|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное**  **учреждение**  **высшего образования**  **«Московский государственный технический университет имени Н.Э. Баумана**  **(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

## КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/05 Современные интеллектуальные**

## программно-аппаратные комплексы.

О Т Ч Е Т

# по лабораторной работе № 3

**Дисциплина:** Системный анализ в управлении

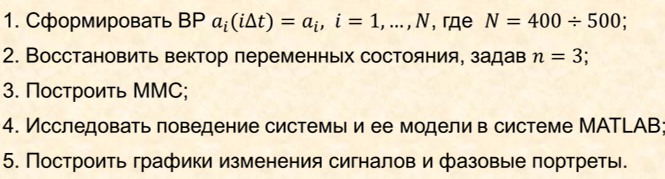
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-41М |  |  |  | И.С. Марчук |
|  | (Группа) |  | (Подпись, дата) |  | (И.О. Фамилия) |
| Преподаватель |  |  |  |  | Д.А. Миков |
|  |  |  | (Подпись, дата) |  | (И.О. Фамилия) |

Москва, 2025

Тема: реконструкция математической модели системы по неполным данным.

Цель работы: исследование алгоритма реконструкции математической модели сложной системы по временному ряду.

Задание: выполнить реконструкцию математической модели системы по временному ряду.



Исходные данные: в качестве регистрируемого сигнала α(t) взять любую гладкую математическую функцию. В результате зависимоть вида y = f(x) можно представить в виде набора точек – временного ряда.

Ход работы:

В качестве гладкой математической функции, описывающей систему по временному ряду, согласно варианту задания, была выбрана следующая функция:



Сформируем временной ряд с функцией y(x)=7sin(x)+1.5

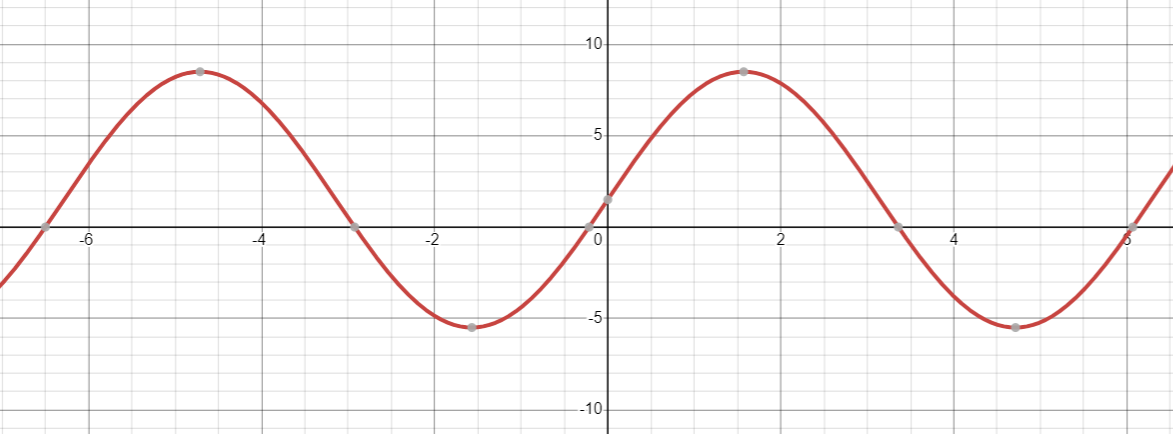


Рисунок 1 – формирование исходного временного ряда y(x).

Восстановление вектора переменного состояния (n = 3)

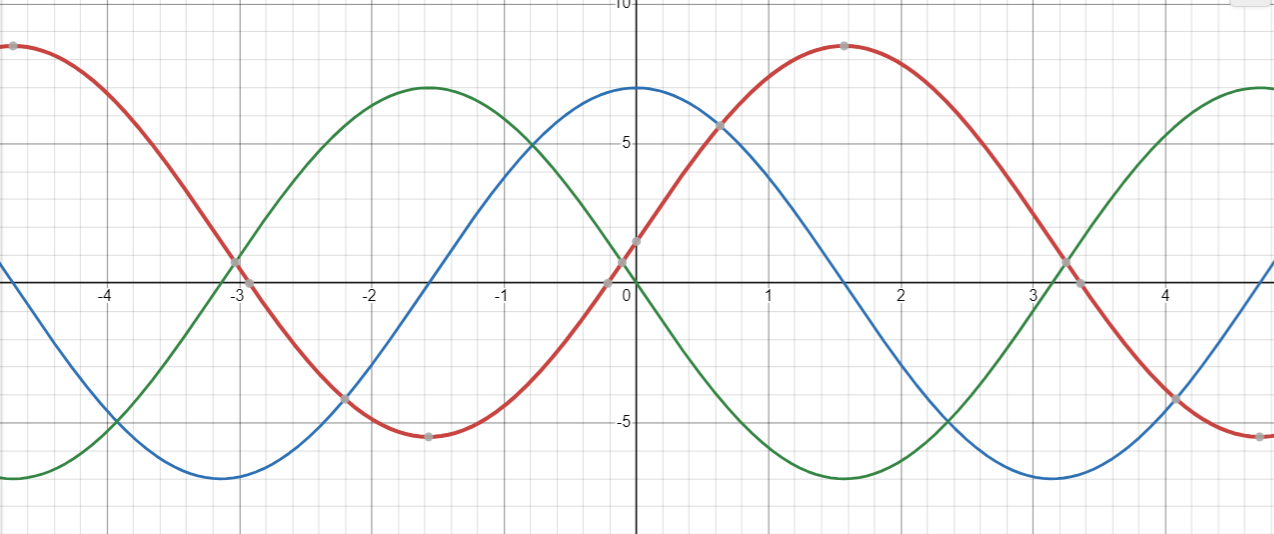


Рисунок 2 – Восстановление вектора переменного состояния (красным – f(x), синим – f’(x), зеленым – f’’(x)).

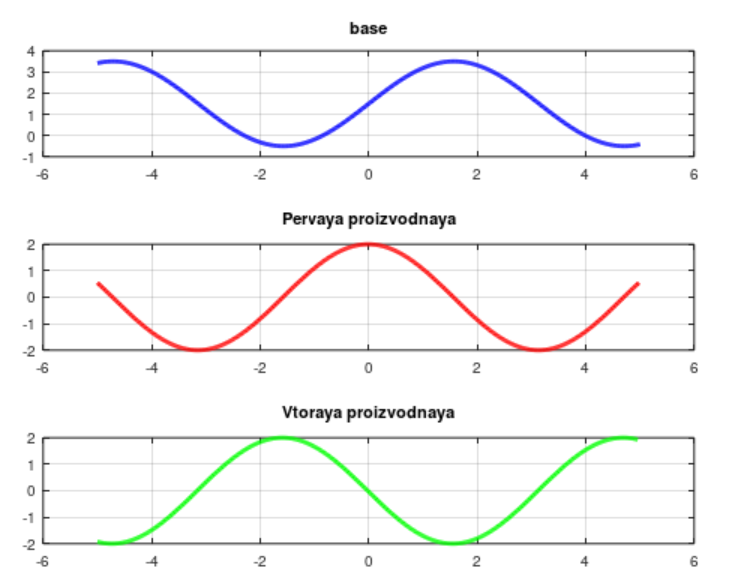


Рисунок 3 – Восстановление вектора переменного состояния в MathLab

Построение ММС

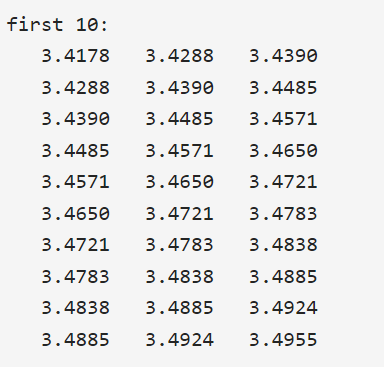


Рисунок 3 – матрица временных срезов.

Исследование поведение системы и ее модели в системе MATLAB через ODE45.

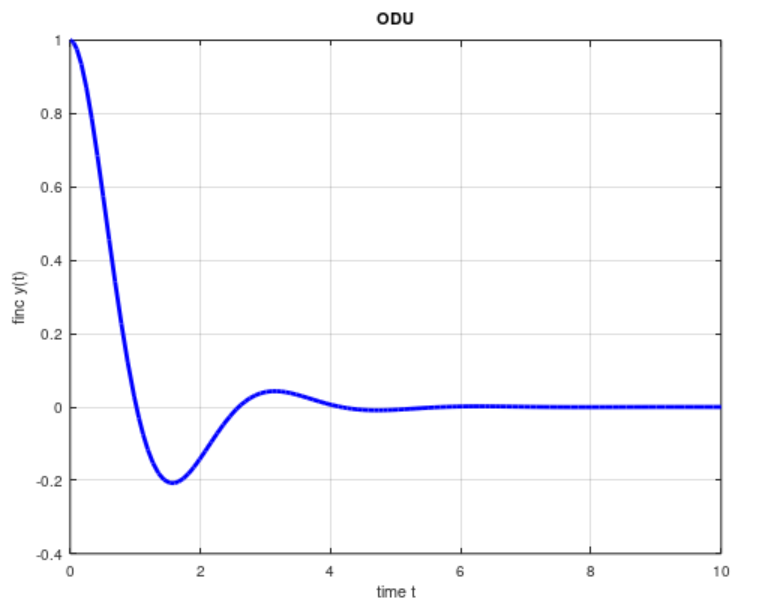


Рисунок 4 – поведение функции сигнала.

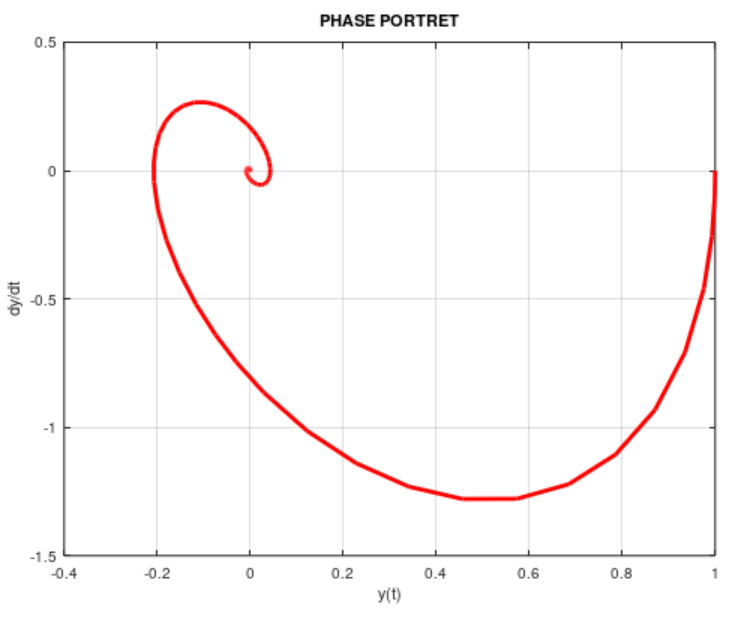


Рисунок 5 – фазовый портрет.

Листнинг кода генерации временного ряда:

% Параметры временного ряда

N = 500; % Количество точек

x = linspace(-5, 5, N); % Временная шкала

% Исходный сигнал (гладкая функция)

y = 2\*sin(x) + 1.5;

% Построение графика временного ряда

figure;

plot(x, y, 'b', 'LineWidth', 2);

grid on;

xlabel('Time');

ylabel('Signal y(x)');

title('base function');

Листнинг кода восстановления вектора переменного состояния:

% Параметры временного ряда

N = 500; % Количество точек

x = linspace(-5, 5, N); % Временная шкала

% Исходный сигнал

y = 2\*sin(x) + 1.5;

% Численное вычисление производных (метод конечных разностей)

dx = x(2) - x(1);

y1 = y; % Исходный сигнал

y2 = diff(y) / dx; % Первая производная

y3 = diff(y2) / dx; % Вторая производная

% Коррекция длины осей для отображения

x2 = x(1:end-1); % Укоротили на 1 точку

x3 = x(1:end-2); % Укоротили на 2 точки

% Построение графиков

figure;

subplot(3,1,1);

plot(x, y1, 'b', 'LineWidth', 2);

title('base');

grid on;

subplot(3,1,2);

plot(x2, y2, 'r', 'LineWidth', 2);

title('Pervaya proizvodnaya');

grid on;

subplot(3,1,3);

plot(x3, y3, 'g', 'LineWidth', 2);

title('Vtoraya proizvodnaya');

grid on;

Листнинг кода генерации матрицы временных срезов:

% Параметры временного ряда

N = 500; % Количество точек

x = linspace(-5, 5, N); % Временная шкала

% Исходный сигнал (гладкая функция)

y = 2\*sin(x) + 1.5;

% Построение графика временного ряда

figure;

plot(x, y, 'b', 'LineWidth', 2);

grid on;

xlabel('Time');

ylabel('Signal y(x)');

title('base function');

% Формирование матрицы временных срезов

y1 = y(1:end-2);

y2 = y(2:end-1);

y3 = y(3:end);

MTC = [y1', y2', y3'];

% Вывод первых 10 строк MTC

disp('first 10:');

disp(MTC(1:10, :));

Листнинг кода поведения функции сигнала и фазового портрета:

function main

% Интервал времени моделирования

tspan = [0 10];

% Начальные условия [y(0), dy/dt(0)]

y0 = [1; 0];

% Установка параметров точности ODE-солвера

options = odeset('RelTol', 1e-6, 'AbsTol', 1e-9);

% Решение системы ОДУ методом Рунге-Кутта (ODE45)

[t, Y] = ode45(@marchuk\_model, tspan, y0, options);

% Построение графика решения ОДУ

figure;

plot(t, Y(:,1), 'b', 'LineWidth', 2);

grid on;

xlabel('time t');

ylabel('finc y(t)');

title('ODU');

% Построение фазового портрета

figure;

plot(Y(:,1), Y(:,2), 'r', 'LineWidth', 2);

grid on;

xlabel('y(t)');

ylabel('dy/dt');

title('PHASE PORTRET');

% Вложенная функция

function dydt = marchuk\_model(~, y)

dydt = zeros(2,1);

dydt(1) = y(2);

dydt(2) = -2\*y(2) - 5\*y(1);

end

end

**Вывод**

Графики исходного сигнала и модели схожи при малых периодах и расходятся с течением времени. Сходство фазовых портретов подтверждает высокую точность построенной модели. Результат реконструкции зависит от формы исходного сигнала (в данном случае удачно выбрана функция) и выбора параметров реконструкции: размерности вектора состояния (разбиение вектора на 500 дает достаточную точность) и степени полинома. Чем больше n и ν, тем точнее результат.