**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 |  |

**Анализ задачи**

Специфика задач на максимальное быстродействие начинает сказываться при записи критерия качества. Для этих задач критерием качества является следующий функционал (5.1)

 (5.1)

Таким образом, требуется найти такое управление, при котором перевод объекта управления из начального состояния в конечное выполняется за минимально возможное время.

Последовательность решения рассматриваемых задач не отличается от процедуры решения других задач, решаемых на основе принципа максимума.

При рассмотрении объектов управления, описываемых линейными уравнениями, задачи максимального быстродействия имеют некоторую особенность. Дело в том, что соответствующая этим задачам функция Гамильтона содержит управление в степени не выше первой и, следовательно, определение максимального значения гамильтониана не может быть выполнено путем приравнивания нулю его первой производной по управлению. Поиск максимального значения гамильтониана в этом случае производится путем анализа возможных комбинаций между управлением и переменными сопряженной системы уравнений. При этом оказывается, что оптимальное управление должно быть максимально по модулю внутри интервала управления и в некоторых его точках мгновенно менять знак в соответствии со знаком некоторой функции от сопряженных переменных. В условиях такого слабого влияния сопряженной системы уравнений на управляющее воздействие возникает возможность вообще отказаться от решения сопряженной системы уравнений и рассматривать моменты смены знака управления (моменты переключения) как самостоятельные переменные.

**Способ 1. 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. Анализируя возможные комбинации значений и u можно сделать вывод о том, что для обеспечения максимальной величины Гамильтониана в зависимости от управления необходимо выполнение следующего соотношения:

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, которая меняет знак не более одного раза. Очевидно, что момент смены знака управления (момент переключения) должен выбираться из условия обеспечения заданных граничных условий для состояний объекта управления.

где Heaviside – функция Хевисайда, - момент переключения.

8. Выполним преобразование по Лапласу

9. Решение системы, полученной после преобразования по Лапласу

10. Найдем момент переключения

Подставим второе уравнение в первое и сделаем замену :

С учетом положительности x получаем:

График, построенный по аналитическим расчетам, представлен на рисунке 1.

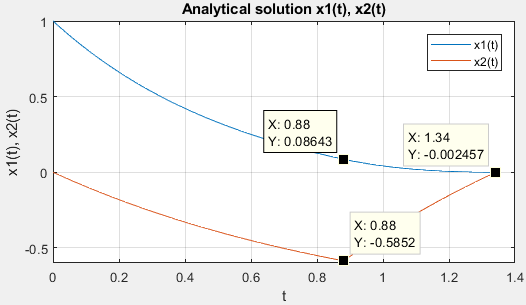


Рисунок 1 – Аналитическое решение

Код, выполняющий построение графика по аналитическим расчетам, представлен на рисунке 2.

|  |
| --- |
| clc; clear; close all;  % 1. Alalytical solution  t\_end = -log((-2+32^0.5)/14); % 1.3425  t\_switch = log(0.5 + 0.5\*(-2+32^0.5)/14)+t\_end; % 0.8814  t = 0 : 0.01 : t\_end;  x1 = exp(-2.\*t) ...  - 0.5.\*(0.5 + 0.5.\*exp(-2\*t) - 1\*exp(-t)) ...  + 0.5.\*(2\*Heaviside(t, t\_switch)).\*(0.5 + 0.5.\*exp(-2\*(t-t\_switch)) - 1\*exp(-(t-t\_switch)));  x2 = -1 + 2.\*Heaviside(t, t\_switch) + exp(-t) - 2.\*Heaviside(t, t\_switch).\*exp(-(t-t\_switch));  figure;  plot(t, x1, t, x2)  grid on; xlabel('t'); ylabel('x1(t), x2(t)');  title('Analytical solution x1(t), x2(t)'); legend('x1(t)', 'x2(t)')  fprintf('1. Alalytical solution: t\_switch = %f, t\_end = %f\n', t\_switch, t\_end) |

Рисунок 2 – Аналитическое решение

**Способ 2. Поиск минимума с помощью Matlab**

Для возможности применения при решении задачи оптимального управления алгоритмов поиска минимума задачу максимального быстро-действия сформулируем следующим образом:

Допустим, что управляющее воздействие является кусочно-постоянной функцией времени, которая меняет знак в момент времени, а перевод объекта управления в конечное состояние происходит в момент времени. Требуется определить такие значения параметров и при которых достигается минимальное значение невязки между фактическими и требуемыми значениями состояний объекта управления в момент. Значение не-вязки вычисляется как сумма квадратов разностей между фактическими и заданными значениями состояний объекта управления в момент времени.

Основной скрипт представлен на рисунке 3. В нем происходит инициализация исходных данных, затем вызов функции fminsearch.

|  |
| --- |
| clc; clear; close all;    global A B X\_START X\_END  X\_START = [1 0];  X\_END = [0 0];  A = [-2 0.5; 0 -1];  B = [0 1];  % 2. Fminsearch solution  figure  x0 = [1 1];  times = fminsearch('switch\_method\_fminsearch\_fun', x0);  fprintf('3. Fminsearch solution: t\_switch = %f, t\_end = %f\n', times(1), times(2))  grid on; legend('x1', 'x2') |

Рисунок 3 – Поиск минимума – основной файл

В качестве аргумента фунции fminsearch передается ссылка на функцию switch\_method\_fminsearch\_fun(), код которой представлен на рисунке 4. В ней происходит вызов функции ode45 и построение графика.

|  |
| --- |
| function error = switch\_method\_fminsearch\_fun(time\_points)  global t\_switch X\_START X\_END  t\_switch = time\_points(1);  t\_end = time\_points(2);  opt = odeset('AbsTol', 1e-7);  [t, x] = ode45('switch\_method\_ode\_fun', [0 t\_end], X\_START, opt);  error = (x(end, 1) - X\_END(1))^ 2 + (x(end, 2) - X\_END(2))^ 2;  plot(t, x); pause(0.1);  end |

Рисунок 4 – Поиск минимума – switch\_method\_fminsearch\_fun.m

В качестве аргумента ode45 используется ссылка на функцию switch\_method\_ode\_fun, код которой представлен на рисунке 5. В ней происходит расчет правых частей исходных уравнений. Также происходит вызов функции calculate\_u.

|  |
| --- |
| function dxdt = switch\_method\_ode\_fun(t, x)  global A B  u = calculate\_u(t);  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];  end |

Рисунок 5 – Поиск минимума – switch\_method\_ode\_fun.m

Код функции calculate\_u представлен на рисунке 6. В ней происходит вычисление значения управляющего воздействия от времени.

|  |
| --- |
| function u = calculate\_u(t)  global t\_switch  Um = 1;  if t < t\_switch  u = -Um;  else  u = Um;  end  end |

Рисунок 6 – Поиск минимума – calculate\_u.m

График переходных процессов представлен на рисунке 7.

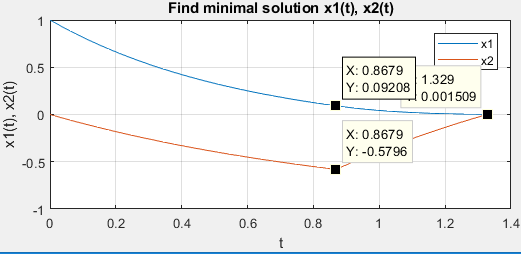


Рисунок 7 – Поиск минимума – график переходных процессов

Результат:

|  |
| --- |
| 3. Fminsearch solution: t\_switch = 0.869956, t\_end = 1.328868 |

**Способ 3. Решение графическим методом**

При использовании этого способа задача оптимального управления решается путем построения линии переключения, геометрического места точек фазового пространства объекта управления, из которых перевод объекта в конечное состояние возможен без переключения знака управления.

Код основного скрипта представлен на рисунке 8. В нем происходят последовательно следующие действия:

1. инициализация исходных данных;

2. построение двух графиков: один с одним знаком управляющего воздействия из начальной точки, второй – с другим знаком и из конечной точки с противоположными знаками правых частей дифференциальных уравнений;

3. расчет времени переключения;

4. построение графиков переходных процессов;

5. расчет времени переходного процесса Т;

6. вывод рассчитанных данных в консоль.

|  |
| --- |
| clc; clear; close all;  global A B X\_START X\_END  X\_START = [1 0];  X\_END = [0 0];  A = [-2 0.5; 0 -1];  B = [0 1];  % 3. Graphic solution  figure; hold on; grid on; axis([-0.2 +1 -1 0]);  title('Graphic solution x2(x1)'); xlabel('x1'); ylabel('x2(x1)');  opt = odeset('AbsTol', 1e-9, 'RelTol', 1e-9);    [t\_inverse, x\_inverse] = ode45('inverse\_ode\_fun', [0 0.5], X\_END, opt);  plot(x\_inverse(:, 1), x\_inverse(:, 2), 'rx-')    [t\_direct, x\_direct] = ode45('direct\_ode\_fun', [0 1], X\_START, opt);  plot(x\_direct(:, 1), x\_direct(:, 2), 'bx-')    t\_switch2 = find\_switch\_time(t\_direct, x\_direct, t\_inverse, x\_inverse);    t\_switch = t\_switch2;  opt = odeset('AbsTol', 1e-7);  [t, x] = ode45('switch\_method\_ode\_fun', [0 2], X\_START, opt);  figure; hold on; grid on;  title('Graphic solution x1(t), x2(t)'); xlabel('t'); ylabel('x1(t), x2(t)');  plot(t, x)    t\_end2 = 0;  for i = 1:length(t)  if x(i, 1) < 0  t\_end2 = t(i);  break;  end  end    fprintf('2. Graphic solution: t\_switch = %f, t\_end = %f\n', t\_switch2, t\_end2) |

Рисунок 8 – Графическое решение – main.m

Построение графика из начальной точки осуществляется с помощью функции ode45 передачей ей в качестве аргумента ссылку на функцию direct\_ode\_fun(), код которой представлен на рисунке 9.

|  |
| --- |
| function dxdt = direct\_ode\_fun(t, x)  global A B  u = -1;  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];  end |

Рисунок 9 – Графическое решение – direct\_ode\_fun.m

Построение графика из конечной точки осуществляется с помощью функции ode45 передачей ей в качестве аргумента ссылку на функцию direct\_ode\_fun(), код которой представлен на рисунке 9.

|  |
| --- |
| function dxdt = inverse\_ode\_fun(t, x)  global A B  u = +1;  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];  end |

Рисунок 10 – Графическое решение – inverse \_ode\_fun.m

Расчет времени переключения осуществляется функцией find\_switch\_time, код которой представлен на рисунке 11. Эта функция выполняет перебор всех значений x1 и x2 с целью поиска точки с минимальной невязкой.

|  |
| --- |
| function t\_switch = find\_switch\_time(t\_direct, x\_direct, t\_inverse, x\_inverse)  t\_switch = 0;  minError = inf;  for directCounter = 1:length(t\_direct)  for inverseCounter = 1:length(t\_inverse)  currentError = (x\_inverse(length(t\_inverse) - inverseCounter + 1, 2) - x\_direct(directCounter, 2))^2 + ...  (x\_inverse(length(t\_inverse) - inverseCounter + 1, 1) - x\_direct(directCounter, 1))^2;  if currentError < minError  minError = currentError;  t\_switch = t\_direct(directCounter);  end  end  end  end |

Рисунок 11 – Графическое решение – find\_switch\_time.m

Фазовый портрет представлен на рисунке 12.

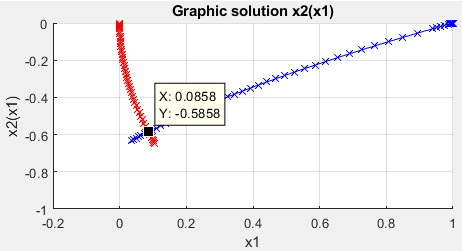


Рисунок 12 – Графическое решение – фазовый портрет

Графики переходных процессов представлены на рисунке 13.

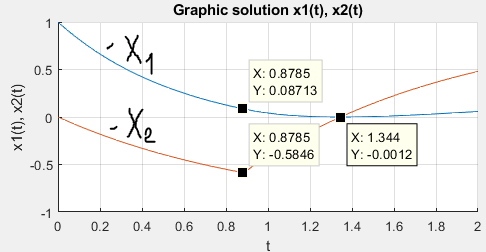


Рисунок 13 – Графическое решение – переходные процессы

Результат выполнения программы представлен на рисунке 14.

|  |
| --- |
| 2. Graphic solution: t\_switch = 0.883351, t\_end = 1.344255 |

Рисунок 14 – Графическое решение – результаты расчетов

**Вывод**

В ходе работы была решена задача максимального быстродействия аналитическим, поисковым и графическими методами. Результаты всех методов (графики переходных процессов и значения времени переключения и время переходного процесса) отличаются на величину не больше, чем 1.31%.

Сложность решения задачи 2-ого порядка относительно задач более высоких порядков, не высокая.

Все результаты расчетов времен представлены на рисунке 15.

Все графики переходных процессов представлены на рисунке 16.

|  |
| --- |
| 1. Alalytical solution: t\_switch = 0.881374, t\_end = 1.342454  2. Graphic solution: t\_switch = 0.883351, t\_end = 1.344255  3. Fminsearch solution: t\_switch = 0.869956, t\_end = 1.328868 |

Рисунок 15 – Все результаты расчетов времен

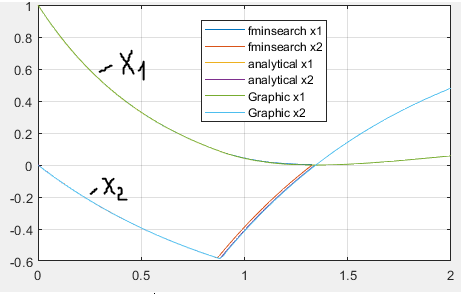


Рисунок 16 – Все графики переходных процессов