.btnClear, 'Enable', 'on')
```

Запустите приложение `mygui` и убедитесь, что установка флагов влияет на отображение сетки при нажатии на кнопку Построить.

Смена состояния флагов сетки не приводит к немедленным изменениям на графике. Пользователь должен перестроить график, нажимая последовательно кнопки Очистить и Построить. Для немедленного реагирования приложения на состояние флагов следует определить их события Callback. Программирование данных событий заключается в проверке

состояния флага и отображении или скрытии соответствующих линий сетки.

Сделайте текущим флаг Сетка по x в редакторе приложений и выберите опцию Edit Callback во всплывающем меню данного объекта. Запрограммируйте событие Callback флага. Используйте аргумент h соответствующих подфункций, содержащий указатель на объект, событие которого обрабатывается в текущий момент времени. Аналогичным образом обработайте событие Callback второго флага Сетка по y.

Листинг. Обработка событий Callback флагов сетки.

```
function varargout = cnbxGridX_Callback (h, eventdata, handles, varargin)
% Проверка флага Сетка по x
if get (h, 'Value')
% Флаг включен, следует добавить линии сетки
set (gca, 'Xgrid', 'on')
else
% Флаг выключен, следует убрать линии сетки
set (gca, 'Xgrid', 'off')
end
function varargout = cnbxGridY_Callback (h, eventdata, handles, varargin)
% Проверка флага Сетка по y
if get (h, 'Value')
% Флаг включен, следует добавить линии сетки
set (gca, 'Ygrid', 'on')
else
% Флаг выключен, следует убрать линии сетки
set (gca, 'Ygrid', 'off')
end
```

Внесенные дополнения позволяют пользователю наносить и убирать сетку по каждой координате при помощи флагов, без перестроения графика функции.

Флаги предоставляют пользователю возможность выбора одной или сразу нескольких опций. Выбор только одной опции осуществляется при помощи переключателей.

*Переключатели.* Переключатели обычно группируются по их предназначению, и пользователь может выбрать только одну опцию. Всегда установлен единственный переключатель из группы. Обработка событий переключателя должна влиять на состояние остальных переключателей группы. Модернизируйте интерфейс приложения *mygui*, предоставьте пользователю возможность выбирать тип маркера (кружок, квадрат ли отсутствие маркера).

Добавьте в окно приложения новую рамку и нанесите на нее три переключателя, установите свойствам Tag значения `rbMarkCirc`, `rbMarkSq`, `rbMarkNone`, а String – соответственно маркеры-круги, маркеры-квадраты, без маркеров.

Состояние переключателя определяется его свойством Value. Если Value равно единице, то переключатель включен, ноль – нет. Задайте в редакторе свойств значение единицы свойству Value переключателя с надписью «без маркеров», он будет включен при запуске программы. Значение свойства Value в версии MATLAB 6.x устанавливается следующим образом. Выделите переключатель и перейдите к его свойствам. В редакторе свойств нажмите кнопку в строке с Value. Появляется окно Value.

Выделите при помощи мыши строку со значением 0.0 и перейдите в режим редактирования значения двойным щелчком мыши. Измените 0.0 на единицу и нажмите ОК. Обратите внимание, что в редакторе свойств значение Value изменилось на единицу и включился переключатель без маркеров на окне приложения в редакторе приложений. Описанным способом устанавливаются значения Value в редакторе свойств. Дальнейшее управление значением Value переключателей должно осуществляться программно в ходе работы приложения *mygui*.

Предположим, что пользователь установил один из переключателей. Происходит обращение к соответствующей подфункции обработки события Callback переключателя, которая обеспечивает:

- изменение типа маркеров линии;
- выключение двух остальных переключателей.

Второе действие программируется достаточно просто – следует занести в Value значение ноль при помощи функции set и указателей на нужные переключатели, хранящихся в структуре handles. Изменение типа маркеров линии не представляет труда: если известен указатель на линию, то достаточно обратиться к свойству линии Marker. Указатель на линию возвращает set в выходном аргументе, его следует записать в некоторую переменную, – например, Hline. Использовать указатель на линию придется в других подфункциях, обрабатывающих событие Callback переключателей. Очень важно понять, что переменная Hline инициализируется при вызове подфункции btnPlot\_Callback, выполняются операторы подфункции, и она заканчивает работу. Все переменные, определенные в подфункции, являются локальными и по окончании работы подфункции не определены.

Обмен данными между подфункциями проще всего осуществить при помощи структуры handles. Подфункция, передающая данные, должна содержать запись данных в новое поле и сохранение структуры командой guidata. Входной аргумент – структура handles всех подфункций теперь содержит обновленное поле, в которое занесено соответствующее значение. Например, в некоторой подфункции можно сохранить массив в поле dat1 структуры handles

```
handles.dat1 = [1.2 3.2 0.1];  
guidata(gcbo, handles)
```

Будем использовать его в другой подфункции  
(handles.dat1)

аналогичным образом сохраняются указатели на объекты, создаваемые в разделах подфункции. Внесите необходимые изменения в подфункцию btnPlot\_Callback и запрограммируйте обработку событий callback переключателей в подфункциях rbMarkCirc\_Callback, rbMarkSq\_Callback, rbMarkNone\_Callback:

сохраните указатель на линию в поле Line структуры handles при построении графика командой plot;

добавьте блоки обработки событий Callback переключателей, каждый из которых устанавливает остальные два переключателя в положение «выключено» и наносит на линию соответствующие маркеры.

Листинг. Обработка событий переключателя в myguiprog.

```
function varargout = btnPlot_Callback(h, eventdata, handles, varargin)  
% Построение графика функции  
x = [-2:0.2:2];  
y = exp(-x.^2);  
% Запись указателя в поле line структуры handles  
handles.line = plot(x,y);  
% Сохранение структуры handles для использования  
% в других подфункциях  
guidata(gcbo, handles);  
...  
function varargout = rbMarkCirc_Callback(h, eventdata, handles, varargin)  
% выбран переключатель маркеры-круги  
set(handles.line, 'Marker', 'o') % размещение маркеров-кругов  
% на линии  
% Установка остальных переключателей в положение выключено  
set(handles.rbMarkSq, 'Value', 0)
```

```

set(handles.rbMarkNone, 'Value', 0)
function varargout = rbMarkSq_Callback(h, eventdata, handles, varargin)
% выбран переключатель маркеры-квадраты
set(handles.line, 'Marker', 's') % размещение маркеров-квадратов
% на линии
% Установка остальных переключателей в положение выключено
set(handles.rbMarkCirc, 'Value', 0)
set(handles.rbMarkNone, 'Value', 0)

```

```

function varargout = rbMarkNone_Callback(h, eventdata, handles, varargin)
% выбран переключатель без маркеров
set(handles.line, 'Marker', 'none') % удаление маркеров с линии
% Установка остальных переключателей в положение выключено
set(handles.rbMarkCirc, 'Value', 0)
set(handles.rbMarkSq, 'Value', 0)

```

Запустите приложение `mygui`, отобразите график функции, нажав кнопку Построить, и убедитесь в том, что возможна установка только одного из переключателей и она приводит к появлению соответствующих маркеров на графике функции. Однако пока еще интерфейс `mygui` имеет ряд недостатков.

Если пользователь использует кнопку Очистить для очистки осей, а затем устанавливает переключатель, то производится обращение к несуществующему объекту линии (сообщение об ошибке выводится в командном меню).

Нажатие на кнопку Построить приводит к получению линии без маркеров вне зависимости от установленного переключателя.

Повторный щелчок по области переключателя вызывает его выключение, но всегда один из переключателей должен быть установлен.

Конечно, первый недостаток является существенным – приложение должно работать без ошибок! Проще всего запретить доступ к переключателям, если нет линии на графике, и разрешить после ее появления. Очевидно, что следует внести изменения в соответствующие подфункции `myguiprog`, обрабатывающие события `Ca;back` кнопок. Нажатие на кнопку Построить должно открыть доступ к группе переключателей, а очистка осей кнопкой Очистить – запрещать доступ. Итак, следует найти указатели на переключатели и установить их свойство `Enabled` в нужное значение `on` или `off`. При запуске приложения все переключатели должны быть недоступны, т.к. пользователь не построил график функции.

Редактор свойств позволяет одновременно установить значение общих свойств (например, `Enable`) целой группы объектов. Выделите щелчком мыши один из переключателей, остальные добавляйте в группу щелчком мыши, удерживая клавишу `<Ctrl>`. В результате должны быть выделены все три переключателя. Теперь перейдите в редактор свойств при помощи всплывающего меню. Вверху окна редактора размещена надпись `Multiply objects selected`, означающая, что проделываемые установки произойдут для свойств сразу всех выделенных объектов. Установите `Enable off`, при запуске приложения `mygui` переключатели недоступны. Осталось дополнить подфункции обработки события `Callback` кнопок Построить и Очистить для программного управления свойством `Enable`. Обратитесь к листингу, содержащему требуемые операторы.

Листинг. Разрешение и запрещение доступа к группе переключателей.

```

function varargout = btnPlot_Callback(h, eventdata, handles, varargin)
% Все переключатели должны стать доступными после
% появления графика
set(handles.rbMarkCirc, 'Enable', 'on')
set(handles.rbMarkSq, 'Enable', 'on')
set(handles.rbMarkNone, 'Enable', 'on')

```

```
%-----
function varargout = btnClear_Callback (h, eventdata, handles, varargin)
% Все переключатели должны стать недоступными после
% очистки осей
set (handles.rbMarkCirc, 'Enable', 'off')
set (handles.rbMarkSq, 'Enable', 'off')
set (handles.rbMarkNone, 'Enable', 'off')
```

Первый недостаток интерфейса приложения `mygui` устранен. Теперь необходимо сделать так, чтобы при построении графика тип маркера отвечал установленному переключателю. Очевидно, что после вывода графика следует найти переключатель со значением `Value`, равным единице, и установить соответствующий тип маркера, согласно листингу.

Листинг. Изменение маркеров при построении графика.

```
function varargout = btnPlot_Callback (h, eventdata, handles, varargin)
...
if get (handles.rbMarkCirc, 'Value')
% Установлен переключатель маркеры-круги
set (handles.line, 'Marker', 'o')
end
if get (handles.rbMarkSq, 'Value')
% Установлен переключатель маркеры-квадраты
set (handles.line, 'Marker', 's')
end
```

Сохраните изменения в `M`-файле и запустите приложение `mygui`. Тип маркеров определяется установленным переключателем при построении графика.

Осталось нерешенной одна проблема. При повторном щелчке по области переключателя он выключается, но всегда должен быть установлен единственный переключатель. Данный недостаток устраняется с привлечением еще одного возможного значения свойства `Enable`. Переключатель со значением `inactive` является неактивным, он выглядит в работающем приложении как доступный переключатель (со значением `on`), но попытка изменить состояние данного переключателя не приводит к успеху. Усовершенствуйте обработку событий согласно следующему алгоритму.

Свойство `Enable` переключателя, событие которого обрабатывается, должно иметь значение `inactive`, а для остальных двух – `on`. Если не задать `on` для других переключателей, то в результате все они станут неактивными.

При нажатии на кнопку Построить свойству `Enable` всех переключателей присваивается значение `on`, а затем определяется установленный в данный момент переключатель и в `Enable` заносится `inactive`.

Выполните подфункции `btnPlot_Callback`, `rbMarkCirc_Callback`, `rbMarkSq_Callback` и `rbMarkNone_Callback` необходимыми операторами, согласно листингу.

Листинг.

```
function varargout = btnPlot_Callback (h, eventdata, handles, varargin)
...
% Поиск установленного переключателя и определение его,
% как неактивного
if get (handles.rbMarkCirc, 'Value')
set (handles.rbMarkCirc, 'Enable', 'inactive')
end
if get (handles.rbMarkSq, 'Value')
set (handles.rbMarkSq, 'Enable', 'inactive')
end
if get (handles.rbMarkNone, 'Value')
```

```

set (handles.rbMarkNone, 'Enable', 'inactive')
end
function varargout = rbMarkCirc_Callback (h, eventdata, handles, varargin)
% Переключатель, событие которого обрабатывается, должен стать
% неактивным, а остальные – активными
set (h, 'Enable', 'inactive')
set (handles.rbMarkSq, 'Enable', 'on')
set (handles.rbMarkNone, 'Enable', 'on')
function varargout = rbMarkSq_Callback (h, eventdata, handles, varargin)
% Переключатель, событие которого обрабатывается, должен стать
% неактивным, а остальные – активными
set (h, 'Enable', 'inactive')
set (handles.rbMarkCirc, 'Enable', 'on')
set (handles.rbMarkNone, 'Enable', 'on')
function varargout = rbMarkNone_Callback (h, eventdata, handles, varargin)
% Переключатель, событие которого обрабатывается, должен стать
% неактивным, а остальные – активными
set (h, 'Enable', 'inactive')
set (handles.rbMarkSq, 'Enable', 'on')
set (handles.rbMarkCirc, 'Enable', 'on')

```

Правильная обработка переключателей (см. листинги 12.4–12.7) требует учета всех ситуаций, которые могут возникнуть при взаимодействии пользователя с приложением. Раскрывающиеся списки реализуют альтернативный способ выбора пользователем только одной из прилагаемых опций.

*Списки.* Модернизируйте интерфейс приложения `туgui`, предоставьте пользователю возможность выбрать цвет линии графика из раскрывающегося списка (синий, красный, зеленый). Перейдите в режим редактирования и добавьте при помощи панели управления раскрывающийся список. В редакторе свойств установите свойство `Tag` в значение `'pmColor'`.

Элементами раскрывающегося списка являются строки, которые вводятся в редакторе свойств. Нажмите кнопку в строке со свойством `String` раскрывающегося списка, появляется окно `String`. Наберите в нем строки `'синий'`, `'красный'`, `'зеленый'` (без кавычек), разделяя их при помощи клавиши `<Enter>`.

Запустите `туgui` и убедитесь, что раскрывающийся список содержит требуемые строки. Выбор различных строк пока не приводит к изменению цвета линии – требуется запрограммировать событие `Callback` раскрывающегося списка.

Обработка события `Callback` раскрывающегося списка состоит в определении выбора пользователя и соответствующем изменении цвета линии. Свойство списка `Value` содержит номер выбранной строки (строки списка нумеруются с единицы). Перейдите к подфункции `pmColor_Callback` и запрограммируйте обработку выбора пользователя. Используйте оператор `switch` для установки цвета линии в зависимости от номера выбранной строки списка, согласно листингу.

Листинг. Обработка выбора пользователя из раскрывающегося списка.

```

function varargout = pmColor_Callback (h, eventdata, handles, varargin)
% Определение номера выбранной строки
Num = get(h, 'Value')
switch Num
case 1
% выбрана первая строка, следует сделать линию синей
set (handles.line, 'Color', 'b');
case 2

```

```

% выбрана вторая строка, следует сделать линию красной
set(handles.line, 'Color', 'r');
case 3
% выбрана третья строка, следует сделать линию зеленой
set(handles.line, 'Color', 'g');
end

```

Запустите приложение, постройте график, нажав на кнопку Построить, и убедитесь в том, что раскрывающийся список позволяет изменять цвет линии графика функции. Не сложно заметить, что интерфейс `mgui` имеет ряд недостатков.

Повторное построение графика не учитывает текущий выбор цвета в раскрывающемся списке.

Выбор цвета при отсутствии линии на графике приводит к ошибке (`handles.line` указывает на несуществующий объект).

Рядом со списком требуется разместить текст, поясняющий назначение списка.

Устраните первый недостаток, поместите в подфункции `btnPlot_Callback` обработки нажатия кнопки Построить блок `switch` для задания цвета построенной линии в зависимости от выбора опции раскрывающегося списка, согласно листингу.

Листинг. Учет выбора опции раскрывающегося списка при построении графика.

```

function varargout = btnPlot_Callback(h, eventdata, handles, varargin)
...
% Определение номера выбранной строки
case 1
set(handles.line, 'Color', 'b');
case 2
set(handles.line, 'Color', 'r');
case 3
set(handles.line, 'Color', 'g');
end

```

Изменение цвета линии при отсутствии графика лишено смысла, поэтому следует запретить доступ пользователя к раскрывающемуся списку и, напротив, разрешить при построении графика. В начале работы приложения список должен быть недоступен для пользователя. Установите в редакторе свойств для раскрывающегося списка `Enable` в `off`. Внесите необходимые изменения в подфункции `btnPlot_Callback` и `btnClear_Callback`, соответствующие нажатию на кнопки, используйте свойство списка `Enable`, согласно листингу.

Листинг. Разрешение и запрещение доступа к раскрывающемуся списку.

```

function varargout = btnPlot_Callback(h, eventdata, handles, varargin)
...
% Разрешение доступа к раскрывающемуся списку
set(handles.pmColor, 'Enable', 'on')
function varargout = btnClear_Callback(h, eventdata, handles, varargin)
...
% Запрещение доступа к раскрывающемуся списку
set(handles.pmColor, 'Enable', 'off')

```

Многие элементы интерфейса, в частности раскрывающиеся списки, следует сопровождать поясняющим текстом. Перейдите в режим редактирования и при помощи панели управления разместите текстовую область над списком. Установите в редакторе свойств `string` в значение цвет линии, `HorisontalAlignment` – в значение `left` для добавленного объекта, используйте кнопки в строках с названиями свойств. Теперь работающее приложение имеет более наглядный интерфейс.

Программирование событий обычных списков производится практически аналогично. Отличие состоит в том, что в обычных списках может быть выделено несколько элемен-

тов. Свойство Value содержит вектор номеров выбранных элементов. Разрешение выбора нескольких элементов определяется значениями свойств Max и Min. Если разность Max — Min больше единицы, то пользователь может выделить несколько строк.

*Полосы скроллинга.* Усовершенствуйте интерфейс приложения mygui, предоставив пользователю возможность устанавливать ширину линии при помощи полосы скроллинга. Добавьте полосу скроллинга в окно приложения и задайте название scrWidth в свойстве Tag полосы. Снабдите полосу скроллинга текстовым пояснением "Толщина линии" так же, как и раскрывающийся список.

Заметим, что простой щелчок мышью в окне приложения при добавлении полосы скроллинга приводит к появлению вертикально расположенной полосы. Вертикальное или горизонтальное направление зависит от соотношения ширины и высоты полосы скроллинга. Измените размер для получения горизонтально расположенной полосы. Проще всего сразу нарисовать прямоугольник полосы скроллинга, удерживая нажатой левую кнопку мыши.

Теперь следует определить соответствие между положением бегунка полосы числовым значением свойства Value.

Проделайте следующие установки из редактора свойств. В Max занесите десять, а в Min — единицу. Свойства Max и Min полосы скроллинга отвечают за границы значений, записываемых в Value, при перемещении бегунка.

Определите начальное положение, записав в Value единицу. Нажмите кнопку в строке с названием свойства и в появившемся окне Value измените значение на единицу.

Обратитесь к свойству SliderStep. Его значением является вектор из двух компонентов, первый из которых определяет относительное изменение Value при нажатии на кнопки со стрелками полосы скроллинга, а второй — при перетаскивании бегунка мышью. Следует установить значение [0.1 0.2] свойства SliderStep для того, чтобы нажатие на кнопки полосы изменяло Value на десять процентов, а щелчок мыши справа или слева от бегунка — на двадцать. Раскройте строку SliderStep щелчком мыши по знаку плюс слева от названия свойства и в появившихся строках x и y введите 0.1 и 0.2.

Удалось запрограммировать событие Callback полосы скроллинга с именем scrwidth, которое состоит в задании ширины линии, равной округленному значению Value. Перейдите к подфункции scrWidth\_Callback и добавьте в ней оператор установки ширины линии, согласно листингу.

Листинг. Обработка события Callback полосы скроллинга.

```
function varargout = scrWidth_Callback(h, eventdata, handles, varargin)
% Получение ширины линии в зависимости от бегунка
% на полосе скроллинга width = get(h, 'Value');
% Установка толщины линии
set(handles.line, 'LineWidth', round(width))
```

Запустите mygui и убедитесь, что полоса скроллинга позволяет легко изменять толщину линии построенного графика. Устраните самостоятельно некоторые недочеты интерфейса. Полоса скроллинга должна быть недоступной после очистки осей кнопкой Очистить, построение графика при помощи кнопки Построить произведите с учетом установленной ширины линии. Данные недостатки исправляются внесением соответствующих изменений в обработку событий callback перечисленных кнопок так же, как и в предыдущих разделах.

*Область ввода текста.* Обычные текстовые области, использовавшиеся на протяжении предыдущих разделов, позволяют лишь вывести некоторый текст в окно приложения. Обмен текстовой информацией между пользователем и приложением осуществляется при помощи областей ввода текста. Предоставьте пользователю возможность размещать заголовок на графике. Текст заголовка пользователь вводит в соответствующей строке.

Добавьте в окно приложения область ввода текста, установите значение editTitle свойству Tag области ввода и снабдите ее пояснением в текстовой области, расположенной

выше. В редакторе свойств удалите из String строку Edit Text, для чего нажмите кнопку в строке с названием свойства и сотрите текст в окне String.

Запрограммируйте событие callback области ввода, перейдите к подфункции editTitle\_Callback и поместите операторы, которые считывают значения свойства string области ввода и наносят заголовок на график, согласно листингу.

Листинг. Обработка события Callback области ввода текста.

```
function varargout = editTitle_Callback(h, eventdata, handles, varargin)
txt = get(h, 'String') ;
title(txt)
```

Запустите приложение mgu1 и нанесите заголовок на график, набрав его в области ввода и нажав клавишу <Enter>. Дополните обработку события, соответствующего нажатию кнопки Очистить, очисткой заголовка при помощи команды title (").

Описанный пример демонстрирует использование области ввода, состоящей из одной строки. Разность значений свойств Max и Min области ввода определяет, позволяет ли данная область ввод многострочного текста.

Если разность больше единицы, то заносимый пользователем текст записывается в массив ячеек текстовых строк, который хранится в свойстве String.

## 6.2. Задание и варианты курсовой работы

Для адаптивной системы управления общего вида

$$\begin{aligned} \frac{dx(t)}{dt} &= \Phi(x(t), x(t-h), u(t), \mathbf{x}) + f_x(t), \quad y(t) = L^T(\mathbf{x})x(t), \\ x(\mathbf{q}) &= j(\mathbf{q}), \quad \mathbf{q} \in [-h; 0], \\ u_k &= u_k(y_k, y_{k-s}, c_k, r_k), \\ y_k &= y(t_k), \quad y_{k-s} = y(t_{k-s}) = y(t_k - h), \\ u(t) &= u_k \quad \text{при } t_k \leq t < t_{k+1}, \end{aligned} \tag{5.1}$$

где  $x(t) \in R^n$  – вектор состояния объекта управления;  $y(t) \in R^l$  – вектор выхода;  $u(t) \in R^m$  – вектор управляющих воздействий;  $f(t) \in R^n$  – вектор возмущающих воздействий;  $c_k$  –

матрица настраиваемых коэффициентов регулятора;  $r_k \in R^m$  – вектор задающих воздействий;  $t_k = kt$  – дискретный аналог времени;  $t = const > 0$  – шаг дискретизации;  $k = 0, 1, 2, \dots$  – номер шага;  $\Phi(x(t), x(t-h), u(t), x)$  – отображение, удовлетворяющее условиям существования и единственности решения уравнений (1) при заданной начальной вектор-функции  $j(q) \in C_h$ ;  $C_h$  – пространство ограниченных непрерывных функций;  $h = const \geq 0$  – запаздывание;  $s = h/t$  – число тактов запаздывания;  $L(x)$  – квазистационарная матрица выхода соответствующего размера;  $x \in \Xi$  – набор всех неизвестных параметров;  $\Xi$  – известное множество возможных значений  $x$ , требуется:

1. Построить SIMULINK-модель (S-модель) гибридной системы адаптивного управления с помощью пакета MATLAB (согласно заданному варианту, используя структурную схему, математическое описание и исходные данные, приведенные в табл. 5.1).

Обеспечить возможность наблюдения следующих функций времени S-модели:  $x(t) \in R^n$ ;  $y(t) \in R^l$ ;  $e_k \in R^n$ ;  $u_k \in R^m$ ;  $f(t) \in R^n$ ;  $C_k$ ;  $r_k \in R^m$ . S-модель представить в виде файла с именем – N\_НомерВарианта\_НомерСхемы\_1.mdl.

2. Сформировать блок S-функции из заданного состава элементов S-модели (см. соответствующую структурную схему) и повторно выполнить все задания п.1. S-модель совместно с S-функцией представить в файле – N\_НомерВарианта\_НомерСхемы\_2.mdl.

3. Объединить S-модели, построенные в п.1 и 2, представив их в виде схемы согласования для выходов систем управления. S-модель представить в файле N\_НомерВарианта\_НомерСхемы\_3.mdl.

4. Разработать графический интерфейс для адаптивной системы управления в виде S-модели (см. результаты п.2), используя возможности подсистемы Guide пакета MATLAB.

5. Интерфейс должен позволять: просматривать результаты моделирования и справочную информацию; изменять время моделирования, параметры объекта управления и эталонной модели, значения коэффициентов контура адаптации, а также шаг дискретизации.

Таблица 5.1.

Номер варианта	Структурная схема	Уравнения математического описания	Состав элементов S-модели, входящих в блок S-функции	Числовые параметры математической модели
1	2	3	4	5
1	рис. 5.2	(5.2) – (5.8)	1	(5.9)
2	рис. 5.2	(5.2) – (5.8)	2	(5.9)
3	рис. 5.2	(5.2) – (5.8)	3	(5.9)
4	рис. 5.2	(5.2) – (5.8)	4	(5.9)
5	рис. 5.2	(5.2) – (5.8)	5	(5.9)
6	рис. 5.2	(5.2) – (5.8)	1-2	(5.9)
7	рис. 5.2	(5.2) – (5.8)	1-3	(5.9)
8	рис. 5.2	(5.2) – (5.8)	2-3	(5.9)
9	рис. 5.2	(5.2) – (5.8)	2-4	(5.9)
10	рис. 5.2	(5.2) – (5.8)	3-4	(5.9)
11	рис. 5.2	(5.2) – (5.8)	3-5	(5.9)
12	рис. 5.2	(5.2) – (5.8)	4-5	(5.9)

Продолжение табл. 5.1.

1	2	3	4	5
13	рис. 5.3	(5.10) – (5.16)	1	(5.17)
14	рис. 5.3	(5.10) – (5.16)	2	(5.17)
15	рис. 5.3	(5.10) – (5.16)	3	(5.17)
16	рис. 5.3	(5.10) – (5.16)	4	(5.17)
17	рис. 5.3	(5.10) – (5.16)	5	(5.17)
18	рис. 5.3	(5.10) – (5.16)	1-2	(5.17)
19	рис. 5.3	(5.10) – (5.16)	1-3	(5.17)
20	рис. 5.3	(5.10) – (5.16)	2-3	(5.17)
21	рис. 5.3	(5.10) – (5.16)	2-4	(5.17)
22	рис. 5.3	(5.10) – (5.16)	3-4	(5.17)
23	рис. 5.3	(5.10) – (5.16)	3-5	(5.17)
24	рис. 5.3	(5.10) – (5.16)	4-5	(5.17)
25	рис. 5.4	(5.18) – (5.24)	1	(5.25)
26	рис. 5.4	(5.18) – (5.24)	2	(5.25)
27	рис. 5.4	(5.18) – (5.24)	3	(5.25)
28	рис. 5.4	(5.18) – (5.24)	4	(5.25)
29	рис. 5.4	(5.18) – (5.24)	5	(5.25)
30	рис. 5.4	(5.18) – (5.24)	1-2	(5.25)
31	рис. 5.4	(5.18) – (5.24)	1-3	(5.25)
32	рис. 5.4	(5.18) – (5.24)	2-3	(5.25)
33	рис. 5.4	(5.18) – (5.24)	2-4	(5.25)
34	рис. 5.4	(5.18) – (5.24)	3-4	(5.25)
35	рис. 5.4	(5.18) – (5.24)	3-5	(5.25)
36	рис. 5.4	(5.18) – (5.24)	4-5	(5.25)
37	рис. 5.5	(5.26) – (5.32)	1	(5.33)
38	рис. 5.5	(5.26) – (5.32)	2	(5.33)
39	рис. 5.5	(5.26) – (5.32)	3	(5.33)
40	рис. 5.5	(5.26) – (5.32)	4	(5.33)
41	рис. 5.5	(5.26) – (5.32)	5	(5.33)
42	рис. 5.5	(5.26) – (5.32)	1-2	(5.33)
43	рис. 5.5	(5.26) – (5.32)	1-3	(5.33)
44	рис. 5.5	(5.26) – (5.32)	2-3	(5.33)
45	рис. 5.5	(5.26) – (5.32)	2-4	(5.33)
46	рис. 5.5	(5.26) – (5.32)	3-4	(5.33)
47	рис. 5.5	(5.26) – (5.32)	3-5	(5.33)
48	рис. 5.5	(5.26) – (5.32)	4-5	(5.33)
49	рис. 5.6	(5.34) – (5.38)	1	(5.39)
50	рис. 5.6	(5.34) – (5.38)	2	(5.39)
51	рис. 5.6	(5.34) – (5.38)	3	(5.39)
52	рис. 5.6	(5.34) – (5.38)	4	(5.39)
53	рис. 5.6	(5.34) – (5.38)	1-2	(5.39)

Продолжение табл. 5.1.

1	2	3	4	5
54	рис. 5.6	(5.34) – (5.38)	1-3	(5.39)
55	рис. 5.6	(5.34) – (5.38)	2-3	(5.39)
56	рис. 5.6	(5.34) – (5.38)	2-4	(5.39)
57	рис. 5.6	(5.34) – (5.38)	3-4	(5.39)
58	рис. 5.7	(5.40) – (5.44)	1	(5.45)
59	рис. 5.7	(5.40) – (5.44)	2	(5.45)
60	рис. 5.7	(5.40) – (5.44)	3	(5.45)
61	рис. 5.7	(5.40) – (5.44)	4	(5.45)
62	рис. 5.7	(5.40) – (5.44)	1-2	(5.45)
63	рис. 5.7	(5.40) – (5.44)	1-3	(5.45)
64	рис. 5.7	(5.40) – (5.44)	2-3	(5.45)
65	рис. 5.7	(5.40) – (5.44)	2-4	(5.45)
66	рис. 5.7	(5.40) – (5.44)	3-4	(5.45)
67	рис. 5.8	(5.46) – (5.53)	1	(5.54)
68	рис. 5.8	(5.46) – (5.53)	2	(5.54)
69	рис. 5.8	(5.46) – (5.53)	3	(5.54)
70	рис. 5.8	(5.46) – (5.53)	4	(5.54)
71	рис. 5.8	(5.46) – (5.53)	5	(5.54)
72	рис. 5.8	(5.46) – (5.53)	1-2	(5.54)
73	рис. 5.8	(5.46) – (5.53)	1-3	(5.54)
74	рис. 5.8	(5.46) – (5.53)	1-4	(5.54)
75	рис. 5.8	(5.46) – (5.53)	2-3	(5.54)
76	рис. 5.8	(5.46) – (5.53)	2-5	(5.54)
77	рис. 5.8	(5.46) – (5.53)	3-4	(5.54)
78	рис. 5.8	(5.46) – (5.53)	3-5	(5.54)
79	рис. 5.8	(5.46) – (5.53)	4-5	(5.54)
80	рис. 5.9	(5.55) – (5.62)	1	(5.63)
81	рис. 5.9	(5.55) – (5.62)	2	(5.63)
82	рис. 5.9	(5.55) – (5.62)	3	(5.63)
83	рис. 5.9	(5.55) – (5.62)	4	(5.63)
84	рис. 5.9	(5.55) – (5.62)	5	(5.63)
85	рис. 5.9	(5.55) – (5.62)	1-2	(5.63)
86	рис. 5.9	(5.55) – (5.62)	1-3	(5.63)
87	рис. 5.9	(5.55) – (5.62)	1-4	(5.63)
88	рис. 5.9	(5.55) – (5.62)	2-3	(5.63)
89	рис. 5.9	(5.55) – (5.62)	2-5	(5.63)
90	рис. 5.9	(5.55) – (5.62)	3-4	(5.63)
91	рис. 5.9	(5.55) – (5.62)	3-5	(5.63)
92	рис. 5.9	(5.55) – (5.62)	4-5	(5.63)

1-я ГСПАУ с ЯЭМ ( $n = 3, l = 3, m = 1$ ).

$$\frac{dx(t)}{dt} = Ax(t) + bu(t) + f(t), \quad y(t) = x(t), \quad (5.2)$$

$$\bar{x}_{k+1} = P_M \bar{x}_k + d_M r_k, \quad \bar{y}_k = \bar{x}_k, \quad (5.3)$$

$$e_k = \bar{x}_k - x_k, \quad z_k = g^T e_k \|e_k\|^q, \quad x_k = x(t_k), \quad (5.4)$$

$$u_k = c_{1,k} r_k + c_{2,k}^T x_k, \quad (5.5)$$

$$c_{1,k} = c_{1,k-1} + h_1 z_k r_k, \quad c_{2,k} = c_{2,k-1} + H_2 z_k x_k, \quad (5.6)$$

$$h_1 = \text{const} > 0, \quad H_2 = \text{diag}\{h_{2i}\}, \quad h_{2i} = \text{const} > 0, \quad i = \overline{1, n}, \quad (5.7)$$

$$u(t) = u_k \text{ при } t_k \leq t < t_{k+1}. \quad (5.8)$$

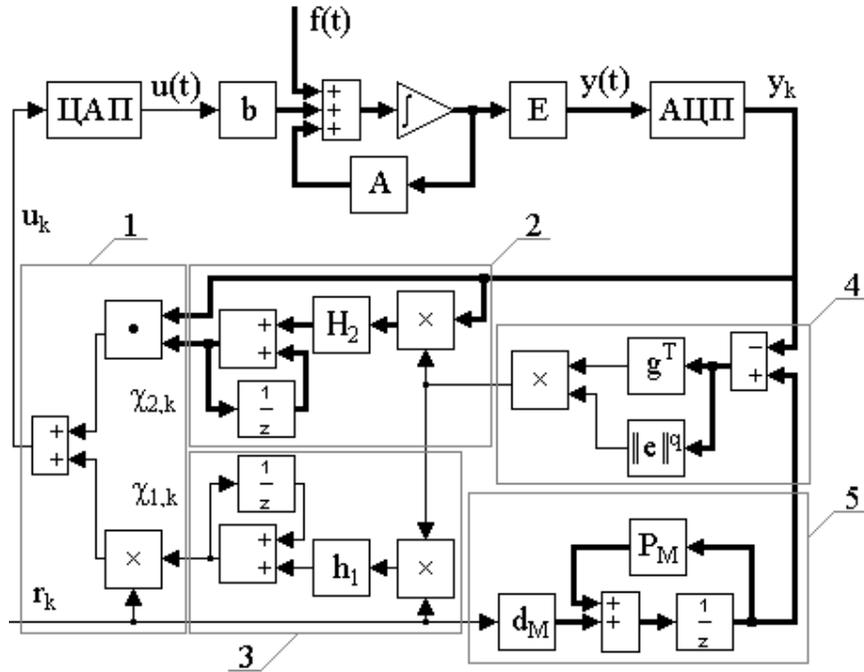


Рис. 5.2. Схема системы (5.2) – (5.8).

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -23 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad f(t) = \begin{pmatrix} 0 \\ 0 \\ \exp(-t) \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad g = \begin{pmatrix} 8 \\ 6 \\ 1 \end{pmatrix},$$

$$P_M = \begin{pmatrix} 0.6301 & 0.2966 & 0.0344 \\ -0.5157 & -0.1607 & -0.0129 \\ 0.1928 & -0.2200 & -0.0450 \end{pmatrix}, \quad d_M = \begin{pmatrix} 0.0247 \\ 0.0344 \\ -0.0129 \end{pmatrix}, \quad \bar{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.9)$$

$$r_k = 0.8 \sin(kt) - 1.1, \quad t = 1, \quad q = 2, \quad h_1 = 320,$$

$$H_2 = \text{diag}\{120000 \quad 50000 \quad 200000\}.$$

2-я ГСПАУ с ЯЭМ ( $n = 3, l = 2, m = 1$ ).

$$\frac{dx(t)}{dt} = Ax(t) + bu(t) + f(t), \quad y(t) = L^T x(t), \quad (5.10)$$

$$\bar{x}_{k+1} = P_M \bar{x}_k + d_M r_k, \quad \bar{y}_k = L^T \bar{x}_k, \quad (5.11)$$

$$v_k = g^T (\bar{y}_k - y_k), \quad y_k = y(t_k), \quad (5.12)$$

$$u_k = c_{1,k} r_k + c_{2,k} y_k, \quad (5.13)$$

$$c_{1,k} = c_{1,k-1} + h_1 n_k \|n_k\|^q r_k, \quad c_{2,k} = c_{2,k-1} + H_2 n_k \|n_k\|^q y_k, \quad (5.14)$$

$$h_1 = \text{const} > 0, \quad H_2 = \text{diag}\{h_{2i}\}, \quad h_{2i} = \text{const} > 0, \quad i = \overline{1, l}, \quad (5.15)$$

$$u(t) = u_k \text{ при } t_k \leq t < t_{k+1}. \quad (5.16)$$

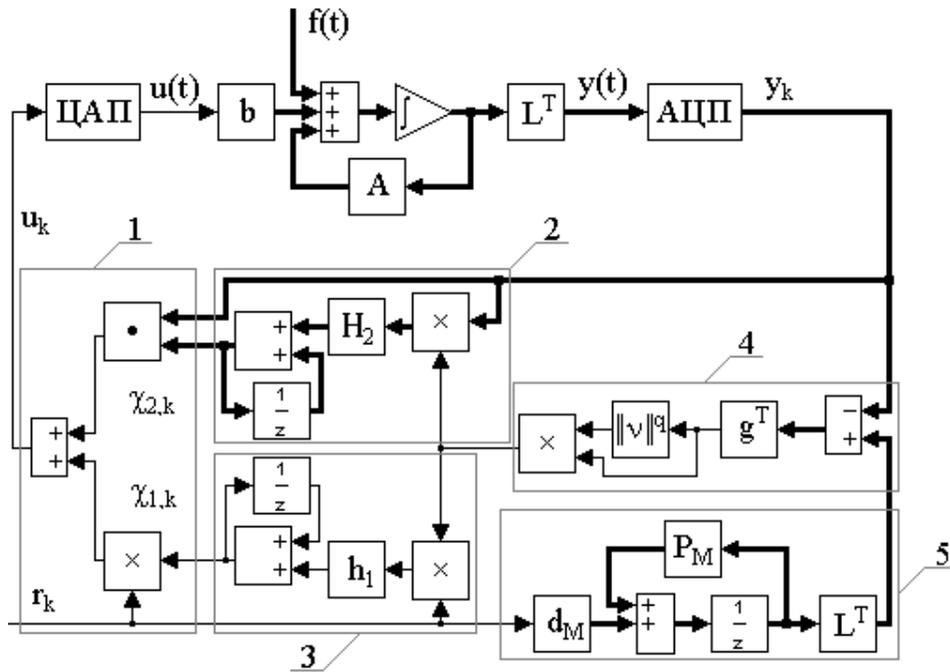


Рис. 5.3. Схема системы (5.10) – (5.16).

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -23 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad L = \begin{pmatrix} 4 & 0 \\ 4 & -2 \\ 0 & 1 \end{pmatrix}, \quad f(t) = \begin{pmatrix} 0 \\ 0 \\ \exp(-t) \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$P_M = \begin{pmatrix} 0.6301 & 0.2966 & 0.0344 \\ -0.5157 & -0.1607 & -0.0129 \\ 0.1928 & -0.2200 & -0.0450 \end{pmatrix}, \quad d_M = \begin{pmatrix} 0.0247 \\ 0.0344 \\ -0.0129 \end{pmatrix}, \quad \bar{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (5.17)$$

$$g = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad r_k = 0.1 \sin(kt) + 1, \quad t = 1, \quad q = 2, \quad h_1 = 110, \quad H_2 = \text{diag}\{3450 \quad 0.6\}.$$

3-я ГСПАУ с ЯЭМ ( $n = 3, l = 3, m = 3$ ).

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + f(t), \quad y(t) = x(t), \quad (5.18)$$

$$\bar{x}_{k+1} = P_M \bar{x}_k + D_M r_k, \quad \bar{y}_k = \bar{x}_k, \quad (5.19)$$

$$e_k = \bar{x}_k - x_k, \quad x_k = x(t_k), \quad z_k = G^T e_k \|e_k\|^q, \quad (5.20)$$

$$u_k = c_{1,k} r_k + c_{2,k} y_k, \quad y_k = y(t_k), \quad (5.21)$$

$$c_{1ji,k} = c_{1ji,k-1} + h_{1ji} r_{i,k} z_{j,k}, \quad h_{1ji} = \text{const} > 0, \quad i, j = \overline{1, m}, \quad (5.22)$$

$$c_{2ji,k} = c_{2ji,k-1} + h_{2ji} y_{i,k} z_{j,k}, \quad h_{2ji} = \text{const} > 0, \quad i = \overline{1, l}, \quad j = \overline{1, m}, \quad (5.23)$$

$$u(t) = u_k \text{ при } t_k \leq t < t_{k+1} \quad (5.24)$$

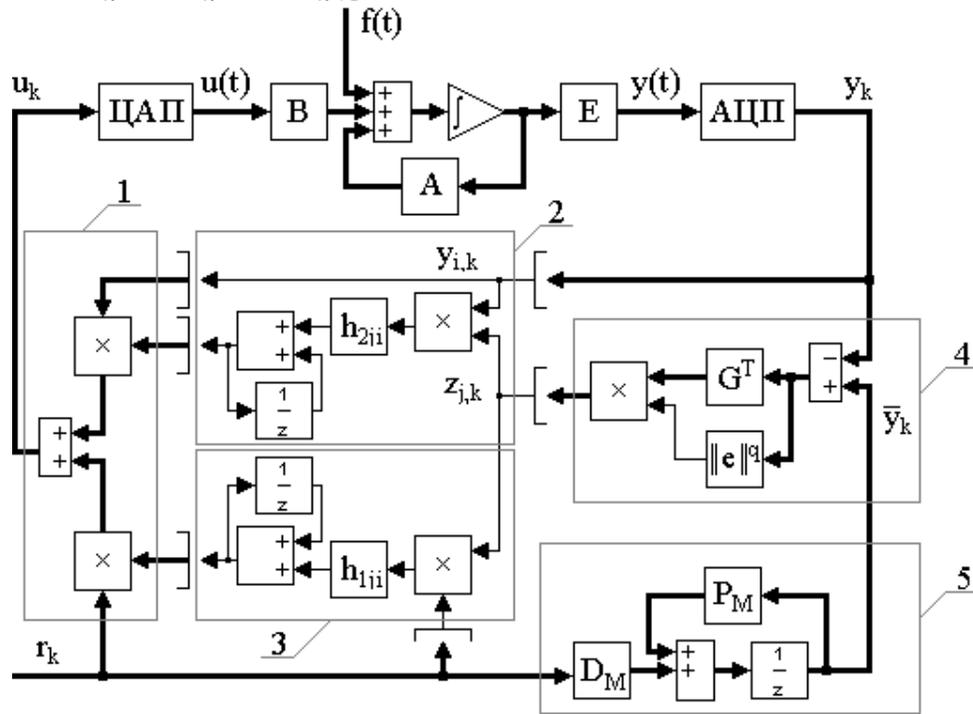


Рис. 5.4. Схема системы (5.18) – (5.24).

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad f(t) = \begin{pmatrix} \exp(-t) \\ \exp(-t) \\ \exp(-t) \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \bar{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$P_M = \begin{pmatrix} 0.9940 & 0.1888 & 0.0135 \\ -0.0807 & 0.8461 & 0.1081 \\ -0.6483 & -1.2693 & 0.1977 \end{pmatrix}, \quad D_M = \begin{pmatrix} 0.1078 & 0.0388 & 0.0020 \\ -0.0032 & 0.3775 & 0.0269 \\ -0.0436 & -0.3078 & 0.2161 \end{pmatrix}, \quad (5.25)$$

$$G = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix}, \quad r_k = \begin{pmatrix} 0.3 \sin(kt) \\ 0 \\ 0.2 \sin(kt) \end{pmatrix}, \quad t = 0.2, \quad q = 4, \quad h_{1ji} = 1, \quad h_{2ji} = 1.$$



5-я ГСПАУ с НЭМ ( $n = 3, l = 2, m = 2$ ).

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + f(t), \quad y(t) = L^T x(t), \quad (5.34)$$

$$e_k = r_* - n_k, \quad n_k = G^T y_k, \quad y_k = y(t_k), \quad (5.35)$$

$$u_k = c_{1,k} r_* + c_{2,k} n_k, \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (5.36)$$

$$c_{1i,k} = c_{1i,k-1} + h_{1i} e_{i,k} \|e_k\|^q r_{i,*}, \quad h_{1i} = \text{const} > 0, \quad (5.37)$$

$$c_{2i,k} = c_{2i,k-1} + h_{2i} e_{i,k} \|e_k\|^q n_{i,k}, \quad h_{2i} = \text{const} > 0, \quad i = \overline{1, m}. \quad (5.38)$$

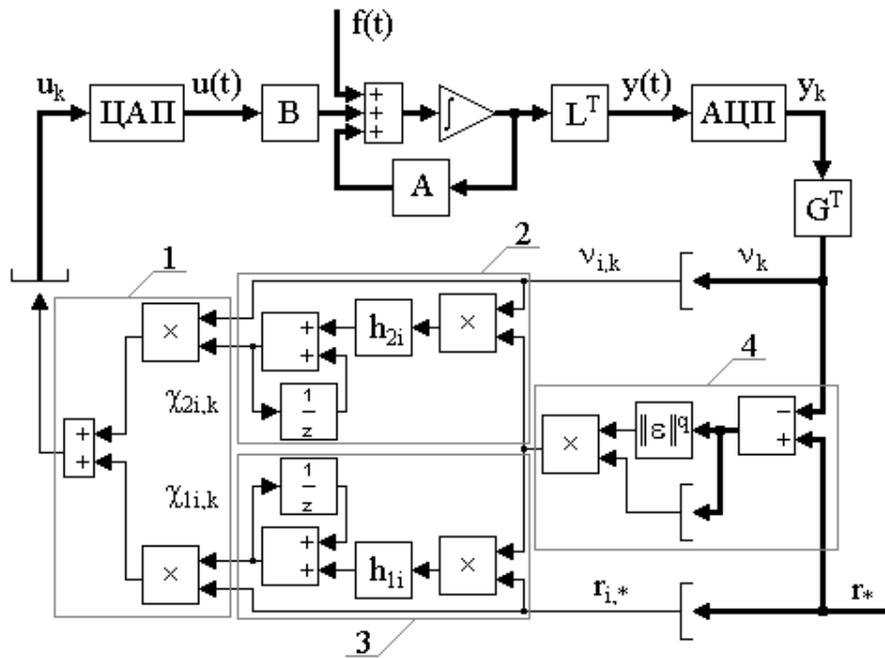


Рис. 5.6. Схема системы (5.34) – (5.38).

$$A = \begin{pmatrix} 4.8 & 0.28 & 2.82 \\ 2.9 & -1.33 & 1.82 \\ 0.04 & 4.77 & -2.22 \end{pmatrix}, \quad B = \begin{pmatrix} 2.8 & 0.99 \\ 0.125 & 1.28 \\ 0.63 & 2.17 \end{pmatrix},$$

$$f(t) = \begin{pmatrix} 0 \\ 0 \\ -0.05 \exp(-t) \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0.01 \end{pmatrix},$$

$$G = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}, \quad L^T = \begin{pmatrix} 1.56 & 0.98 & 0.55 \\ 0.79 & 3.42 & 0.95 \end{pmatrix}, \quad r_* = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (5.39)$$

$$h_{21} = 3000, \quad h_{22} = 160,$$

$$t = 0.2, \quad q = 4.$$

6-я ГСПАУ с НЭМ ( $n = 3, l = 2, m = 1$ ).

$$\frac{dx(t)}{dt} = Ax(t) + bu(t) + f(t), \quad y(t) = L^T x(t), \quad (5.40)$$

$$e_k = r_* - n_k, \quad n_k = g^T y_k, \quad y_k = y(t_k), \quad (5.41)$$

$$u_k = c_{1,k} r_* + c_{2,k} n_k, \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (5.42)$$

$$c_{1,k} = c_{1,k-1} + h_1 e_k \|e_k\|^q r_*, \quad h_1 = \text{const} > 0, \quad (5.43)$$

$$c_{2,k} = c_{2,k-1} + h_2 e_k \|e_k\|^q n_k, \quad h_2 = \text{const} > 0. \quad (5.44)$$

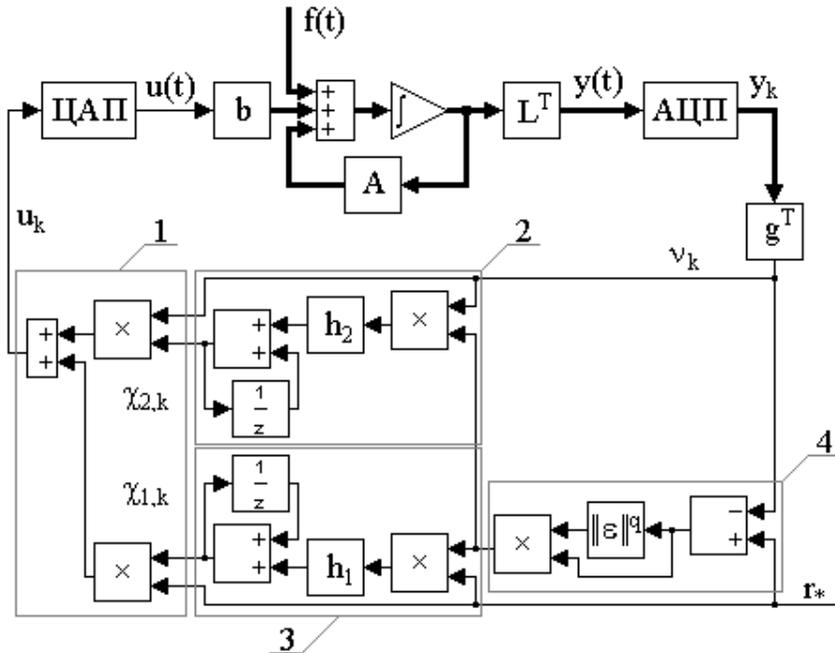


Рис. 5.7. Схема системы (5.40) – (5.44).

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -26 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad L = \begin{pmatrix} 4 & 0 \\ 4 & -2 \\ 0 & 1 \end{pmatrix},$$

$$f(t) = \begin{pmatrix} 0 \\ 0 \\ 0.1 \sin(t) \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (5.45)$$

$$g^T = (2 \ 1), \quad r_* = 1, \quad h_1 = 1, \quad h_2 = 0.9,$$

$$t = 5, \quad q = 3.$$

7-я ГСПАУ с запаздыванием ( $n = 3, l = 2, m = 2$ ).

$$\frac{dx(t)}{dt} = Ax(t) + \Gamma x(t-h) + Bu(t) + f(t), \quad y(t) = L^T x(t), \quad (5.46)$$

$$x(q) = j(q), \quad q \in [-h; 0], \quad (5.47)$$

$$u_k = c_{1,k} r_* + c_{2,k} n_k + c_{3,k} n_{k-s}, \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (5.48)$$

$$n_k = n(t_k), \quad n_{k-s} = n(t_{k-s}) = n(t_k - h). \quad (5.49)$$

$$c_{1i,k} = c_{1i,k-1} + h_{1i} e_{i,k} \|e_k\|^q r_{i,*}, \quad h_{1i} = \text{const} > 0, \quad i = \overline{1, m}, \quad (5.50)$$

$$c_{2i,k} = c_{2i,k-1} + h_{2i} e_{i,k} \|e_k\|^q n_{i,k}, \quad h_{2i} = \text{const} > 0, \quad i = \overline{1, m}, \quad (5.51)$$

$$c_{3i,k} = c_{3i,k-1} + h_{3i} e_{i,k} \|e_k\|^q n_{i,k-s}, \quad h_{3i} = \text{const} > 0, \quad i = \overline{1, m}, \quad (5.52)$$

$$e_k = r_* - n_k, \quad n_k = G^T y_k, \quad y_k = y(t_k). \quad (5.53)$$

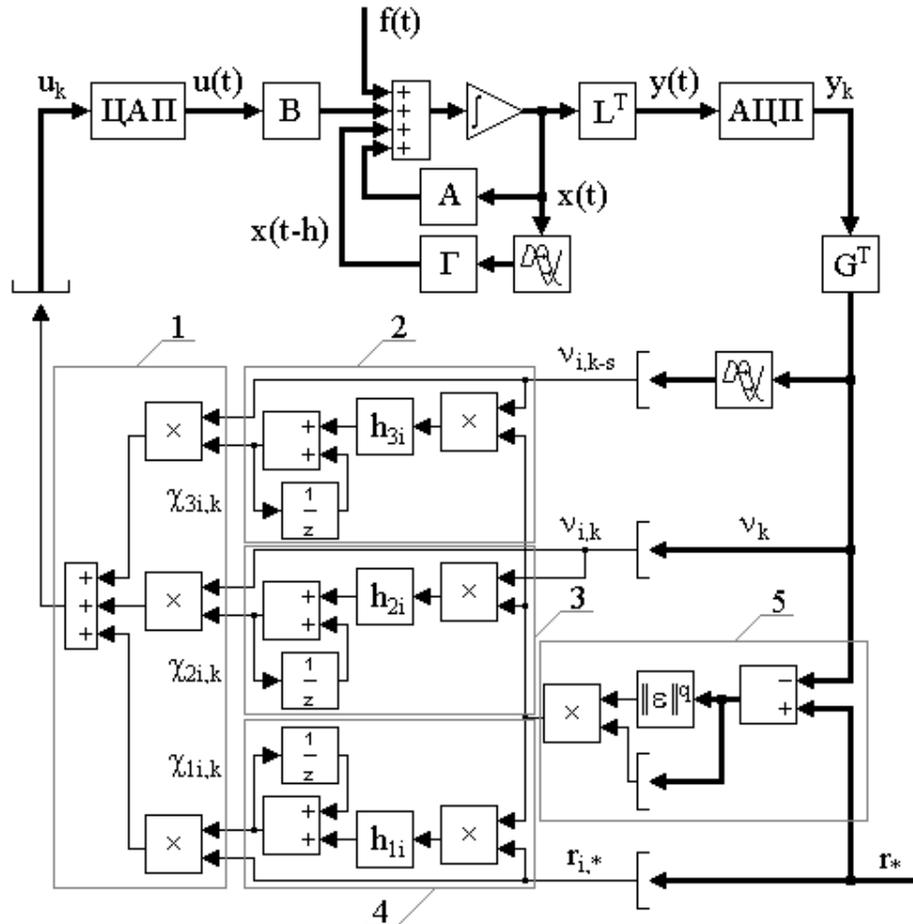


Рис. 5.8. Схема системы (5.46) – (5.53).

$$A = \begin{pmatrix} 4.8 & 0.28 & 2.82 \\ 2.9 & -1.33 & 1.82 \\ 0.04 & 4.77 & -2.22 \end{pmatrix}, \quad B = \begin{pmatrix} 2.8 & 0.99 \\ 0.125 & 1.28 \\ 0.63 & 2.17 \end{pmatrix}$$

$$\tilde{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1.3 & -0.3 & 1.4 \end{pmatrix}, \quad f(t) = \begin{pmatrix} 0 \\ 0 \\ -0.05 \exp(-t) \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0.01 \end{pmatrix} \quad (5.54)$$

$$G = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}, \quad L^T = \begin{pmatrix} 1.56 & 0.98 & 0.55 \\ 0.79 & 3.42 & 0.95 \end{pmatrix}, \quad r_* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$h = 2, \quad q = 4,$$

$$h_{21} = 1750, \quad h_{22} = 150, \quad h_{31} = 125000, \quad h_{32} = 35000,$$

$$t = 0.2.$$

8-я ГСПАУ с запаздыванием ( $n = 3, l = 2, m = 1$ ).

$$\frac{dx(t)}{dt} = Ax(t) + \Gamma x(t-h) + bu(t) + f(t), \quad y(t) = L^T x(t), \quad (5.55)$$

$$x(q) = j(q), \quad q \in [-h; 0], \quad (5.56)$$

$$u_k = c_{1,k} r_* + c_{2,k} n_k + c_{3,k} n_{k-s}, \quad u(t) = u_k \text{ при } t_k \leq t < t_{k+1}, \quad (5.57)$$

$$n_k = n(t_k), \quad n_{k-s} = n(t_{k-s}) = n(t_k - h), \quad (5.58)$$

$$c_{1,k} = c_{1,k-1} + h_1 e_k \|e_k\|^q r_*, \quad h_1 = \text{const} > 0, \quad (5.59)$$

$$c_{2,k} = c_{2,k-1} + h_2 e_k \|e_k\|^q n_k, \quad h_2 = \text{const} > 0, \quad (5.60)$$

$$c_{3,k} = c_{3,k-1} + h_3 e_k \|e_k\|^q n_{k-s}, \quad h_3 = \text{const} > 0, \quad (5.61)$$

$$e_k = r_* - n_k, \quad n_k = g^T y_k, \quad y_k = y(t_k) \quad (5.62)$$

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -23 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.1 & 0.3 & 0.3 \end{pmatrix}$$

$$f(t) = \begin{pmatrix} 0 \\ 0 \\ \exp(-t) \end{pmatrix}, \quad L = \begin{pmatrix} 4 & 0 \\ 4 & -2 \\ 0 & 1 \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad g = \begin{pmatrix} 2 \\ 0.9 \end{pmatrix}$$

$$r_* = 1, \quad h = 1, \quad q = 4, \quad (5.63)$$

$$h_1 = 0.5, \quad h_2 = 1, \quad h_3 = 0.03,$$

$$t = 6.$$

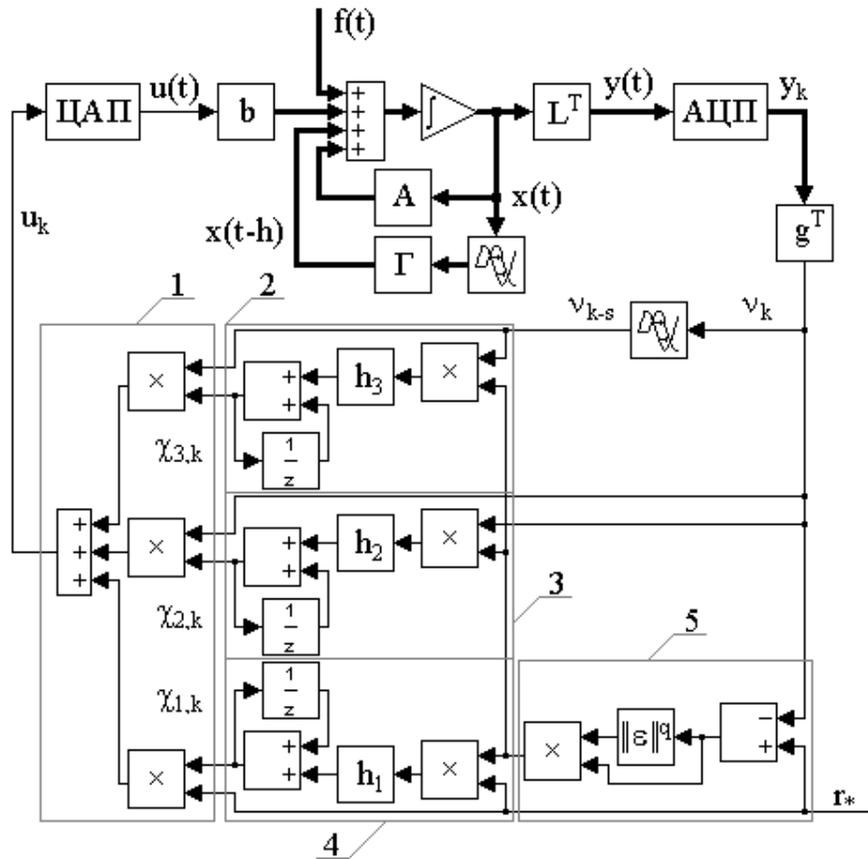


Рис. 5.9. Схема системы (5.55) – (5.62).

## 7. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ И КОМПЛЕКТ ЗАДАНИЙ

### ЧАСТЬ 1. МЕТОД ПРОСТРАНСТВА СОСТОЯНИЙ. СЛУЧАЙНЫЕ ПРОЦЕССЫ. ГАРМОНИЧЕСКАЯ ЛИНЕАРИЗАЦИЯ НЕЛИНЕЙНОСТЕЙ. (лабораторные работы)

#### ЛАБОРАТОРНАЯ РАБОТА №1 (4-часовая)

ТЕМА. *Использование типовых математических моделей и их преобразований при исследовании свойств линейных моделей.*

ЦЕЛЬ. *Расчет и построение характеристик линейной системы управления во временной и частотной областях с использованием типовых форм записи и преобразований ее математической модели.*

ТЕКУЩИЙ КОНТРОЛЬ ЗНАНИЙ.

1. Охарактеризуйте назначение **m**-файлов: **conv(a,b)**, **tf2ss(num,den)**, **ctrb(a,b)**, **obsv(a,c)**.

2. Поясните суть методов прямого, последовательного и параллельного программирования.

3. Сформулируйте критерии управляемости и наблюдаемости Калмана.

4. Охарактеризуйте назначение **m**-файлов **yprime(t,y)**, **ode45('имя файла', t0,tfinal,y0,tol,trace)**.

5. Дайте определение основных временных и частотных характеристик динамической системы.

6. Приведите в общем виде уравнения частотных характеристик: **mag**, **phase**, **re** и **im**.

7. Охарактеризуйте назначение **m**-файлов **bode(a,b,c,d,1,w)**, **nyquist(a,b,c,d,1,w)**.

ИСХОДНЫЕ ДАННЫЕ.

Исследуется динамическая колебательная система четвертого порядка, описываемая передаточной функцией вида

$$W(p) = R(p)/D(p), \quad (1)$$

где  $R(p)$  и  $D(p)$  – полиномы вида:

$$R(p)=r_0*(r_1*p+1); D(p)=(d_0*p+1)*(d_1*p+1)*(d_2^2*p^2+2*d_2*d_3*p+1). \quad (2)$$

НОМЕР ВАРИАНТА И ЗНАЧЕНИЯ ПАРАМЕТРОВ.

№01:  $r_0=10; r_1=0.18; d_0=0.26; d_1=0.08; d_2=0.44; d_3=0.03;$

№02:  $r_0=50; r_1=0.32; d_0=0.06; d_1=0.25; d_2=0.52; d_3=0.21;$

№03:  $r_0=30; r_1=0.16; d_0=0.13; d_1=0.31; d_2=0.26; d_3=0.11;$

№04:  $r_0=20; r_1=0.22; d_0=0.84; d_1=0.36; d_2=0.82; d_3=0.49;$

№05:  $r_0=40; r_1=0.56; d_0=0.17; d_1=0.06; d_2=0.12; d_3=0.55;$

№06:  $r_0=20; r_1=0.78; d_0=0.86; d_1=0.18; d_2=0.24; d_3=0.01;$

№07:  $r_0=70; r_1=0.62; d_0=0.76; d_1=0.35; d_2=0.32; d_3=0.28;$

№08:  $r_0=60; r_1=0.56; d_0=0.63; d_1=0.41; d_2=0.46; d_3=0.34;$

№09:  $r_0=80; r_1=0.32; d_0=0.54; d_1=0.56; d_2=0.52; d_3=0.33;$

№10:  $r_0=90; r_1=0.16; d_0=0.47; d_1=0.36; d_2=0.62; d_3=0.65;$

№11:  $r_0=40; r_1=0.98; d_0=0.2; d_1=0.018; d_2=0.4; d_3=0.033;$

№12:  $r_0=60; r_1=0.82; d_0=0.0; d_1=0.295; d_2=0.2; d_3=0.421;$

№13:  $r_0=80; r_1=0.76; d_0=0.1; d_1=0.381; d_2=0.6; d_3=0.211;$

№14:  $r_0=90; r_1=0.62; d_0=0.8; d_1=0.376; d_2=0.2; d_3=0.349;$

№15:  $r_0=80; r_1=0.56; d_0=0.1; d_1=0.066; d_2=0.2; d_3=0.455;$

№16:  $r_0=60; r_1=0.48; d_0=0.8; d_1=0.158; d_2=0.4; d_3=0.561;$

№17:  $r_0=40; r_1=0.32; d_0=0.7; d_1=0.345; d_2=0.2; d_3=0.428;$

№18:  $r_0=20; r_1=0.26; d_0=0.6; d_1=0.431; d_2=0.6; d_3=0.334;$

№19:	$r_0=40; r_1=0.12; d_0=0.5; d_1=0.526; d_2=0.2; d_3=0.233;$
№20:	$r_0=70; r_1=0.06; d_0=0.4; d_1=0.316; d_2=0.2; d_3=0.165;$
№21:	$r_0=130; r_1=0.6; d_0=0.13; d_1=0.31; d_2=0.126; d_3=0.1;$
№22:	$r_0=120; r_1=0.2; d_0=0.84; d_1=0.36; d_2=0.282; d_3=0.4;$
№23:	$r_0=140; r_1=0.6; d_0=0.17; d_1=0.06; d_2=0.312; d_3=0.5;$
№24:	$r_0=120; r_1=0.8; d_0=0.86; d_1=0.18; d_2=0.424; d_3=0.1;$
№25:	$r_0=170; r_1=0.2; d_0=0.76; d_1=0.35; d_2=0.532; d_3=0.2.$

#### ЗАДАНИЕ и КОММЕНТАРИЙ.

1. Учитывая, что исходное математическое описание системы задано в виде (1), и используя возможности функции `conv(a, b)`, ВВЕДИТЕ в вашу программу числитель (под именем **num**) и знаменатель (по имени **den**) передаточной функции (1), с учетом выражений (2) и согласно данным вашего варианта.

2. Поскольку MatLab для расчетов использует векторно-матричную форму записи, то для работы с линейными системами необходимо описать их в пространстве состояний:

$$\frac{dx}{dt} = a*x + b*u, \quad y = c*x + d*u. \quad (3)$$

ВЫПОЛНИТЕ в вашей программе этот переход с помощью функции **tf2ss(num,den)**.

3. ОСУЩЕСТВИТЕ для исследуемой системы (форма записи (3)) ее проверку на управляемость и наблюдаемость по соответствующим критериям, используйте для формирования матрицы управляемости функцию **ctrb(a,b)**, а для построения матрицы наблюдаемости – функцию **obsv(a,c)**.

4. ОБЪЯВИТЕ глобальными следующие переменные: **a,b,c,d**, и ВЫПОЛНИТЕ численное интегрирование системы уравнений (3), используя функции: **yprime(t,y)**, **ode45('имя файла',t0,tfinal,y0,tol,trace)**.

*Замечание 1.* Для исполнения процедуры **yprime(t,y)**, СОЗДАЙТЕ отдельный **m**-файл с описанием системы (3). При этом задайте внешнее воздействие в виде  $u(t)=1$ .

*Замечание 2.* Внутри создаваемого **m**-файла ОБЪЯВИТЕ глобальными переменные: **a,b,c,d**.

*Замечание 3.* Перед использованием функции **ode45** ЗАДАЙТЕ и ВВЕДИТЕ в программу значения переменных: **t0, tfinal, y0, tol, trace**.

5. ВЫПОЛНИТЕ построение и сохранение графика переходного процесса в системе (3). ОСУЩЕСТВИТЕ построение и сохранение фазового портрета системы (3) по переменным пространства состояний  $y(3)$  и  $y(4)$ .

*Замечание.* После получения изображения рисунка – например, в окне **Fig.1** – для получения очередного рисунка рекомендуется принудительно открыть очередное окно.

6. Для расчета частотных характеристик системы (3) ВЫБЕРИТЕ и ЗАДАЙТЕ в программе частотный диапазон исследования, используя в качестве обозначения частоты переменную **w**.

7. Применяя функцию **bode(a,b,c,d,1,w)**, для системы (3) ВЫПОЛНИТЕ расчет частотных характеристик **mag, phase**. ПОСТРОЙТЕ годограф системы, вначале в полярных координатах (см. функцию **polar**), а затем, используя значения **mag, phase** и ВЫЧИСЛИВ значения характеристик **re, im**, ПОСТРОЙТЕ тот же годограф в прямоугольных координатах.

*Замечание.* При использовании для расчетов **phase** обратите внимание на ее размерность.

8. ВЫПОЛНИТЕ для той же системы построение и сохранение графиков основных частотных характеристик: **mag, phase, re** и **im**.

9. ПОСТРОЙТЕ и сохраните графики характеристик **mag, phase** в логарифмическом масштабе.

*Замечание.* При построении графика для **mag** используйте функцию **loglog(w,mag)**, а при построении в полулогарифмическом масштабе по оси абсцисс графика для **phase** – воспользуйтесь функцией **semilogx(w,phase)**.

#### ТРЕБОВАНИЯ К ОТЧЕТУ.

1. Первый лист отчета – титульный (см. приложение 2).
2. Форма изложения произвольная, но в отчете следует отразить все этапы выполнения лабораторной работы и привести как полную ее программу на языке **Matlab**, так и все графики (формат и размеры графических изображений должны иметь экономичный формат).

*Замечание.* Графики должны быть информативными, т.е. отражать временные или частотные характеристики во всем диапазоне изменения свойств системы управления.

3. Отчет УСТАНОВЛЕННОЙ ФОРМЫ должен быть отправлен по **e-mail: mads@freemail.amursu.ru** НЕ ПОЗДНЕЕ 10 ДНЕЙ после выполнения лабораторной работы.

4. В отчете указать личный электронный почтовый адрес студента.  
ДОПУСК К РАБОТЕ.

К следующей работе студент допускается только при условии сдачи на проверку отчета по предыдущей работе и получении допуска от преподавателя.

### ЛАБОРАТОРНАЯ РАБОТА №2 (4-часовая)

*ТЕМА. Применение теории случайных функций для анализа линейных систем управления.*

*ЦЕЛЬ. Расчет и построение вероятностных характеристик выходного сигнала линейной системы управления при прохождении через нее стационарных случайных сигналов с известными корреляционными функциями (спектральными плотностями).*

#### ТЕКУЩИЙ КОНТРОЛЬ ЗНАНИЙ.

1. Охарактеризуйте назначение **m**-файлов: **cloop(num0,den0,sign)**, **trapz(x,y)**.
2. Дайте определения и охарактеризуйте назначение корреляционной функции и спектральной плотности стационарных случайных процессов.
3. В чем суть гипотезы эргодичности для стационарных случайных процессов?
4. Почему знание корреляционной функции и спектральной плотности позволяет полностью охарактеризовать свойства стационарных случайных процессов?

#### ИСХОДНЫЕ ДАННЫЕ И КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

К замкнутой системе управления с единичной отрицательной обратной связью, у которой передаточная функция разомкнутой системы имеет вид

$$W(p) = Q(p)/D(p), \quad (1)$$

где  $Q(p)$  и  $D(p)$  – полиномы вида:

$$Q(p)=q_0*(q_1*p+1); D(p)=(d_0*p+1)*(d_1*p+1)*((d_2)^2*p^2+2*d_2*d_3*p+1)(d_4*p+1), (2)$$

приложена внешнее возмущение (помеха) с известной корреляционной функцией  $R(T)$ .

Рассматриваются два типа помех (приложенных ко входу системы управления при нулевом задающем воздействии), у которых корреляционные функции соответствуют, во-первых, случаю "телеграфного сигнала" (моменты изменения знака сигнала заданной амплитуды распределены по закону Пуассона) и, во-вторых, случаю "нерегулярной качки", когда корреляционная функция аппроксимируется соответствующими периодическими функциями.

Для случая эргодического сигнала с распределением по закону Пуассона корреляционная функция имеет вид:

$$R1(T)=dd*\exp(-aa|T|), \quad (3)$$

а для случая эргодического сигнала типа "нерегулярной качки" рассматриваются два варианта описания корреляционной функции: во-первых,

$$R2(T)=dd*[\exp(-aa|T|)]*\cos(bb*T); \quad (4)$$

и, во-вторых,

$$R3(T)=dd*[\exp(-aa|T|)]*[\cos(bb*T)+(aa/bb)*\sin(bb*T)]. \quad (5)$$

В соотношения (3) – (5) аргументом функций  $R(T)$  является интервал времени  $T$ . При этом в математическое описание  $R(T)$  входят следующие константы:  $aa$  – среднее число перемен знака за единицу времени,  $dd$  – дисперсия,  $bb$  – резонансная частота для случаев (4) и (5).

С помощью преобразования Фурье можно показать, что спектральные плотности случайных сигналов с известными корреляционными функциями могут быть получены в результате соответствующих вычислений, которые для выражения (3) дают соотношение:

$$S1(w)=2*aa*dd/(w^2+(aa)^2), \quad (6)$$

а для выражений (4) и (5) соответственно соотношения:

$$S2(w)=aa*dd*[1/((bb-w)^2+(aa)^2)+1/((bb+w)^2+(aa)^2)], \quad (7)$$

$$S3(w)=(aa*dd/bb)*[(2*bb-w)/((bb-w)^2+(aa)^2)+(2*bb+w)/((bb+w)^2+(aa)^2)]. \quad (8)$$

Как известно, для расчета спектральной плотности выходного сигнала в замкнутой системе управления используется следующая формула:

$$S_{\text{выход}}(w)=(\text{mod}[\Phi(jw)])^2*S_{\text{вход}}(w), \quad (9)$$

где  $\Phi(jw)$  – обозначение частотной передаточной функции замкнутой системы.

*Замечание.* Для удобства используемых обозначений пусть в дальнейшем спектральные плотности выходных сигналов, в отличие от входных (имеющих нумерацию 1, 2, 3), будут иметь соответственно нумерацию 4, 5, 6.

При этом, по известной спектральной функции случайного сигнала на выходе системы управления, его дисперсию можно определить как значение следующего интеграла:

$$D_{\text{выход}}=(1/\pi)*\text{Integral}[S_{\text{выход}}(w)]dw, \quad (10)$$

где  $\pi$  – обозначение числа пи. Выражение (10), строго говоря, вычисляется на частотном интервале  $[0, \text{inf}]$ , ( $\text{inf}$  – плюс бесконечность), но на практике исследователь всегда ограничивается конечным интервалом, т.е. тем значением частоты, начиная с которого можно полагать выполненным условие

$$S_{\text{выход}}(w)=0.$$

По известному значению  $D_{\text{выход}}$  просто вычислить среднеквадратическое отклонение

$$Q_{\text{выход}}=(D_{\text{выход}})^{0.5}; \quad (11)$$

*Замечание.* Для обозначения дисперсий и среднеквадратических отклонений выходных сигналов, рекомендуем использовать, по аналогии с предыдущим, нумерацию 4, 5, 6.

#### НОМЕР ВАРИАНТА И ЗНАЧЕНИЯ ПАРАМЕТРОВ.

№01:  $q_0=100; q_1=1.8; d_0=0.6; d_1=0.8; d_2=0.4; d_3=1.3; d_4=.1; aa=13; dd=.00025; bb=15;$   
 №02:  $q_0=500; q_1=2.1; d_0=0.6; d_1=0.5; d_2=0.5; d_3=1.1; d_4=.2; aa=14; dd=.00015; bb=20;$   
 №03:  $q_0=300; q_1=1.6; d_0=0.3; d_1=0.1; d_2=0.2; d_3=1.1; d_4=.1; aa=11; dd=.00035; bb=16;$   
 №04:  $q_0=200; q_1=2.2; d_0=0.4; d_1=0.6; d_2=0.8; d_3=1.9; d_4=.2; aa=12; dd=.00055; bb=18;$   
 №05:  $q_0=400; q_1=1.5; d_0=0.7; d_1=0.6; d_2=0.1; d_3=1.5; d_4=.1; aa=13; dd=.00025; bb=17;$   
 №06:  $q_0=200; q_1=1.7; d_0=0.6; d_1=0.8; d_2=0.2; d_3=1.1; d_4=.2; aa=12; dd=.00015; bb=19;$   
 №07:  $q_0=200; q_1=1.6; d_0=0.4; d_1=0.5; d_2=0.3; d_3=1.8; d_4=.1; aa=11; dd=.00035; bb=20;$   
 №08:  $q_0=200; q_1=1.5; d_0=0.3; d_1=0.1; d_2=0.4; d_3=1.4; d_4=.2; aa=12; dd=.00055; bb=15;$   
 №09:  $q_0=300; q_1=1.3; d_0=0.4; d_1=0.6; d_2=0.5; d_3=1.3; d_4=.1; aa=11; dd=.00045; bb=16;$

- №10:  $q_0=400; q_1=1.1; d_0=0.7; d_1=0.6; d_2=0.6; d_3=1.5; d_4=.2; aa=13; dd=.00025; bb=17;$   
 №11:  $q_0=400; q_1=1.9; d_0=0.2; d_1=0.8; d_2=0.4; d_3=1.3; d_4=.1; aa=12; dd=.00015; bb=18;$   
 №12:  $q_0=300; q_1=1.8; d_0=0.0; d_1=0.5; d_2=0.2; d_3=1.1; d_4=.2; aa=11; dd=.00035; bb=19;$   
 №13:  $q_0=200; q_1=1.7; d_0=0.1; d_1=0.1; d_2=0.6; d_3=1.1; d_4=.1; aa=12; dd=.00055; bb=18;$   
 №14:  $q_0=300; q_1=1.6; d_0=0.8; d_1=0.6; d_2=0.2; d_3=1.9; d_4=.2; aa=11; dd=.00045; bb=17;$   
 №15:  $q_0=100; q_1=1.5; d_0=0.1; d_1=0.6; d_2=0.2; d_3=1.5; d_4=.1; aa=13; dd=.00025; bb=16;$   
 №16:  $q_0=500; q_1=1.4; d_0=0.8; d_1=0.8; d_2=0.4; d_3=1.1; d_4=.2; aa=12; dd=.00015; bb=15;$   
 №17:  $q_0=400; q_1=1.3; d_0=0.7; d_1=0.5; d_2=0.2; d_3=1.8; d_4=.1; aa=11; dd=.00035; bb=20;$   
 №18:  $q_0=200; q_1=1.2; d_0=0.6; d_1=0.1; d_2=0.6; d_3=1.4; d_4=.2; aa=12; dd=.00055; bb=19;$   
 №19:  $q_0=400; q_1=1.1; d_0=0.5; d_1=0.6; d_2=0.2; d_3=1.3; d_4=.1; aa=11; dd=.00045; bb=18;$   
 №20:  $q_0=300; q_1=1.0; d_0=0.4; d_1=0.6; d_2=0.2; d_3=1.5; d_4=.2; aa=13; dd=.00025; bb=17;$   
 №21:  $q_0=300; q_1=1.6; d_0=0.1; d_1=0.1; d_2=0.2; d_3=1.1; d_4=.1; aa=12; dd=.00015; bb=16;$   
 №22:  $q_0=200; q_1=1.2; d_0=0.8; d_1=0.6; d_2=0.8; d_3=1.4; d_4=.2; aa=11; dd=.00035; bb=15;$   
 №23:  $q_0=400; q_1=1.6; d_0=0.1; d_1=0.6; d_2=0.1; d_3=1.5; d_4=.1; aa=12; dd=.00055; bb=16;$   
 №24:  $q_0=200; q_1=1.8; d_0=0.8; d_1=0.8; d_2=0.2; d_3=1.1; d_4=.2; aa=11; dd=.00045; bb=17;$   
 №25:  $q_0=200; q_1=1.2; d_0=0.7; d_1=0.5; d_2=0.3; d_3=1.2; d_4=.1; aa=13; dd=.00025; bb=18;$

#### ЗАДАНИЕ.

1. ВВЕДИТЕ в вашу программу математическое описание разомкнутой системы управления, опираясь на уравнения (1), (2) и в соответствии с числовыми данными вашего варианта.

*Замечание.* Рекомендуем для передаточной функции разомкнутой системы использовать имена **num0**; **den0**, а для замкнутой системы имена **num**; **den**.

2. Используя функцию **cloop(num0,den0,sign)**, где  $sign=-1$  и  $sign=1$  описание отрицательной и положительной обратных связей, ПОЛУЧИТЕ в явном виде описание (**num** и **den**) передаточной функции замкнутой системы управления  $\Phi(p)$ .

3. ВЫПОЛНИТЕ переход от передаточной функции замкнутой системы управления  $\Phi(p)$  к ее описанию в пространстве состояний

$$\dot{x}/dt = a*x + b*u, \quad y = c*x + d*u. \quad (12)$$

4. Для определения частотных характеристик системы ЗАДАЙТЕ частотный диапазон исследования и ВЫПОЛНИТЕ для системы (12) РАСЧЕТ и ПОСТРОЕНИЕ ее частотной характеристики – **mag**.

5. ЗАДАЙТЕ временной интервал изменения переменной T, РАССЧИТАЙТЕ и ПОСТРОЙТЕ графики корреляционных функций входных сигналов (3), (4) и (5).

*Замечание.* для удобства пусть в дальнейшем корреляционные функции выходных сигналов, в отличие от входных (имеющих нумерацию: 1, 2, 3), будут иметь соответственно нумерацию: 4, 5, 6.

6. ЗАДАЙТЕ частотный интервал изменения переменной  $w$ , РАССЧИТАЙТЕ и ПОСТРОЙТЕ графики спектральных плотностей входных сигналов (6), (7) и (8).

*Замечание 1.* Поскольку спектральные плотности – четные функции, то при расчетах достаточно использовать только область положительных частот.

Используя формулу (9), РАССЧИТАЙТЕ и ПОСТРОЙТЕ графики спектральной плотности выходного сигнала системы (11) для случаев, когда спектральная плотность входного сигнала имеет описание в виде (6), (7) и (8).

*Замечание 2.* Поскольку функции  $S_{\text{вход}}(w)$  являются строками соответствующего размера, а функция  $(\text{mod}[\Phi(jw)])^2$  – это столбец (того же размера), то для вычисления значений функций  $S_{\text{выход}}(w)$  рекомендуем воспользоваться оператором **for**.

7. Используя формулы (10) и (11), ВЫЧИСЛИТЕ значения дисперсий и среднеквадратических отклонений, для всех вариантов сигналов на выходе замкнутой системы управления.

*Замечание.* Для вычисления интеграла (10) рекомендуем воспользоваться функцией **trapz(x,y)**.

### ЛАБОРАТОРНАЯ РАБОТА №3 (4-часовая)

ТЕМА. *Анализ систем управления с типовыми нелинейностями.*

ЦЕЛЬ. *Имитационное моделирование режима автоколебаний и анализ устойчивости непрерывных систем управления первого типа с нелинейными элементами (с помощью точных и приближенных методов исследования).*

ИСХОДНЫЕ ДАННЫЕ И КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

Рассматриваются замкнутые системы управления, прямой канал которых состоит из последовательно соединенных нелинейной (НЧ) и линейной (ЛЧ) частей, причем ЛЧ описывается передаточной функцией вида

$$W(p) = y(p)/u(p) = Q(p)/D(p), \quad (1)$$

где  $y(p)$  и  $u(p)$  – соответственно изображения по Лапласу выходного и входного сигналов,  $Q(p)$  и  $D(p)$  – полиномы, представляющие собой следующие выражения:

$$Q(p) = q_0 * (q_1 * p + 1),$$

$$D(p) = (d_0 * p + 1) * (d_1 * p + 1) * ((d_2)^2 * p^2 + 2 * d_2 * d_3 * p + 1) * (d_4 * p + 1). \quad (2)$$

В свою очередь НЧ является статической и ее описание задается как графически, так и аналитически в виде некоторой функциональной зависимости

$$u = F(e), \quad (3)$$

где  $F(-e) = -F(e)$  – нечетная функция;  $e = e(t)$  – ошибка, формируемая как рассогласование между задающим воздействием  $r = r(t)$  и регулируемым выходом системы управления  $y = y(t)$ , т.е.  $y = r - y$ .

*Замечание.* Далее для простоты  $r(t) = 0$ , т.е.  $u = -F(y)$ .

Далее рассматриваются следующие простейшие нелинейные звенья:

1. Идельное реле.
2. Реле с зоной нечувствительности.
3. Нелинейный элемент с зоной насыщения.
4. Нелинейный элемент с зоной насыщения и зоной нечувствительности.

*Замечание 1.* Структурная схема исследуемой системы и графические изображения с матописанием рассматриваемых нелинейных элементов приведены в файле **pril3.doc**, который помещен в папку **PRIL**.

*Замечание 2.* Все ссылки на рисунки и формулы, расположенные в файле **pril3.doc**, начинаются с индекса (П).

ДЛЯ ВСЕХ ВАРИАНТОВ параметры нелинейных элементов, частота и соответствующее начальное условие имеют следующие значения:

$$bb = 0.99; m = 0.3; cc = 1.9; aa = 1.2; w_0 = 0.8 * \pi; y_0(1) = -0.03.$$

*Замечание.* Исходные данные для уравнений (1) и (2) взять из лаб. работы №2.

ЗАДАНИЕ.

1. Рассматривается отдельно взятый нелинейный элемент. Пусть сигнал  $x(t) = aa * \sin(w_0 * t)$  поступает на вход нелинейного элемента. Согласно исходным данным РАССЧИТАЙТЕ и выполните ПОСТРОЕНИЕ функций  $y(t)$ , используя уравнения (П.1), (П.3), (П.5), (П.7).

2. ВВЕДИТЕ в вашу программу математическое описание линейной части системы управления, опираясь на уравнения (1), (2), и ВЫПОЛНИТЕ переход в пространство состояний

$$\frac{dx}{dt} = a * x + b * u, \quad y = c * x + d * u. \quad (4)$$

3. ОБЪЯВИТЕ глобальными следующие переменные: a, b, c, cc, bb, m и ВЫПОЛНИТЕ численное интегрирование системы управления (3), (4) (см. структурную схему на рис. П.5), при начальном условии  $y_0 = [y_0(1) \ 0 \ 0 \ 0 \ 0]$  для СЛЕДУЮЩИХ СЛУЧАЕВ задания нелинейной части:

идельного реле; реле с зоной нечувствительности;

нелинейный элемент с зоной насыщения;

нелинейный элемент с зоной насыщения и зоной нечувствительности.

ПОСТРОЙТЕ графики выходного сигнала  $y(t)$  для всех рассматриваемых систем управления.

4. ПОВТОРИТЕ РАСЧЕТЫ и ПОСТРОЕНИЯ п.4 для случаев, когда коэффициент усиления линейной части системы принимает следующие значения:  $q_0 * 0.005$  и  $q_0 * 0.0005$ .

5. СРАВНИТЕ между собой графики, полученные для различных коэффициентов усиления, результат сравнения **ОБОСНУЙТЕ**.

6. Для ТРЕХ СЛУЧАЕВ задания нелинейностей (т.е. для идеального реле; реле с зоной нечувствительности; нелинейного элемента с зоной насыщения) **ВОСПОЛЬЗУЙТЕСЬ** их описанием, которое получается с помощью гармонической линеаризации, и выполните **ПОСТРОЕНИЕ** всех характеристик  $M_n(aa)$ , заданных в виде:

$$M_n(aa) = -(q(aa))^{-1}, \quad (5)$$

где  $q(aa)$  – коэффициент гармонической линеаризации (см. файл **pril3.doc**).

7. Совместите **ПОСТРОЕНИЕ** каждой из характеристик  $M_n(aa)$  с годографом исследуемой системы и **ОБОСНУЙТЕ** возможность возникновения или исчезновения автоколебательных режимов в нелинейной системе управления для каждого из рассматриваемых типов нелинейных элементов.

*Замечание.* Моделирование осуществить при  $q_0 * 0.01$ .

## ЧАСТЬ 2. СОЗДАНИЕ S-ФУНКЦИЙ (лабораторные работы)

S-функции используются при создании новых библиотечных блоков, блоков, обеспечивающих взаимодействие SIMULINK с аппаратными средствами компьютера, при создании блоков на основе математических уравнений, блоков, реализующих анимационные возможности MATLAB, а также при подключении к модели SIMULINK существующего программного кода языков высокого уровня [22, 24, 27].

S-функции обладают дополнительными возможностями, которые включают работу с разными типами данных (целыми, действительными и комплексными числами различной степени точности), использование матриц в качестве входных и выходных переменных (MATLAB S-функции могут оперировать только векторами в качестве входных и выходных переменных), а также большой набор внутренних функций (callback-методов).

### *Программирование в SIMULINK пользовательских функций.*

SIMULINK-функции (S-functions, S-функции). SIMULINK-функции – описание блока на одном из языков программирования: MATLAB, C, C++, Ada, или Fortran. Набор стандартных блоков SIMULINK достаточно обширен, однако в практике моделирования встречаются ситуации, когда нужного блока нет либо структурное моделирование делает модель слишком сложной. В этом случае необходимо использовать технологию S-функций для создания нужного блока. С помощью языков программирования пользователь может создать описание блока и подключить его к SIMULINK-модели, при этом, с точки зрения взаимодействия пользователя с моделью, блок на основе S-функции ничем не отличается от стандартного библиотечного блока SIMULINK. S-функции, созданные на C, C++, Ada или

Fortran, компилируются в исполняемые (\*.dll) файлы, за счет чего обеспечивается повышенная скорость выполнения таких блоков. S-функции обладают дополнительными возможностями, включая работу с разными типами данных, использование матриц в качестве входных и выходных переменных (MATLAB S-функции могут оперировать только векторами в качестве входных и выходных переменных),

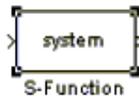


Рис. 4.1. Пиктограмма блока S-function.

а также большой набор внутренних функций (callback-методов). S-функция подключается к модели SIMULINK с помощью библиотечного блока S-function (из библиотеки User-Defined Function). На рис. 4.1 приведена пиктограмма блока S-Function, а на рис. 4.2 показаны окно S-модели с

блоком S-function и его диалоговое окно.

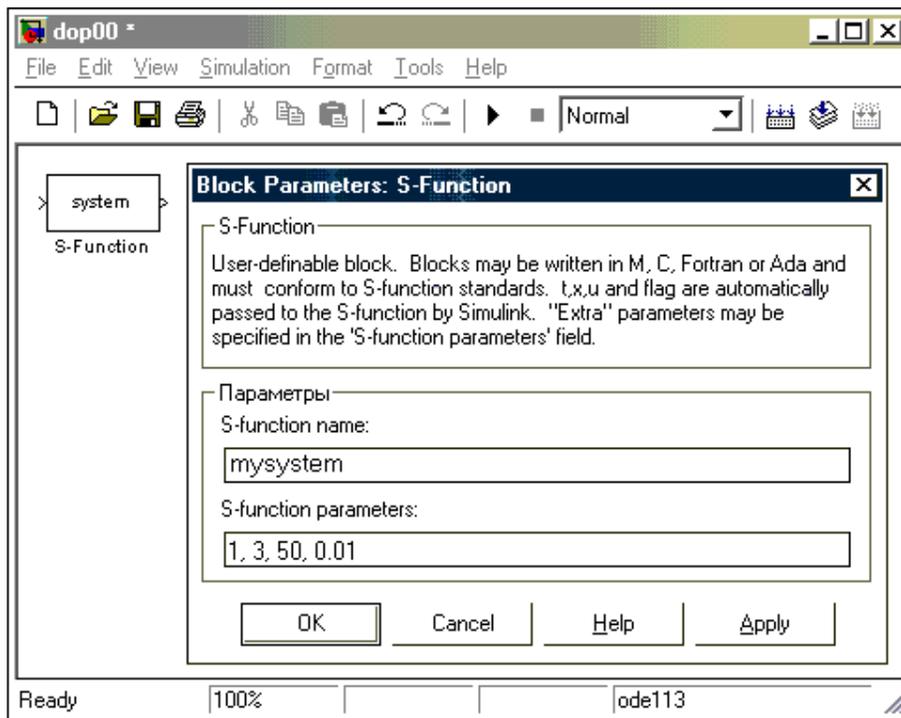


Рис. 4.2. Окно модели с вызванным блоком S-функции и его окно диалога.

Блок S-function Builder создает S-функции на языке C, – таким образом, инструментарий SIMULINK предоставляет два способа создания S-функций на языке C: за счет автоматического построителя S-Function Builder; с использованием шаблона (вручную), аналогично созданию S-функций на языке MATLAB. Наибольшими возможностями обладает второй способ, зато первый прост и наименее трудоемок, поэтому с помощью S-Function Builder можно легко и быстро создать требуемую S-функцию. В обоих случаях S-функция будет создана в виде dll-файла, что повышает быстродействие функции при ее выполнении.

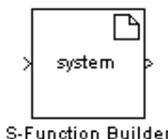


Рис. 4.3. Пиктограмма блока S-Function Builder.

Двойной щелчок на пиктограмме блока S-Function Builder (рис. 4.3) открывает его диалоговое окно. Окно S-Function Builder с активной вкладкой Initialization показано на рис. 4.4.

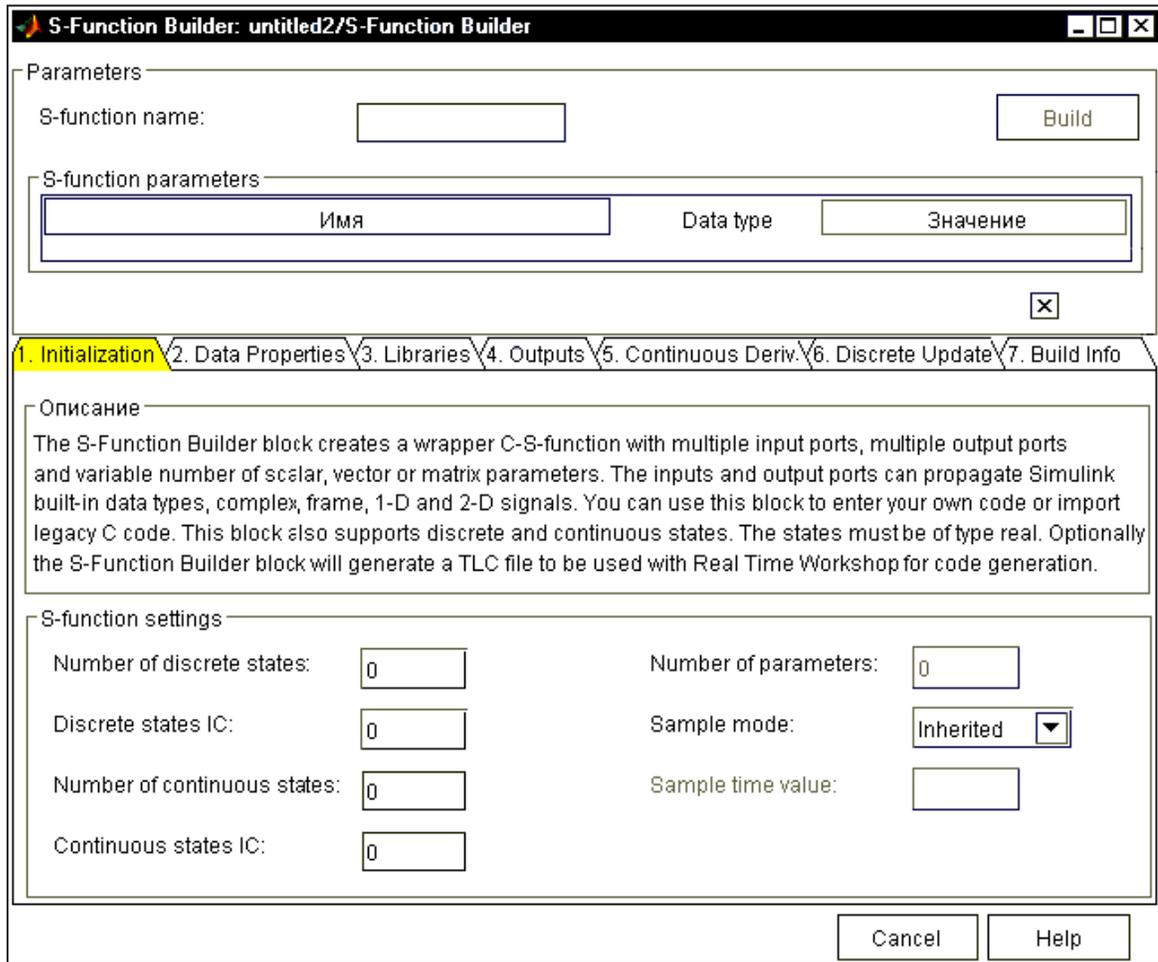


Рис. 4.4. Окно S-Function Builder с вкладкой Initialization.

В этом окне содержится панель параметров (Parameters), в которой находятся графа для ввода имени S-функции (S-function name) и графа для ввода начальных значений параметров, передаваемых через окно диалога блока S-function (S-function parameters). На этой панели для запуска процесса компиляции S-функции имеется кнопка Build. Среднюю часть окна занимают шесть вкладок, предназначенных для задания свойств S-функции и фрагментов ее кода. Первая вкладка – Initialization (инициализация), на ней задаются следующие свойства S-функции:

- Input port width – число входных переменных (размерность входного вектора);
- Output port width – число выходных переменных (размерность выходного вектора);
- Number of parameters – количество параметров, передаваемых в S-функцию через окно диалога блока;
- Sample time – шаг модельного времени, принимающий значения (из списка): Inherited (наследуемый); Continuous (непрерывный); Discrete (дискретный);
- Discrete sample time value – шаг модельного времени для дискретной системы (параметр доступен для ввода значения, когда выбран дискретный шаг расчета);
- Number of discrete states – число дискретных переменных состояния;
- Discrete states IC – начальные значения дискретных переменных состояния, которые вводятся через запятую;
- Number of continuous states – число непрерывных переменных состояния;
- Continuous states IC – начальные значения непрерывных переменных состояния, также вводятся через запятую.

## ЛАБОРАТОРНАЯ РАБОТА №1 (2 ЧАСА)

Тема. S-функции – средство программирования функций пользователя в Simulink.

Цель. Познакомиться с интерфейсом системы Simulink позволяющим работать с S-функциями. Рассмотреть существующие примеры S-функций в демонстрационной версии S-function demo.

Задание. Установить связь S-функции пользователя и .m файла из демонстрации и проверить работу S-блока.

Используя шаблон sfuntmpl.m (см. приложение 2) определить, какие методы необходимо использовать для программирования линейных непрерывных моделей систем, дискретных моделей систем, гибридных моделей систем.

Описать процедуры изменения иконки блока S-функции, его оформления, формы и надписи.

Контрольные вопросы.

1. Для чего могут быть использованы S-функции при моделировании сложных систем?

2. Какими возможностями обладают S-функции пользователя?

3. Как S-функция подключается к модели Simulink?

4. С какими типами данных можно работать в S-функциях?

Требования к отчету по оформлению лабораторных работ.

Титульный лист приведен в приложении 3.

В отчете следует отразить этапы выполнения лабораторной работы: описание математической модели системы; блок S-функции в .m файле и в .mdl файле (листинг программы и структурную Simulink-диаграмму системы) сохранить в электронной форме; результаты тестирования S-функции вынести в отчет.

### ***Разработка S-функций непрерывных систем управления***

Математическое описание S-функции. Создаваемые блоки могут быть непрерывными, дискретными или гибридными системами.

Simulink-блок (рис. 4.5), описывается наборами входных переменных  $u$ , переменных состояния  $x$  и выходных переменных  $y$ .

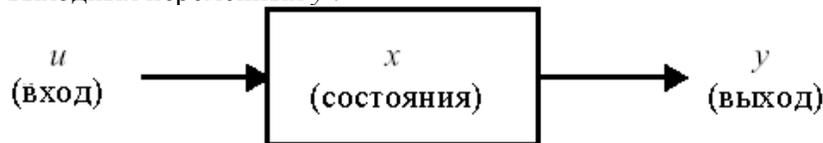


Рис. 4.5. Общий вид Simulink-блока.

S-функция подключается к модели Simulink с помощью библиотечного блока S-function (библиотека User-Defined Function), его параметрами являются:

S-function name – имя S-функции, которое не должно совпадать с именем модели (mdl-файла);

S-function parameters – параметры S-функции, передаваемые в нее через окно диалога. Параметры записываются в окне диалога в том же порядке, что и в заголовке S-функции.

Основные Callback-методы S-функции и основные понятия S-функции приведены в приложении 4.

Создание S-функций на языке MATLAB для модели непрерывной системы. Наиболее простой и быстрый путь создать S-функцию – это написать ее на языке MATLAB с использованием файла-шаблона. И хотя создание S-функций на языке MATLAB имеет некоторые ограничения (например, MATLAB S-функция может иметь только по одному входному и выходному порту, а передаваемые и принимаемые через эти порты данные могут быть только скалярами и векторами типа double), способ является наилучшим с точки зрения изучения механизма работы S-функции. Шаблон S-функции приведен в приложении 2.

Модель непрерывной системы. С помощью MATLAB в виде S-функции пользователь может задать систему обыкновенных дифференциальных уравнений (ODE), уравнения дискретной системы и (или) любой алгоритм, описывающий работу Simulink-блока.

Модель непрерывной системы, описываемой уравнениями пространства состояния:

$$\begin{aligned}x' &= Ax + Bu, \\ y &= Cx + Du,\end{aligned}\tag{4.1}$$

где параметры модели (значения матриц A, B, C, D) задаются в теле S-функции и передаются в callback-методы через их заголовки в качестве дополнительных параметров.

Данная S-функция моделирует непрерывную систему с двумя входами ( $u \in R^2$ ), двумя выходами ( $y \in R^2$ ) и двумя переменными состояниями ( $x \in R^2$ ).

Базовая форма синтаксиса S-функции выглядит следующим образом:

```
function [sys,x0,str,ts] = sfuntmpl(t,x,u,flag),
```

где t – текущее время; x – вектор переменных состояний системы; u – вектор входных сигналов; flag – целое число (флаг), определяющее, какая функция внутри S-функции выполняется при вызове; p1,..., pn – параметры S-функции, задаваемые в окне диалога блока "S-function".

Результат, возвращаемый (вычисляемый) S-функцией в момент времени t, зависит от значения переменной flag, значения вектора состояния системы x и текущего значения вектора входного сигнала u.

Значения матриц передаются в callback-методы через их заголовки в качестве дополнительных параметров. Задание матриц, например, может быть следующим:

задание матрицы состояния

```
A=[-0.09 -0.01
    1 0];
```

задание матрицы управления (входа)

```
B=[ 1 -7
    0 -2];
```

задание матрицы выхода

```
C=[ 0 2
    1 -5];
```

задание матрицы обхода системы

```
D=[-3 0
    1 0];
```

switch flag.

В зависимости от значения переменной flag происходит вызов того или иного метода:

case 0,

```
[sys,x0,str,ts]=mdlInitializeSizes(A,B,C,D);
```

case 1,

```
sys=mdlDerivatives(t,x,u,A,B,C,D);
```

case 3,

```
sys=mdlOutputs(t,x,u,A,B,C,D);
```

case { 2, 4, 9 }

```
sys=[];
```

otherwise

```
error(['Unhandled flag = ',num2str(flag)]);
```

end

```
function [sys,x0,str,ts]=mdlInitializeSizes(A,B,C,D)
```

```
sizes = simsizes;
```

```
sizes.NumContStates = 2; % Число непрерывных переменных состояния.
```

```

sizes.NumDiscStates = 0; % Число дискретных переменных состояния.
sizes.NumOutputs    = 2; % Число выходных переменных (размерность
                        % выходного вектора).
sizes.NumInputs     = 2; % Число входных переменных (размерность
                        % входного вектора).
sizes.DirFeedthrough = 1; % Прямой проход. Есть проход входного сигнала
% на выход.
% (матрица D не пустая).
sizes.NumSampleTimes = 1; % Размерность вектора шагов модельного
% времени.
sys = simsizes(sizes);
x0 = zeros(2,1); % Задание вектора начальных значений переменных
% состояния. Начальные условия нулевые.
str = []; % Параметр зарезервирован для будущего использования.
ts = [0 0]; % Матрица из двух колонок, задающая шаг модельного
% времени и смещение.

```

Функция для расчета значений производных вектора состояния непрерывной части системы имеет вид:

```

function sys=mdlDerivatives(t,x,u,A,B,C,D)
sys = A*x + B*u;

```

Функция для расчета значений выходных сигналов имеет вид:

```

function sys=mdlOutputs(t,x,u,A,B,C,D)
sys = C*x + D*u;

```

Пример модели с S-функцией csfunc.m приведен на рис. 4.6.<sup>1</sup>

## ЛАБОРАТОРНАЯ РАБОТА №2 (4 ЧАСА)

**Тема.** Создание Simulink-функции на языке MATLAB для модели непрерывной системы. **Цель.** Создать блок Simulink-функции и написать S-функцию в виде \*.m файла для модели непрерывной системы.

**Задание.** Создать S-функцию на языке MATLAB с использованием файл-шаблона sfuntmpl.m. Уравнения пространства состояния моделируемой непрерывной системы (4.1) заданы следующими параметрами:

1.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -24 \ -26 \ -9]$ ,  $B = [0 \ 0 \ 1]$ ,  $C = [1 \ 0 \ 1]$ ,  $D = 0$ ;
2.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -14 \ -26 \ -10]$ ,  $B = [0 \ 0 \ 1]$ ,  $C = [1 \ 1 \ 1]$ ,  $D = 1$ ;
3.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -10 \ -16 \ -10]$ ,  $B = [0 \ 0 \ 1]$ ,  $C = [1 \ 0 \ 0, \ 0 \ 0 \ 1]$ ,  $D = [1 \ 2]$ ;
4.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -5 \ -7 \ -3]$ ,  $B = [0 \ 0 \ 1]$ ,  $C = [1 \ 0 \ 0, \ 0 \ 0 \ 1]$ ,  $D = [2 \ 1]$ ;
5.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -10 \ -17 \ -2]$ ,  $B = [0 \ 0 \ 1]$ ,  $C = [1 \ 0 \ 1, \ 1 \ 1 \ 1]$ ,  $D = [1 \ 1]$ ;
6.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -2 \ -13 \ -6]$ ,  $B = [0 \ 0 \ 1]$ ,  
 $C = [1 \ 0 \ 0, \ 0 \ 1 \ 0, \ 0 \ 0 \ 1]$ ,  $D = [1 \ 2 \ 0]$ ;
7.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -3 \ -13 \ -5]$ ,  $B = [0 \ 0 \ 1]$ ,  
 $C = [1 \ 0 \ 0, \ 0 \ 1 \ 0, \ 0 \ 0 \ 1]$ ,  $D = [1 \ 1 \ 1]$ ;

<sup>1</sup> Взяты из книги И.В. Черных "Simulink: Инструмент моделирования динамических систем". [http://www.matlab.ru/simulink/book1/16.asp#16\\_4](http://www.matlab.ru/simulink/book1/16.asp#16_4).

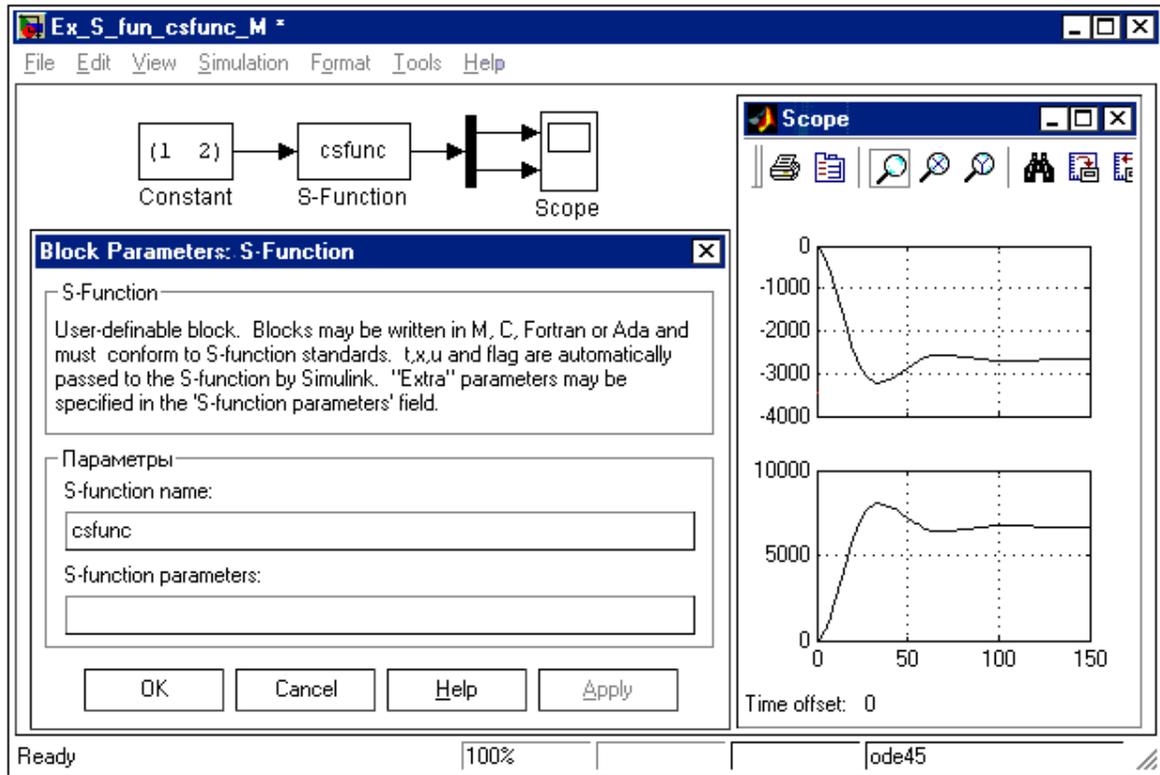


Рис. 4.6. Модель с S-функцией csfunc.m.

8.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -4 \ -6 \ -10]$ ,  $B = [0 \ 0 \ 1]$ ,  $C = [1 \ 0 \ 0, \ 0 \ 0 \ 1]$ ,  
 $D = [2 \ 1]$ ;
9.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -24 \ -26 \ -9]$ ,  $B = [0 \ 0 \ 1]$ ,  
 $C = [1 \ 0 \ 0, \ 0 \ 1 \ 0, \ 0 \ 0 \ 1]$ ,  $D = [0 \ 0 \ 0]$ ;
10.  $A = [0 \ 1 \ 0, \ 0 \ 0 \ 1, \ -4 \ -6 \ -5]$ ,  $B = [0 \ 0 \ 1]$ ,  
 $C = [1 \ 0 \ 0, \ 0 \ 1 \ 0, \ 0 \ 0 \ 1]$ ,  $D = [1 \ 1 \ 1]$ .

Сделать математическое описание модели непрерывной системы и проверку на устойчивость.

Контрольные вопросы.

1. Какие callback-методы применяются при создании модели непрерывной системы?
2. Какая функция используется для расчета значений вектора состояния непрерывной системы?
3. Какая функция пригодна для расчета значений производных вектора состояния непрерывной системы?
4. Какая функция используется для расчета значений вектора выхода непрерывной системы?
5. Как входные переменные можно задавать через окно диалога S-функции?

**Разработка S-функций гибридных систем управления**

В общем случае создаваемые на языке MATLAB блоки могут быть как непрерывными, так и дискретными или гибридными системами.

Модель дискретной системы, описываемой уравнениями пространства состояния, дана в файле dsfunc.m. С помощью уравнений пространства состояния моделируется дискретная система вида

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n), \\ y(n) &= Cx(n) + Du(n). \end{aligned} \tag{4.2}$$

Данная S-функция моделирует дискретную систему с двумя входами, двумя выходами и двумя переменными состояниями. Параметры модели (значения матриц A, B, C, D), как и в случае непрерывной системы, задаются в теле S-функции и передаются в callback-методы через их заголовки в качестве дополнительных параметров. Ниже приведен текст этой S-функции.

```
% S-функция dsfunc:
function [sys,x0,str,ts] = dsfunc(t,x,u,flag)
% Задание матриц:
A = [0.9135 0.1594 -0.7971 0.5947]; .
B = [0.05189 0 0.4782 0];
C = [0 1 1 0];
D = [0.01 0 0 -0.02];
switch flag,
case 0, [sys,x0,str,ts] = mdlInitializeSizes(A,B,C,D);
% Расчет значений вектора состояний дискретной части системы
case 2, sys = mdlUpdate(t,x,u,A,B,C,D);
% Расчет значений вектора выходных сигналов непрерывной части системы
case 3, sys = mdlOutputs(t,x,u,A,C,D);
case {1, 4, 9}
sys=[];
otherwise error(['unhandled flag = ',num2str(flag)]); end
function [sys,x0,str,ts] = mdlInitializeSizes(A,B,C,D)
sizes = simsizes; sizes.NumContStates = 0;
% Число непрерывных переменных состояний.
sizes.NumDiscStates = size(A,1); % Число дискретных переменных состояний.
sizes.NumOutputs = size(D,1);
% Число выходных переменных (размерность выходного вектора).
Sizes.NumInputs = size(D,2);
% Число входных переменных (размерность входного вектора).
sizes.DirFeedthrough = 1;
% Прямой проход. Проход входного сигнала на выход (матрица D не пустая).
sizes.NumSampleTimes = 1;
% Размерность вектора шагов модельного времени.
sys = simsizes(sizes);
x0 = zeros(sizes.NumDiscStates,1); % Задание начальных условий (нулевых).
str = []; % Параметр зарезервирован для будущего использования.
ts = [0.2 0]; % Матрица из двух колонок, задающая шаг времени и смещение.
% Функция расчета значений вектора состояния дискретной части системы.

function sys = mdlUpdate(t,x,u,A,B,C,D)
sys = A*x+B*u;
```

Функция для расчета значений вектора выходных сигналов.

```
function sys = mdlOutputs(t,x,u,A,C,D)
sys = C*x+D*u;
```

Пример модели с S-функцией dsfunc показан на рис. 4.7.<sup>1</sup>

<sup>1</sup> Взят из книги И.В. Черных "SIMULINK: Инструмент моделирования динамических систем". [http://www.matlab.ru/simulink/book1/16.asp#16\\_4](http://www.matlab.ru/simulink/book1/16.asp#16_4).

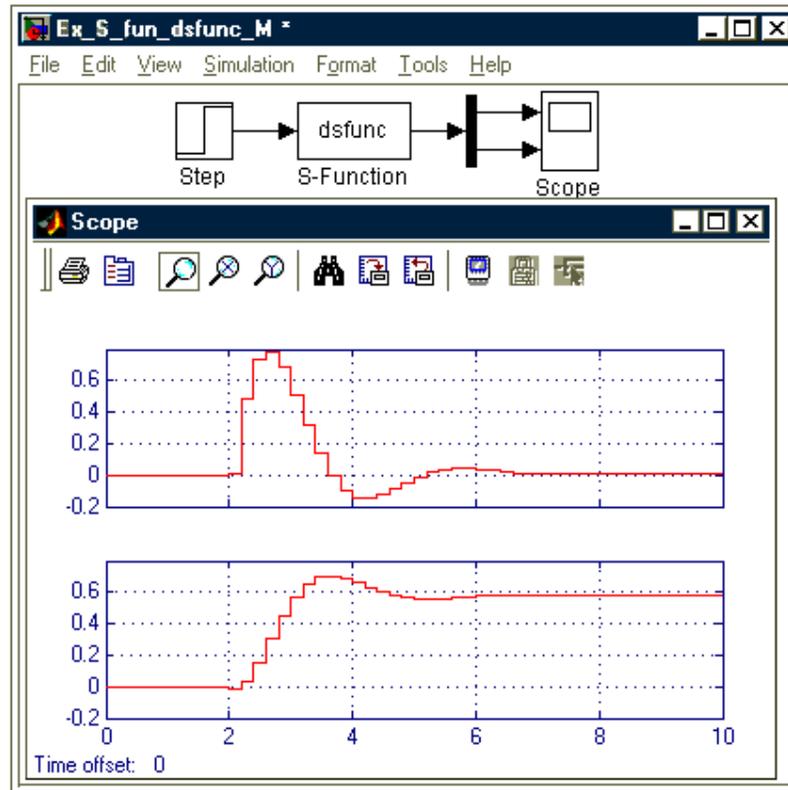


Рис. 4.7. Модель с S-функцией dsfunc.m.

Модель гибридной системы. Модель гибридной системы (комбинации непрерывной и дискретной частей) дана в файле mixedm.m. Рассматриваемая S-функция моделирует систему, состоящую из последовательно включенных интегратора ( $1/s$ ) и блока задержки ( $1/z$ ). Особенность S-функции для гибридной системы в том, что вызов callback-методов для расчета дискретной части системы выполняется в те же моменты времени, что и для непрерывной ее части, поэтому пользователь при написании S-функции должен предусмотреть проверку правильности вызова callback-методов, рассчитывающих дискретную часть системы. Текст этой S-функции следующий.

```
% S-функция dsfunc
function [sys,x0,str,ts] = mixedm(t,x,u,flag)
% Шаг модельного времени и смещение для дискретной части системы:
dperiod = 1;
doffset = 0;
switch flag
% В зависимости от значения переменной flag вызывается тот или иной метод:
%===== %
% Инициализация %
%===== %
case 0
[sys,x0,str,ts]=mdlInitializeSizes(dperiod,doffset);
%===== %
% Расчет производных %
%===== %
case 1
sys=mdlDerivatives(t,x,u);
%===== %
```

```

% Расчет значений вектора состояний дискретной части системы
%=====
case 2,
    sys=mdlUpdate(t,x,u,dperiod,doffset);
%=====
% Расчет значений вектора выходных сигналов
%=====
case 3
    sys=mdlOutputs(t,x,u,doffset,dperiod);
%=====
% Завершение расчета
%=====
case 9
    sys = [];
%=====
% Неизвестное значение переменной flag
%=====
otherwise
    error(['unhandled flag = ',num2str(flag)]);
end
% Окончание mixedm
%=====
% mdlInitializeSizes
% Функция инициализации
% Расчет начальных условий, значений вектора модельного времени,
% размерности матриц
%=====
function [sys,x0,str,ts]=mdlInitializeSizes(dperiod,doffset)
sizes = simsizes;
sizes.NumContStates = 1; % Число непрерывных переменных состояния.
sizes.NumDiscStates = 1; % Число дискретных переменных состояния.
sizes.NumOutputs = 1;
% Число выходных переменных (размерность выходного вектора).
sizes.NumInputs = 1;
% Число входных переменных (размерность входного вектора).
sizes.DirFeedthrough = 0;
% Прямой проход. Прохода входного сигнала на выход нет.
sizes.NumSampleTimes = 2;
% Размерность вектора шагов модельного времени.
% Шаги времени задаются для непрерывной и дискретной частей системы.
sys = simsizes(sizes);
x0 = zeros(2,1);
% Задание вектора начальных значений переменных состояния.
% Начальные условия нулевые
str = []; % Параметр зарезервирован для будущего использования.
ts = [0 0]; % Шаг модельного времени для непрерывной части системы.
dperiod doffset]; % Шаг модельного времени для дискретной части системы.
% Окончание mdlInitializeSizes
%=====
% mdlDerivatives
% Функция для расчета значений производных вектора состояния
%
```

```

% непрерывной части системы %
%===== %
function sys=mdlDerivatives(t,x,u)
sys = u;
% Окончание mdlDerivatives
%===== %
% mdlUpdate %
% Функция расчета значений вектора состояния дискретной части системы %
%===== %
function sys=mdlUpdate(t,x,u,dperiod,doffset)
% Расчет значения переменной состояния дискретной части системы
% выполняется в соответствии с дискретным шагом расчета
% (погрешность по времени выбрана равной 1e-15).
if abs(round((t - doffset)/dperiod) - (t - doffset)/dperiod) < 1e-15
    sys = x(1);
else
    sys = [];
end
% Окончание mdlUpdate
%===== %
% mdlOutputs %
% Функция для расчета значений вектора выходных сигналов %
%===== %
function sys=mdlOutputs(t,x,u,doffset,dperiod)
% Расчет значения выходного сигнала системы
% выполняется в соответствии с дискретным шагом расчета
% (погрешность по времени выбрана равной 1e-15).
% Если условие, заданное оператором if истинно, то выходной сигнал блока
% изменяется. Иначе выходной сигнал остается равным значению
% на предыдущем шаге.
if abs(round((t - doffset)/dperiod) - (t - doffset)/dperiod) < 1e-15
    sys = x(2);
else
    sys = [];
end
% Окончание mdlOutputs

```

### ЛАБОРАТОРНАЯ РАБОТА №3 (4 ЧАСА)

**Тема.** Создание S-функции на языке MATLAB для моделей дискретной и гибридной систем.

**Цель.** Создать блок S-функции и написать S-функцию в виде \*.m файла для модели дискретной системы.

**Задание.** Создать S-функцию на языке MATLAB, использовать при этом файл-шаблона sfuntmpl.m. Уравнения пространства состояния дискретной системы (4.2) заданы со следующими параметрами:

1.  $A = [0.9135 \ 0.1594, -0.7971 \ 0.5947]$ ,  $B = [0.051 \ 0, 0.47 \ 0]$ ,  
 $C = [1 \ 0, 0 \ 1]$ ,  $D = [0.01 \ 0, 0 \ -0.02]$ ;
2.  $A = [0.91 \ 0.15, -0.79 \ 0.59]$ ,  $B = [0 \ 1, 1 \ 0]$ ,  $C = [1 \ 1, 1 \ 1]$ ,  
 $D = [0.01 \ 0, 0 \ -0.02]$ ;
3.  $A = [0.5 \ -0.5, 0.2 \ 0.5]$ ,  $B = [0 \ 1]$ ,  $C = [1 \ 0]$ ,  $D = [1.1]$ ;

4.  $A = [0.1 \ 0.02, -0.05 \ 0.15]$ ,  $B = [1 \ 0]$ ,  $C = [1 \ 1]$ ,  $D = [5]$ ;
5.  $A = [0.5 \ -0.5, 0.2 \ 0.5]$ ,  $B = [0 \ 1]$ ,  $C = [1 \ 2]$ ,  $D = [0.02]$ ;
6.  $A = [0.1 \ 0.02, -0.05 \ 0.15]$ ,  $B = [0 \ 1, 1 \ 0]$ ,  $C = [1 \ 0, 0 \ 1]$ ,  
 $D = [0.01 \ -0.02]$ ;
7.  $A = [0.51 \ -0.53, 0.21 \ 0.51]$ ,  $B = [1 \ 0, 0 \ 1]$ ,  $C = [1 \ 0, 0 \ 1]$ ,  
 $D = [0.03 \ 0, 0 \ -0.01]$ ;
8.  $A = [0.1 \ 0.02, -0.05 \ 0.1]$ ,  $B = [0.051 \ 0, 0.47 \ 0]$ ,  $C = [1 \ 0, 0 \ 1]$ ,  
 $D = [0.01 \ 0, 0 \ -0.01]$ ;
9.  $A = [0.95 \ 0.154, -0.771 \ 0.57]$ ,  $B = [0.01 \ 0, 0.47 \ 0]$ ,  
 $C = [1 \ 0, 0 \ 1]$ ,  $D = [0.02 \ 1, 1 \ 0.3]$ ;
10.  $A = [0.52 \ -0.512, 0.21 \ 0.56]$ ,  $B = [0 \ 1.1, 0.1 \ 0.2]$ ,  $C = [1 \ 1, 2 \ 1]$ ,  
 $D = [0.02 \ 1, 1 \ -0.03]$ .

Изменить иконку блока S-функции в соответствии с вариантом задания.

Создать S-функцию на языке MATLAB для модели гибридной системы с единичной обратной связью.

Сделать математическое описание модели дискретной системы и проверку на устойчивость.

#### Контрольные вопросы.

1. Какие callback-методы используются при создании модели дискретной системы?
2. Какие callback-методы применяются при создании модели гибридной системы?
3. Какая функция пригодна для расчета значений вектора состояния дискретной части гибридной системы?

### ***Разработка S-функций дискретных систем управления***

Пример S-функции. Математическое описание сумматора в дискретной системе изображается уравнением

$$y(n) = Cx(n) + Du(n). \quad (4.3)$$

Модель дискретной системы, описываемой уравнениями (4.3), приведена в файле `discret_ad.m`, структура ее представлена на рис. 4.8.

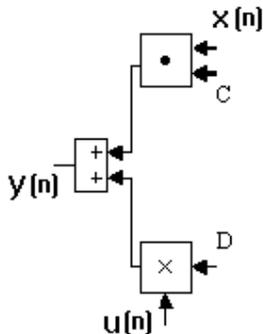


Рис. 4.8. Структурная схема дискретной системы.

Данная S-функция моделирует дискретную систему с двумя векторными входами  $C$ ,  $x$  и двумя скалярными входами  $D$  и  $u(n)$ , одним дискретным выходом  $y(n)$ . Значения векторов  $C$ ,  $x$  и константы  $D$  задаются в теле S-функции и передаются в callback-методы через их заголовки в качестве дополнительных параметров. На вход S-функции подается сигнал

$$u(n) = 0.8 \sin(nT), \quad T = 1.$$

Ниже приведен текст этой S-функции.

```
%S-функция discret_ad:
function [sys,x0,str,ts] = discret_ad(t,x,u,flag)
% Example M-file S-function implementing discrete equations:
%     y(n) = C*x(n) + D*u(n)
%
% See sfuntmpl.m for a general S-function template.
```

```
%
% Generate a discrete linear system:
C=[0 4.2382];
D=[-0.8141]
x=[0.2sin(0.1t);0.5cos(t)];

switch flag
```

```

%%%%%%%%%%
% Initialization %
%%%%%%%%%%
case 0,
[sys,x0,str,ts] = mdlInitializeSizes(C,D);
%%%%%%%%%%
% Update %
%%%%%%%%%%
case 2,
sys = [];
%%%%%%%%%%
% Output %
%%%%%%%%%%
case 3,
sys = mdlOutputs(t,x,u,C,D);
%%%%%%%%%%
% Terminate %
%%%%%%%%%%
case 9,
sys = []; % do nothing
%%%%%%%%%%
% Unexpected flags %
%%%%%%%%%%
otherwise
error(['unhandled flag = ',num2str(flag)]);
end
%end dsfunc
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
function [sys,x0,str,ts] = mdlInitializeSizes(C,D)

sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = size(C,1);
sizes.NumOutputs = size(D,1);
sizes.NumInputs = size(D,2);
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = ones(sizes.NumDiscStates,1);
str = [];
ts = [1 0];
% end mdlInitializeSizes
%=====
% mdlOutputs
% Return the output vector for the S-function
%=====
function sys = mdlOutputs(t,x,u,C,D)

```

```
sys = C*x+D*u;
%end mdlOutputs
```

Для функций  $x(n)$ ,  $u(n)$  и в соответствии с формулой (4.3) на рис. 4.9 изображен результат моделирования S-функции `discret_ad`, а на рис. 4.10 приведен результат ее тестирования.

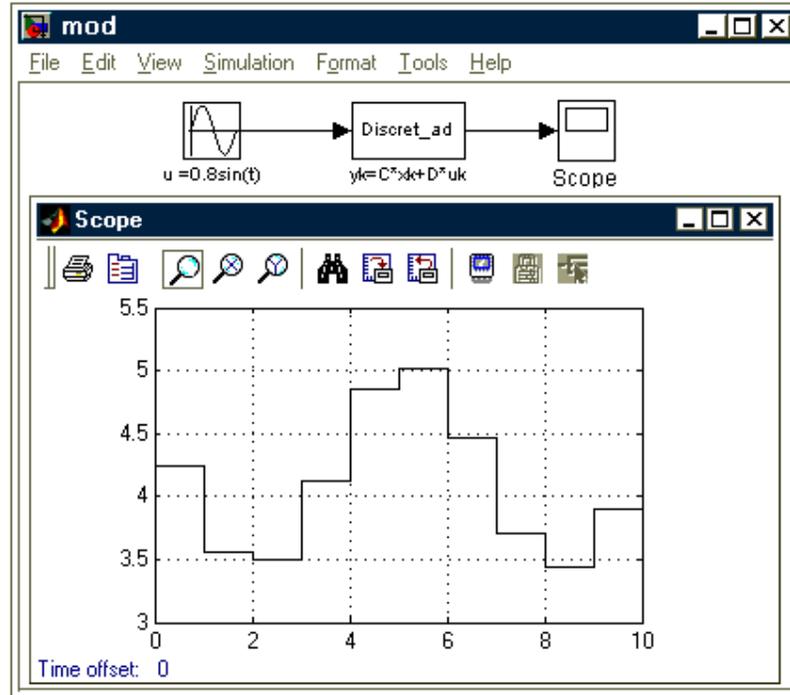


Рис. 4.9. Моделирование S-функции `discret_ad`.

#### ЛАБОРАТОРНАЯ РАБОТА №4 (4 ЧАСА)

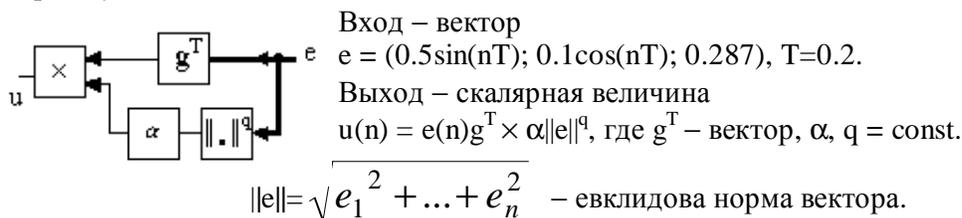
**Тема.** Построение S-функции для фрагмента структурной схемы дискретной модели.

**Цель.** Создать блок S-функции и описать S-функцию в виде \*.m файла для заданного фрагмента структурной схемы дискретной модели.

**Задание.** Создать S-функцию для расчета параметров фрагмента структурной схемы дискретной модели, входные параметры модели ввести через окно диалога S-функции.

Для тестирования созданной S-функции построить соответствующую Simulink-диаграмму.

1.



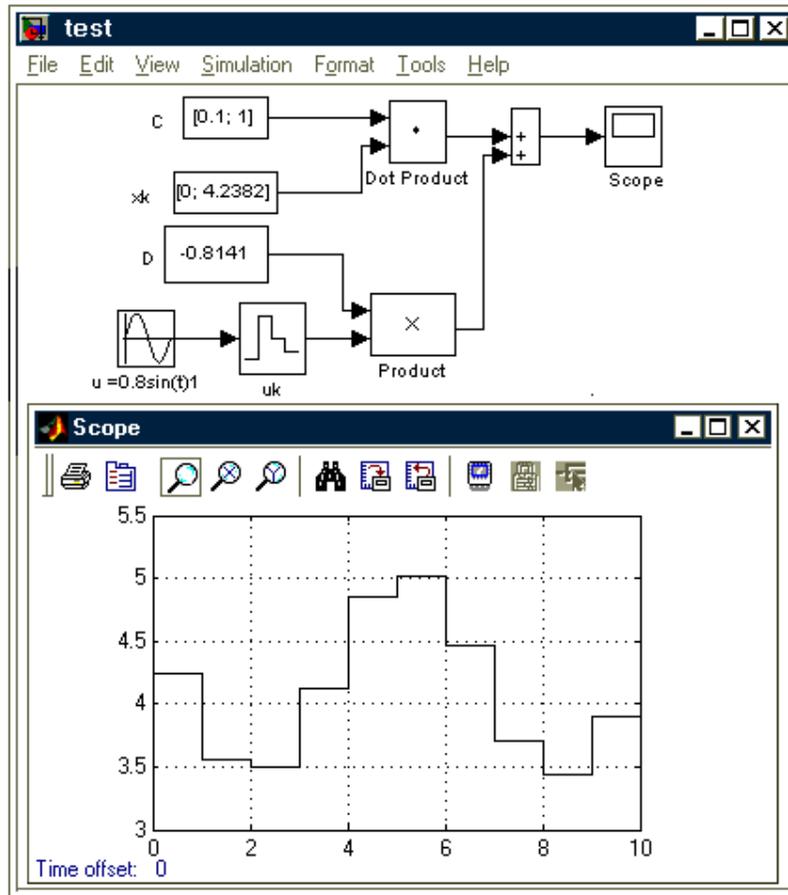
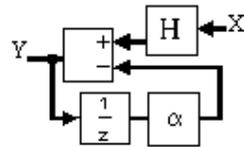
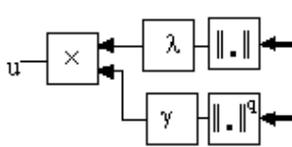
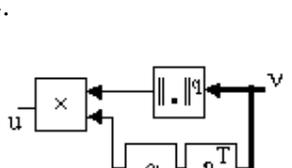
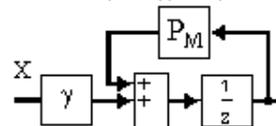


Рис. 4.10. Тестирование S-функции discret\_ad.

2. 

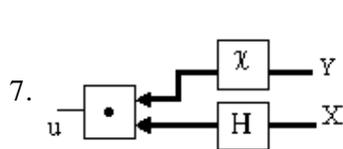
Вход – вектор  
 $X = (\sin(nT); 0.5\sin(nT) - 2.1; 0.1\cos(nT))$ ,  $T = 0.3$ .  
 Выход – вектор  $Y(n+1) = H \times X - \alpha Y(n)$ , где  $H$  – матрица (3×3),  $\alpha = \text{const}$ .
3. 

Вход – два вектора:  $e = (1.21nT; 0.5nT - 1)$ ,  
 $g = (0.2\sin(nT); \sin(nT); \cos(nT))$ ,  $T = 0.2$ .  
 Выход – скалярная величина  $u(n) = \gamma \|g\| \times \lambda \|e\|^q$ , где  $\|e\| = \sqrt{e_1^2 + \dots + e_n^2}$  – евклидова норма вектора,  $\gamma, \alpha, q = \text{const}$ .
4. 

Вход – вектор  $v = (6.2nT + 0.1; 0.3nT; 4.28nT)$ ;  $T = 0.1$ .  
 Выход – скалярная величина  $u(n) = \|v\|^q \times \alpha v g^T$ ,  
 где  $g^T$  – вектор,  $\|v\| = \sqrt{v_1^2 + \dots + v_n^2}$  – евклидова норма вектора,  $\alpha, q = \text{const}$ .
5. 

Вход – вектор  
 $X = (0.3\sin(nT); 2.1\sin(nT); 0.5\cos(nT))$ ,  $T = 0.7$ .  
 Выход – вектор  $Y(n+1) = P_M \times Y(n) + \gamma X$ , где  $P_M$  – диагональная матрица (3×3),  $\gamma = \text{const}$ .
6. 

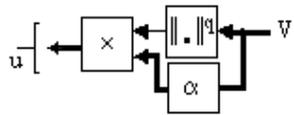
Вход – два вектора  
 $X = (5.35e^{-nT}; 0.3\sin(\pi nT + 1.5); 4e^{-nT})$ ,  
 $Y = (1.3\sin(nT); 0.1\sin(nT); 0.5nT)$ ,  $T = 0.5$ .



Выход – скалярная величина  
 $u(n) = \chi Y(n) \times (H \times X(n))$ , где  $\chi = \text{const.}$ ,  
 Вход – вектор

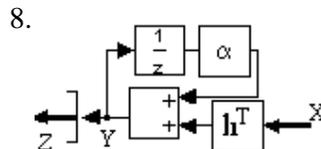
$V = (1.5 \sin(nT-0.4); 0.8 \sin(nT); |0.5nT-1|)$ ,  $T=0.2$ .

Выход – скалярная величина



$u(n) = (\|V\| \times \alpha V^T) \times h^T$ , где  $h$  – вектор,

$\|V\| = \sqrt{v_1^2 + \dots + v_n^2}$  – евклидова норма вектора,  $\alpha, q = \text{const.}$

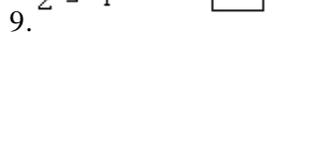


Вход – вектор

$X = (1.1nT; 0.7nT-1; 0.2 \sin(nT-1); 7.5nT)$ ,  $T=0.6$ .

Выход – вектор

$Z(n) = Y(n+1)h$ ,  $Y(n+1) = (h^T X + \alpha Y(n))$ , где  $h$  – вектор,  $\alpha = \text{const.}$

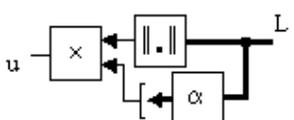


Вход – вектор

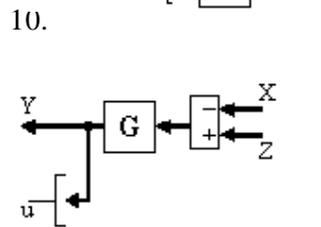
$L = (0.01e^{-nT}; 0.7nT+1; 0.25 \sin(nT+0.1))$ ,  $T=0.3$ .

Выход – скалярная величина

$u(n) = \|L\| \alpha L^T m$ , где  $m$  – вектор,



$\|L\| = \sqrt{l_1^2 + \dots + l_n^2}$  – евклидова норма вектора,  
 $\alpha = \text{const.}$



Вход – два вектора

$Z = (0.01e^{nT}; 0.51 \sin(nT+0.1); 0.1 \sin(\pi nT+0.6))$ ,  $X = (5.3nT; 0.7e^{-nT}-1.2; 0.3 \sin(nT))$ ,  $T=0.9$ .

Выход – вектор  $Y(n) = (\alpha Z - X) \times G^T$  и

скалярная величина  $u(n) = \|Y(n)\|^t$ , где  $G$  – диагональная матрица,

$\|Y\| = \sqrt{y_1^2 + \dots + y_n^2}$  – евклидова норма вектора,  $\alpha, t = \text{const.}$

### Контрольные вопросы.

1. Какие callback-методы были использованы при реализации S-функции модели дискретной системы?

2. Какие значения переменных или констант можно передавать как параметры через окно диалога S-функции?

3. Какие стандартные методы или функции использованы для расчета значений входных и выходных векторов дискретной системы?

## **8. ПЕРЕЧЕНЬ ПРОГРАММНЫХ ПРОДУКТОВ, ИСПОЛЬЗУЕМЫХ В ПРАКТИКЕ ВЫПУСКНИКОВ И УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ**

Программный продукт – математический пакет MatLab.

*Еремин Е.Л., Еремина В.В., Семичевская Н.П., Шевко Д.Г.* Алгоритмы и S-модели гибридных систем адаптивного управления (практикум в среде SIMULINK). Благовещенск: Амурский гос. ун-т, 2005. 205 с.

Допущено Учебно-методическим объединением вузов по университетскому политехническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальностям «Информационные системы и технологии» и «Автоматизированные системы обработки информации и управления»

Книга посвящена вопросам синтеза и компьютерного моделирования гибридных систем прямого адаптивного управления с использованием математического пакета MATLAB и среды визуального моделирования SIMULINK. Примеры разработки и использования S-моделей адаптивных систем или их фрагментов рассмотрены в рамках выполнения лабораторных и курсовой работ при проведении вычислительного эксперимента.

## **9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРИМЕНЕНИЮ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ ПРЕПОДАВАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ**

В качестве методических указаний по применению современных информационных технологий для преподавания МАДС используется учебное пособие – *Еремин Е.Л., Еремينا В.В., Семичевская Н.П., Шевко Д.Г.* Алгоритмы и S-модели гибридных систем адаптивного управления (практикум в среде SIMULINK). Благовещенск: Амурский гос. ун-т, 2005. 205 с., содержание которой следующее:

1. ГИБРИДНЫЕ СИСТЕМЫ АДАПТИВНОГО УПРАВЛЕНИЯ .....
  - 1.1. Характеристика проблемы управления в условиях априорной неопределенности .....
  - 1.2. Способы построения гибридных систем адаптации .....
  - 1.3. Формализация задачи синтеза адаптивных систем управления ..
  - 1.4. Критерий гиперустойчивости и синтез нелинейно преобразованных гибридных систем прямого адаптивного управления .....
2. ПАКЕТ MATLAB И SIMULINK МОДЕЛИРОВАНИЕ .....
3. СИНТЕЗ И S-МОДЕЛИ АДАПТИВНЫХ СИСТЕМ .....
4. СОЗДАНИЕ S-ФУНКЦИЙ (лабораторные работы) .....
5. СОЗДАНИЕ MATLAB-ПРИЛОЖЕНИЙ (курсовая работа) .....
6. НЕСТАЦИОНАРНО-НЕЛИНЕЙНЫЕ СИСТЕМЫ (самостоятельная работа)

## 10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ МЕЖСЕССИОННОГО И ЭКЗАМЕНАЦИОННОГО КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ

1. Межсессионная аттестация студентов проводится дважды в семестр на 6 и 8 неделях 8-го семестра.
2. Аттестационная оценка складывается из оценок, полученных на аттестационных занятиях по лабораторным работам, собеседованиях и при защите курсовой работы:
  - Первое аттестационное занятие.* Проверка знаний и навыков студентов по анализу линейных и нелинейных систем автоматического управления при прохождении случайных сигналов
  - Второе аттестационное занятие.* Проверка знаний и навыков студентов по созданию S-функций.
  - Первое собеседование.* Гибридные системы со скалярным управлением.
  - Второе собеседование.* Гибридные системы с векторным управлением.
  - Защита курсовой работы.* Создание MATLAB приложений.
3. Организация аттестации студентов, проводится в соответствии с положением АмГУ о курсовых экзаменах и зачетах\*

---

\* 2.1. Организация аттестации студентов в университете по специальностям и направлениям высшего профессионального образования регламентируется рабочим учебным планом, расписанием учебных занятий и программами учебных дисциплин, утверждаемыми в установленном в университете порядке.

Контроль за качеством освоения образовательных программ осуществляется путем текущей внутрисеместровой аттестации, ректорской контрольной аттестации, промежуточной аттестации студентов в форме курсовых экзаменов и зачетов и итоговой аттестации выпускников.

2.2. Курсовые экзамены и зачеты проводятся по дисциплинам утвержденного учебного плана по соответствующим специальностям и направлениям высшего профессионального образования. Знания, умения и навыки обучающегося определяются оценками "отлично", "хорошо", "удовлетворительно", "неудовлетворительно", "зачтено" и "незачтено".

2.3. Студенты, обучающиеся по основным программам высшего профессионального образования, сдают в течение учебного года не более 10 экзаменов и 12 зачетов. В это число не входит аттестация по физической культуре и факультативным дисциплинам.

Студенты, обучающиеся в сокращенные сроки (по индивидуальным планам), в течение учебного года сдают не более 20 экзаменов и 24 зачетов.

2.4. Сроки проведения курсовых зачетов и экзаменов (экзаменационная сессия) и начало очередного учебного семестра устанавливаются графиком учебного процесса, утвержденным проректором по учебной работе.

Расписание экзаменов составляется в соответствии с графиком учебного процесса, утверждается проректором по учебно-научной работе и доводится до сведения преподавателей и студентов не позднее, чем за две недели до начала сессии. Расписание составляется таким образом, чтобы на подготовку к экзаменам по каждой дисциплине было отведено не менее 3 дней, исключая день предыдущего экзамена. По согласованию с деканами и заведующими соответствующих кафедр отдельные экзамены (зачеты) могут проводиться в течение семестра по завершении преподавания дисциплины.

## 11. ФОНД ТЕСТОВЫХ И КОНТРОЛЬНЫХ ЗАДАНИЙ ДЛЯ ОЦЕНКИ КАЧЕСТВА ЗНАНИЙ ПО ДИСЦИПЛИНЕ

### Контрольные вопросы для проверки знаний по применению программной среды MatLab

1. Охарактеризуйте назначение m-файлов: `conv(a, b)`, `tf2ss(num,den)`, `ctrb(a,b)`, `obsv(a,c)`.
2. Поясните суть методов прямого, последовательного и параллельного программирования.
3. Сформулируйте критерии управляемости и наблюдаемости Калмана.
4. Охарактеризуйте назначение m-файлов `yprime(t,y)`, `ode45('имя файла',t0,tfinal,y0,tol,trace)`.
5. Дайте определение основных временных и частотных характеристик динамической системы.
6. Приведите в общем виде уравнения частотных характеристик: `mag`, `phase`, `re` и `im`.
7. Охарактеризуйте назначение m-файлов `bode(a,b,c,d,1,w)`, `nyquist(a,b,c,d,1,w)`.
8. Охарактеризуйте назначение m-файлов: `cloop(num0,den0,sign)`, `trapz(x,y)`.
9. Дайте определения и охарактеризуйте назначение корреляционной функции и спектральной плотности стационарных случайных процессов.
10. В чем суть гипотезы эргодичности для стационарных случайных процессов?
11. Почему знание корреляционной функции и спектральной плотности позволяют полностью охарактеризовать свойства стационарных случайных процессов?
12. Для чего могут быть использованы S-функции при моделировании сложных систем?
13. Какими возможностями обладают S-функции пользователя?
14. Как S-функция подключается к модели Simulink?
15. С какими типами данных можно работать в S-функциях?
16. Какие callback-методы применяются при создании модели непрерывной системы?
17. Какая функция используется для расчета значений вектора состояния непрерывной системы?
18. Какая функция пригодна для расчета значений производных вектора состояния непрерывной системы?
19. Какая функция используется для расчета значений вектора выхода непрерывной системы?
20. Как входные переменные можно задавать через окно диалога S-

функции? Какие callback-методы используются при создании модели дискретной системы?

21. Какие callback-методы применяются при создании модели гибридной системы?
22. Какая m-функция пригодна для расчета значений вектора состояния дискретной части гибридной системы?
23. Какие callback-методы были использованы при реализации S-функции модели дискретной системы?
24. Какие значения переменных или констант можно передавать как параметры через окно диалога S-функции?
25. Какие стандартные методы или функции использованы для расчета значений входных и выходных векторов дискретной системы?

Тестовые задания по проверке остаточных знаний по дисциплине «МАДС» для специальностей 230102

1. Для случая эргодического сигнала с распределением по закону Пуассона корреляционная функция вид  $R(T) = dd \cdot \exp(-aa|T|)$ , укажите соответствующую ей спектральную плотность:

- a)  $S(w) = \frac{2 \cdot aa \cdot dd}{w^2 + (aa)^2}$ ,      b)  $S(w) = \frac{aa \cdot dd}{w^2 + (aa)^2} + \frac{1}{(bb+w)^2 + (aa)^2}$       c)  $S(w) = \frac{aa \cdot dd}{bb} \cdot \left[ \frac{2 \cdot bb - w}{(bb-w)^2 + (aa)^2} + \frac{2 \cdot bb + w}{(bb+w)^2 + (aa)^2} \right]$       d)  $S(w) = \frac{aa \cdot dd}{bb} \cdot \left[ \frac{2 \cdot bb + w}{(bb+w)^2 + (aa)^2} \right]$ .

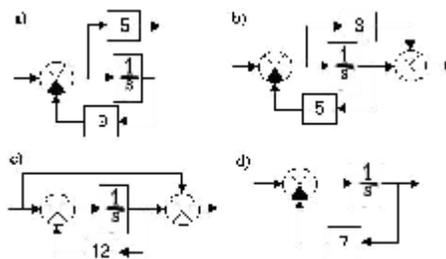
2. Укажите уравнение переходного процесса аperiodического звена с коэффициентом передачи равным единице

- a)  $h(t) = T_1 \left( T_2 - e^{-\frac{t}{T_2}} \right)$       b)  $h(t) = \frac{T_1}{T_2} \cdot e^{-\frac{t}{T_2}}$       c)  $h(t) = \frac{T_2}{T_1} \cdot e^{-\frac{t}{T_1}}$       d)  $h(t) = 1 - e^{-\frac{T_1}{T_2} t}$

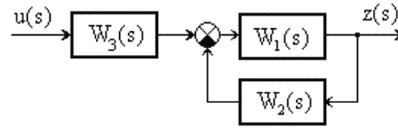
3. Сигнал, обладающий свойством  $w(t) = \frac{dh(t)}{dt}$  называется:

- a) Дельта-функцией      б) функцией Дирака      с) весовой функцией      d) функцией Ляпунова

4. Укажите структурную схему упругого звена:



5. Общая передаточная функция звена, представленного на рисунке, имеет вид

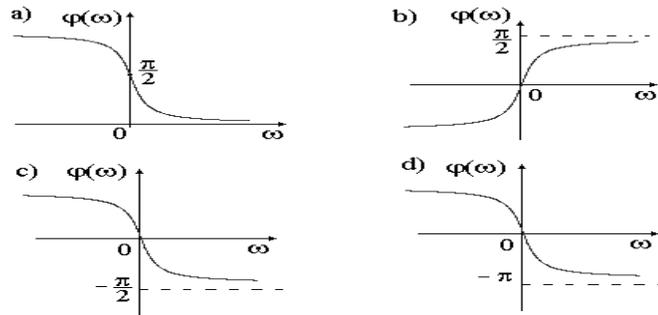


- a)  $W(s) = \frac{W_1(s) \cdot W_2(s) \cdot W_3(s)}{1 - W_1(s) \cdot W_2(s)}$       b)  $W(s) = \frac{W_1(s) \cdot W_3(s)}{1 + W_1(s) \cdot W_2(s)}$       c)  $W(s) = \frac{W_1(s) \cdot W_2(s)}{1 + W_1(s) \cdot W_2(s) \cdot W_3(s)}$       d)  $W(s) = \frac{1 + W_1(s) \cdot W_2(s)}{W_1(s) \cdot W_2(s) \cdot W_3(s)}$

6. Уравнение переходного процесса для интегрирующего звена с замедлением имеет вид

- a)  $h(t) = k \left[ 1 - e^{-\frac{t}{T}} \right]$       b)  $h(t) = k \left[ e^{-\frac{t}{T}} - 1 - t \right]$       c)  $h(t) = k \left[ t - T + T e^{-\frac{t}{T}} \right]$       d)  $h(t) = k \left[ 1 - T \cdot t - T e^{-\frac{t}{T}} \right]$

7. Укажите график фазовой частотной характеристики динамического звена второго порядка



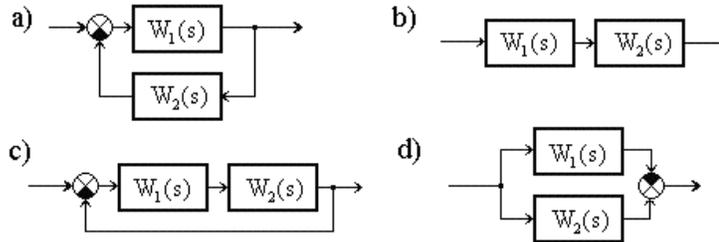
8. Если передаточная функция имеет вид  $W(p) = \frac{k}{T_2^2 p^2 + T_1 p + 1}$ ,  $T_1^2 - 4T_2^2 < 0$ , то  $x$  - это

- a) степенью устойчивости      b) коэффициентом затухания      c) степенью затухания      d) собственной частотой колебаний

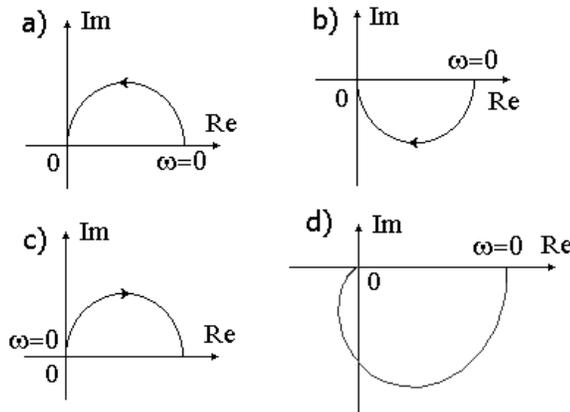
9. Переходной процесс колебательного звена имеет вид

- a)  $h(t) = 1 + \frac{g}{x \cdot I} \cdot e^{-g \cdot t} \cdot \sin(I t + j)$       b)  $h(t) = k \cdot \left[ 1 - \frac{g}{x \cdot I} \cdot e^{-g \cdot t} \cdot \sin(I t + j) \right]$       c)  $h(t) = -\frac{g}{x \cdot I} \cdot e^{-g \cdot t} \cdot \sin(I t - j)$       d)  $h(t) = k + 1 + \frac{g}{x \cdot I} \cdot e^{-g \cdot t} \cdot \sin(I t + j)$

10. Укажите схему, на которой представлено параллельное соединение звеньев



11. Укажите годограф звена второго порядка ( $\omega > 0$ )



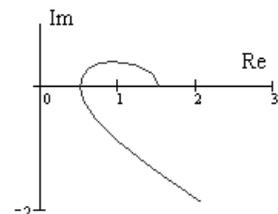
12. Для звена с передаточной функцией  $W(j\omega) = k \cdot \frac{\text{Re } Q(j\omega) + j \text{Im } Q(j\omega)}{\text{Re } R(j\omega) + j \text{Im } R(j\omega)}$ , укажите уравнение ее фазовой частотную характеристики

$$j(\omega) = \arctg \frac{\text{Re } R(j\omega)}{\text{Re } Q(j\omega)} - \arctg \frac{\text{Im } R(j\omega)}{\text{Im } Q(j\omega)}, \quad j(\omega) = -\arctg \frac{\text{Im } Q(j\omega)}{\text{Re } Q(j\omega)} + \arctg \frac{\text{Im } R(j\omega)}{\text{Re } R(j\omega)},$$

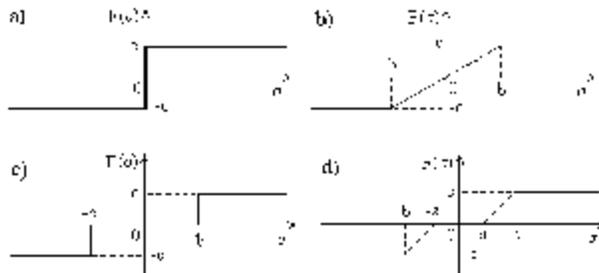
$$j(\omega) = \arctg \frac{\text{Im } Q(j\omega)}{\text{Im } R(j\omega)} - \arctg \frac{\text{Re } Q(j\omega)}{\text{Re } R(j\omega)}, \quad j(\omega) = \arctg \frac{\text{Im } Q(j\omega)}{\text{Re } Q(j\omega)} - \arctg \frac{\text{Im } R(j\omega)}{\text{Re } R(j\omega)}.$$

13. Если годограф Михайлова имеет вид, то система является:

- a) абсолютно устойчивой;
- b) устойчивой;
- c) неустойчивой
- d) условно устойчивой



14. Укажите, какой нелинейный элемент описывается уравнением  $F(s) = \begin{cases} c, & s > b, \\ \frac{c}{b} \cdot s, & -b \leq s \leq b, \\ -c, & s < -b. \end{cases}$



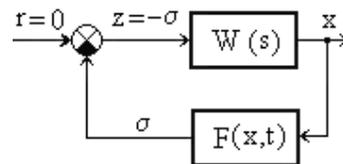
15. Если характеристическое уравнение системы имеет вид  $D(p) = a_0 p^5 + a_1 p^4 + a_2 p^3 + a_3 p^2 + a_4 p + a_5$ , то укажите верную матрицу Рауса-Гурвица

a)  $\begin{vmatrix} a_1 & a_3 \\ a_0 & a_2 \end{vmatrix}$ , b)  $\begin{vmatrix} a_1 & a_3 & a_5 & 0 & 0 \\ a_0 & a_2 & a_4 & 0 & 0 \\ 0 & a_1 & a_3 & a_5 & 0 \\ 0 & a_0 & a_2 & a_4 & 0 \\ 0 & 0 & a_1 & a_2 & a_5 \end{vmatrix}$ , c)  $\begin{vmatrix} a_1 & a_3 & a_5 \\ a_0 & a_2 & a_4 \\ 0 & a_1 & a_3 \end{vmatrix}$ , d)  $\begin{vmatrix} a_1 & a_3 & a_5 & 0 & 0 \\ a_0 & a_2 & a_4 & 0 & 0 \\ a_1 & a_2 & a_3 & a_4 & a_5 \\ a_0 & a_1 & a_2 & a_3 & a_4 \\ 0 & 0 & a_1 & a_2 & a_5 \end{vmatrix}$

16. Если функция сохраняет один и тот же знак, но может обращаться в нуль не только в начале координат, но и в других точках области, то она в данной области является

- a) знакоопределенной, b) знакопостоянной, c) знакопеременной, d) функцией Ляпунова

17. Укажите верное интегральное неравенство В.М. Попова, которое должно выполняться  $\forall t \geq 0$ , согласно критерию гиперустойчивости, чтобы нелинейная система, изображенная на рисунке, была асимптотически гиперустойчивой



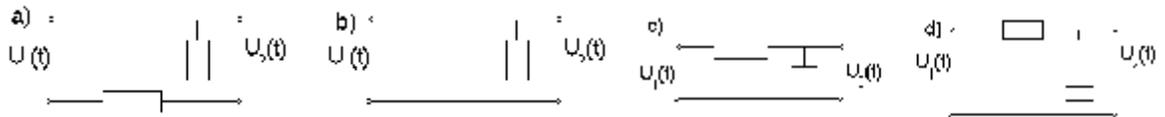
a)  $h(0,t) = -\int_0^t z(s)x(s)ds > -g_0^2 = const,$  b)

$h(0,t) = \int_0^t z(s)x(s)ds < -g_0^2 = const,$

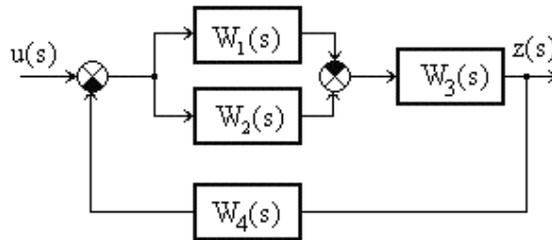
c)  $h(0,t) = -\int_0^t z(s)x(s)ds \geq -g_0^2 = const,$

d)  $h(0,t) = -\int_{x(0)}^{x(t)} z(s)x(s)ds \geq g(t) .$

18. Укажите схему апериодического звена первого порядка



19. Общая передаточная функция звена, представленного на рисунке, имеет вид

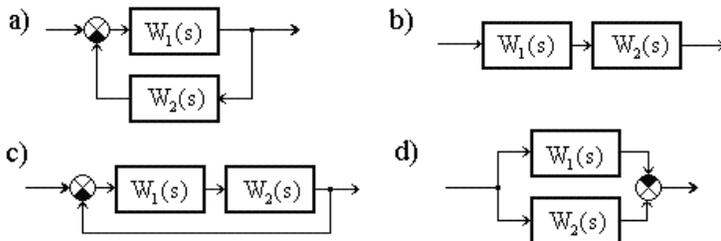


$W(s) = \frac{W_1(s) - W_2(s) \cdot W_3(s)}{1 + (W_1(s) - W_2(s) \cdot W_3(s)) \cdot W_4(s)},$  b)  $W(s) = \frac{(W_2(s) - W_1(s)) \cdot W_3(s)}{1 + (-W_1(s) + W_2(s)) \cdot W_3(s) \cdot W_4(s)},$

c)  $W(s) = \frac{(W_1(s) - W_2(s)) \cdot W_3(s)}{1 + (W_1(s) - W_2(s)) \cdot W_3(s) \cdot W_4(s)},$  d)  $W(s) = \frac{(W_2(s) - W_1(s)) \cdot W_3(s) \cdot W_4(s)}{1 + (-W_1(s) + W_2(s)) \cdot W_3(s)}.$

20. Укажите схему, на которой представлено соединение с итоговой передаточной функцией

ей  $W(s) = \frac{W_1(s)}{1 + W_1(s)W_2(s)}$



## 12. КОМПЛЕКТ ЭКЗАМЕНАЦИОННЫХ БИЛЕТОВ

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 1

1. Коэффициенты гармонической линеаризации релейных элементов с гистерезисом.
2. Уравнение Эйлера-Лагранжа и метод множителей Лагранжа в задаче оптимального управления с закрепленными концами и фиксированным временем.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 2

1. Расчет установившихся ошибок при случайных стационарных воздействиях.
2. Вариационные методы оптимального управления (вывод уравнения Эйлера).

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 3

1. Методы анализа периодических режимов (основной алгебраический метод и с использованием графиков коэффициентных гармонической линеаризации) и их устойчивости.
2. Пример задачи оптимального управления с закрепленными концами и фиксированным временем.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 4

1. Расчет по минимуму среднеквадратической ошибки.
2. Вывод условий трансверсальности и запись уравнений Эйлера-Лагранжа в задаче с подвижными концами и фиксированным временем.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 5

1. Спектральная плотность случайных процессов.
2. Статические и скоростные ошибки автоколебательных систем.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 6

1. Случайные процессы и корреляционные функции.
2. Методы анализа периодических режимов (использование годографа Михайлова и метод коэффициентных соотношений) и их устойчивости.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 7

1. Вероятностные характеристики дискретных и непрерывных случайных величин.
2. Применение прямого метода Ляпунова в задаче адаптивного управления.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

### Экзаменационный билет № 8

1. Стационарные случайные процессы и их характеристики.
2. Пример задачи оптимального управления с подвижными концами и фиксированным временем.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

Экзаменационный билет № 9

1. Частотный метод определения автоколебаний и их устойчивости.
2. Построение робастных законов управления в условиях априорной неопределенности.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

Экзаменационный билет № 10

1. Прохождение случайных сигналов через линейную систему.
2. Приближенные методы анализа нелинейных систем первого типа на основе гармонической линеаризации.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры  
« \_\_\_\_ » \_\_\_\_\_ 200\_ г.  
Заведующий кафедрой  
Утверждаю: \_\_\_\_\_

Кафедра ИУС  
Факультет МиИ  
Курс III  
Дисциплина МАДС

Экзаменационный билет № 11

1. Метод пространства состояний (прямое, последовательное и параллельное программирование).
2. Коэффициенты гармонической линеаризации релейных элементов без гистерезиса.

### **13. КАРТА КАДРОВОЙ ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ**

Лектор – профессор, д-р техн. наук  
Еремин Евгений Леонидович

Руководитель лаб. работ – доцент, канд. техн. наук  
Семичевская Наталия Петровна