МОДЕЛИРОВАНИЕ И ИДЕНТИФИКАЦИЯ ОБЪЕКТОВ УПРАВЛЕНИЯ

Методические указания к выполнению лабораторных работ

для студентов специальности «Автоматизация и управление»

Алматы 2019

СОСТАВИТЕЛЬ: Шортанбаева А.Т. Моделирование и идентификация       объектов управления. Методические указания к выполнению лабораторных работ для студентов всех форм обучения специальности      050702 - Автоматизация и управление. – Алматы, 2019 Казахский Национальный Университет имени Аль-Фараби:

  Методические указания предназначены для выполнения лабораторных работ по дисциплине «Моделирование и идентификация объектов управления» и охватывают два основных подхода к моделированию объектов управления: аналитический и экспериментальный (идентификация). Методические указания включают 7 работ. Для выполнения этих работ (кроме работы 5) используются различные инструментарии программной системы MatLab. Это система предназначена для практического моделирования динамических систем и содержит множество визуальных средств моделирования и исследования таких систем. При выполнении 5 лабораторной работы для обработки результатов экспериментов используется программа, разработанная на кафедре «Искусcтвенный интеллект BIG DATA».

**Содержание**

|  |  |
| --- | --- |
|   |   |
| Введение |   |
| 1 Лабораторная работа № 1. Исследование линейных стационарных систем |  5 |
|  1.1 Основные виды моделей линейных динамических объектов | 7 |
|    1.2 Исследование линейных стационарных систем | 7 |
|  1.3 Задание на выполнение лабораторной работы | 10 |
|            1.4 Требования к отчету | 11 |
|            1.5 Варианты заданий | 12 |
|            1.6 Контрольные вопросы | 13 |
| 2 Лабораторная работа № 2. Аналитические методы моделирования объектов с сосредоточенными параметрами |  13 |
|           2.1 Описание процесса | 13 |
|  2.2 Моделирование систем в пакете Simulink | 14 |
|    2.3 Реализация блок-диаграммы модели | 14 |
|    2.4 Задание на выполнение лабораторной работы | 16 |
|            2.5 Требования к отчету | 16 |
|            2.6 Варианты заданий | 16 |
|            2.7 Контрольные вопросы | 17 |
| 3 Лабораторная работа № 3. Моделирование объекта регулирования уровня жидкости в резервуаре |  18 |
|  3.1 Описание объекта | 18 |
|  3.2 Подсистема Statefiow | 19 |
|           3.3 Задание на выполнение лабораторной работы | 21 |
|  3.4 Требования к отчету | 21 |
|    3.5 Варианты заданий | 21 |
|  3.6 Контрольные вопросы | 22 |
| 4 Лабораторная работа № 4. Идентификация объекта с помощью переходной функции |  22 |
|             4.1 Графический метод идентификации с помощью переходной функции |  22 |
|    4.2 Задание на выполнение лабораторной работы | 23 |
|    4.3 Требования к отчету | 24 |
|            4.4 Варианты заданий | 24 |
|            4.5 Контрольные вопросы | 25 |
| 5 Лабораторная работа №5 Идентификация линейного объектас помощью частотной характеристики |  25 |
|          5.1 Определение коэффициентов передаточной функциис помощью частотных характеристик |  25 |
|    5. 2 Программа обработки результатов измерений | 26 |
|    5.3 Задание на выполнение лабораторной работы                             | 27 |
|    5.4 Требования к отчету | 27 |
|            5.5 Варианты заданий | 27 |
|            5.6 Контрольные вопросы | 28 |
| 6 Лабораторная работа № 6. Параметрическая идентификация линейных систем |  28 |
|            6.1 Постановка задачи | 28 |
|  6.2 Идентификация параметров | 28 |
|            6.3 Верификация модели | 30 |
|    6.4 Задание на выполнение лабораторной работы | 30 |
|    6.5 Требования к отчету | 31 |
|            6.6 Варианты заданий | 31 |
|            6.7 Контрольные вопросы | 31 |
| 7 Лабораторная работа № 7. Непараметрическая идентификация динамических объектов |  31 |
|            7.1 Постановка задачи | 31 |
|    7.2 Аппроксимация дискретной импульсной переходной функции | 32 |
|    7.3 Пакет подгонки кривых Curve Fitting Toolbox | 33 |
|    7.4 Задание на лабораторную работу | 34 |
|    7.5 Порядок выполнения работы | 35 |
|    7.6 Требования к отчету | 35 |
|            7.7 Варианты заданий | 35 |
|            7.8 Контрольные вопросы | 37 |
| Список литературы | 37 |

**Введение**

Математическое моделирование - это средство для изучения процессов, протекающих в различных системах и устройствах без их создания «в натуре». Существует два принципиально различных подхода к построению математических моделей.

Первый подход основан на выборе моделей с учетом основных физико-химических закономерностей, определяющих течение исследуемого процесса. Такие модели называются *аналитическими* моделями процесса. Моделирование в этом случае основано на ограниченности числа фундаментальных законов природы и принципе подобия, означающем, что явления различной физической природы могут описываться одинаковыми математическими зависимостями. Аналитические методы определения характеристик объектов регулирования основаны на составлении их дифференциальных уравнений, базирующемся на использовании основных физических законов: сохранения массы, энергии и количества движения. Как правило, этим  методом удается получить нелинейное уравнение объекта, аналитическое решение которого в общем случае не может быть получено. Следующим шагом является *линеаризация* полученного уравнения, то есть переход к линейной математической модели объекта. Линеаризация обычно проводится путем разложения нелинейных зависимостей в ряд Тейлора в окрестности исходного стационарного режима с сохранением только линейных частей разложения и последующим вычитанием уравнений статики. Полученная таким образом линейная модель объекта справедлива лишь при малых отклонениях от исходного стационарного режима.

Второй подход базируется на концепции "черного ящика", то есть постулируется, что внутренняя структура объекта неизвестна, да и не должна интересовать исследователя. Вся информация получается только в результате наблюдений за объектом при пассивном и активном эксперименте. Полученные таким образом модели называются *эмпирическими* (экспериментальными). При построении эмпирических моделей используются реализации входных и выходных переменных объекта. Задачи восстановления таких моделей 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. Для выполнения этих работ (кроме работы 5) используются различные инструментарии программной системы MatLab. Эта система выгодно выделяется среди систем компьютерного моделирования. Она является одной из старейших систем автоматизации математических расчетов и построена на расширенном представлении и применении матричных операций, что нашло отражение в названии системы - MATrix LABoratory (матричная лаборатория).

Отметим те средства системы, которые будут использоваться для выполнения лабораторных работ.

Любые (подчас весьма сложные) вычисления можно выполнять в режиме *прямых вычислений*(в командном режиме), то есть без подготовки программы. Это превращает MatLab в необычайно мощный научный калькулятор, который способен производить не только обычные для калькуляторов вычисления, но и операции с векторами, матрицами, комплексными числами, рядами и полиномами. Причем работа в этом режиме носит диалоговый характер. 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. Несмотря на значительные возможности работы в режиме калькулятора, сложные вычисления требуют оформления их в виде программ. Для использования процедур пользователь создает *М-файл* в специальном окне редактора программ. Рекомендуется выполнить упражнения, приведенные в [6], п.п.1 и 2, чтобы освоить основные приемы программирования и работы в командном окне системы MatLab.

В лабораторной работе 1 для создания и анализа моделей линейных стационарных систем используется командное окно системы и пакет *Control System Toolbox.* Пакет Control Toolbox предоставляет широкий набор процедур, осуществляющих анализ системы автоматического управления с самых различных точек зрения и, прежде всего, определение откликов системы на внешние воздействия, как во временной, так и в частотной области.

Популярности системы MatLab способствует ее мощное расширение *Simulink*, предоставляющее пользователю удобные и простые средства, в том числе визуального объектно-ориентированного программирования, для блочного моделирования линейных и нелинейных динамических систем. В лабораторных работах 2 и 3 модели исследуемых процессов реализуются в виде блок-диаграмм пакета Simulink, и исследование модели проводится в визуальном режиме. Этот пакет также используется в лабораторной работе 4 для идентификации объекта графическим методом.

Моделирование систем и устройств, поведение которых зависит от внешних событий, в системе MatLab + Simulink реализуется с помощью специального расширения *Stateflow*. Этот пакет используется при выполнении лабораторной работы 3.

В лабораторной работе 5 идентификация объекта с помощью частотной характеристики проводится в среде программы, разработанной на кафедре «Инженерная кибернетика».

Для параметрической идентификации систем (лабораторная работа 6) используется графическая оболочка *ident* программного продукта MatLab.

Задачи непараметрической идентификации систем рассмотрены в лабораторной работе 7. Для численного решения задачи непараметрической идентификации используется командное окно системы. Для аппроксимации дискретных значений искомой импульсной переходной функции используется пакет подгонки кривых *Curve Fitting Toolbox.*

**1 Лабораторная работа №1. Исследование линейных стационарных систем**

Цель работы: создание моделей динамических систем в командном окне MatLab; анализ моделей средствами пакета Control System Toolbox.

1.1 Основные виды моделей линейных динамических объектов

В целом ряде случаев реакция линейной стационарной системы на то или иное достаточно простое воздействие определяется ее теоретическими характеристиками. Основными из них являются:

- *передаточная функция W(p)*(определяется с помощью преобразования Лапласа *L{∙}*);

- *импульсная характеристика* *g(t)*– реакция предварительно невозмущенного объекта на входной сигнал в виде δ-функции.

При нулевых начальных условиях связь между выходным и входным сигналами описывается интегралом свертки

                   

- *переходная функция* *h(t)*– реакция предварительно невозмущенного объекта на входной сигнал в виде единичного скачка.

Из теории управления известны следующие соотношения между этими характеристиками

*L{w(t)} = W(p), g(t)= h’(t),;*

 - *частотные характеристики*объекта, которые определяются

комплексным коэффициентом передачи *W(jω)= W(p)*|*p=jω.*

Под *моделью* будем понимать информацию о наиболее существенных характеристиках объекта, выраженную в той или иной форме.

Приведем основные виды моделей линейных непрерывных стационарных динамических объектов.

1.     Дифференциальные уравнения.

  Наиболее универсальная модель, основанная на дифференциальных уравнениях, описывается выражением



где *na* - порядок модели (*na > nb*);

*ai*и *bj*- постоянные коэффициенты (параметры модели);

*u(j)(t)* и *y(i)(t)* - производные соответственно входного и выходного сигналов.

2.     Уравнения переменных состояния.

 При выборе *n* координат системы (объекта) в качестве переменных ее состояния (такими координатами, например, могут быть выходной сигнал      *y(t)*и *n-1*его производных) *xi(t), i=1,2,...,n,*данную систему можно описать уравнениями для переменных состояния

                            ,

где X(t) = [*x1(t), x1(t),..., xn(t)*]T - вектор-столбец переменных состояния; A,B,C и D при скалярных *u(t)*и *y(t)* – соответственно матрица размера

*n*x*n*, векторы размера *n*x*1*x*1* и скаляр (при векторных *u(t)*и*y(t)* – матрицы соответствующих размеров).

Приведенные модели являются эквивалентными, то есть, зная любую из них, можно получить все остальные.

3.     Разностные уравнения.

 Для объектов, функционирование которых представляется для дискретного времени *tk=kT*(в данном случае *T* – интервал дискретизации), то есть для дискретных объектов, наиболее общим видом описания является разностное уравнение (аналог дифференциального)

*yk + a1yk-1 + ... + amyk-na = b1uk + b2uk-1 + b3uk-2 + ... + bnbuk-nb+1*

         где  *yk-i = y*[*(k – i)T*]*, uk-j = u*[*(k – j)T*].

4.     Z-преобразование.

 Связь между сигналами может быть отражена также через дискретную свертку

                        

         где g*i* – ординаты весовой решетчатой функции объекта, или, с использованием аппарата Z-преобразования

                            ****

         где z = e pT, через дискретную передаточную функцию

                                 ,

которая определяется на основании разностного уравнения после применения к обеим частям этого уравнения Z-преобразования

         (1 + a1z-1 + a2z-2 + …+ anaz-na)Y(z) = (b1 + b2z-1 + b3z-2 + …+ bnbz—nb+1)U(z)

Оператор z-1 = e-pT представляет собой оператор задержки, то есть

           z -1uk = uk-1, z-2uk = uk-2и т.д.

Отметим, что на практике в большинстве случаев измерение непрерывных сигналов производится в дискретные моменты времени, что представляет определенное удобство при последующей обработке данных на компьютере. Непрерывные объекты можно приближенно отображать дискретными моделями. При этом для перехода от непрерывных моделей к дискретным, применяется либо Z-преобразование, либо замена производных в дифференциальном уравнении, описывающем непрерывный объект, конечными разностями (этот подход дает приемлемую точность только при малых интервалах дискретизации и небольшом порядке уравнения).

Ниже приведены несколько распространенных дискретных моделей дискретных объектов для временной области, учитывающих действие шума наблюдения (моменты дискретного времени обозначены тем же символом t, что и непрерывное время, здесь t = 0, 1, 2…).

5.     Модели авторегрессии.

  Модель авторегрессии AR (AutoRegressive) считается самым простым описанием

*A(z) y(t) = e(t),*

         где

*A(z) = 1 + a1z-1+ a2z-2 +...+ anaz-na.*

Более сложная – ARX-модель (AutoRegressive with eXternal input)

*A(z) y(t) = B(z) u(t) + e(t)*

или в развернутом виде

*y(t) + a1y(t-1) +...+ anay(t-n) = b1u(t) + b2u(t-1) +...+ bnbu(t-m) + e(t).*

Здесь и ниже *e(t)* – дискретный белый шум, *B(z) =b1+ b2z-1 +...+ bnbz-nb+1.*

         ARMAX-модель (AutoRegressive-MovingAverage with eXternal input) – модель авторегрессии скользящего среднего

*A(z) y(t) = B(z) u(t - nk) + C(z) e(t)*

         где *nk -*величина задержки (запаздывания),

*C(z) = 1 +c1z-1+ c2z-2 +...+ bncz-nc.*

         Модель «вход-выход» (в англоязычных источниках такая модель называется «Output-Error», то есть «выход-ошибка», сокращенно ОЕ)

                            **

         где *F(z) = 1 +f1z-1+ f2z-2 +...+ fnfz-nf.*

         Модель Бокса-Дженкиса (BJ)

                            **

         где *D(z) = 1 +d1z-1+d2z-2 +...+ dndz-nd.*

         Все эти модели можно рассматривать как частные случаи обобщенной параметрической линейной структуры

                            **

при этом все они допускают расширение для многомерных объектов (имеющих несколько входов и выходов).

6.     Модель для переменных  состояния (State Space).

Она имеет вид

*x(t +1) =*A*x(t) +*B*u(t) +*K*e(t,*

*y(t) =*C*x(t) +*D*u(t) + v(t)*

         где A, B, C, D - матрицы соответствующих размеров, *v(t)* - коррелированный шум наблюдения.

  1.2 Исследование линейных стационарных систем

В системе MatLab для исследования линейных стационарных систем используется пакет *Control System Toolbox*. Пакет реализует методы исследования динамических систем, основанные на использовании передаточных функций и моделей переменных состояния. Ввод модели в среду пакета возможен в трех формах – в форме матриц пространства состояния, в виде коэффициентов числителей и знаменателей передаточных функций и в форме задания нулей, полюсов и коэффициента передачи системы. Надо отметить, что здесь линейные,  инвариантные во времени системы с постоянными параметрами (в русском варианте ЛСС - линейные стационарные системы) названы *LTI-объектами* (Linear Time Invariant).

         Некоторые процедуры создания LTI -моделей:

*- ss*    - создает модель пространства состояния по заданным матрицам уравнений состояния системы;

*- dss*- создает аналогичную модель  по описанию пространства состояния более общего вида, когда уравнения переменных состояния не          разрешены относительно производных;

*- tf*- создает модель по заданным передаточным функциям системы;

*- zpk*-  создает модель по заданным нулям, полюсам и коэффициентам передачи системы;

*- filt***-**создает модель по дискретным передаточным функциям,

записанным в форме полиномов от z—1;

*- set*    - присваивает значения некоторым другим полям объектов

(названиям входов и выходов, названиям системы и т.д.).

Указанные процедуры позволяют создавать как непрерывные, так и дискретные модели. Кроме этого существует инструментарий  преобразования моделей из одной указанной формы в другую.

         Удобным средством просмотра временных и частотных характеристик модели является интерактивный обозреватель *LTI-Viewer*. Для вызова графического интерфейса из командного окна используется функция *ltiview,* при этом на экране появляется рабочее окно обозревателя. Для обмена данными между рабочими средами MatLab и LTI-Viewer используются команды меню окна обозревателя *File/Import* и *File/Export*. После загрузки модели в окне обозревателя отобразится график переходной функции модели. С помощью обозревателя легко вычисляются и отображаются в виде графиков и диаграмм частотные и временные характеристики, нули и полюсы системы.

         1.3 Задание к выполнению лабораторной работы

         1.3.1 В командном окне MatLab введите вектора коэффициентов

числителя *num* и знаменателя *den* передаточной функции объекта первого порядка.

         1.3.2 Используя команду *model = tf(num,den)* создайте модель объекта.

         1.3.3 Преобразуйте непрерывную модель в дискретную

                            *disc=c2d(model, 0.01)*, здесь 0.01 – шаг дискретизации

и обратно в непрерывную

*cont=d2c(disc).*

         1.3.4 Убедитесь, что при обратном преобразовании получен исходный

вариант модели.

         В результате этих действий в рабочей области сохранены три модели: *model, disc, cont.*

1.3.5 Загрузите файл с демонстрационными LTI-моделями, набрав в

командной строке *load LTIexamples*.

         1.3.6 В командной строке наберите *ltiview*. Выберите в появившемся окне *File/Import*. Откроется диалоговое окно выбора модели для загрузки. Ваши модели тоже будут в этом списке.

         1.3.7 Загрузите свою модель*model*. Появится окно с графиком

переходной функции.

         1.3.8 Используя указатель мыши, щелчки мышью на графике,

контекстные меню, выполните различные операции (ввиду их очевидности,  описание этих операций не приводится).

         1.3.9 Активизируйте меню *Edit/Plot Configutation*, это приведет к появлению диалогового окна, позволяющего выбрать набор интересующих характеристик для их отображения. Возможен выбор до 6 характеристик одновременно. На рисунке 1.1 показаны четыре характеристики.

         1.3.10 Используя контекстные меню, установите: надписи на графике; размеры графика; стиль графика и другие возможные свойства.

         1.3.10 Выполните следующие действия:

         - выбором меню *File/Toolbox* установите параметры инструментария;

          - выбором меню *Preferences File/LineSstyle* установите стиль

графического изображения;

         - выбором меню *Preferences File/LTI Viewer Preferences* установите

параметры векторов времени и частот.

         1.4  Требования к отчету

       Отчет по лабораторной работе должен содержать:

- листинг командного окна с моделями систем;

- график переходной функции системы с установленными свойствами

(пояснительные надписи, сетка, цвет и тип шрифтов и т.д.).

         

                   Рисунок 1.1 - Графики переходной и частотных характеристик

         1.5 Варианты заданий

         Параметры передаточной функции объекта первого порядка выбираются студентом самостоятельно.

         1.6 Контрольные вопросы

         1.6.1 Что собой представляет система Matlab?

         1.6.2  Назовите основные элементы командного окна системы Matlab.

         1.6.3 Какие режимы работы в этой системе вы изучили?

         1.6.4 Что является основным объектом системы?

         1.6.6 Какие зарезервированные переменные в системе Matlab вы знаете?

         1.6.7 Назначение пакета Control System Toolbox?

         1.6.8 Что такое LTI-объект?

         1.6.9  Перечислите известные вам процедуры создания LTI –моделей.

         1.6.10 Перечислите известные вам процедуры преобразования моделей.

**2 Лабораторная работа  № 2.  Аналитические методы моделирования объекта с сосредоточенными параметрами**

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

2.1 Описание процесса

Объектом исследования является мельница, в которую поступает входной поток исходного перерабатываемого материала , содержащего крупные фракции. Выходной поток из мельницы  пропорционален массе материала в мельнице М

                     

где   -  константа пропорциональности.

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

                     

где .

Здесь m - масса мелкой фракции в мельнице; С - доля мелкой фракции; k0 (1/час) - константа скорости измельчения; М+ - характерная масса, соответствующая максимальной скорости измельчения (максимальной производительности мельницы). Значение М+= 10 т; k0 = 1,5 (1/час).

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

                     ;

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

                    .

Система дифференциальных уравнений, описывающих рассматриваемый процесс имеет вид:

,

                     .

В лабораторной работе исследуется производительность мельницы как функция выходного потока .

2. 2 Моделирование систем в пакете Simulink

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

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.

Чтобы запустить пакет необходимо из командного окна MatLab выбрать команду *File/New/Model* или нажать соответствующую кнопку на панели инструментов. При этом открывается окно *Simulink Library Browser* с перечнем разделов библиотеки Simulink. Выбрав в этом окне *File/New/Model* создаем пустое окно *untitled* – заготовку для создания новой модели. Вид экрана приведен на рисунке 2.1.

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.

2.3 Реализация модели процесса

Модель процесса представляет собой систему дифференциальных уравнений второго порядка для определения масс крупной M(t) и мелкой



  Рисунок 2.1 – Окно пакета Simulink для создания новой модели

фракций m(t). Для преобразования этих уравнений в блок-схемы используется следующая идея. Вначале формируются правые части уравнений, считая, что M и m известны, в результате получаются dM/dt и dm/dt. Проинтегрировав эти выражения, получаем значения M(t) и m(t). Полученные величины теперь можно использовать для формирования правых частей уравнений (обратной связью). Итак, формирование блок-схемы состоит из следующих шагов:

а) в основу блок-схемы каждого уравнения положен блок *Integrator*, на его вход подаются величины, полученные в результате расчета правых частей уравнений; выходы этих блоков представляют искомые значения M(t) и m(t);

б) сформировать отдельным блоком в виде подсистемы (блок *Subsystem*) правую часть первого уравнения, используя в качестве входных процессов полученные значения M(t), а также функцию входного потока Φ0; вторая подсистема (для второго уравнения) в качестве входных процессов использует полученные значения M(t) и m(t).

       В подсистемах для формирования выражений правых частей уравнений используется блок *fcn.*  Этот блок 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),...,*нумерация компонентов вектора в том порядке, в каком они поданы на блок Mux; выражение в окне настройки блока формируется по правилам языков программирования;

в) используя созданные подсистемы, связать сформированные процессы M(t) и m(t) с соответствующими входами подсистем, а выход подсистем связать с окном просмотра результатов моделирования *Scope*;

г) выход второй подсистемы m(t) связать с блоком *Gain* и получить исследуемую производительность ; отразить Р в смотровом окне *Scope*;

д) теперь необходимо сформировать входной поток; здесь также используется блок *fcn,*в котором формируется выражение для входного потока (по варианту); так как входной поток задан как явная функция времени, то на вход этого блока *fcn* надо подать сигнал с блока *Clock.*

2.4 Задание к выполнению лабораторной работы

2.4.1 В пакете Simulink системы MatLab построить блок-диаграмму динамической модели процесса. Сохранить файл с моделью.

2.4.2 Провести имитационные эксперименты на модели (по варианту). Найти наиболее производительные режимы.

2.5 Требования к отчету

  Отчет по лабораторной работе должен содержать:

  - блок-диаграммы модели с поясняющими комментариями.

       - результаты проведенных исследований.

 2.6 Варианты заданий

 В вариантах 1-8 коэффициент   = 0.5 (1/час).

 2.6.1 Исследовать производительность мельницы как функцию входного потока Ф0(t) = a\*exp(-bt). Найти наиболее производительные режимы.

 2.6.2 Исследовать производительность мельницы как функцию входного потока Ф0(t) = a\*exp(bt). Найти наиболее производительные режимы.

2.6.3 Исследовать производительность мельницы как функцию входного потока Ф0(t) = a1\*t2 + a2t + a3. Найти наиболее производительные режимы.

 2.5.4 Исследовать производительность мельницы как функцию входного потока Ф0(t) = a1t + a2. Найти наиболее производительные режимы.

 2.6.5 Исследовать производительность мельницы как функцию входного потока Ф0(t) = sqrt(a1t + a2). Найти наиболее производительные режимы.

 2.6.6 Исследовать производительность мельницы как функцию входного потока Ф0(t) = a1/(a2t + a3).  Найти наиболее производительные режимы.

 2.6.7 Исследовать влияние на производительность мельницы частоты  периодически изменяющегося входного потока Ф0(t) = Ф0\* + 2.5cos(t),  где  Ф0\* = 5 т/час. Требуется найти наиболее «опасные» частоты колебаний  входного потока.

   2.6.8 Исследовать влияние на производительность мельницы амплитуды А периодически изменяющегося входного потока Ф0 = Ф0\* + Аcos(0.2t), где Ф0\* = 5 т/час. Требуется найти амплитуды наиболее «опасных» колебаний входного потока.

2.6.9 Исследовать влияние на производительность мельницы частоты   периодически изменяющегося коэффициента : = 0 + 0.25cos(t), где 0 = 0.5 (1/час). Выбрать оптимальный режим изменения .

2.6.10 Исследовать влияние на производительность мельницы амплитуды 1  периодически изменяющегося коэффициента : = 0 + 1cos(0.2t), где 0 = 0.5 (1/час). Выбрать оптимальный режим изменения .

2.7 Контрольные вопросы

2.7.1 Каковы основные типы математических моделей?

2.7.2 В чем отличие аналитических методов моделирования от

экспериментальных?

2.7.3 Какие законы сохранения используются для вывода уравнений

модели рассматриваемого процесса?

2.7.4 К какому классу относится модель рассматриваемого процесса?

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

2.7.6 Объясните назначение пакета Simulink?

2.7.7 Что собой представляет подсистема в Simulink?

2.7.8 Объясните назначение блоков Clock, Integrator,  fcn, Gain?

2.7.9 Какие варианты просмотра результатов экспериментов на модели вы знаете?

2.7.10 Какой буквой обозначается входной сигнал блока fcn*?*

**3 Лабораторная работа №3. Моделирование объекта регулирования уровня жидкости в резервуаре**

Цель работы: освоение пакета событийного моделирования Stateflow системы MatLab.

3.1 Описание объекта

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

                       ,

где F – площадь поперечного сечения резервуара, ρ – плотность.

При составлении уравнений математической модели объекта используются:

- уравнение материального баланса системы

**                                                               (3.1)

- уравнение, отражающее закон  сохранения движения

                           .                                             (3.2)

         Примем за координаты состояния системы уровень жидкости в резервуаре  Н и поток на выходе Gc.

Статическая модель системы будет иметь вид

,

.                                                              (3.3)

Динамика системы описывается следующим дифференциальным уравнением:

          ,                                     (3.4)

здесь для Gc остается справедливым соотношение (3.2).

В лабораторной работе требуется в различные временные отрезки общего интервала моделирования провести расчеты по статической (3.3) или динамической (3.4) моделям. Переходные процессы в системе возникают в результате изменения входных переменных Gn(t) и fc.

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

3.2 Подсистема Statefiow

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 Statefiow-диаграммы необходимо сначала создать модель в Simulink или открыть уже существующую. Затем нужно выбрать в библиотеке блок, обозначающий диаграмму Statefiow, и перенести его в свою модель. После двойного щелчка по нему мышью, откроется окно *Statefiow**(Сhart)* (рисунок 3.1), в котором, пользуясь меню и панелью инструментов, можно описывать состояния, связи между ними, определять переменные, условия переключения и другие, необходимые для управления элементы.



Рисунок 3.1 - Пример простой SF-диаграммы

На диаграмме Statefiow 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 получает значение “истина” условие перехода, ведущего к нему, 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.

         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 мышь 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;

*сondition –* булевское 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 Statefiow - *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.*

3.3 Задание к выполнению лабораторной работы

3.3.1 В пакете Simulink системы MatLab постройте блок-диаграммы статической и динамической модели объекта, оформите их как подсистемы.

3.3.2 Подключите блок Chart к модели в Simulink.

3.3.3 Разработайте SF-диаграмму для проведения имитационных

экспериментов на модели:

  -  временные интервалы (t0 - t1), (t2 - t3), (t4 - t5) отражают три состояния равновесия резервуара;

  - интервал (t1 – t2) соответствует переходному процессу в  резервуаре, вызванному изменением притока жидкости Gn в момент t1;

          - переходной процесс на интервале (t3 – t4) вызван изменением

проходного сечения сливного отверстия в момент t3.

 3.4 Требования к отчету

    Отчет по лабораторной работе должен содержать:

 - подсистемы блок-диаграмм моделей с поясняющими комментариями;

 -  SF-диаграмму, описывающую функционирование системы в

различных временных интервалах;

 - графики проведенных имитационных экспериментов.

 3.5 Варианты заданий

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № варианта | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| F | 2.0 | 1.5 | 1.0 | 2.2 | 1.8 | 2.4 | 1.2 | 1.9 | 2.1 | 1.2 |
| fc | 0.6 | 0.3 | 0.2 | 0.5 | 0.4 | 0.6 | 0.2 | 0.3 | 0.7 | 0.1 |
| Gп |  350 | 300 | 250 | 270 | 330 | 390 | 270 | 340 | 460 | 370 |

Моменты времени t0, t1, t2, t3, t4, t5 выбираются студентом самостоятельно.

3.6 Контрольные вопросы

3.6.1 К какому классу относится разработанная модель?

3.6.2 Какие переменные моделируемого объекта являются входами модели?

3.6.3 Какие переменные моделируемого объекта являются выходами модели?

3.6.4 Дайте определение статической и динамической модели.

3.6.5 Из-за чего может возникнуть переходной процесс в

рассматриваемом объекте?

3.6.6 Для чего предназначен пакет Stateflow?

3.6.7 Что такое SF-диаграмма?

3.6.8 Объясните понятие «состояние» в пакете Stateflow.

3.6.9 Что такое переходы?

3.6 10 Как создать или изменить данные в пакете Stateflow?

**4 Лабораторная работа №4. Идентификация объекта с помощью переходной функции**

Цель работы: освоение методов графической идентификации линейных динамических объектов.

 4.1 Графический метод идентификации с помощью переходной функции

 Первые реализованные в системах управления методы идентификации были основаны на использовании частотных, ступенчатых и импульсных воздействий. Эти методы (прямые методы идентификации) требуют специальных входных сигналов, а именно: ступенчатых сигналов для идентификации по переходной функции (ступенчатой переходной функции), импульсных входных сигналов для идентификации по импульсной переходной функции и синусоидальных входных сигналов с различными частотами для определения частотной характеристики. Из трех типов входных сигналов, о которых говорилось выше, ступенчатый входной сигнал является наиболее простым для применения.

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 графический метод идентификации с использованием переходных функций применяется  к процессам первого порядка.

Пусть дан график переходного процесса. В момент времени t0=0 входная величина *x* скачком изменяется на величину *a*. Надо записать уравнение объекта. Искомое уравнение для объекта первого порядка имеет вид

  или   

Общее выражение для решения этого уравнения



Надо определить параметры T и k. Это можно выполнить следующими способами:

а) при t → ∞, имеем y(t) = k\*a, то есть через ординату y(t)  асимптоты (ордината асимптоты K=ka) можно определить k. Коэффициент k представляет собой соотношение между установившейся величиной выходного сигнала системы и амплитудой входного сигнала.

При t = T функция y(t) = K(1-e-1) = K(1-0.37) = 0.63K. Постоянная времени T системы первого порядка равна отрезку времени, за которое переходная функция достигает 63% своей установившейся величины;

б) продифференцируем решение



тогда

  ,

здесь α – угол наклона касательной к графику функции при t = 0. Тогда . Итак, величина T  - отрезок оси, равный расстоянию от начала координат до точки, в которой касательная пересекается с асимптотой. Коэффициент k определяется, как описано выше.

4.2 Задание к выполнению лабораторной работы

4.2.1 В окно редактора моделей Simulink загрузите из папки Work/Objects4 свой «объект» (по варианту).

4.2.2 Установите в качестве входного сигнала ступенчатый сигнал, график выходного сигнала отразите в смотровом окне.

4.2.3 Используя полученный график, определите обоими способами параметры передаточной функции объекта первого порядка.

4.2.4 По восстановленным моделям, подавая на их входы тот же ступенчатый сигнал, получите выходные сигналы (общий вид получившейся диаграммы приведен на рисунке 4.1).

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

 4.3 Требования к отчету

    Отчет по лабораторной работе должен содержать:

- смотровое окно с переходным процессом, на котором отмечены касательная к графику переходной функции, 63% -ая отметка и установившееся состояние;

- блок-диаграммы объекта и обеих моделей (аналогично рисунку 4.1);

         - графики переходных процессов объекта и обеих моделей в одном смотровом окне;

         - сравнительный анализ графиков.

  

  Рисунок 4. 1 - Блок-диаграмма сравнения выходов объекта и моделей

4.4 Варианты заданий

         Выбираются из папки Work/Objects4 пакета Simulink.

4.5 Контрольные вопросы

4.5.1 Что такое идентификация?

4.5.2 Какие методы называются прямыми методами идентификации?

4.5.3 Какая функция используется для графической идентификации линейного объекта?

4.5.4 Какой сигнал на входе объекта использовался для проведения процедуры идентификации?

4.5.5 Как провели касательную к графику переходного процесса в смотровом окне Scope?

4.5.6 Какой блок используется для того, чтобы отразить все графики в одном смотров окне?

4.5.7 Какой из способов идентификации дает более точный результат? Почему?

**5 Лабораторная работа №5. Идентификация линейного объекта с помощью частотной характеристики**

Цель работы: освоение методов идентификации с помощью сигналов определенного вида.

5.1 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) = A0sin(ωt),*

то установившееся измеренное значение выходного сигнала

*y(t) = A1sin[ωt + φ(ω)] + n(t),*

где *n(t)*  - ошибка измерения;

 *A1/ A0 =│W(jω)│;*

*φ = Arg[W(jω)].*

Частотная характеристика W(jω) определяется путем подачи синусоидальных входных сигналов *A0sin(ωt)* на различных частотах *ω* и записи соответствующих выходных сигналов *A1sin[ωt + φ].* С целью получения необходимой частотной характеристики, величины *A1/A0*и*φ* определяются для каждой рассматриваемой частоты *ω.*  То есть по записям входного и выходного сигналов определяют отношение амплитуд на частоте *ωi*и получают *│W(jωi)|.* Фазовый сдвиг *φ(ωi)* 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(ωi) = A(ωi) cos [φ(ωi)],

Q (ωi) = A(ωi) sin[φ(ωi)].

Эти данные используются для определения коэффициентов (a1,…,am-1,b0,…,bl) передаточной функции:

                W(p) = (bl pl+ bl-1pl-1 +…+ b0)/ (pm+ a1pm-1+ a2pm-2+ am-1).

 Для вычисления коэффициентов в выражении для передаточной функции заменим p на jω, обозначим (приравняем) его P(ω) + jQ(ω) и получим выражение, справедливое для всех значений ω. Из этого комплексного выражения, приравнивая коэффициенты при действительной и мнимой частях,  получим систему из двух уравнений. Подставляя различные значения ωi (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 повторных вычислениях значения коэффициентов будут сильно отличаться.

   5. 2 Программа обработки результатов измерений

Для определения коэффициентов передаточной функции необходимо решать системы уравнений большого порядка. В лабораторной работе для этого используется программа Lab6, разработанная на языке Delphi[[1]](http://libr.aues.kz/facultet/tef/kaf_ik/16/umm/ik_1.htm%22%20%5Cl%20%22_ftn1%22%20%5Co%20%22). Основное окно программы приведено на рисунке 5.1.

Работа в этом окне проводится следующим образом:

- введите значения частот *ωi* и определенные для каждой частоты значения амплитуд и фазовых сдвигов;

 - нажмите на кнопку «Рассчитать P(jω) и Q(jω)», 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 5.1 – Основное окно программы обработки результатов экспериментов

  5.3 Задание к выполнению лабораторной работы

5.3.1 Из папки  Work/Objects5 загрузите свой «объект»  в окно пакета Simulink (по варианту).

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

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

Т а б л и ц а 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ωi | ω1 | ω2 | … | ω10 |
| A(ωi) |   |   |   |   |
| φ(ωi) |   |   |   |   |

5.3.4 Загрузите программу Lab6.

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

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

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

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

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

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

      5.311 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).

 5.4  Требования к отчету

         Отчет по работе должен содержать:

- графики, полученные экспериментально на объекте;

- амплитудные и фазовые характеристики объекта;

- заполненную таблицу 1;

- значения коэффициентов для различных порядков системы;

- результаты имитационных экспериментов на модели.

5.5 Варианты заданий

         Варианты выбираются из папки work\Objects5 пакета Simulink.

         5.6 Контрольные вопросы

      5.6.1  В чем заключаются прямые методы идентификации?

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

         5.6.3 Какие объекты могут быть идентифицированы прямыми  методами?

      5.6.4 Как определяется фазовый сдвиг выходного сигнала?

5.6.5 Как была определена амплитуда выходного сигнала?

5.6.6 Насколько точно определена модель, в чем причина разницы?

**6 Лабораторная работа № 6. Параметрическая идентификация линейных систем**

         Цель работы: освоить методы параметрической идентификации систем, применить для процедуры идентификации возможности командного окна и  графического инструментария идентификации системы MatLab.

6.1 Постановка задачи

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. Критерий качества идентификации характеризует степень адекватности модели объекту в рамках согласованных допущений и ограничений. Очень 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. Оценивание параметров выполняется на основе алгоритма идентификации, определяющего правила поиска оценок.

Наконец, для того, чтобы проверить, насколько точно построенная модель имитирует или предсказывает данные наблюдений, необходимо сравнить их при одинаковых воздействиях. Эта процедура называется *верификацией* модели.

6.2 Идентификация параметров

В лабораторной работе для идентификации линейной системы, используется графическая интерактивная программа (GUI — graphic user interface) *ident*, входящая в состав инструментария идентификации пакета Matlab.

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

Идентификация объекта проводится следующим образом:

а) 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;*

б) 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;

          в) 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);

г) для оценивания параметрической модели в раскрывающемся меню *Estimate* выбирается строка *Parametric model*; задайте структуру модели в появившемся окне (поле *Structure*); откройте окно для редактирования порядка модели (кнопка *Order Editor*), установите значения констант, затем щелкните на кнопку *Estimate*. Результатом выполнения последнего пункта являются оценки соответствующих параметров модели. Для того, чтобы эти оценки отобразить в командном окне Matlab, надо дважды щёлкнуть правой клавишей мыши по пиктограмме модели и в открывшемся окне нажать кнопку “*Present*”.

         6.3 Верификация модели

Для того, чтобы убедиться, насколько точно построенная модель описывает объект, необходимо проанализировать её поведение. Эта процедура называется подтверждением или верификацией модели.

 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.

6.4 Задание к выполнению лабораторной работы

6.4.1 Загрузить в Simulink блок-диаграмму «объекта» из папки Work/Param (по варианту).

6.4.2 Определить входные сигналы и подать их на вход объекта;  рассчитать реакцию (выходные сигналы).

6.4.3  Результаты эксперимента, то есть измеренные значения входных и выходных сигналов  передать в рабочую область программы; для этого вывести сигналы в блоки *To Workspace*, в которых задать имена соответствующих переменных.

6.4.4 Провести параметрическую идентификацию объекта, то есть, оценить неизвестные параметры объекта по имеющимся реализациям входных и выходных сигналов. Оценка проводится по среднеквадратичному критерию, то есть в списке возможных моделей надо выбрать *Parametric Model*.

Для выполнения процедуры идентификации:

- набрать в командном окне команду *ident*;

- в появившемся окне командой *Data-Import* загрузить данные

эксперимента из рабочей области;

- выделить некоторый диапазон экспериментальных данных для

идентификации объекта;

- выберите структуру модели – ARX,  порядок модели - первый, получите оценки параметров модели;

- проведите верификацию модели, используя другой диапазон

экспериментальных данных.

         6.4.5 Отразите параметры восстановленной модели в командном окне программы.

         6.4.6 По рассчитанным параметрам соберите блок-диаграмму модели.

      6.4.7 Подайте на вход модели те же самые сигналы, что и на «объекте» и сравните отклики объекта и модели (в одном смотровом окне).

      6.4.8 Проведите анализ полученных результатов.

  6.5 Требования к отчету

Отчет по работе должен содержать:

- результаты экспериментов на объекте;

- результаты идентификации: искомые параметры модели,

сравнительные графики выходов объекта и модели;

- анализ полученных результатов.

6.6 Варианты заданий

Варианты заданий выбираются из папки *Work/Param* пакета Simulink.

         6.7 Контрольные вопросы

6.7.1 Что такое параметрическая идентификация?

6.7.2 Для чего предназначена графическая оболочка ident?

6.7.3 Что такое оценка параметра?

6.7.4 Какой критерий используется для оценивания параметров

линейной модели?

6.7.5 Что такое верификация модели?

6.7.6 Как можно в пакете ident визуально оценить правильность восстановленной модели?

**7 Лабораторная работа № 7. Непараметрическая идентификация**

**динамических объектов**

Цель работы: освоить методы непараметрической идентификации систем, и пакет подгонки кривых Curve Fitting Toolbox  системы MatLab.

         7.1 Постановка задачи

Непараметрическая модель определяется в общем случае непрерывной функцией, но она может быть задана точками или в виде разложения в ряд по какой-либо системе функций. 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.

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

 .                                                   (7.1)

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



здесь ∆ - интервал дискретизации;

n∆ - дискретное время;

N – количество интервалов.

Это уравнение справедливо для τ = ∆, …, N∆.

         Подставляя в это уравнение N значений τ, получим систему из N уравнений. Это система для определения значений импульсной переходной функции в дискретные моменты ∆, …, N∆.

         Итак, получаем систему уравнений

*Ag = b*                            (7.2)

где  А – квадратная матрица;

                 

b = [b1, b2,…,bm]T -  вектор-столбец с компонентами  bi=Ryx(τ)/∆, i = 1,...,m;

g = [g1, g2,…,gm]T – вектор-столбец искомых значений импульсной переходной функции.

Решение этой системы  позволяет определить дискретные значения ординат весовой функции g(τ)  в точках t, 2t,…, 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.

 7.2 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

                   

         где φk(t) – заданная система функции;

         N – порядок разложения;

         ak – коэффициенты ряда, которые вычисляются по формуле

                    .

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

 7.3 Пакет подгонки кривых Curve Fitting Toolbox

  Пакет подгонки кривых Curve Fitting Toolbox обеспечивает под­гонку кривых с использованием метода наименьших квадратов. 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; Ploting... - 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):

- Custom equations — функция регрессии, задаваемая пользователем;

- Exponential — экспоненциальная репрессия;

- Fourier — приближение рядом Фурье;

- Gaussian — приближение кривой гауссиана;

- interpolant — интерполяция с выбором метода;

- Polynomial — полиномиальная регрессия;

- Power — степенная регрессия;

- Rational — регрессия рациональной функцией с выбором ее числителя и знаменателя;

- Smoothing Spline — регрессия сглаживающими сплайнами;

- Sum of Sin Func — регрессия суммой синусоидальных функций;

  - Weibull — регрессия кривой Вейбулла.

 Как видно из этого перечисления, представлен внушительный набор возмож­ных видов приближений. Особенно важно, что среди них 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.

7.4 Задание на лабораторную работу

   Зарегистрированы входные и выходные переменные исследуемого объекта в течение определенного интервала времени. По результатам этих измерений вычислены автокорреляционная и взаимнокорреляционная функции (см. таблицу вариантов). Требуется определить численным методом из уравнения Винера-Хопфа импульсную переходную функцию. Затем полученные дискретные значения этой функции следует аппроксимировать полиномами различных порядков и выбрать наилучшую степень аппроксимации.

7.5 Порядок выполнения работы

7.5.1 Записать систему линейных алгебраических уравнений (7.2) для своего варианта.

         7.5.2 Решить  систему (7.2), используя командное окно системы MatLab:

         -  введите компоненты матрицы A и вектора b в командное окно системы;

         - проверьте существование решения системы (7.2), набрав в командной строке det(A); как известно, для существования решения необходимо, чтобы  det(A) ≠ 0;

         - получите решение системы, применив процедуру g = b\*inv(A).

7.5.3 Используя пакет Curve Fitting Tool, выполните аппроксимацию полученных дискретных значений полиномами различных порядков. 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.

7.5.4 Завершив подгонку, выведите графики исходных то­чек, кривых приближения и погрешностей.

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

7.6 Требования к отчету

         Отчет по работе должен содержать:

- систему алгебраических уравнений для определения дискретных значений импульсной переходной функции;

- результат решения этой системы;

- результаты подгонки;

- обоснованный выбор наилучшего аппроксимирующего полинома.

7.7 Варианты заданий

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант 1 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 | t, мин | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ∞ |
|   | 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 |

7.8 Контрольные вопросы

7.8.1 В чем суть непараметрической идентификации?

7.8.2  Какие объекты описывает уравнение Винера-Хопфа,

7.8.3 Почему возможно представить уравнение Винера-Хопфа системой

алгебраических уравнений?

7.8.4 Что такое аппроксимация функции?

7.8.5 Объясните назначение пакета Curve Fitting Tool.

       7.8.6 Какая из функций дает лучшие результаты приближения?

**Список литературы**

1.  Дейч А.М Методы идентификации динамических объектов.-

М.: Энергия, 1979.

2. Советов Б.Я., Яковлев С.А. Моделирование систем. - М.: Высшая школа, 1991.

3. Лазарев Ю. MatLab 5.x. – Киев: «Ирина», BHV, 2000.

 4. Дьяконов В. П. MatLab 6/6.1/6.5 + Simulink 4/5. Основы применения. –

М.: Солон-ПРЕСС,  2004.

5.  Matlab 6.5 SP1/7.06. Simulink 5/6 в математике и моделировании. – М.: СОЛОН-Пресс, 2005.

6. Ибраева Л.К.Основные приемы работы в среде Matlab. Методический практикум. – Алматы: АИЭС, 2004.

        7. [www.exponenta.ru](http://www.exponenta.ru/)

        8. [www.matlab.ru](http://www.matlab.ru/)

Лида Куандыковна Ибраева