# Лабораторная работа

# «Исследование нелинейных систем методом фазовых портретов»

# 1. Краткие теоретические сведения

*Динамическая система* – система произвольной природы (физической, химической, биологической, социальной, экономической и т.д.), для которой однозначно определено понятие состояния, и это состояние меняется во времени согласно некоторому детерминированному закону*.*

Состояние системы определяется совокупностью некоторых независимых величин. Эти величины принято называть *переменными состояния, динамическими переменными, фазовыми* *переменными.* Состав переменных состояния определяется спецификой конкретной динамической системы (ДС). Например, переменными состояния механической системы являются мгновенные значения положений и скоростей всех составляющих ее материальных точек относительно выбранной системы отсчета; для электрического устройства – силы токов в его цепях или падения напряжений на отдельных элементах, для химической реакции – концентрации реагентов и т.д.

Число переменных состояния отождествляется с порядком системы, т.е. вектор состояния имеет определенную размерность: dim **x**=*n* . Геометрическое пространство в ортогональном базисе переменных состояния называется *пространством состояния* или *фазовым пространством*. Мгновенному состоянию ДС, т.е. состоянию в отдельно выбранный момент времени, соответствует точка в его пространстве состояний, которую принято называть *изображающей точкой (ИТ) или фазовой точкой*. При изменении переменных состояния во времени ИТ совершает движение в пространстве состояний. Линию в пространстве состояний, которая связывает начальное состояние ДС со всеми ее последующими состояниями с течением времени («след» ИТ) принято называть *фазовой траекторией*. Совокупность фазовых траекторий, полученных при различных начальных условиях, есть *фазовый портрет* ДС.

*Состоянием равновесия* ДС называют такое ее состояние, при котором переменные состояния системы не изменяются во времени. Каждому состоянию равновесия ДС соответствует точка в ее пространстве состояний, которую обычно называют точкой равновесия, неподвижной точкой или особой точкой.

ДС сопоставляется ее *математическая модель*, т.е. некоторое формальное математическое описание закона изменения состояния системы во времени. Можно рассматривать математическую модель ДС как *оператор эволюции*, устанавливающий соответствие между состоянием системы в начальный момент времени с состояниями в последующие моменты времени. Математическая модель может быть задана различными способами: в виде системы дифференциальных, интегральных и интегро-дифференциальных уравнений, дискретных отображений, графов и т.д.

Если состояние системы и оператор эволюции определены для любого момента времени, то ее называют *системой с непрерывным временем* или *непрерывной системой*. Часто непрерывную систему называют *потоком*. Такое понятие основано на аналогии движения системы с течением жидкости: совокупность движущихся ИТ системы можно рассматривать как совокупность движущихся молекул жидкости, т.е. поток. Математическая модель непрерывных систем обычно задается в виде системы обыкновенных дифференциальных уравнений:

|  |  |
| --- | --- |
|  | (1) |

или в компактной векторной форме

|  |  |
| --- | --- |
|  | (2) |

Данная модель характеризует изменение переменных состояния во времени и обычно называется *моделью в переменных состояния.* Множество точек равновесия непрерывной ДС определяется из очевидного условия ***x**i*** = 0, *i* =1*n* и является решением системы алгебраических уравнений:

|  |  |
| --- | --- |
|  | (3) |

Точки равновесия можно расценивать как «организующие центры» динамики системы в пространстве состояний. Таким образом, определив эти точки и оценив характер фазовых траекторий в их окрестности нередко (но далеко не всегда!) можно построить приблизительный фазовый портрет и тем самым получить общую картину поведения ДС.

# 2. Техника моделирования динамических систем в MATLAB

Аналитическое исследование динамики нелинейных систем ограничено возможностями существующего математического аппарата. Поэтому в большинстве случаев исследователям приходиться обращаться к численным методам и полагаться на корректность средств компьютерного моделирования.

Далее будут рассмотрены вопросы компьютерного моделирования динамических систем (ДС) в популярном пакете математических вычислений MATLAB. Акцент, прежде всего, будет сделан на синтаксисе основных команд, используемых при компьютерном моделировании ДС и рациональной организации вычислительного процесса. В качестве иллюстрации будут приведены фрагменты программ. Предполагается, что представленные сведения послужат необходимым минимумом для успешного решения различных задач компьютерного моделирования динамики систем.

## Общие сведения об организации вычислительного процесса в MATLAB

Математические вычисления в MATLAB можно проводить либо в диалоговом режиме простых вычислений путем записи необходимых выражений в командную строку рабочего окна, либо в программном режиме, когда перечень необходимых действий формируется в виде соответствующей программы. Программы MATLAB хранятся в специальных файлах, имеющих расширение \*.m (m-файлах) и уникальное имя, начинающееся с латинской буквы, не содержащее букв кириллицы, специальных символов и пробелов. Для подготовки, редактирования и отладки m-файлов служит специальный многооконный редактор. Он выполнен как типичное приложение Windows. Для создания нового m-файла нужно воспользоваться меню интерфейса, выбрав File –> New –> m-file/

M-файлы делятся на два класса: файлы-сценарии и файлы-функции.

*Файл-сценарий,* именуемый также Script-файлом, является просто записью серии команд без входных и выходных параметров. Для запуска файла-сценария из командной строки MATLAB достаточно указать его имя в этой строке и нажать клавишу ввода, либо воспользоваться «горячей клавишей» F5 из окна редактора. Переменные, используемые в файлах-сценариях, являются глобальными, т. е. они действуют одинаково в командах сессии и внутри файла-сценария.

*Файл-функция* является типичным объектом языка программирования системы MATLAB. Одновременно он является полноценным модулем с точки зрения структурного программирования, поскольку содержит входные и выходные параметры и использует аппарат локальных переменных. Структура такого модуля с одним выходным параметром выглядит следующим образом:

function var=name(Cnиcoк формальных napaмeтpов) Тело файла с любыми выражениями vаг=выражение

Таким образом, файл-функция всегда начинается с объявления function, после которого указывается имя выходной переменной var, имя самой функции и список ее входных параметров. Если выходных параметров больше, то они указываются в квадратных скобках после слова function. Функция возвращает свое значение и может использоваться в виде name (Список фактических параметров) в математических выражениях и вызываться как в командном режиме, так и из других m-файлов. Вызов функции происходит по уникальному имени соответствующего m-файла, а не по имени, указанному в заголовке функции (в данном случае name).

Практически любой вычислительный эксперимент предполагает выполнение последовательности действий, связанных с формированием исходных данных, проведением математических расчетов и получением результата. В этой связи рационально организовать эту последовательность в виде соответствующего файла-сценария, в котором при необходимости осуществляется вызов встроенных функций MATLAB или функций пользователя.

## Особенности компьютерного моделирования динамических систем

Основной целью компьютерного моделирования ДС, очевидно, является оценка временной эволюции системы, т.е. выявление характера изменения ее переменных состояния во времени. Таким образом, происходит восстановление динамики системы по ее математической модели. В случае непрерывных ДС этот процесс фактически выливается в процедуру численного интегрирования дифференциальных уравнений при конкретных начальных условиях на заданном временном интервале. Для дискретных систем организуется итерационная процедура, позволяющая вычислять последующее состояние системы на основании текущего состояния по заданной рекуррентной форме. Результаты вычислений необходимо представить в наглядной и удобной для анализа форме. Наибольшей иллюстративностью в этом плане обладают соответствующие графики – фазовый портрет системы (для систем 2-го и 3-го порядка) и графики изменения переменных состояния во времени (графики переходных процессов).

Отмеченные особенности моделирования ДС вызывают необходимость введения в структуру файла-сценария трех основных блоков: блока формирования исходных данных, блока вычисления и блока визуализации. В блоке исходных данных задаются значения параметров системы, ее начальное состояние и время моделирования, в вычислительном блоке на основании численной процедуры формируется массив переменных состояния на заданном временном интервале, а блок визуализации организует построение необходимых графиков.

*Модель ДС* формируется в отдельном файле-функции, который вызывается основной программой по его имени. Модель непрерывной системы представляется в виде модели в переменных состояния, т.е. в виде правых частей системы обыкновенных дифференциальных уравнений и записывается в соответствующем m-файле, т.н. ODE функции.

Рассмотрим пример создания ODE функции для простого математического маятника, динамика которого описывается следующей моделью в переменных состояния:

|  |  |
| --- | --- |
|  |  |

где *x*1 – угол отклонения маятника от вертикальной оси, *x*2 – угловая скорость вращения маятника относительно точки подвеса, *g* – ускорение свободного падения, *l* – длина маятника, *k* – коэффициент трения.

Файл-функция, соответствующий этой модели, организуется следующим образом.

% pendulum.m function y = pendulum(t, x)

g=9.81; l=1; k=0.01;

y = [x(2); -g/l\*sin(x(1))-k\*x(2)];

Помимо переменных состояния модель ДС часто содержит параметры, т.е. условно постоянные величины. В представленном выше примере параметрами модели маятника являются ускорение свободного падения, длина маятника и коэффициент трения. Параметры системы могут задаваться в том же файле, что и модель или в файле-сценарии. В последнем случае они передаются в файл-функцию, если они объявлены глобальными переменными. Для этого в начале файла сценария, и каждого из используемых файлов-функций записывается команда:

global a1 a2 a3 a4 …….

Целью моделирования системы может являться так называемый параметрический анализ, заключающийся в оценке влияния изменения параметров на общую динамику системы. Эта возможность должна быть предусмотрена при организации вычислительной процедуры. Значения параметров могут быть заданы и изменены как в самой программе путем присвоения им численного значения (например, l=3), так и с помощью команды интерактивного ввода input. Например, для интерактивного ввода значения длины маятника можно воспользоваться командой:

l=input(‘Введите значение длины маятника l=’)

После выполнения этой команды в командном окне MATLAB появится соответствующий запрос. Рационально задавать все параметры ДС в начале файла-сценария и при необходимости изменять их значение в теле этого файла с помощью соответствующих команд присвоения или интерактивного ввода.

Численное интегрирование ДС в MATLAB осуществляется с помощью решателей, каждый из которых реализует определенный алгоритм численного интегрирования системы обыкновенных дифференциальных уравнений. Под численным интегрированием системы обыкновенных дифференциальных уравнений (ОДУ) понимают нахождение приближенного решения задачи Коши, т.е. приближенных значений функций *xi* (*t*) на заданном временном интервале. Применение того или иного решателя определяется свойствами ДС и требованиями к точности полученных результатов и быстродействию вычислительной процедуры. В MATLAB используются различные решатели. Для начальной пробы можно порекомендовать решатель ode45, реализующий одношаговый явный метод Рунге-Кутта 4-го и 5-го порядка и обычно демонстрирующий хорошие результаты.

Вызов решателя, осуществляющего выполнение процесса численного интегрирования системы ОДУ, производится командой

[t,x] = solver(@Fname,tspan,x0)

Здесь: solver – имя решателя (например, ode45), Fname – имя файла-функции, содержащего модель динамической системы; tspan – интервал времени интегрирования, задаваемый в форме вектора-строки [tmin tmax]; x0 – вектор начальных условий, число элементов которого должно совпадать с размерностью системы уравнений; t – выходной массив времени (вектор-столбец); x – выходной массив переменных состояния системы, в общем случае представляющий собой матрицу с количеством столбцов равным порядку системы.

Например, для численного интегрирования маятника на временном интервале от 0 до 20 секунд при единичных начальных условиях следует воспользоваться командой:

[t,x] = ode45(@pendulum,[0,20],[1,1])

Результатом выполнения этой команды будут вектор столбец t и матрица x с двумя столбцами, в каждом из которых расположены значения соответствующей переменной на заданном временном интервале. Необходимо заметить, что число элементов массива времени и число строк массива переменных состояния заранее неизвестно, а определяется самой процедурой численного интегрирования. Соответствие времени и значений переменных задается построчно, т.е. в i-ой строке массива x находятся значения переменных в момент времени t[i].

Интервал времени интегрирования tspan и вектор начальных условий x0 могут быть заданы непосредственно в команде вызова решателя или перед ней с помощью соответствующих команд присвоения или интерактивного ввода. Для двухмерных систем начальные условия удобно задавать в поле фазового портрета с помощью команды графического ввода данных ginput(1). Эта команда позволяет вводить данные с помощью мыши из активного графического окна. Для этого сначала создается графическое окно, определяется диапазон изменения переменных в виде пределов осей абсцисс и ординат графика. После этого непосредственно используется команда графического ввода. После ее вызова в текущих осях появляется перекрестие, управляемое движением мыши. Щелчок левой кнопки мыши в нужной точке плоскости, формирует вектор с координатами [x y], т.е. вектор начальных условий. Данный алгоритм действий реализуется следующей последовательностью команд:

figure(1)

axis([-x1max x1max -x2max x2max]); x0 = ginput(1)

Двумя основными формами графического представления динамики системы являются графики переходных процессов переменных состояния, т.е. графики функций *xi* (*t*) , полученные при определенных начальных условиях, и ее фазовый портрет как совокупность фазовых траекторий, полученных для набора начальных условий. Построение графиков переходных процессов не требуется ничего большего, кроме как правильного использования команды plot. Например:

plot(t,x(:,1)) – график функции *x*1(*t*) ; plot(t,x(:,2)) – график функции *x*2(*t*); plot(t,x(:,1),’r’ t,x(:,2),’b’) – графики функций *x*1(*t*) и *x*2(*t*) в одних координатных осях красным и синим цветом соответственно;

Одна фазовая траектория также строится весьма просто:

plot(x(:,1),x(:,2)) – график траектории на фазовой плоскости; plot3(x(:,1),x(:,2),x(:,3))) – график траектории в трехмерном фазовом пространстве.

Для получения фазового портрета необходимо производить численное интегрирование системы на множестве начальных условий и строить соответствующие фазовые траектории. Таким образом, в файле-сценарии должна быть организована процедура многократного вызова решателя и применения команды plot. Разумеется многократный запуск программы на решение – процесс весьма утомительный. Его можно успешно избежать, если использовать в теле файла-сценария операторы цикла.

Любая фазовая траектория характеризуют движение состояния системы во времени. Очень наглядно видеть это движение при визуализации результатов. Для отображения движения точки на плоскости или в трехмерном пространстве в MATLAB есть команды comet и comet3. Движущая точка представляется ядром кометы с выделенным цветом хвостом. Синтаксис этой команд для построения фазовых траекторий:

comet(x(:,1),x(:,2)) comet3(x(:,1),x(:,2), ,x(:,3))).

Вот фактически все минимальные сведения для создания программ моделирования ДС. Структура файла-сценария задается разработчиком в соответствии с целями вычислительного эксперимента.

## Пример программы моделирования непрерывных систем в MATLAB

Представленный ниже программный модуль может использоваться при моделировании непрерывных систем 2-го порядка. Вычислительный процесс организован таким образом, что при вызове программы создается графическое окно с пустыми координатными осями фазового портрета и активным перекрестием мыши. Щелчком мыши задаются начальные условия, начинается процесс моделирования, в результате которого в текущем окне строится фазовая траектория, соответствующая данным начальным условиям *x*2(*x*1), а в другом окне графики переходных процессов *x*1(*t*) и *x*2(*t*). В теле программы с помощью команды while 1…end организован бесконечный цикл. Таким образом, описанная процедура задания начальных условий, численного интегрирования и построения графиков будет повторяться столько раз, сколько это необходимо пользователю. Прерывание программы производится в начале цикла (при появлении перекрестия мыши) щелчком вне поля координатных осей (серая область графического окна). В примере для конкретности используется модель и параметры простого маятника. Чтобы промоделировать другую ДС, нужно создать соответствующий файл-функцию с моделью системы, записать его под уникальным именем, которое вызывается решателем, и изменить список параметров. Текст приведенной ниже программы сопровождается комментариями, начинающимися с символа % и дающие описание используемых команд и их блоков.

% *simoDS2 – программа моделирования непрерывной ДС 2-го порядка* close all, clear all % *Закрытие графических окон и очистка рабочей области памяти*

while 1 % *Организация бесконечного цикла* figure(1) %

% *Задание пределов осей графического окна (поля фазового портрета)* x1max=5; x2max=5;

axis([-x1max x1max -x2max x2max]);

% *Построение сетки, обозначение осей, заголовка графика и фиксация % свойств графического окна*

grid on

xlabel('x1');

ylabel('x2');

title('Phase portrait')

hold on

% *Блок численного интегрирования*

x0 = ginput(1) % *Задание начальных условий*

% *Организация прерывания*

if abs(x0(1)) > x1max | abs(x0(2)) > x2max, break, end tmax=20; % *Время интегрирования*

[t,x]=ode45(@pendulum,[0 tmax],x0); % *Вызов решателя*

% *Блок визуализации вычислений* figure(1)

*% Визуализация точки старта траектории (начального состояния)*

start= line('color',[0 0 1],'Marker','.','markersize',14,'xdata',x0(1),'ydata',x0(2));comet(x(:,1),x(:,2)) % *Анимация фазовой траектории* plot(x(:,1),x(:,2),'b') % *Построение фазовой траектории*

% *Построение графиков переходных процессов переменных состояния*

figure(2)

for i=1:2

subplot(2,1,i),plot(t,x(:,i),’b’) xlabel('t');

grid on

if i==1

ylabel('x1');

 else

 ylabel('x2');

 end

 end

end % *Конец основного цикла*

Представленная программа может использоваться при моделировании ДС более высокого порядка. В этом случае уже невозможно использовать команду графического ввода, и начальные условия задаются численно, например, введением в начале цикла команды интерактивного ввода:

x0 =input(‘Initial conditions x0=’);

Прерывание цикла можно организовать путем соответствующего запроса в его конце или начале. Программа прервется по ошибке, если вместо задания вектора начальных условий просто нажать клавишу ввода. Блок визуализации в зависимости от желания пользователя может содержать различные команды двухмерной и трехмерной графики.

## 3. Задание к лабораторной работе

1. Создать файл-функцию с моделью непрерывной динамической системы (таблица 1) и файл сценарий для ее компьютерного моделирования в пакете MATLAB.
2. Провести моделирование непрерывной динамической системы, получить ее фазовый портрет и графики изменения переменных состояния во времени.

