**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра КСУ**

отчет

**по идз №5**

**по дисциплине «ПРОЕКТИРОВАНИЕ**

**ОПТИМАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ»**

Тема: Максимальное быстродействие

**Вариант 12**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4491 | Пономарев Д.А. |  |
| Преподаватель | Ветчинкин А.С. |  |

Санкт-Петербург

2018

**Максимальное быстродействие**

**Исходные данные**

По каждому варианту необходимо оценит сложность решения задачи максимального быстродействия аналитическим методом, и получить решения поисковым и графическими методами.

Для всех вариантов граничные значения состояний объекта управления , модуль максимального значения управления равен 1.

Исходные данные заданы в таблице 1.

Таблица 1. Исходные данные к заданию

|  |  |
| --- | --- |
| Вариант | Матрицы объекта управления и полином |
| 12 |  |

**Аналитическое решение задачи**

1. Запишем исходную систему уравнений в общем виде:

2. Гамильтониан в общем виде:

3. Определим зависимость оптимального управления от переменных сопряженной системы уравнений в общем виде:

4. Сформируем систему сопряженных уравнений в общем виде:

5. Итоговая система в общем виде:

6. Итоговая система для варианта задания:

7. Анализируем. Видим, что искомое управляющее воздействие имеет вид прямоугольной волны, кото-рая меняет знак не более одного раза. Очевидно, что момент смены знака управления (момент переключения) должен выбираться из условия обес-печения заданных граничных условий для состояний объекта управления. для определения моментов переключения может быть использовано не-сколько способов.

**Численное решение задачи с помощью Matlab**

Программа будет состоять из четырех файлов: main.m, ode\_fun.m, calculate\_error.m и calculate\_u.m.

В первом файле main.m, код которого представлен на рисунке 1, происходит 3 действия: ввод исходных данных системы, численное решение задачи с помощью функции fminsearch, аргументом которой является ссылка на функцию calculate \_error(), а также аналитическое решение с помощью функции dsolve.

|  |
| --- |
| clear; close all; clc;  global A B X\_START X\_END TIME\_END  X\_START = [1 0]; X\_END = [0 0]; TIME\_END = 2;  A = [-1 2; 0 -1]; B = [1 1];    % Numerical solution:  ksi0 = [1 1];  fminsearch('calculate\_error', ksi0)    % Analytical solution:  S = dsolve('Dx1=-x1+2\*x2+0.5\*ksi1+0.5\*ksi2', ...  'Dx2 = -x2 + 0.5\*ksi1+ 0.5\*ksi2', ...  'Dksi1 = ksi1', ...  'Dksi2 = -2\*ksi1 + ksi2', ...  'x1(0) = 1', x2(0) = 0', 'x1(2) = 0', 'x2(2) = 0');  S.x1, S.x2, S.ksi1, S.ksi2 |

Рисунок 1 – Код скрипта main.m

Второй файл calculate\_error.m содержит код функции, ссылка на которую передается аргументом в функцию fminsearch(). Код данного файла представлен на рисунке 2. В данной функции происходит вызов функции ode45, построение графика переходных процессов системы уравнений, а также расчет ошибки, которая вычисляется как сумма квадратов разностей желаемых значений переменных x в момент времени 2 и фактических значений.

|  |
| --- |
| function error = calculate\_error(ksi0)  global X\_START X\_END TIME\_END  [t, x] = ode45('ode\_fun', [0 TIME\_END], [X\_START ksi0]);    error = [x(end, 1) - X\_END(1), x(end, 2) - X\_END(2)];  error = error(1)^2 + error(2)^2;    plot(t, [x calculate\_u([x(:,3) x(:,4)])])  legend('x1', 'x2', 'ksi1', 'ksi2', 'u')  grid on; xlabel('t'); ylabel('x, ksi, u');  pause(0.1)  end |

Рисунок 2 – Код файла calculate\_error.m

Третий файл ode\_fun.m, содержит в себе код функции, ссылка на которую передается в фунцию ode45(). Функция ode\_fun() вычисляет правые значения системы из четырех уравнений. Ее код представлен на рисунке 3.

|  |
| --- |
| function dxdt = ode\_fun(t, x)  global A B  u = calculate\_u([x(3), x(4)]);  dxdt = [A(1,1)\*x(1) + A(1,2)\*x(2) + B(1)\*u; ...  A(2,1)\*x(1) + A(2,2)\*x(2) + B(2)\*u; ...  -A(1,1)\*x(3) - A(2,1)\*x(4); ...  -A(1,2)\*x(3) - A(2,2)\*x(4)];  end |

Рисунок 3 – Код файла ode\_fun.m

В последнем файле осуществляется вызов функции calculate\_u, которая возвращает значение оптимального управления по значению сопряженных переменных. Ей код представлен на рисунке 4.

|  |
| --- |
| function u = calculate\_u(ksi)  global B  u = 0.5\*(ksi(:,1).\*B(1) + ksi(:,2).\*B(2));  end |

Рисунок 4 – Код файла calculate\_u.m

График переходных процессов и управляющего воздействия представлен на рисунке 5.

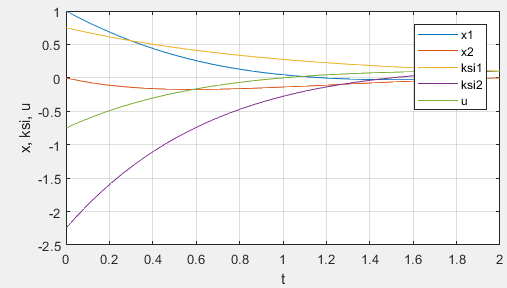


Рисунок 5 – График переходных процессов и управляющего воздействия

Начальные значения представлены на рисунке 6.

|  |
| --- |
| -0.1073 -0.2225 |

Рисунок 6 – Начальные значения сопряженных переменных

**Аналитическое решение задачи с помощью Matlab**

Аналитическое решение, полученное с помощью функции dsolve, представлено на рисунке 7.

|  |
| --- |
| ans =  (2\*exp(t))/(exp(4) + 3) - (2\*(exp(-t)/2 + t\*exp(-t)))/(exp(4) + 3) - ((exp(4) - 1)\*(exp(-t)/4 + (t\*exp(-t))/2 + (t^2\*exp(-t))/2))/(exp(4) + 3) + (exp(-t)\*(5\*exp(4) + 7))/(4\*(exp(4) + 3))    ans =  exp(t)/(exp(4) + 3) - exp(-t)/(exp(4) + 3) - (t\*exp(-t)\*(exp(4) - 1))/(2\*(exp(4) + 3))    ans =  (4\*exp(t))/(exp(4) + 3)    ans =  -(exp(-t)\*(exp(4) - 1))/(exp(4) + 3) |

Рисунок 7 – Результат работы программы

**Вывод**

В ходе работы было определено оптимальное управляющее воздействие как функция времени

построены графики оптимального управляющего воздействия и состояний объекта управления (рисунок 5), при этом задача решена двумя способами (численным и аналитическим методом).