

Содержание

	с.
Введение	4
1 Лабораторная работа №1. Моделирование объекта с сосредоточенными параметрами аналитическим методом	5
2 Лабораторная работа №2. Исследование статических и динамических процессов в резервуаре	11
3 Лабораторная работа №3. Прямые методы идентификации линейных объектов первого порядка	16
4 Лабораторная работа №4. Идентификация линейного объекта с помощью частотной характеристики	20
5 Лабораторная работа №5. Параметрическая идентификация линейных динамических объектов	24
6 Лабораторная работа №6. Идентификация линейных объектов средствами пакета System Identification Toolbox	29
7 Лабораторная работа №7. Непараметрическая идентификация динамических объектов	34
Список литературы	39

Введение

Существует два принципиально различных подхода к построению математических моделей объектов управления. Первый подход основан на составлении дифференциальных уравнений моделей, базирующихся на использовании основных физических законов: сохранения массы, энергии и количества движения. Этот подход называется аналитическим, а разработанные модели - *аналитическими* моделями объекта. Вторым подходом базируется на концепции "черного ящика", то есть вся информация об объекте получается только в результате наблюдений за ним при пассивном и активном эксперименте. Полученные таким образом модели называются *эмпирическими* (экспериментальными). Задачи восстановления таких моделей (динамических характеристик объектов управления) изучаются в *теории идентификации*. Экспериментальные методы определения динамических характеристик делятся на три основные группы: прямые методы идентификации, параметрическая и непараметрическая идентификации.

Предлагаемые методические указания к лабораторным работам по дисциплине «Моделирование и идентификации объектов управления» включают 7 работ и предназначены для освоения методов аналитического моделирования и идентификации. Для выполнения этих работ в основном используются различные инструментариумы программной системы MatLab:

- режим *прямых вычислений* (в командном режиме), то есть без подготовки программы. Командное окно часто используется и при работе в различных пакетах системы MatLab. Например, данные между отдельными пакетами не передаются напрямую, а только лишь через командное окно;

- создание *M-файла* в специальном окне редактора программ для использования процедур;

- пакет Simulink, предоставляющий пользователю удобные и простые средства, в том числе визуального объектно-ориентированного программирования, для блочного моделирования линейных и нелинейных динамических систем;

- специальное расширение Stateflow системы MatLab + Simulink, которое позволяет моделировать системы и устройства, поведение которых зависит от внешних событий. Пакет имеет свой графический интерфейс пользователя, позволяющий создавать графические диаграммы, которые служат для визуального представления работы моделируемой системы;

- пакет System Identification Toolbox, содержащий графический пользовательский интерфейс для построения корреляционных, спектральных и параметрических моделей различных порядков;

- пакет подгонки кривых Curve Fitting Toolbox программы Matlab для решения задачи аппроксимации функций.

Идентификация объекта с помощью частотной характеристики проводится в среде программы, разработанной на кафедре «Инженерная кибернетика».

1 Лабораторная работа №1. Моделирование объекта с сосредоточенными параметрами аналитическим методом

Цель работы: приобретение навыков применения основных законов сохранения при моделировании объектов с сосредоточенными параметрами; освоение приемов моделирования и анализа объектов в пакете Simulink программы MatLab.

1.1 Задание к лабораторной работе

В процессе выполнения лабораторной работы студент должен:

- изучить аналитические методы моделирования объектов управления с сосредоточенными параметрами [1,2];
- ознакомиться с приемами моделирования объектов в пакете Simulink системы MatLab [1,4];
- ознакомиться с моделируемым процессом;
- построить блок-диаграмму динамической модели рассматриваемого процесса в пакете Simulink;
- с целью исследования провести имитационные эксперименты на модели;
- выполнить анализ полученных результатов; сделать выводы.

1.2 Описание процесса

Объектом исследования является мельница, в которую поступает входной поток исходного перерабатываемого материала \hat{O}_0 , содержащего крупные фракции. Выходной поток из мельницы $\hat{O}_{\text{вых}}$ пропорционален массе материала в мельнице M

$$\Phi_{\text{вых}} = \alpha M,$$

где α - константа пропорциональности.

В мельнице происходит измельчение крупной фракции со скоростью W , пропорциональной массе крупной фракции, описываемой следующей зависимостью

$$W = k(M)(1 - C)M,$$

где $C = \frac{m}{M}$, $k(M) = \frac{k_0(2M^+ - M)}{M^+}$.

Здесь m - масса мелкой фракции в мельнице;
 C - доля мелкой фракции;
 k_0 (1/час) - константа скорости измельчения;
 M^+ - характерная масса, соответствующая максимальной скорости измельчения (максимальной производительности мельницы).
Значение $M^+ = 10 \text{ т}$; $k_0 = 1,5$ (1/час).

Производительность мельницы является функцией выходного потока $P = C \cdot \Phi_{\text{вых}}$, то есть $P = \alpha \cdot m(t)$.

Математическая модель процесса в виде системы дифференциальных уравнений выводится на основе аналитических методов моделирования. При составлении уравнений математической модели процесса используются следующие уравнения баланса:

а) баланс общей массы в мельнице

$$\frac{dM}{dt} = \Phi_0 - \Phi_{\text{вых}};$$

б) баланс массы мелких фракций (готового продукта) в мельнице

$$\frac{d(CM)}{dt} = W - \tilde{N} \cdot \hat{O}_{\text{аио}}.$$

В конечном итоге, окончательный вид системы дифференциальных уравнений, описывающих рассматриваемый процесс, имеет вид:

$$\begin{cases} \frac{dM}{dt} = \Phi_0 - \alpha M, \\ \frac{dm}{dt} = 0,15 \cdot (10 - M) \cdot (M - m) - \alpha \cdot m. \end{cases} \quad (1.1)$$

1.3 Построение блок-диаграммы модели объекта в пакете Simulink

Пакет моделирования динамических систем Simulink предназначен для математического моделирования линейных и нелинейных динамических систем и устройств, представленных своей функциональной блок-схемой.

Для построения функциональной блок-схемы моделируемых устройств пакет Simulink системы MatLab имеет обширную библиотеку блочных компонентов и удобный редактор блок-схем. Ввод характеристик исследуемых систем производится в диалоговом режиме, путем графической сборки схемы соединений стандартных элементарных звеньев. В результате такой сборки образуется модель исследуемой системы, которую называют *S-моделью*. Модель хранится в файле с расширением *.mdl*.

Simulink автоматизирует следующий наиболее трудоемкий этап моделирования: он составляет и решает сложные системы алгебраических и дифференциальных уравнений, описывающих заданную функциональную схему (модель), обеспечивая удобный и наглядный визуальный контроль за поведением созданного пользователем виртуального устройства.

Для построения *S-модели* используются блоки из библиотеки Simulink. Библиотека блоков Simulink представляет собой набор визуальных объектов, используя которые можно собирать, как из кубиков, произвольную конструкцию. Для любого блока можно получать требуемое число копий и

использовать каждую из них автономно. Более того, практически для всех блоков существует возможность индивидуальной настройки: пользователь может изменить как внутренние параметры блоков (например, количество входов), так и внешнее оформление (размер, цвет, имя и т. д.). На порядок соединения блоков друг с другом также не накладывается никаких ограничений. Конечно, при связывании блоков необходимо соблюдать определенные правила, которые обусловлены в основном логикой работы самой модели, а не специальными требованиями Simulink.

Состав библиотеки может быть пополнен пользовательскими блоками.

Модель рассматриваемого процесса представляет собой систему дифференциальных уравнений для определения масс крупной $M(t)$ и мелкой фракций $m(t)$.

Для преобразования уравнений модели (1.1) в блок-диаграмму используется следующая идея. Вначале формируются правые части уравнений, считая, что переменные M и m известны. Для этого используются блоки Fcn (для каждого уравнения отдельный экземпляр блока), в которых формируется по правилам языков программирования выражение для правых частей уравнений. Блок Fcn имеет только один входной порт. Если надо подать на вход блока несколько сигналов, их надо объединить в один вектор блоком Mux . Входной сигнал (скаляр или вектор) блока Fcn обозначается буквой u , если этот сигнал является вектором, то его компоненты обозначаются $u(1), u(2), \dots$, нумерация компонентов вектора в том порядке, в каком они поданы на блок Mux . В результате формирования правых частей уравнений модели, получаются значения для выражений dM/dt и dm/dt .

Проинтегрировав эти выражения с помощью специального блока $Integrator$, получаем искомые значения $M(t)$ и $m(t)$. Полученные величины теперь можно использовать для формирования правых частей уравнений (обратной связью). Одновременно их можно вывести на смотровое окно $Scope$.

1.4 Порядок выполнения лабораторной работы

1.4.1 Запустите пакет Simulink из командного окна MatLab выбором команды *File/New/Model* или нажав соответствующую кнопку на панели инструментов.

1.4.2 В открывшемся окне создайте пустое окно *untitled* – заготовку для новой модели. Вид экрана приведен на рисунке 1.1. Начинайте собирать блок-диаграмму модели рассматриваемого процесса.

Примечание - блокам в блок-диаграммах всегда давайте осмысленные имена (имена переменных, фигурирующих в описании процесса и в уравнениях). Это поможет в дальнейшем легко ориентироваться в диаграмме.

1.4.3 Сформируйте входной поток Φ_0 :

- разместите в окне диаграммы блоки Mux (для объединения переменных в один вектор), Fcn (для формирования выражения входного

потока), *Constant* (для задания значений коэффициентов выражения). Так как входной поток задан как явная функция времени, необходим еще блок *Clock*;

- настройте количество входов блока *Mux*, соедините его с блоком *Fcn*;
- откройте окно настройки блока *Fcn*, запишите выражение для входного потока, используя в качестве входного вектора выход блока *Mux*; выход блока *Fcn* пока оставьте свободным.

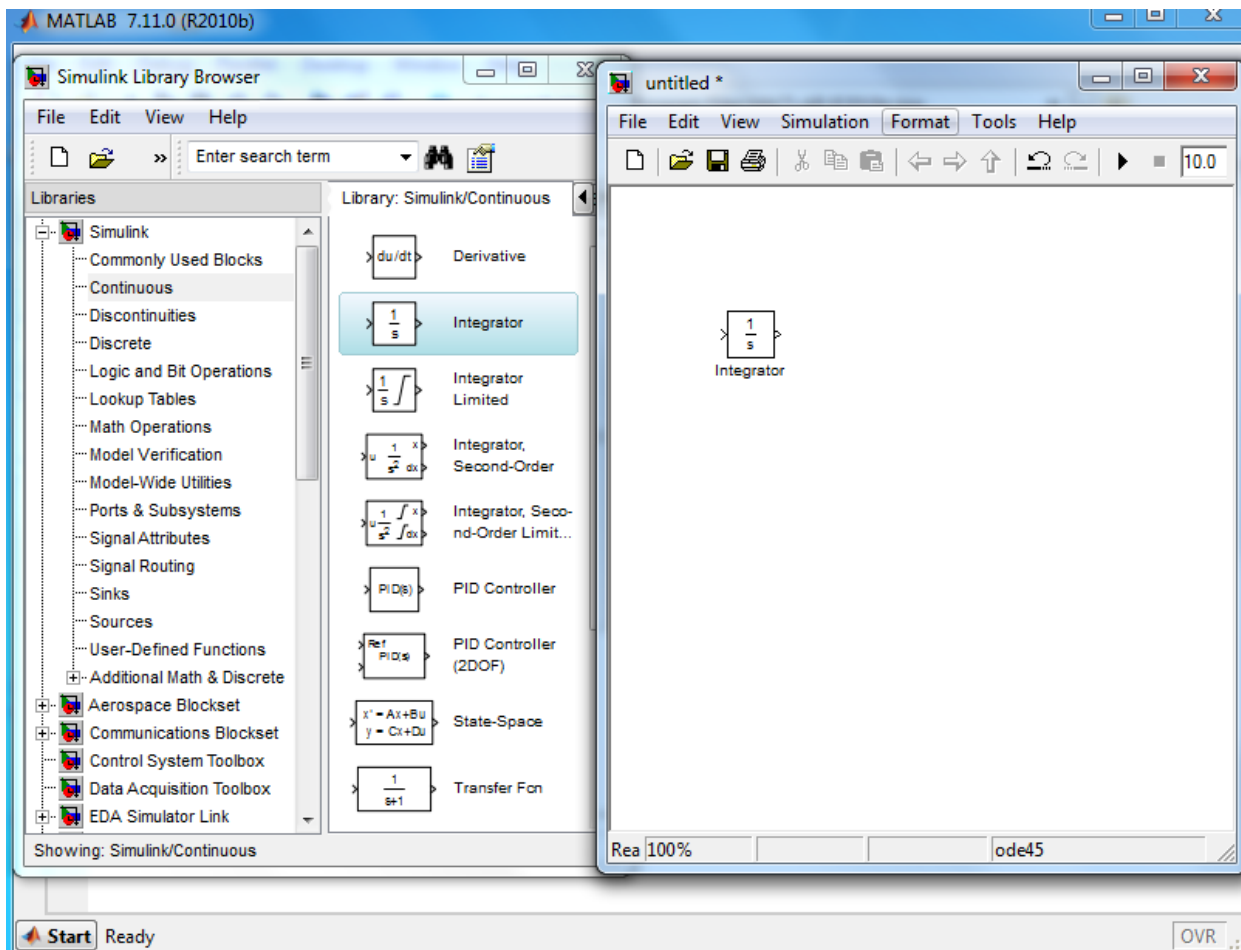


Рисунок 1.1 – Окно пакета Simulink для создания новой модели

1.4.4 Соберите блок-диаграмму для первого уравнения модели процесса:

- разместите в окне диаграммы блоки *Mux* (для объединения всех входных переменных в один вектор), *Fcn* (для формирования выражения правой части уравнения), *Integrator* (для интегрирования правой части уравнения) и *Scope* (для просмотра результата интегрирования);
- соедините созданные блоки в порядке перечисления;
- настройте количество входов блока *Mux* (на один из входов блока *Mux* необходимо подать сигнал с блока *Fcn* предыдущего пункта);
- настройте блок *Fcn*, используя в качестве входных переменных функцию входного потока Φ_0 , а также значения $M(t)$, получившиеся в результате моделирования;
- выход блока *Fcn* подайте на вход блока *Integrator*;

- выход блока *Integrator* выведите в смотровое окно *Scope*, а также обратной связью на вход блока *Mux*.

1.4.5 Аналогично п.1.4.3 соберите диаграмму для второго уравнения; здесь в качестве входных переменных используются значения $M(t)$, полученные из диаграммы первого уравнения и значения $m(t)$, получившиеся в результате моделирования; выход блока *Fcn* также подается на блок *Integrator*, выходом которого будет искомая величина

1.4.6 Выход $m(t)$ диаграммы для второго уравнения свяжите с блоком *Gain* и получите исследуемую производительность $P = \alpha \cdot m(t)$; отразите P в смотровом окне *Scope*.

1.4.7 Сохраните файл с моделью.

1.4.8 Проведите имитационные эксперименты на модели. В процессе экспериментов необходимо менять значения коэффициентов входного потока (в блоках *Constant*), запускать диаграмму на выполнение и отслеживать графики выходной переменной.

Выбор значений коэффициентов производите не произвольно, а выделите области возможных, недопустимых, особых и т.д. значений переменных и исследуйте тенденцию поведения выходной переменной в этих диапазонах.

1.4.9 Найдите наиболее производительные режимы.

1.5 Содержание отчета

Отчет по лабораторной работе должен содержать:

- цель и задание к лабораторной работе;
- уравнения модели рассматриваемого процесса;
- блок-диаграмму модели с поясняющими комментариями;
- листинги наиболее значимых результатов проведенных экспериментов;
- выводы по работе.

1.6 Варианты заданий

1.6.1 Исследовать производительность мельницы как функцию входного потока $\Phi_0(t) = a \cdot \exp(-b \cdot t)$. Найти наиболее производительные режимы.

1.6.2 Исследовать производительность мельницы как функцию входного потока $\Phi_0(t) = a \cdot \exp(b \cdot t)$. Найти наиболее производительные режимы.

1.6.3 Исследовать производительность мельницы как функцию входного потока $\Phi_0(t) = a_1 \cdot t^2 + a_2 \cdot t + a_3$. Найти наиболее производительные режимы.

1.6.4 Исследовать производительность мельницы как функцию входного потока $\Phi_0(t) = a_1 \cdot t + a_2$. Найти наиболее производительные режимы.

1.6.5 Исследовать производительность мельницы как функцию входного потока $\Phi_0(t) = \text{sqr}(a_1 \cdot t + a_2)$. Найти наиболее производительные режимы.

1.6.6 Исследовать производительность мельницы как функцию входного потока $\Phi_0(t) = \frac{a_1}{a_2 \cdot t + a_3}$. Найти наиболее производительные режимы.

1.6.7 Исследовать влияние на производительность мельницы частоты ω периодически изменяющегося входного потока $\Phi_0(t) = 2.5 \cos(\omega \cdot t)$. Требуется найти наиболее «опасные» частоты колебаний входного потока.

1.6.8 Исследовать влияние на производительность мельницы амплитуды A периодически изменяющегося входного потока $\Phi_0(t) = A \cdot \cos(0,2 \cdot t)$. Требуется найти амплитуды наиболее «опасных» колебаний входного потока.

1.6.9 Входной поток задается выражением $\Phi_0(t) = \alpha \cdot t$. Исследовать влияние на производительность мельницы частоты ω периодически изменяющегося коэффициента α : $\alpha = 2.5 \cos(\omega \cdot t)$. Выбрать оптимальный режим изменения α .

1.6.10 Входной поток задается выражением $\Phi_0(t) = \alpha \cdot t$. Исследовать влияние на производительность мельницы амплитуды A периодически изменяющегося коэффициента α : $\alpha = A \cdot \cos(0,3 \cdot t)$. Выбрать оптимальный режим изменения α .

1.7 Контрольные вопросы

1.7.1 Каковы основные типы математических моделей?

1.7.2 В чем отличие аналитических методов моделирования от экспериментальных?

1.7.3 Какие законы сохранения используются для вывода уравнений модели рассматриваемого процесса?

1.7.4 К какому классу относится модель рассматриваемого процесса?

1.7.5 Как по виду модели определить, является ли она динамической?

1.7.6 Объясните назначение пакета Simulink.

1.7.7 Объясните порядок создания блок-диаграммы для дифференциального уравнения.

1.7.8 Объясните назначение блоков Clock, Integrator, Fcn, Gain, Mux, Scope, Constant для реализуемой диаграммы.

1.7.9 Почему для реализации входного потока необходимо использование блока Clock?

1.7.10 Какой буквой обозначается входной сигнал блока Fcn?

2 Лабораторная работа №2. Исследование статических и динамических характеристик процесса в резервуаре

Цель работы: закрепление навыков моделирования и анализа динамических характеристик объектов в пакете Simulink; освоение пакета событийного моделирования Stateflow системы MatLab.

2.1 Задание к лабораторной работе

В процессе выполнения лабораторной работы студент должен:

- изучить пакет событийного моделирования Stateflow системы MatLab [1,4];
- ознакомиться с моделируемым объектом;
- построить блок-диаграммы статической и динамической моделей рассматриваемого объекта в пакете Simulink;
- в различные временные отрезки общего интервала моделирования провести расчеты по статической и динамической моделям;
- провести имитационные эксперименты по моделированию управления объектом.

2.2 Описание объекта

Объектом исследования является резервуар с независимым $G_n(t)$ притоком жидкости и зависимым стоком $G_c(t)$. Поток на выходе определяется величиной уровня жидкости над сливным отверстием H и площадью проходного сечения сливного отверстия f_c . Количество жидкости в резервуаре, находящейся над плоскостью сливного отверстия, определяется из соотношения

$$M = F \cdot H \cdot \rho,$$

где F – площадь поперечного сечения резервуара;

ρ – плотность.

При составлении уравнений математической модели объекта используются:

- уравнение материального баланса системы:

$$\frac{dM(t)}{dt} = G_i(t) - G_c(t); \quad (2.1)$$

- уравнение, отражающее закон сохранения движения:

$$G_c(t) = \mu \cdot f_c \cdot \sqrt{2 \cdot g \cdot \rho \cdot H}. \quad (2.2)$$

Примем за координаты состояния объекта уровень жидкости в резервуаре H и поток на выходе G_c .

Статическая модель объекта будет иметь вид:

$$H = \frac{1}{2g\rho\mu^2} \cdot \frac{G_n^2}{f_c^2},$$

$$G_c = G_n = const. \quad (2.3)$$

Динамика объекта описывается следующим дифференциальным уравнением:

$$\frac{dH(t)}{dt} = \frac{1}{F\rho} [G_n(t) - \mu\sqrt{2g\rho} \cdot \sqrt{H(t)} \cdot f_c(t)], \quad (2.4)$$

здесь для G_c остается справедливым соотношение (2.2).

В лабораторной работе требуется в различные временные отрезки общего интервала моделирования провести расчеты по статической (2.3) или динамической (2.4) моделям. Переходные процессы в объекте возникают в результате изменения входных переменных $G_n(t)$ и f_c .

Для переключения между этими моделями, а также для передачи новых значений параметров в модель необходимо использовать подсистему *Stateflow*.

2.3 Подсистема *Stateflow* для моделирования управления объектом

Основное назначение пакета Simulink заключается в осуществлении имитационного моделирования систем. При этом воздействие внешних событий (ситуаций) на поведение систем не учитывается или учитывается на уровне временных зависимостей источников воздействия. Между тем, многие системы работают в условиях, когда возникновение тех или иных событий может существенным образом повлиять на их работу. Моделирование систем, поведение которых зависит от внешних событий (ситуаций) называется *событийным* или *ситуационным*. В пакете Simulink системы MatLab такое моделирование реализуется с помощью специального расширения *Stateflow*. Пакет имеет свой графический интерфейс пользователя, позволяющий создавать графические диаграммы, называемые *SF-диаграммами*. Эти диаграммы служат для визуального представления работы моделируемой системы. Это достигается анализом всех стадий ее работы с указанием активных и пассивных в данное время блоков и переходов между ними по результатам анализа тех или иных условий.

Для создания *Stateflow*-диаграммы необходимо сначала создать модель в Simulink или открыть уже существующую. Затем нужно выбрать в библиотеке блок, обозначающий диаграмму *Stateflow*, и перенести его в свою модель. После двойного щелчка по нему мышью, откроется окно *Stateflow (Chart)* (см.рисунок 2.1), в котором, пользуясь меню и панелью инструментов, можно описывать состояния, связи между ними, определять переменные, условия переключения и другие, необходимые для управления элементы.

На диаграмме Stateflow различают две основные группы элементов: графические и неграфические. Все графические элементы диаграммы представлены на левой панели основного окна графического редактора карт состояний.

Важным объектом SF-диаграммы является *состояние (state)*. Каждое состояние описывает один режим работы событийно-управляемой системы. Состояние становится активным, если условие перехода, ведущего к нему, получает значение “истина” или если это состояние является начальным. Каждое состояние на диаграмме имеет своего родителя и может иметь потомков. Если состояние является единственным, то его родителем является сама SF-диаграмма (*корневая диаграмма*).

Для создания графического образа состояния необходимо, выбрав на панели соответствующую кнопку, щелкнуть кнопкой мыши в том месте, где его необходимо разместить.

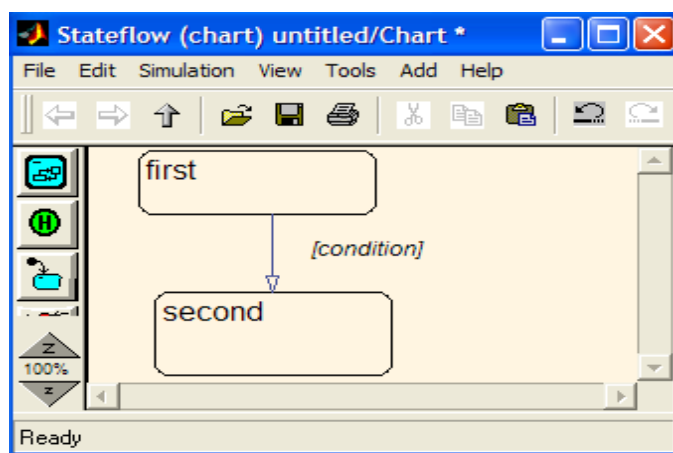


Рисунок 2.1 - Пример простой SF-диаграммы

Состояние имеет текстовые метки, которые определяют действия, выполняемые во время его активности. Имя состояния вводится первым. При описании самого состояния могут быть определены следующие действия:

- *name* – имя состояния;
- *entry* – действие, выполняемое на входе в состояние;
- *during* - действие, выполняемое, пока состояние активно;
- *exit* – действие, выполняемое на выходе из состояния;
- *on: {имя события}* - действие, выполняемое в момент появления события (имя которого указано в угловых скобках) при условии, что система будет находиться в данном состоянии.

Еще один важный графический объект SF-диаграммы - *переходы (Transition)*. Переходы отражают связь одного объекта с другим и представляются обычно стрелками, начало которых относится к состоянию-источнику, а конец – к состоянию-адресату. Переходы не имеют своей кнопки на панели инструментов. Для того чтобы нарисовать переход, достаточно, установив мышью у состояния-источника, нажать ее левую кнопку и,

удерживая ее, протянуть появившуюся стрелку до состояния-адресата. Переходы имеют метки, описывающие условия, при которых происходит переход от одного состояния к другому. Текст метки имеет следующий формат

Event [condition] {condition_action}/transition_action,

где *Event* – определяет события, которые инициирует переход. Если событие не указано, то переход инициируется при выполнении булевского выражения. Если событий, инициирующих данный переход, несколько, то они все указываются, при этом разделяются оператором *OR*;

condition – булево выражение, инициирующее переход, становясь истинным;

condition_action – действие, выполняемое после того, как стало истинным условие перехода, но до того, как весь переход определился истинным (и определилось состояние-адресат);

transition_action – определяет действия, совершаемые при переходе, если все описанные ранее условия истинны и уже определено состояние-адресат.

Важнейшее понятие пакета *Stateflow* - *события (Event)* – это неграфические объекты, управляющие диаграммой. Для визуализации события можно использовать метки переходов. Все события, имеющие отношение к диаграмме *Stateflow*, должны быть определены. Различают следующие виды событий: входные; выходные; локальные; импортируемые из источника, внешнего относительно диаграмм *Stateflow* и *Simulink*; экспортируемые адресату, внешнему относительно диаграмм *Stateflow* и *Simulink*.

Редактор *SF*-диаграммы имеет меню *Add/Event*, с помощью которого можно задать тип события и указать его свойства.

Данные – это неграфические объекты на диаграмме *Stateflow*, предназначенные для хранения числовых данных. Переменные можно использовать на любом уровне иерархии. Для создания и модификации данных следует использовать пункт меню *Add/Data*.

2.4 Порядок выполнения лабораторной работы

2.4.1 В пакете *Simulink* постройте блок-диаграммы статической и динамической модели объекта (значения параметров и исходные значения переменных объекта выбираются студентом самостоятельно).

2.4.2 Оформите их как подсистемы – *Subsystem*, то есть *S*-модели, содержащие модели более низкого уровня. Один из путей создания подсистемы - выделить все блоки, которые должны входить в подсистему и выбрать соответствующий пункт из контекстного меню.

Связь между подсистемой с основной системой осуществляется путем ввода в подсистему стандартных блоков *In* (Вход) и *Out* (Выход). Все

величины, которые формируются в основной модели, и используются в подсистеме, должны попадать в подсистему через блоки *In*, а величины, сформированные в подсистеме и затем используемые в основной системе, должны выходить из подсистемы через блоки *Out*. Напомним, что в блок-диаграммах переменным (блокам) рекомендуется давать осмысленные имена.

2.4.3 Ознакомьтесь с назначением и порядком использования блока *Switch* [1,3]; разместите в окне диаграммы модели этот блок.

2.4.4 Используйте блок *Switch* для переключения между статической и динамической моделями: на управляющий порт этого блока подается системное время (блок *Clock*), на информационные блоки подаются выходы статической и динамической моделей.

2.4.5 Проведите имитационные эксперименты на разработанной диаграмме:

- временной интервал ($t_0 - t_1$) отражает состояние равновесия;
- временной интервал ($t_1 - t_2$) отражает переходной процесс в резервуаре (значения t_0, t_1, t_2 выбираются студентом самостоятельно в соответствии с общим временем моделирования);

- выведите графики переходных процессов в блок *Scope*.

2.4.6 Смоделируйте управление объектом. Перенесите в окно модели блок *Chart*. Откройте этот блок и разработайте *SF*-диаграмму для модели:

- разместите в окне три экземпляра объекта *state*;
- в первом экземпляре этого объекта установите исходные значения переменных G_w, G_c, f_c ;

- соедините состояния объектами *Transition*; событием, инициирующим переход, является системное время: по истечении времени t_1 изменяется приток жидкости в резервуар; по истечении времени t_2 изменяется площадь проходного сечения сливного отверстия;

- сформируйте перечень входных (системное время) и выходных (изменяющиеся в процессе моделирования значения переменных объекта) данных блока.

2.4.7 Подсистему динамической модели объекта скопируйте в другое место этого же окна блок-диаграммы. Подключите блок *Chart* к динамической модели резервуара.

2.4.8 Переходной процесс отразите на блоке *Scope*.

2.4.9 Проведите имитационные эксперименты по управлению объектом на разработанной диаграмме.

2.5 Содержание отчета

Отчет по лабораторной работе должен содержать:

- цель и задание к лабораторной работе;
- уравнения статической и динамической модели резервуара;
- блок-диаграммы модели с поясняющими комментариями;
- результаты экспериментов, проведенных на этих моделях;

- *SF*-диаграмму, описывающую функционирование системы в различных временных интервалах;
- графики проведенных имитационных экспериментов;
- выводы по работе.

2.6 Варианты заданий

Значения параметров и исходные значения переменных объекта выбираются студентом самостоятельно.

2.7 Контрольные вопросы

- 2.7.1 К какому классу относится разработанная модель?
- 2.7.2 Какие переменные моделируемого объекта являются входами модели?
- 2.7.3 Какие переменные моделируемого объекта являются выходами модели?
- 2.7.4 Дайте определение статической и динамической модели.
- 2.7.5 Из-за чего может возникнуть переходной процесс в рассматриваемом объекте?
- 2.7.6 Для чего предназначен пакет Stateflow?
- 2.7.7 Что такое *SF*-диаграмма?
- 2.7.8 Объясните понятие «состояние» в пакете Stateflow.
- 2.7.9 Что такое переходы?
- 2.7.10 Как создать или изменить данные в пакете Stateflow?

3 Лабораторная работа №3. Прямые методы идентификации линейных объектов первого порядка

Цель работы: освоение методов идентификации линейных динамических объектов с помощью переходной функции⁷

3.1 Задание к лабораторной работе

- В процессе выполнения лабораторной работы студент должен:
- изучить методы графической идентификации линейного объекта первого порядка [1,2];
 - собрать блок-диаграмму для идентификации объекта с помощью переходной функции;
 - определить параметры передаточной функции объекта графическим методом идентификации;
 - оценить точность идентификации.

3.2 Метод идентификации с помощью переходной функции

Прямые методы идентификации требуют специальных входных сигналов: ступенчатых сигналов для идентификации по переходной функции, импульсных входных сигналов для идентификации по импульсной переходной функции и синусоидальных входных сигналов с различными частотами для определения частотной характеристики.

Во многих случаях для определения передаточной функции системы можно использовать запись ее переходной функции. Такой способ применим к большинству типов линейных систем (1 и 2 порядков и к апериодическим системам высшего порядка). Наиболее корректно метод идентификации с использованием переходных функций применяется к процессам первого порядка.

Рассматривается объект первого порядка. Искомое уравнение для объекта первого порядка имеет вид

$$T \frac{dy(t)}{dt} + y(t) = kx(t) \quad \text{или} \quad W(p) = \frac{k}{Tp + 1}. \quad (3.1)$$

На объекте проводятся эксперименты: в момент времени $t_0=0$ входная величина x скачком изменяется на величину a и записывается график переходного процесса. Требуется, используя этот график, определить параметры T и k передаточной функции.

Общее выражение для решения дифференциального уравнения (3.1) имеет вид:

$$y(t) = k \cdot a \cdot \left(1 - e^{-\frac{t}{T}} \right). \quad (3.2)$$

Функция (3.2) описывает график переходного процесса. Для простоты примем, что $a=1$ (единичный скачок).

Определение параметров передаточной функции можно выполнить следующими способами:

а) из выражения (3.2) при $t \rightarrow \infty$, имеем $y(t) = k \cdot a$. С другой стороны при $t \rightarrow \infty$ имеем установившуюся величину выходного сигнала, то есть через ординату установившегося состояния объекта (на графике) можно определить параметр k .

При $t = T$ функция $y(t) = k \cdot (1 - e^{-1}) = k \cdot (1 - 0.37) = 0.63 \cdot k$. Это означает, что постоянная времени T системы первого порядка равна отрезку времени, за которое переходная функция достигает 63% своей установившейся величины;

б) коэффициент k определяется, как описано в п.а).

Для определения параметра T продифференцируем выражение (3.2)

$$y' = \frac{dy(t)}{dt} = k \cdot a \cdot \frac{1}{T} \cdot e^{-\frac{t}{T}}.$$

В начальной точке

$$\frac{dy(0)}{dt} = k \cdot a \cdot \frac{1}{T},$$

то есть (из геометрического смысла производной)

$$\operatorname{tg} \alpha = k \cdot a \cdot \frac{1}{T},$$

здесь α – угол наклона касательной к графику функции при $t = 0$.

С другой стороны, касательная, перпендикуляр, опущенный из точки пересечения касательной с установившимся состоянием на ось абсцисс, и ось абсцисс образуют прямоугольный треугольник, в котором $\operatorname{tg} \alpha$ – отношение противолежащего катета к прилежащему, то есть прилежащий катет в этом треугольнике и будет величиной T .

Итак, величина $T = \frac{k \cdot a}{\operatorname{tg} \alpha}$ – отрезок оси абсцисс, равный расстоянию от начала координат до точки, в которой касательная пересекается с асимптотой.

3.3 Порядок выполнения лабораторной работы

3.3.1 В окно редактора моделей Simulink загрузите из папки *Work/Objects3* свой «объект» (по варианту).

3.3.2 Установите в качестве входного сигнала ступенчатый сигнал, выходной сигнал отразите в смотровом окне *Scope*.

3.3.3 Используя полученный в *Scope* график переходного процесса, определите двумя способами (а и б) параметры передаточной функции объекта первого порядка. В результате получатся две модели в виде передаточных функций. Запишите выражения для этих моделей.

3.3.4 По восстановленным моделям, подавая на их входы тот же ступенчатый сигнал, получите выходные сигналы (общий вид получившейся диаграммы приведен на рисунке 3.1).

3.3.5 Сравните графики переходных функций «объекта» и моделей. Какая из моделей дает более точный результат? С чем это связано?

3.4 Содержание отчета

Отчет по лабораторной работе должен содержать:

- цель и задание к лабораторной работе;

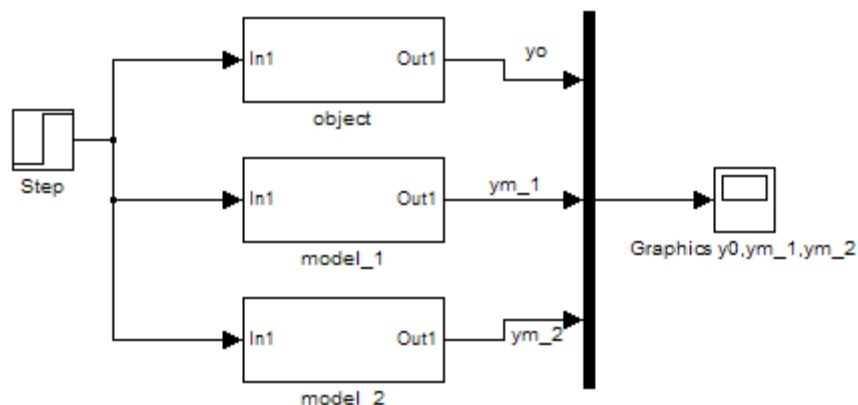


Рисунок 3.1 - Блок-диаграмма сравнения выходов объекта и моделей

- смотровое окно с переходным процессом, на котором отмечены касательная к графику переходной функции, 63% -ая отметка и установившееся состояние;
- значения параметров передаточных функций, определенные обоими способами;
- блок-диаграммы объекта и обеих моделей (аналогично рисунку 3.1);
- графики переходных процессов объекта и обеих моделей в одном смотровом окне;
- сравнительный анализ графиков; выводы.

3.5 Варианты заданий

Выбираются из папки *Work/Objects3* пакета *Simulink*.

3.6 Контрольные вопросы

- 3.6.1 Что такое идентификация?
- 3.6.2 Какие методы называются прямыми методами идентификации?
- 3.6.3 Какая функция используется для графической идентификации линейного объекта?
- 3.6.4 Какой сигнал на входе объекта использовался для проведения процедуры идентификации графическим методом?
- 3.6.5 Как определяется параметр k передаточной функции?
- 3.6.6 Объясните способы определения параметра T передаточной функции.
- 3.6.7 Как провели касательную к графику переходного процесса в смотровом окне *Scope*?
- 3.6.8 Какой блок используется для того, чтобы отразить все графики в одном смотровом окне?
- 3.6.9 Какой из способов идентификации дает более точный результат? Почему?

3.6.10 Для каких объектов можно применить рассмотренный метод идентификации?

4 Лабораторная работа №4. Идентификация линейного объекта с помощью частотной характеристики

Цель работы: освоение методов идентификации с помощью гармонических сигналов.

4.1 Задание на лабораторную работу

В процессе выполнения лабораторной работы студент должен:

- изучить частотные методы идентификации линейного объекта [1,2,3];
- собрать блок-диаграмму для идентификации объекта;
- провести эксперименты; определить значения амплитуд и фазовых сдвигов выходного сигнала;
- ознакомиться с программой обработки результатов экспериментов;
- используя эту программу, определить АЧХ и ФЧХ объекта;
- определить параметры передаточной функции объекта;
- оценить точность идентификации.

4.2 Определение коэффициентов передаточной функции с помощью частотных характеристик

Частотные характеристики широко используются для анализа и синтеза систем регулирования. Но они также дают возможность определения уравнений объекта.

Как известно, если на вход объекта подается синусоидальное воздействие

$$x(t) = A_0 \cdot \sin(\omega \cdot t),$$

то установившееся измеренное значение выходного сигнала

$$y(t) = A_1 \cdot \sin[\omega \cdot t + \varphi(t)] + n(t),$$

где $n(t)$ - ошибка измерения;

$$A_1/A_0 = |W(j\omega)|;$$

$$\varphi = \text{Arg}[W(j\omega)].$$

Частотная характеристика $W(j\omega)$ определяется путем подачи синусоидальных входных сигналов $A_0 \sin(\omega t)$ на различных частотах ω и записи соответствующих выходных сигналов $A_1 \sin[\omega t + \varphi]$. С целью получения необходимой частотной характеристики, величины A_1/A_0 и φ определяются для каждой рассматриваемой частоты ω . То есть по записям входного и

выходного сигналов определяют отношение амплитуд на частоте ω_i и получают $W(j\omega)$. Фазовый сдвиг $\varphi(\omega_i)$ получают из сравнения положения максимумов кривых $x(t)$ и $y(t)$.

Полученные частотные характеристики дают возможность определить уравнение объекта. Суть метода заключается в следующем. Для каждой частоты определяются действительные и мнимые частотные характеристики

$$P(\omega_i) = A(\omega_i) \cdot \cos[\varphi(\omega_i)],$$

$$Q(\omega_i) = A(\omega_i) \cdot \sin[\varphi(\omega_i)].$$

Эти данные используются для определения коэффициентов $(a_1, a_2, \dots, a_{n-1}, b_0, b_1, \dots, b_m)$ передаточной функции:

$$W(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_0}{p^n + a_{n-1} p^{n-1} + a_{n-2} p^{n-2} \dots + a_1}.$$

Для вычисления коэффициентов передаточной функции, заменим p на $j\omega$, запишем функцию в виде суммы действительной и мнимой частей $P(\omega_i) + j \cdot Q(\omega)$ и получим выражение, справедливое для всех значений ω . Из этого комплексного выражения, приравнявая коэффициенты при действительной и мнимой частях, получим систему из двух уравнений. Подставляя различные значения ω_i (столько значений, чтобы система получилась квадратной) в эти уравнения, получим систему для определения неизвестных коэффициентов.

Для уточнения полученных значений коэффициентов, повторяем вычисления для других частот и возьмем среднее из двух вычислений.

Если истинный порядок объекта выше предполагаемого, то в первоначальных и в повторных вычислениях значения соответствующих коэффициентов будут сильно отличаться.

4.3 Программа обработки результатов измерений

Для определения коэффициентов передаточной функции необходимо решать системы уравнений большого порядка. В лабораторной работе для этого используется программа Lab4, разработанная на языке Delphi. Основное окно программы приведено на рисунке 4.1.

Работа в этом окне проводится следующим образом:

- введите значения частот ω_i и определенные для каждой частоты значения амплитуд и фазовых сдвигов;
- нажмите на кнопку «Рассчитать $P(j\omega)$ и $Q(j\omega)$ », в таблице появятся еще две строки с рассчитанными значениями действительных и мнимых характеристик;
- переключателем установите порядок объекта;
- из раскрывающихся списков выберите значения частот;

- нажмите на кнопку «Рассчитать систему»;
- в правой части нижней окна программы появятся значения рассчитанных коэффициентов передаточной функции объекта.

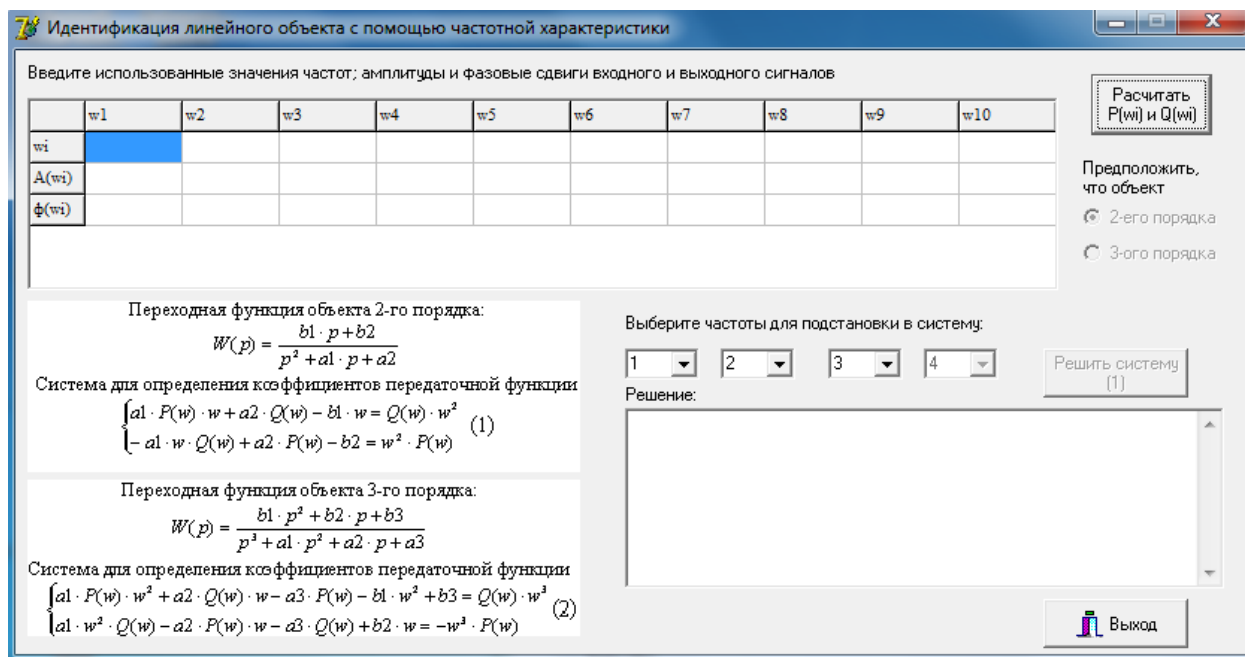


Рисунок 4.1 – Основное окно программы обработки результатов экспериментов

4.4 Задание к лабораторной работе

4.4.1 Из папки Work/Objects4 загрузите свой «объект» в окно пакета Simulink (по варианту).

4.4.2 На вход «объекта» подайте синусоидальный сигнал с известной амплитудой, на выходе установите смотровое окно Scope.

4.4.3 Заполните таблицу 1. Для этого, меняя значения частот входного сигнала, записывайте соответствующие выходные сигналы. Для каждой частоты определите амплитудные и фазовые характеристики объекта.

4.4.4 Загрузите программу Lab4.

4.4.5 Данные из таблицы 1 перенесите в окно программы. Рассчитайте P(jω) и Q(jω).

4.4.6 Выберите порядок объекта – 2, установите значения частот, рассчитайте систему (1).

4.4.7 Для уточнения полученных значений коэффициентов, повторите вычисления для других частот (порядок объекта – 2) и возьмите среднее из двух вычислений.

4.4.9 Выберите порядок объекта – 3, установите значения частот, рассчитайте систему (2).

4.4.7 Для уточнения полученных значений коэффициентов, повторите вычисления для других частот (порядок объекта – 3) и возьмите среднее из двух вычислений.

4.4.10 Выберите наиболее подходящий порядок объекта и запишите выражение для восстановленной передаточной функции

4.4.11 В пакете Simulink соберите модель объекта с полученной передаточной функцией. Подайте на вход модели те же самые сигналы, что и на «объекте» и сравните отклики объекта и модели (в одном смотровом окне).

Т а б л и ц а 1

ω_i	ω_1	ω_2	...	ω_{10}
$A(\omega_i)$				
$\varphi(\omega_i)$				

4.5 Содержание отчета

Отчет по лабораторной работе должен содержать:

- цель и задание к лабораторной работе;
- блок-диаграмму для проведения экспериментов;
- графики, полученные экспериментально на объекте;
- заполненную таблицу 1;
- систему уравнений для определения коэффициентов передаточной функции;
- значения коэффициентов модели для различных порядков объекта;
- блок-диаграмму для сравнения выходов объекта и модели;
- сравнительные результаты имитационных экспериментов на модели;
- выводы по работе.

4.6 Варианты заданий

Варианты выбираются из папки Work/Objects4 пакета Simulink.

4.7 Контрольные вопросы

4.7.1 В чем заключаются прямые методы идентификации?

4.7.2 Какие сигналы, кроме гармонических, могут быть использованы в прямых методах?

4.7.3 Какие объекты могут быть идентифицированы частотным методом?

4.7.4 Объясните суть частотного метода идентификации.

4.7.5 Каким образом выбирается порядок модели?

4.7.6 Как определяется фазовый сдвиг выходного сигнала?

4.7.7 Как была определена амплитуда выходного сигнала?

4.7.8 Как получается система уравнений для определения коэффициентов передаточной функции?

4.7.9 Как уточняются значения коэффициентов модели?

4.7.10 Насколько точно определена модель, в чем причина разницы?

5 Лабораторная работа №5. Параметрическая идентификация линейных динамических объектов

Цель работы: освоение методов параметрической идентификации объектов, аппроксимации функций и решения задачи минимизации функционала невязки.

5.1 Задание к лабораторной работе

В процессе выполнения лабораторной работы студент должен:

- ознакомиться с методами параметрической идентификации линейных динамических объектов [1,2];
- изучить процедуру минимизации среднеквадратичного критерия [1,3];
- провести эксперименты на «объекте»: на вход объекта подать заданный входной сигнал, зарегистрировать числовые значения выходного сигнала;
- используя пакет подгонки кривых Curve Fitting Toolbox, выполнить аппроксимацию выходного сигнала выбранными уравнениями;
- определить структуру линейной динамической модели объекта;
- записать вид функционала невязки для модели выбранной структуры;
- используя результаты экспериментов, решить задачу минимизации этого функционала; определить параметры модели;
- оценить точность построенной модели.

5.2 Параметрическая идентификация динамических объектов

В параметрическом случае модель определяется набором параметров, которые необходимо оценить в процессе идентификации. Задачу параметрической идентификации можно сформулировать так: на основе экспериментальных данных указать значения вектора параметров, при которых модель наилучшим образом (или достаточно точно) в определенном смысле аппроксимирует оператор объекта.

Построение модели начинается с выбора структуры модели, определяющей взаимосвязь наблюдаемых данных через совокупность параметров. После этого формируются входные воздействия и подаются на объект, и измеряются отклики на эти воздействия (выходные сигналы). Затем входные и выходные сигналы и выбранная структура используются для оценки значений параметров в соответствии с принятым критерием качества. В качестве критерия часто используется среднеквадратичный критерий, в соответствии с которым ищутся такие оценки параметров, которые

обеспечивают минимальный средний квадрат разности выходных сигналов модели и объекта при одном и том же входном воздействии.

В одномерном случае связь между входной величиной $x = x(t)$ и выходом $y = y(t)$ представляет собой обыкновенное дифференциальное уравнение:

$$\sum_{i=0}^p a_i \cdot y^{(i)} = \sum_{j=0}^l b_j \cdot x^{(j)}, \quad (5.1)$$

где

$$y^{(i)} = \frac{d^i y}{dt^i}, \quad x^{(j)} = \frac{d^j x}{dt^j}, \quad a_p = 1, \quad l \leq p$$

вместе с начальными условиями для $\frac{d^i y(0)}{dt^i}$, $i = 0, 1, \dots, n-1$.

Модель определяется $(p+l+1)$ параметрами $c = (a_0, \dots, a_{p-1}, b_0, \dots, b_l)$.

Исходной информацией для построения процедуры идентификации является вид идентифицируемой модели (7.11) и наблюдения

$$\{X_t, Y_t\} \text{ в промежутке } [0, T]. \quad (5.2)$$

Функция невязки строится в виде среднего квадрата разности правой и левой части уравнения (5.1) при подстановке туда функций X_t , Y_t – наблюдений объекта:

$$Q(c) = \int_0^T \left[\sum_{i=0}^p a_i \cdot Y^{(i)} - \sum_{j=0}^l b_j \cdot X^{(j)} \right]^2 dt. \quad (5.3)$$

Задача минимизации функционала (5.3) решается обычными методами поиска минимума гладкой функции. Приравнивая нулю производные функции (5.3) по всем неизвестным параметрам получим систему линейных уравнений, решение которой является решением задачи минимизации. Полученную систему можно решить стандартными вычислительными методами. Однако для вычисления коэффициентов системы необходимо знать производные входного x_t и выходного y_t сигналов объекта. Если эти сигналы заданы в аналитической форме нет никаких проблем в определении производных любого порядка. В лабораторной работе входные сигналы заданы в виде некоторых функций от времени. Значения выходных сигналов в ходе эксперимента получаются в виде массива данных. Поэтому необходимо использовать разложение в ряд по системе заданных в аналитическом виде функций. Аппроксимацию функции будем выполнять с помощью пакета подгонки кривых Curve Fitting Toolbox.

5.3 Пакет подгонки кривых Curve Fitting Toolbox

Работу с пакетом надо начинать с загрузки данных. Данные могут быть сформированы в командном окне системы в соответствующих векторах.

Основное окно пакета Curve Fitting Tool открывается командой *cftool* (в командном окне MatLab). Вначале оно имеет пустое окно графики и 4 кнопки над ним: *Data...* - загрузка данных; *Fitting...* - выполнение подгонки; *Plotting...* - графическая визуализация подгонки; *Analysis...* - анализ результатов подгонки.

Импорт данных. Обычно работа с окном *Curve Fitting Tool* начинается с уточнения типа загружаемых данных, для чего необходимо активизировать кнопку *Data...* Это ведет к появлению окна импорта данных *Data*. В этом окне с помощью соответствующих вкладок можно установить тип множеств данных, просмотреть их, скорректировать и сгладить. Нажатие кнопки *Apply* и закрытие окна приводит к загрузке скорректированных данных в основное окно Curve Fitting Tool. В результате точки данных появляются в поле графики этого окна.

Выполнение подгонки заданного типа. Выбор кнопок *Fitting/New Fit* позволяет начать процедуру подгонки.

Открывающийся список *Type of fit* позволяет установить любой из следующих типов подгонки (регрессии):

- 1) Custom equations — функция регрессии, задаваемая пользователем.
- 2) Exponential — экспоненциальная регрессия.
- 3) Fourier — приближение рядом Фурье.
- 4) Gaussian — приближение кривой гауссиана.
- 5) Interpolant — интерполяция с выбором метода.
- 6) Polynomial — полиномиальная регрессия.
- 7) Power — степенная регрессия.
- 8) Rational — регрессия рациональной функцией с выбором ее числителя и знаменателя.
- 9) Smoothing Spline — регрессия сглаживающими сплайнами;
- 10) Sum of Sin Func — регрессия суммой синусоидальных функций;
- 11) Weibull — регрессия кривой Вейбулла.

Также есть возможность задания пользователем функции регрессии любого вида, разумеется, на основе встроенных функций *MatLab*.

Если нажать кнопку *Fit Options...* окна подгонки *Fitting*, можно задать ряда опций подгонки. В окне опций подгонки возможна установка различных алгоритмов подгонки, пределов изменения производных, максимального значения функции и начальных значений параметров регрессии. Все эти величины существенно влияют на сходимость и скорость подгонки.

Графическая визуализация регрессии. Кнопка *Plotting...* окна Curve Fitting Tool открывает окно установок графики.

Анализ результатов регрессии. Важным этапом приближения является анализ результатов. Он проводится при активизации кнопки *Analysis...* окна

Curve Fitting Tool. При этом открывается окно анализа *Analysis*. В левой части окна имеется ряд опций анализа: выполнения вычислений, вывода графиков производных и интеграла для функции регрессии и др. Нужные опции задаются установкой знака птички у их названия. После этого нажатие кнопки *Apply* формирует таблицу результатов анализа сверху окна и выводит окно с графическим представлением результатов анализа.

Основным критерием выбора кривой регрессии является минимум погрешности *Residuals* в заданном диапазоне изменения аргумента, отсутствие резких выбросов кривой погрешности на краях этого диапазона или в отдельных его областях и т. д. Нередко важным является соответствие кривой регрессии возможному закону расположения точек исходных данных. Рекомендуется опробовать несколько кривых регрессии для заданного набора точек.

5.4 Порядок выполнения лабораторной работы

5.4.1 Реализуйте «измерительный стенд» для проведения процедуры идентификации:

- загрузите в Simulink блок-диаграмму «объекта» из папки *Work/Objects5* (по варианту);

- определите входной сигнал (по варианту) и подайте его на вход объекта; получите выходные сигналы;

- значения выходного сигнала передайте в рабочую область программы; для этого выведите сигнал в блок *To Workspace*, в котором задайте имя соответствующей переменной.

5.4.2 Загрузите пакет *Curve Fitting Tool*, перенесите данные о выходном сигнале в окно пакета; по виду расположения точек графика сигнала, выберите структуру аппроксимирующего полинома.

Выполните аппроксимацию полученных значений выходного сигнала различными функциями: полиномами различных порядков или некоторыми выбранными уравнениями.

Для задания своего уравнения регрессии в списке видов регрессии надо выбрать позицию *Custom equations*. Это приводит к появлению окна задания регрессии *Create Custom Equations*. Это окно имеет две вкладки. На первой вкладке *Linear Equations* можно задать параметры уравнения, линейного относительно коэффициентов регрессии. Заметим, что сама зависимость при этом может быть нелинейной. На другой вкладке *General Equations* можно установить произвольное нелинейное уравнение регрессии, т. е. осуществить нелинейную регрессию.

5.4.3 Завершив подгонку, выберите функцию, которая наилучшим образом аппроксимирует выходной сигнал. Запишите выражение для этой функции (с рассчитанными коэффициентами).

5.4.4 Определите структуру линейной модели (запишите уравнение).

5.4.5 Запишите вид функционала невязки: вместо функции входа и ее производных подставьте выражения для входного сигнала и его производных; вместо функции выхода и ее производных подставьте полученное в результате аппроксимации выражение и его соответствующие производные.

5.4.6 Приравнявая нулю производные функционала невязки по всем неизвестным параметрам модели запишите систему уравнений для определения коэффициентов этой модели; преобразуйте полученную систему к стандартному виду системы линейных алгебраических уравнений.

5.4.7 Полученную систему линейных алгебраических уравнений решите с помощью процедуры пакета Matlab (в командном окне). Для системы стандартного вида

$$Ax = b,$$

решение находится следующим образом

$$x = \text{inv}(A) \cdot b.$$

Здесь A - матрица коэффициентов системы линейных алгебраических уравнений;

b – вектор правых частей системы;

x – вектор неизвестных (это неизвестные параметры модели).

Предварительно массивы A (матрица) и b (вектор-столбец) надо ввести в командное окно пакета, а также проверить условие существования решения $\det(A) \neq 0$.

5.4.8 Используя полученные значения параметров, напишите искомое уравнение динамической модели.

5.4.9 К диаграмме, реализованной в п.5.4.1, добавьте блок с моделью; на его вход подайте тот же входной сигнал; выходы «объекта» и модели отразите в одном смотровом окне. Сравните результаты.

5.4.10 Сделайте выводы по работе.

5.5 Содержание отчета

Отчет по лабораторной работе должен содержать:

- цель и задание к лабораторной работе;
- блок-диаграмму «измерительного стенда»;
- выведенные в рабочее пространство системы Matlab результаты экспериментов на объекте;
- последовательность выполнения процедуры аппроксимации выходного сигнала; выбранный аппроксимирующий полином;
- вид функционала невязки для определения параметров модели;
- условие минимизации этого функционала;
- система линейных алгебраических уравнений для определения параметров модели; решение этой системы уравнений;
- полученное уравнение модели;

- сравнительные графики выхода объекта и модели;
- выводы по работе.

5.6 Варианты заданий

5.6.1 Варианты «объектов» выбираются из папки Work/Objects5 пакета Simulink.

5.6.2 Структурные параметры модели и входные сигналы выбираются из таблицы.

№ варианта	Выражение для входного сигнала	Структурные параметры модели
1	$x(t) = t+1$	$p=1, l=0$
2	$x(t) = 3t+2$	$p=2, l=1$
3	$x(t) = t^2+t+1$	$p=3, l=1$
4	$x(t) = t-7$	$p=2, l=1$
5	$x(t) = 2t^2-6$	$p=3, l=2$
6	$x(t) = t$	$p=2, l=1$
7	$x(t) = 2t+4$	$p=1, l=0$
8	$x(t) = t^2-t+5$	$p=3, l=1$
9	$x(t) = 4t+3$	$p=2, l=1$
10	$x(t) = t^2+2$	$p=3, l=2$

5.7 Контрольные вопросы

- 5.7.1 Что такое параметрическая идентификация?
- 5.7.2 Что такое оценка параметра?
- 5.7.3 Какой критерий используется для оценивания параметров модели?
- 5.7.4 Как можно получить в рабочем окне пакета Matlab значения выходного сигнала, сформированного в Simulink?
- 5.7.5 Дайте определение аппроксимации функций.
- 5.7.6 Какой критерий используется для аппроксимации функций?
- 5.7.7 Какой пакет использовался для аппроксимации функций?
- 5.7.8 Что такое функционал невязки?
- 5.7.9 Как решается задача минимизации функционала невязки?
- 5.7.10 С чем связано различие между выходами объекта и модели?

6 Лабораторная работа №6. Идентификация линейных объектов средствами пакета System Identification Toolbox

Цель работы: освоить методы идентификации объектов с помощью графического инструментария идентификации системы MatLab.

6.1 Задание к лабораторной работе

В процессе выполнения лабораторной работы студент должен:

- ознакомиться с методами параметрической идентификации линейных динамических объектов [1,2,3];
- провести эксперименты на объекте;
- выполнить предварительную обработку данных эксперимента;
- провести идентификацию корреляционной модели;
- оценить частотные характеристики объекта;
- для параметрической модели выбрать структуру модели объекта; используя результаты экспериментов и выбранную структуру, определить параметры модели;
- оценить точность построенных моделей.

6.2 Графический инструментарий System Identification Toolbox

Пакет System Identification Toolbox системы Matlab содержит средства для создания математических моделей линейных динамических систем на основе наблюдаемых входных и выходных данных. В лабораторной работе для идентификации линейной системы, используется графический интерактивный интерфейс *ident*, входящий в состав инструментария идентификации.

Чтобы загрузить графическую программу идентификации, в командном окне необходимо набрать *ident*. В результате загрузки на экране появится панель (главное окно) программы.

Идентификация объекта проводится следующим образом:

- для загрузки данных эксперимента выберите раскрывающееся меню *Data-Import...*; в появившемся диалоговом окне введите имена входных и выходных переменных модели; также можно ввести интервал времени, текст комментария. Затем нажмите кнопку *Import*, и данные отобразятся в иконке графической оболочки. Закройте окно импорта данных. Обратите внимание, что данные также появятся в полях *Working Data* и *Validation Data*;
- флажок *Time Plot* позволяет отразить окно с графиками входов и выходов. Это окно имеет меню для управления. Вы можете выделить мышью прямоугольную область на графике для увеличения фрагмента графика, щелчком левой кнопки мыши можно увеличить/уменьшить масштаб, двойной щелчок возвращает исходный масштаб. Чтобы очистить окно графиков, щелкните на иконке с данными. В дальнейшем, чтобы работать с новыми данными, надо переместить их мышью в область *Working Data*.

Двойной щелчок на иконке данных вызывает еще одно окно, где можно изменить название массива данных;

- команда *Select Range* раскрывающегося меню *Preprocess* позволяет выбрать подмножество из массива данных. Это можно сделать через редактируемые текстовые поля появившегося окна или, выделяя мышью прямоугольную область на графиках сигналов. Нажатие на *Insert* размещает выбранные данные в новой иконке. Для верификации модели обычно используются данные, которые не использовались для идентификации параметров. Можно разделить исходные массивы значений сигналов на две половины, одна из которых используется для идентификации, а вторая - для подтверждения модели. В реальной работе, особенно если приходится иметь дело со сложными объектами, для верификации используются несколько наборов различных данных.

Для идентификации выбранные данные переносятся в область *Working Data*, для верификации модели соответствующие данные переносятся в область *Validation Data* (перетаскиванием с помощью мыши);

- для оценки модели выбранного типа в раскрывающемся меню *Estimate* выбирается соответствующая строка; в появляющихся (в соответствии с выбранным типом модели) диалоговых окнах необходимо установить требуемые параметры по нахождению оценки модели;

- при выполнении операций предыдущего пункта в основном окне интерфейса (в его правой части – *Model Views*) появятся значки с именами моделей.

Верификация модели: для того, чтобы убедиться, насколько точно построенная модель описывает объект, необходимо проанализировать её поведение. Для анализа и подтверждения модели программа *ident* предоставляет пользователю широкий набор средств. Эти процедуры выполняются с помощью установки флажков в окнах, расположенных в правой нижней части панели.

Для сравнения наблюдаемых и моделируемых выходных сигналов установите флажок "*Model output*". В открывшемся окне представлены графики измеренного (линия чёрного цвета) и рассчитанного по модели выходных сигналов. Для рассмотрения отдельных фрагментов необходимо, удерживая в нажатом положении левую клавишу мыши, выделить на графике прямоугольную область, которая будет отображена на всё графическое окно.

6.3 Порядок выполнения лабораторной работы

6.3.1 Реализуйте «измерительный стенд» для проведения процедуры идентификации:

- загрузите в Simulink блок-диаграмму «объекта» из папки *Work/Objects6* (по варианту);

- подайте на вход объекта ступенчатый сигнал $x=a$ (значение a выбирается произвольно); получите выходные сигналы;

- результаты эксперимента, то есть измеренные значения входных и выходных сигналов, передайте в рабочую область программы; для этого выведите сигналы в блоки *To Workspace*, в которых задайте имена соответствующих переменных.

6.3.2 Для начала процедуры идентификации наберите в командном окне команду *ident*. Появится диалоговое окно интерфейса.

6.3.3 Загрузите данные эксперимента из рабочей области:

- выберите в раскрывающемся списке команду *Data- Import*;
- в появившемся окне введите имена входного и выходного сигналов, начальное время - 0, интервал дискретизации – 0.08; нажмите *Import*, а затем закройте окно импорта данных;

- проведите исследование исходных данных, установив флажок *Time Plot*.

6.3.4 Проведите предварительную обработку сигналов исследуемого объекта, исключив из них постоянную составляющую:

- *Preprocess-Remove Means*; в поле *Data Views* появится информация о новых данных;

- опять активизируйте окно *Time Plot* и посмотрите изображения сигналов без постоянной составляющей;

- выберите *Preprocess-Select Range* и в появившемся окне укажите диапазон данных; новые данные отразятся в окне интерфейса;

- эти новые данные перетащите в область *Working Data*.

6.3.5 Повторите операцию задания диапазона с целью формирования данных для проверки модели (например, из оставшейся части начального диапазона), эти данные тоже отразятся в окне интерфейса; перетащите их мышью в область *Validation Data*.

6.3.6 Оцените корреляционную модель:

- в раскрывающемся списке *Estimate* выберите *Correlation Model*;

- в появившемся окне можно задать данные фильтра (ничего не меняйте), нажмите кнопку *Estimate*;

- в окне интерфейса *Model Views* появится значок *imp*, то есть выбранная модель построена;

- установите флажок *Transient Response*, и в отдельном окне появится график переходной функции, найденный корреляционным методом;

- используя меню *Optoins*, вместо графика переходной функции можно вывести график импульсной характеристики.

6.3.7 Оцените частотные характеристики объекта:

- выберите *Estimate-Spectral Model*;

- повторите операции предыдущего пункта;

- в окне интерфейса *Model Views* появится значок *spad*, представляющий модель в виде частотных характеристик, найденных спектральным методом;

- отразите результат, установив флажок *Frequency Response*.

6.3.8 Оцените параметрическую модель:

- выберите *Estimate-Parametric model*. Появляется окно задания структуры модели (в поле *Structure*); по умолчанию предлагается модель типа ARX (можно выбрать другую модель из раскрывающегося списка) с параметрами $na = 4$, $nb = 4$, $nk = 1$. Параметры na и nb определяют полрядки правых и левых частей уравнения регрессии. Параметр nk означает наличие в объекте запаздывания с величиной $nk \cdot T = 1 \cdot 0,08 \text{ c} = 0,08 \text{ c}$. Для изменения параметров в окне редактирования порядка модели (кнопка *Order Editor*), устанавливаются значения констант;

- оставьте предлагаемые параметры модели и нажмите на кнопку *Estimate*. В результате появится значок с названием модели *arx441*;

- повторите действия по оцениванию параметрической модели для ARX-модели с параметрами $na = 2$, $nb = 2$, $nk = 3$. $nk=3$ означает запаздывание $nk \cdot T = 3 \cdot 0,08 \text{ c} = 0,24 \text{ c}$. Появится еще один значок модели *arx223*;

6.3.9 Для сравнения полученных моделей установите флажок *Transient Response*; здесь отразятся графики переходных процессов, соответствующие трем построенным моделям.

6.3.10 Сравнение параметрических моделей можно выполнить, установив флажок *Model output*.

6.3.11 Чтобы оценки параметров модели отразить в командном окне Matlab, надо дважды щёлкнуть правой клавишей мыши по пиктограмме модели и в открывшемся окне нажать кнопку «*Present*».

6.4 Содержание отчета

Отчет по лабораторной работе должен содержать:

- цель и задание к лабораторной работе;
- блок-диаграмму «измерительного стенда»;
- результаты экспериментов на объекте, выведенные в рабочее пространство системы Matlab;
- окна интерфейса *ident*, отражающие этапы загрузки, обработки и размещения данных в области идентификации и верификации данных;
- результат идентификации объекта корреляционным методом;
- результат идентификации объекта спектральным методом;
- сравнительные графики полученных моделей;
- результат идентификации объекта параметрическими моделями различных порядков;
- сравнительные графики параметрических моделей;
- оценки параметров модели в командном окне Matlab;
- выводы по работе.

6.5 Варианты заданий

Варианты выбираются из папки *Work/Objects5* пакета Simulink.

6.6 Контрольные вопросы

- 6.6.1 Что такое идентификация?
- 6.6.2 Модели каких объектов можно оценить с помощью пакета System Identification Toolbox?
- 6.6.3 Для чего предназначена графическая оболочка *ident*?
- 6.6.4 Какой блок пакета Simulink используется для обмена данными между различными пакетами системы Matlab?
- 6.6.5 Как загружаются данные эксперимента в графический интерфейс идентификации?
- 6.6.6 Для чего используется предварительная обработка данных эксперимента?
- 6.6.7 Как в окне интерфейса можно получить график переходной функции?
- 6.6.8 Как можно выполнить сравнение параметрических моделей различной структуры?
- 6.6.9 Что такое верификация модели?
- 6.6.10 Как можно в пакете *ident* оценить точность восстановленной модели?

7 Лабораторная работа №7. Непараметрическая идентификация динамических объектов

Цель работы: освоить методы непараметрической идентификации систем и навыки работы в пакете подгонки кривых Curve Fitting Toolbox системы MatLab.

7.1 Задание к лабораторной работе

В процессе выполнения лабораторной работы студент должен:

- изучить методы непараметрической идентификации и методы аппроксимации функций [1,2,3];
- изучить приемы работы в пакете Curve Fitting Toolbox;
- сформировать систему уравнений модели объекта для непараметрической идентификации объекта;
- выполнить процедуру непараметрической идентификации для определения дискретных значений импульсной переходной функции объекта;
- используя пакет Curve Fitting Toolbox, получить сглаженные решения задачи непараметрической идентификации.

7.2 Использование корреляционных функций для непараметрической идентификации объекта

В простейшем случае под непараметрической идентификацией понимается определение некоторых функций в операторе объекта. Специфика линейного динамического объекта однозначно определяется его реакцией на единичное импульсное воздействие. Это обстоятельство и лежит в основе определения *непараметрической* модели, которая характеризуется импульсной (весовой) переходной функцией.

Методы определения динамических характеристик исследуемого объекта при помощи подачи на вход искусственного возмущения определенного типа и замера реакции часто неприменимы по многим причинам. Рассмотрим метод, основанный на статистических представлениях. Статистический метод позволяет в качестве источника информации использовать случайные естественные сигналы идентифицируемого объекта. При этом может возникнуть ряд погрешностей, так как вероятность того, что случайные сигналы имеют необходимый частотный спектр, очень мала. Поэтому при статистическом методе идентификации широко используются корреляционные и спектральные функции.

При идентификации линейного стационарного динамического объекта статистическими методами весовая функция определяется из уравнения Винера-Хопфа

$$R_{yx}(\tau) = \int_0^T g(t) R_{xx}(\tau - \theta) d\theta. \quad (7.1)$$

Один из способов решения этого уравнения – численный. Суть численного метода заключается в возможности представления этого уравнения системой линейных алгебраических уравнений. Для этого заменим интеграл конечной суммой:

$$R_{yx}(\tau) = \sum_{n=0}^N g(n\Delta) R_{xx}(\tau - n\Delta) \cdot \Delta,$$

здесь Δ - интервал дискретизации;

$n\Delta$ - дискретное время;

N – количество интервалов.

Это уравнение справедливо для $\tau = \Delta, \dots, N\Delta$.

Подставляя в это уравнение N значений τ , получим систему из N уравнений. Это система для определения значений импульсной переходной функции в дискретные моменты $\Delta, \dots, N\Delta$.

Итак, получаем систему уравнений

$$Ag = b, \quad (7.2)$$

где A – квадратная матрица;

$$A = \begin{vmatrix} R_x(1) & R_x(2) & R_x(3) & \dots & R_x(N) \\ R_x(0) & R_x(1) & R_x(2) & \dots & R_x(N-1) \\ R_x(1) & R_x(0) & R_x(1) & \dots & R_x(N-2) \\ \dots & \dots & \dots & \dots & \dots \\ R_x(N) & R_x(N-1) & R_x(N-2) & \dots & R_x(0) \end{vmatrix},$$

$b = [b_1, b_2, \dots, b_m]^T$ - вектор-столбец с компонентами $b_i = R_{yx}(\tau)/\Delta$, $i = 1, \dots, m$;

$g = [g_1, g_2, \dots, g_m]^T$ - вектор-столбец искомых значений импульсной переходной функции.

Решение этой системы позволяет определить дискретные значения ординат весовой функции $g(\tau)$ в точках $t, 2t, \dots, mt$. Эти решения получаются с большими погрешностями, так как вместо истинных значений корреляционных функций используются их оценки, а сама система плохо обусловлена. Хотя полученные значения импульсных переходных функций имеют малую среднеквадратичную ошибку, близкую к минимуму, ценность их невелика, так как эти функции не соответствуют физическому смыслу процессов в объекте. Физический смысл имеют гладкие решения. Поэтому используются различные процедуры сглаживания импульсной переходной функции.

7.3 Аппроксимация дискретной импульсной переходной функции

Для получения сглаженных импульсных переходных функций, используется процедура приближения значений этой функции с помощью какого-либо аппроксимирующего полинома. Чаще всего используется разложение по какой-либо системе ортогональных полиномов

$$g(t) = \sum_{k=0}^N a_k \varphi_k(t),$$

где $\varphi_k(t)$ – заданная система функции;

N – порядок разложения;

a_k – коэффициенты ряда, которые вычисляются по формуле

$$a_k = \sum_{t=0}^m g_k \varphi_k(t).$$

Функции разложения выбираются таким образом, чтобы достичь хорошего приближения при небольшом их числе. Целесообразно выбирать ортогональные функции, преобразуемые по Лапласу (полиномы Чебышева, Лагерра, Лежандра).

7.4 Порядок выполнения лабораторной работы

7.4.1 Зарегистрированы входные и выходные переменные исследуемого объекта в течение определенного интервала времени. По результатам этих измерений вычислены автокорреляционная и взаимокорреляционная функции входного и выходного сигналов (см. таблицу вариантов).

7.4.2 Запишите систему линейных алгебраических уравнений (7.2) для своего варианта.

7.4.3 Решите систему (7.2), используя командное окно системы *MatLab* (аналогично п.6.4.6 лабораторной работы №6).

7.4.4 Используя пакет *Curve Fitting Tool*, выполните аппроксимацию полученных дискретных значений ортогональными полиномами различных порядков (для четных вариантов используются полиномы Лагерра, для нечетных - полиномы Лежандра).

7.4.5 Выберите полином, который наилучшим образом аппроксимирует импульсную весовую функцию.

7.4.6 Используя полученную импульсную переходную функцию, запишите непараметрическую модель

$$y(t) = \int_0^T g(\theta) \cdot x(t - \theta) d\theta .$$

7.4.7 По полученному уравнению модели соберите блок-диаграмму.

7.4.8 Проведите эксперименты, подавая на вход модели различные сигналы; выходные сигналы отразите в смотровом окне *Scope*.

7.5 Содержание отчета

Отчет по лабораторной работе должен содержать:

- цель и задание к лабораторной работе;
- систему алгебраических уравнений для определения дискретных значений импульсной переходной функции;
- массив результатов решения этой системы;
- последовательность выполнения аппроксимации дискретных значений импульсной переходной функции;
- аппроксимирующие выражения для импульсной переходной функции;
- обоснованный выбор наилучшего аппроксимирующего полинома.
- блок-диаграмму непараметрической модели;
- график выходного сигнала модели.

7.6 Варианты заданий

№ варианта	t, мин	0	1	2	3	4	5	6	∞
Вариант 1	Rx(t)	1	0,37	0,16	0,05	0,02	0,01	0,005	0
	Rxy(t)	0,1788	0,4729	0,3866	0,2565	0,1454	0,06	0,03	0
Вариант 2	Rx(t)	1	0,47	0,2	0,1	0,05	0,01	0,005	0
	Rxy(t)	0,2376	0,5128	0,1343	0,2848	0,155	0,003	0,003	0
Вариант 3	Rx(t)	1	0,53	0,26	0,15	0,1	0,02	0,003	0
	Rxy(t)	0,3049	0,5792	0,5575	0,4152	0,2176	0,002	0,002	0
Вариант 4	Rx(t)	1	0,5	0,3	0,2	0,09	0,03	0,003	0
	Rxy(t)	0,3978	0,7428	0,7030	0,5960	0,3200	0,001	0,001	0
Вариант 5	Rx(t)	1	0,52	0,28	0,18	0,1	0,02	0,005	0
	Rxy(t)	0,4872	0,9122	0,8998	0,7320	0,4088	0,002	0,002	0
Вариант 6	Rx(t)	1	0,46	0,22	0,12	0,1	0,03	0,004	0
	Rxy(t)	0,3825	0,8008	0,7846	0,6214	0,3794	0,001	0,002	0
Вариант 7	Rx(t)	1	0,38	0,19	0,04	0,02	0,01	0,005	0
	Rxy(t)	0,1735	0,4398	0,3637	0,2415	0,1026	0,003	0,002	0
Вариант 8	Rx(t)	1	0,45	0,2	0,1	0,05	0,01	0,004	0
	Rxy(t)	0,2531	0,5655	0,5020	0,3365	0,1445	0,003	0,002	0
Вариант 9	Rx(t)	1	0,39	0,19	0,06	0,01	0,001	0,0001	0
	Rxy(t)	0,2428	0,6008	0,5659	0,4181	0,1166	0,0005	0,003	0
Вариант 10	Rx(t)	1	0,35	0,18	0,05	0,02	0,01	0,005	0
	Rxy(t)	0,2495	0,6569	0,5835	0,3389	0,1058	0,002	0,003	0

7.7 Контрольные вопросы

7.7.1 В чем суть непараметрической идентификации?

7.7.2 Дайте определение импульсной переходной функции.

7.7.3 Дайте определение автокорреляционной и взаимокорреляционной функций.

7.7.4 С какой целью используются корреляционные функции при непараметрической идентификации?

7.7.5 Какие объекты описывает уравнение Винера-Хопфа?

7.7.6 Как преобразуется уравнение Винера-Хопфа в систему алгебраических уравнений?

7.7.7 Дайте определение ортогонального полинома.

7.7.8 Преимущества аппроксимации ортогональными полиномами?

7.7.9 Объясните назначение пакета Curve Fitting Tool.

7.7.10 Какая из функций дает лучшие результаты приближения?

Список литературы

1. Ибраева Л.К., Хисаров Б.Д. Моделирование и идентификация объектов управления. Учебное пособие. – Алматы: АИЭС, 2009 – 209 с.
2. Ибраева Л.К. Моделирование и идентификация объектов управления. Конспект лекций для студентов всех форм обучения специальности 5В0702 – Автоматизация и управление. - Алматы: АИЭС, 2009 – 84 с.
3. Сыздыков Д.Ж. Идентификация в системах управления. – Алматы: Изд-во «Эверо», 2007.
4. MATLAB 6.5 SP1/7.06 Simulink 5/6 в математике и моделировании. – М7: СОЛОН-Пресс, 2005.
5. www.exponenta.ru
6. www.matlab.ru