

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

Кафедра КСУ

**ОТЧЕТ**  
по курсовой работе  
по дисциплине «Проектирование оптимальных систем управления»  
**ВАРИАНТ 1**

Студент гр. 9492

\_\_\_\_\_

Викторов А.Д.

Преподаватель

\_\_\_\_\_

Мирошников А.Н.

Санкт-Петербург

2024

## ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Виктор А.Д.

Группа 9492

Тема работы: Проектирование алгоритма управления динамическим объектом на примере водоизмещающего судна

### Исходные данные:

Таблица 1 – Данные судна

Параметры	Скорость хода	Длина по ватерлинии	Коэффициенты математической модели					
	$V_0$ , м/с	$L$ , м	$r_{21}$	$r_{31}$	$q_{21}$	$q_{31}$	$s_{21}$	$s_{31}$
Значения	2,57	99,6	-0,58	6,16	0,80	-7,23	-0,34	-3,5

Косвенный метод решения задачи оптимизации — минимизация квадратичного функционала.

Содержание пояснительной записки: Содержание, Введение, Математическое описание объекта управления, Прямые методы, Косвенные методы, Сравнение алгоритмов управления, Список использованных источников

Студент гр. 9492

\_\_\_\_\_ Виктор А.Д.

Преподаватель

\_\_\_\_\_ Мирошников А.Н.

## **АННОТАЦИЯ**

В курсовой работе осуществляется проектирование алгоритмов управления динамическим объектом на примере водоизмещающего судна. Проектирование алгоритма управления состоит из следующих этапов:

- математическое описание объекта управления;
- математическая формулировка цели управления;
- выбор метода решения поставленной оптимизационной задачи;
- оценка вариантов решения задачи.

Рассмотрены и сравнены следующие методы проектирования оптимальной системы управления:

- методом минимизации интегрального квадратичного функционала;
- методом, основанном на теореме от N интервалах;
- методе параметрической оптимизации линейного закона управления.

Спроектирован алгоритм управления рулем судна, который обеспечивает минимальное время устранения начального значения угла рыскания равного  $10^\circ$ .

## **SUMMARY**

The aim of this work is to develop an algorithm for controlling a dynamic object, specifically a displacement ship, with the goal of minimizing the time required to eliminate the initial yaw angle value. To solve the optimization problem, both direct methods (including the N-interval theorem-based method and parametric optimization method) and an indirect method (minimization of integral quadratic functional) were utilized. A comparative analysis of these methods was conducted.

## СОДЕРЖАНИЕ

АННОТАЦИЯ .....	3
SUMMARY .....	3
СОДЕРЖАНИЕ .....	4
ВВЕДЕНИЕ .....	5
ОБЩИЕ СВЕДЕНИЯ .....	6
МЕТОД, ОСНОВАННЫЙ НА ТЕОРЕМЕ ОБ N ИНТЕРВАЛАХ .....	8
МЕТОД ПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ ЛИНЕЙНОГО ЗАКОНА УПРАВЛЕНИЯ .....	12
МЕТОД МИНИМИЗАЦИИ ЛИНЕЙНОГО КВАДРАТИЧНОГО ФУНКЦИОНАЛА .....	14
ПРОВЕРКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ НА ЧУВСТВИТЕЛЬНОСТЬ .....	16
ЗАКЛЮЧЕНИЕ .....	17

## **ВВЕДЕНИЕ**

Целью курсовой работы является проектирование оптимальных систем управления различными методами и сравнение результатов проектирования на примере разработки алгоритма управления судном, который обеспечивает минимальное время устранения начального значения угла рыскания равного  $10^\circ$ .

Курсовая работа содержит этапы математического описания объекта управления, проектирования оптимальной системы управления, минимизирующей время переходного процесса, различными прямыми и косвенными методами, сравнения полученных результатов и их анализ.

## ОБЩИЕ СВЕДЕНИЯ

Динамика судна, как и любого физического тела, подчиняется второму закону Ньютона. Силы и моменты, действующие на судно, в свою очередь, описываются законами гидродинамики. Соотношения между кинематическими параметрами движения ( $\varphi$  - угол рыскания,  $\omega_y$  - угловая скорость рыскания,  $\beta$  - угол дрейфа,  $\delta_e$  - угол перекладки руля) показаны на рисунке 1.

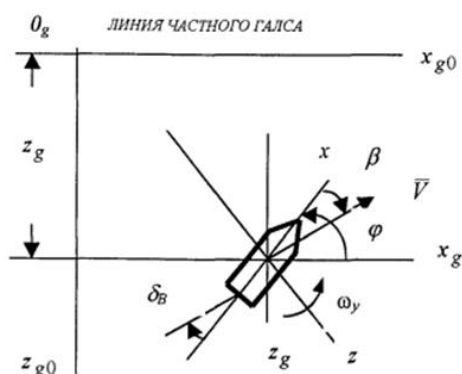


Рисунок 1.1 – Динамика судна

В общем случае, зависимость сил и моментов, действующих на судно от параметров движения носит нелинейный характер. Однако предположение о малых значениях угла дрейфа и угловой скорости рыскания и постоянстве линейной скорости движения судна позволяют линеаризовать эти зависимости и описать динамику в виде системы линейных дифференциальных уравнений относительно углов рыскания, дрейфа, угловой скорости рыскания, угла перекладки руля и одного нелинейного соотношения, отражающего тот факт, что руль не может поворачиваться на произвольный угол при произвольном сигнале управления. Для большинства современных судов максимальный угол перекладки руля равен  $35^\circ$ . Упомянутые соотношения, записанные относительно нормированного времени ( $\tau$ ), имеют вид (1). При записи (1), кроме предположений о малости углов не учитывалось действие на судно ветро-волновых возмущений. т.е. математическая модель соответствует движению судна на тихой воде:

$$\begin{cases} \frac{d\omega}{d\tau} = -r_{31}\omega + q_{31}\beta + s_{31}\delta_e; \\ \frac{d\beta}{d\tau} = -r_{21}\omega - q_{21}\beta - s_{21}\delta_e; \\ |\delta_e| \leq 35^\circ; \end{cases}$$

где:  $\omega = \frac{\omega_y}{\Omega}$  – относительная скорость рыскания;  $\beta$  – угол дрейфа;  $\delta$  – угол перекладки руля.

Значение нормирующей частоты:

$$\Omega = \frac{V_0}{L} = \frac{2.57}{99.6} = 0.0258$$

Математическая модель судна записывается в виде:

$$\frac{d}{dt} \begin{bmatrix} \omega_y \\ \beta \\ \varphi \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_y \\ \beta \\ \varphi \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \\ 0 \end{bmatrix} \delta_e$$

$$|\delta_e| \leq 35^\circ$$

Соотношение между параметрами имеет вид:

$$a_{11} = -r_{31}\Omega = -6.16 \cdot 0.0258 = -0.159$$

$$a_{12} = -q_{31}\Omega^2 = 7.23 \cdot 0.0258^2 = 0.005$$

$$a_{21} = -r_{21} = 0.58$$

$$a_{22} = -q_{21}\Omega = -0.80 \cdot 0.0258 = -0.021$$

$$b_{11} = -s_{31}\Omega^2 = 3.5 \cdot 0.0258^2 = 0.002$$

$$b_{21} = -s_{21}\Omega = 0.34 \cdot 0.0258 = 0.009$$

## МЕТОД, ОСНОВАННЫЙ НА ТЕОРЕМЕ ОБ N ИНТЕРВАЛАХ

Выполнив вычисления параметров модели получим следующую модель в форме переменных состояния:

$$\frac{d}{dt} \begin{bmatrix} \omega_y \\ \beta \\ \varphi \end{bmatrix} = \begin{bmatrix} -0.159 & 0.005 & 0 \\ 0.58 & -0.021 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_y \\ \beta \\ \varphi \end{bmatrix} + \begin{bmatrix} 0.002 \\ 0.009 \\ 0 \end{bmatrix} \delta_\epsilon$$

Собственные числа модели:

$$\begin{vmatrix} -0.159 - \lambda & 0.005 & 0 \\ 0.58 & -0.021 - \lambda & 0 \\ 1 & 0 & -\lambda \end{vmatrix} = 0$$

$$\lambda = \begin{bmatrix} -0.244 \\ -0.125 \\ 0 \end{bmatrix}$$

Т.к. у собственных чисел ХП системы отсутствуют мнимые составляющие, то для системы можно применить теорему об N – интервалах и искать управление в виде кусочно-постоянной функции с двумя моментами переключения.

Результат, получаемый с помощью поисковых методов, может существенно зависеть от выбора начальной точки поиска. В связи с этим при проектировании алгоритма управления на основе теоремы об N интервалах рекомендуется выполнять мероприятия, направленные на определение начальной комбинации искомых параметров.

Для определения начальной точки поиска воспользуемся графическим методом нахождения моментов переключения. Искать переключение будем в плоскости  $(\omega_y, \beta)$ , тем самым найденные моменты переключения позволяет

перевести систему из состояния  $\begin{bmatrix} \omega_{y0} \\ \beta_0 \\ \varphi_0 \end{bmatrix}$  в  $\begin{bmatrix} \omega_y(T) \\ \beta(T) \\ \varphi_i \neq \varphi(T) \end{bmatrix}$ . В дальнейшем,

воспользуемся алгоритмом Нелдера – Мида для нахождения конечного

состояния системы  $\begin{bmatrix} \omega_y(T) \\ \beta(T) \\ \varphi(T) \end{bmatrix}$ , начальной точкой поиска в котором и будет



выступать точка, полученная с помощью графического метода. В качестве целевой функции будет выступать следующий оптимум:

$$J_i = (\varphi(T) - \varphi_{fi})^2 + \omega_y^2 + \beta^2$$

На каждом шаге итеративной процедуры в качестве начального набора параметров ( $t_1$ ,  $t_2$ ,  $T$ ), при поиске функции минимума  $J_i$  используется результат предыдущего шага, а значение  $\varphi_{fi}$  систематически приближается к заданному конечному значению угла рыскания  $\varphi(T) = 0$ .

Код для реализации приведен в Листинге А1. На рисунке 2.1 представлена фазовая траектория, которая была получена графическим методом. Вектор моментов переключения:

$$\begin{bmatrix} t_1 \\ t_2 \\ T \end{bmatrix} = \begin{bmatrix} 19 \\ 47 \\ 48 \end{bmatrix}$$

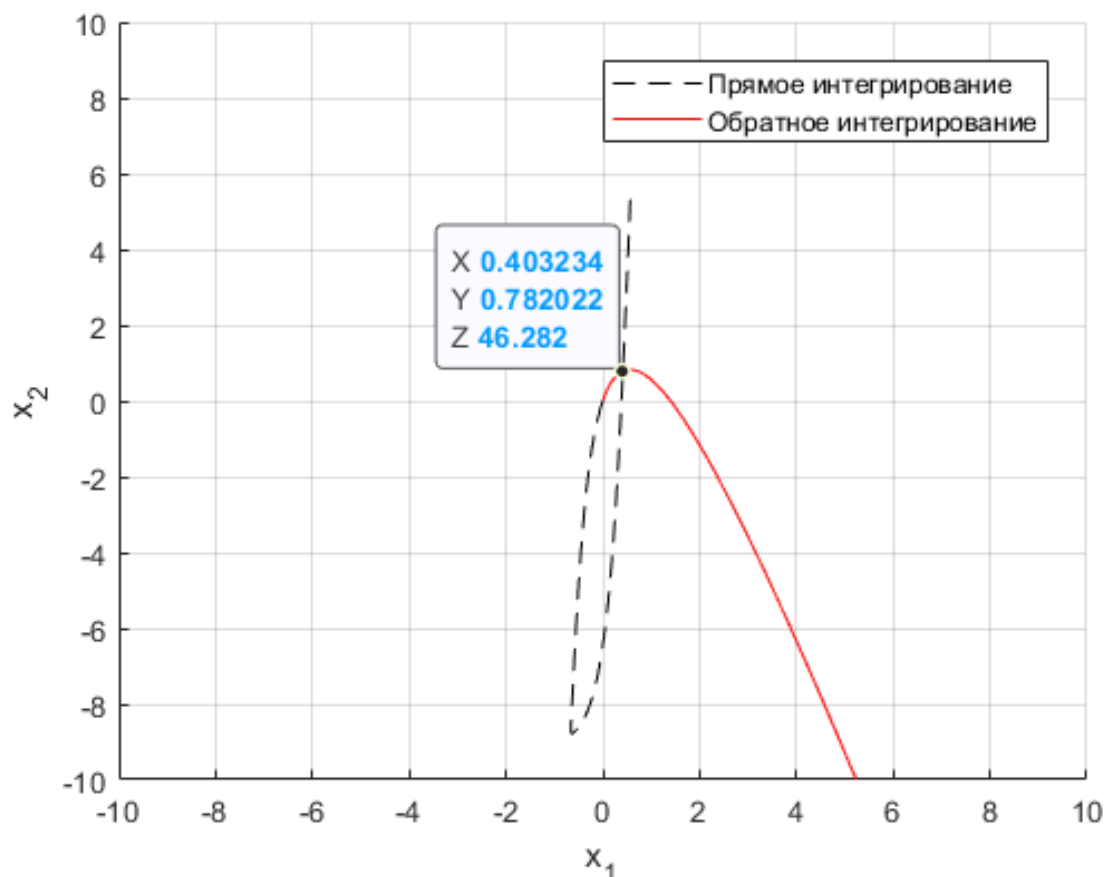


Рисунок 2.1 – Фазовая траектория

Данный вектор моментов переключения переводит систему из точки (0, 0, 10) в точку (0,46 4,65 7,48).

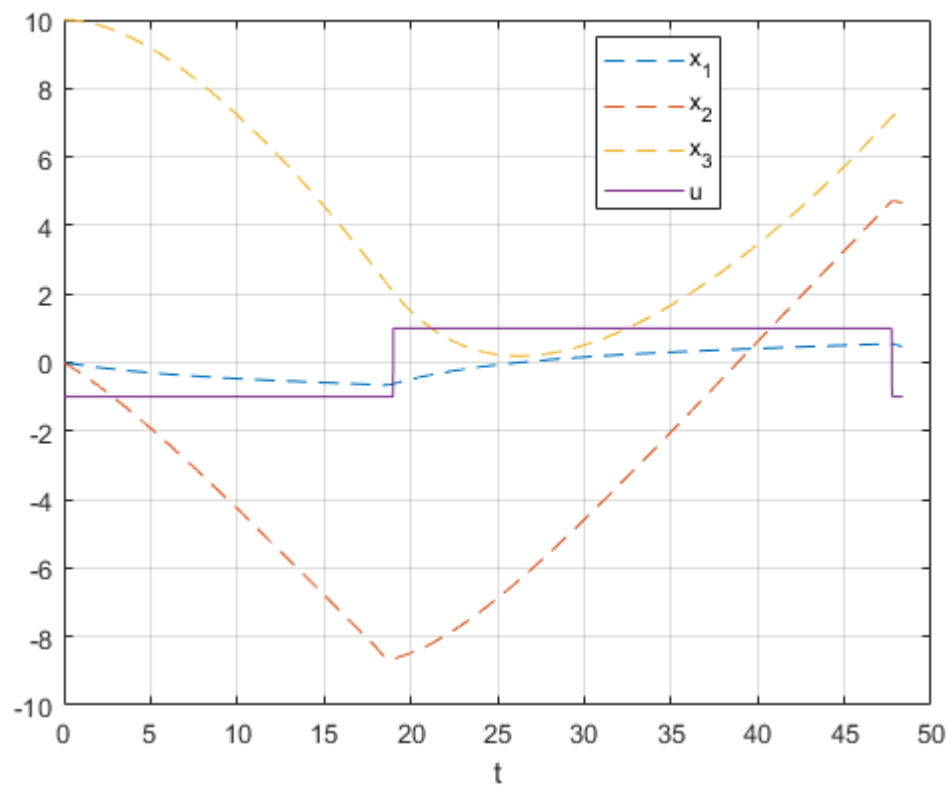


Рисунок 2.2 – Переходные процессы и управляющий сигнал

После нескольких итераций метода Нелдера – Мида получается следующий вектор моментов переключения:

$$\begin{bmatrix} t_1 \\ t_2 \\ T \end{bmatrix} = \begin{bmatrix} 23.59 \\ 47.75 \\ 48.36 \end{bmatrix}$$

Переходные процессы и сигнал управления для данных моментов переключения представлен на рисунке 2.2.

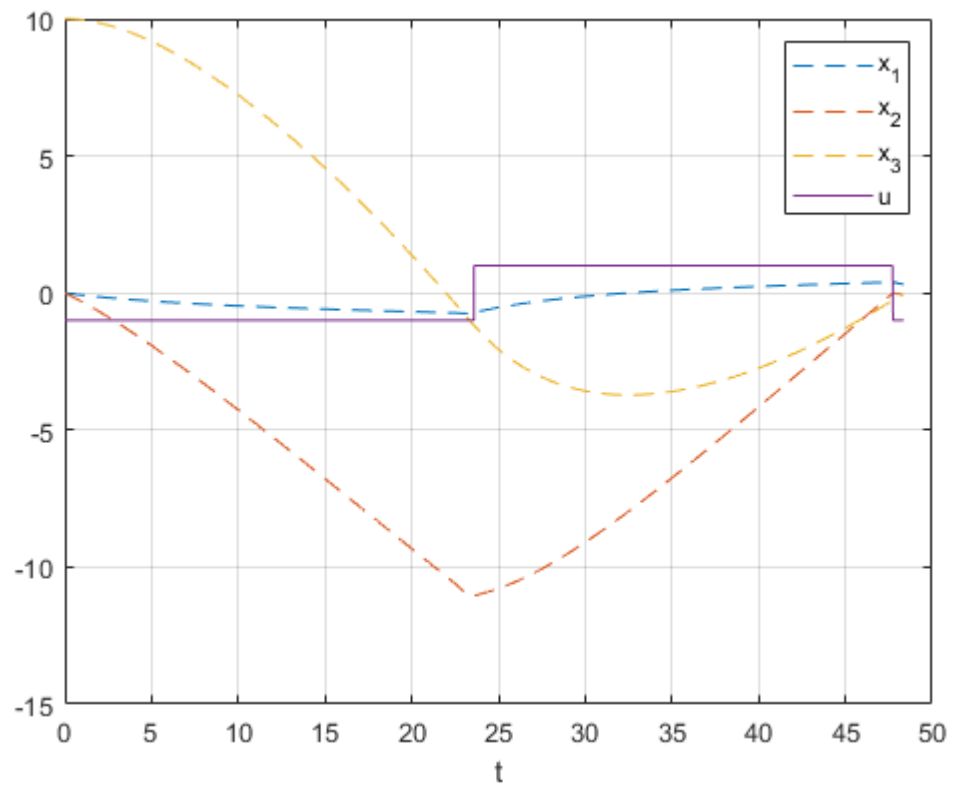


Рисунок 2.2 – Переходные процессы и сигнал управления после оптимизации

Таким образом, удалось перевести систему из состояния  $(0, 0, 10)$  в состояние  $(0.31, -0.07, -0.04)$ . за время, равное 48.36

## МЕТОД ПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ ЛИНЕЙНОГО ЗАКОНА УПРАВЛЕНИЯ

Метод параметрической оптимизации линейного закона управления заключается в поиске таких значений параметров линейного закона управления, которые обеспечивают перевод объекта управления в заданное состояние за минимальное время и последующее удержание объекта в этом состоянии.

Одним из достоинств этого метода является возможность включения в закон управления только тех переменных состояния, которые соответствуют достаточно точно измеряемым физическим величинам. В случае водоизмещающего судна наиболее точно из принятых в рассмотрение физических величин измеряются угол рыскания и угловая скорость рыскания.

С математической точки зрения задача параметрической оптимизации заключается в том, чтобы для алгоритма управления  $\delta_e = -K_1\varphi - K_2\omega_y$  найти такие значения параметров  $K_1$  и  $K_2$  при которых время перевода судна из начальной точки в конечную происходит за минимальное время.

Поскольку в рассматриваемом случае изменение состояния судна вблизи целевой точки носит экспоненциальный характер, то теоретически время перехода в целевое состояние не ограничено. В связи с этим предлагается принимать за момент окончания процесса управления момент времени, после которого абсолютное значение угла рыскания не превышает 5% от начального значения

Код реализации метода параметрической оптимизации линейного закона управления представлена на Листинге А2.

Результатом работы программы стали коэффициенты

$$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 46.97 \\ 139.72 \end{bmatrix}$$
$$T = 21.31$$

На рисунке 3.1 приведены переходный процесс и управляющее воздействие для системы соответственно.

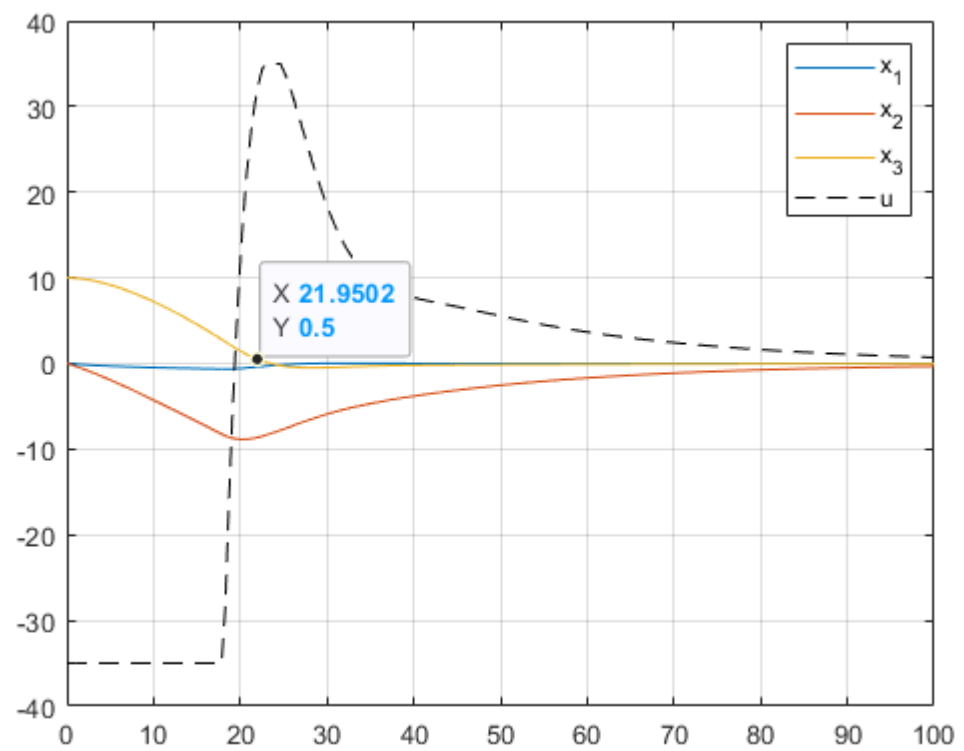


Рисунок 3.1 – Переходный процесс

## МЕТОД МИНИМИЗАЦИИ ЛИНЕЙНОГО КВАДРАТИЧНОГО ФУНКЦИОНАЛА

Интегральный квадратичный функционал для решаемой задачи записывается в виде:

$$J = \int_0^{\infty} (\lambda \varphi^2 + \delta_e^2) dt$$

В общем случае его минимизация не эквивалентна минимизации времени переходного процесса, однако можно показать, что переходные процессы в системах, оптимальных по интегральным квадратичным функционалам, ускоряются при увеличении значений весовых множителей. Следовательно, можно приблизиться к задаче максимального быстродействия, решив задачу поиска такой величины весового множителя, при котором время переходного процесса минимально.

Для вычисления параметров алгоритма управления используем функцию LQR.

В том случае, когда объект управления описывается линейным матричным дифференциальным уравнением, имеющим вид:

$$\frac{dx}{dt} = Ax + Bu$$

где  $x$  – вектор состояния объекта управления,  $u$  – вектор управляющих воздействий, а функционал качества имеет вид:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

где  $Q$  и  $R$  – весовые матрицы. При этом управление определяется как  $u = Kx$ , где  $K$  – искомая матрица коэффициентов обратных связей по переменным состояния, которая находится в результате применения функции LQR:

$$K = \text{lqr}(A, B, Q, R)$$

Общий алгоритм решения задачи синтеза алгоритма управления может быть реализован с помощью функции FMINSEARCH. Код для реализации данного метода представлен на Листинге А3.

В процессе моделирования (код представлен в приложении В) были получены значения  $\lambda = 5.69 \cdot 10^3$ . Соответствующий переходный процесс и управляющее воздействие представлены на рисунке 4.1. Время переходного процесса при этом составило 21.79 с.

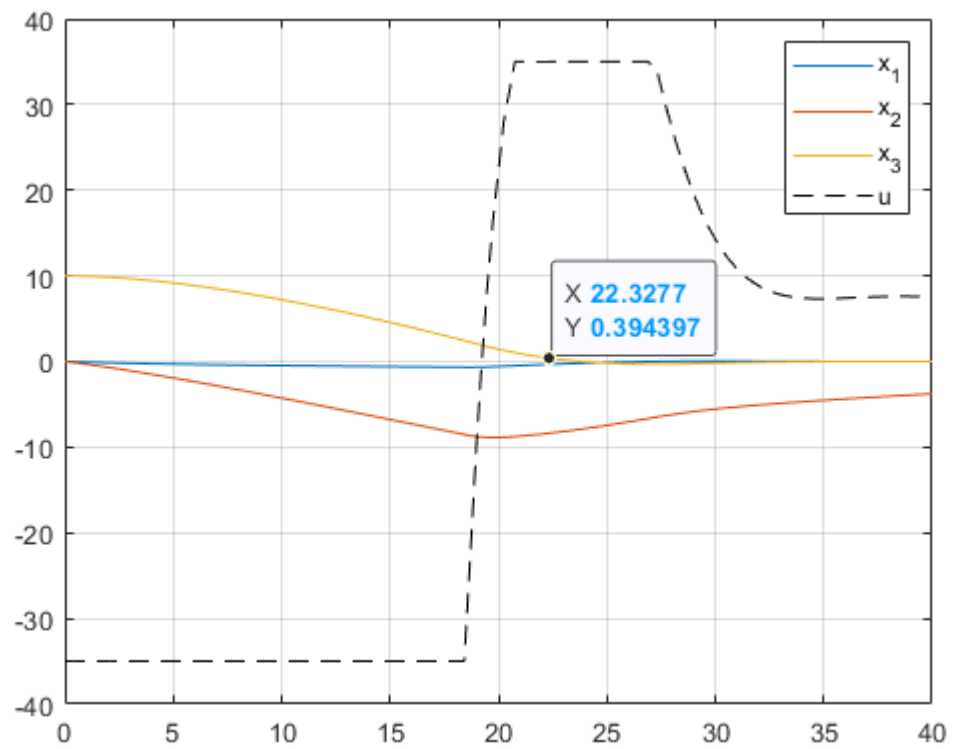


Рисунок 4.1 – Переходные процессы.

Матрица коэффициентов обратных связей:

$$\begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} = \begin{bmatrix} 196.84 \\ 2.02 \\ 75.40 \end{bmatrix}$$

## ПРОВЕРКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ НА ЧУВСТВИТЕЛЬНОСТЬ.

Изменим изначальное значение скорости  $r_{21}$  на 10% и -10%.

$$r_{21}^{+10\%} = r_{21} \cdot 1.1 = -0.58 \cdot 1.1 = -0.638$$

$$r_{21}^{-10\%} = r_{21} \cdot 0.9 = -0.58 \cdot 0.9 = -0.522$$

Применим результаты методов параметрической оптимизации линейного закона управления №1 и минимизации линейного квадратичного функционала №2. Результаты моделирования приведены в таблице 5.1

Таблица 5.1. Времена переходных процессов

Метод	Скорость	Время переходного процесса, с	Изменение времени ПП, %
1	$r_{21}$	21.32	-
	$r_{21}^{+10\%}$	21.41	+0.4%
	$r_{21}^{-10\%}$	21.77	+2.1%
2	$r_{21}$	21.79	-
	$r_{21}^{+10\%}$	21.99	+0.9%
	$r_{21}^{-10\%}$	21.94	+0.7%

Как видно из полученных результатов, представленные методы чувствительны к неопределенностям параметров. Изменение параметров математической модели судна приводит к соответствующему изменению времени переходного процесса.

В случаях использования обоих методов удалось добиться схожих результатов. Проектирование первого метода оказалось сложнее, так как получаемые значения коэффициентов сильно зависели от начальной точки поиска. Однако в плане реализации данный метод проще, так как использует обратную связь лишь по двум переменным состояниям, значения которых могут быть измерены достаточно точно.



## ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была достигнута цель по отработке начального угла рысканья в  $10^\circ$  при использовании трех методов:

- метод, основанный на теореме об N интервалах;
- метод параметрической оптимизации линейного закона управления;
- метод минимизации линейного квадратичного функционала.

Был осуществлен анализ чувствительности времени переходного процесса к изменению параметров объекта управления.

Каждый из методов имеет свои указанные преимущества и недостатки, так что выбор какого-либо из них должен осуществляться исходя из располагаемых ресурсов и требований к системе управления.

## ПРИЛОЖЕНИЕ А

*Листинг 0 – параметры системы:*

```
V0 = 2.57; % м/с скорость хода
L = 99.6; % м, длина по ватерлинии
r21 = -0.58;
r31 = 6.16;
q21 = 0.80;
q31 = -7.23;
s21 = -0.34;
s31 = -3.5;
WW = V0/L; %Нормирующая частота

%% Соотношение параметров в нормированном и нормальном времени
a11 = -r31*WW;
a12 = -q31*WW^2;
a21 = -r21;
a22 = -q21*WW;
b11 = -s31*WW^2;
b21 = -s21*WW;

%% Матрицы мат.модели с-мы в реальном времени
A = [a11 a12 0; a21 a22 0; 1 0 0];
B = [b11; b21; 0];
%% Собственные значения системы
% eig(A);
```

*Листинг 1 – Оптимизация по теореме об N-интервалах:*

```
clear, clc
close all;
initial_data;

x0 = [0 0 10];
Um = 35;
T = 50;
t_sw_1 = 19;
direct_time = @(t,x) A*[x(1); x(2); x(3)]+B*Um*(-(t<t_sw_1)+(t>=t_sw_1));
[t1,x1] = ode45(direct_time, 0:0.001:T, x0);

reverse_time = @(t,x) A*[x(1); x(2); x(3)]-B*Um;
[t2,x2] = ode45(reverse_time, T:-0.001:0, [0 0 0]);

figure(1)
plot3(x1(:,1), x1(:,2), t1, '--k', x2(:,1), x2(:,2), t2, 'r'); grid on
legend('Прямое интегрирование', 'Обратное интегрирование')
xlabel('x_1'); ylabel('x_2')
axis([-10 10 -10 10])
view([0 90])

t_sw_2 = 47
T_off = 48

u = @(t) Um*(-(t<t_sw_1)+(t>=t_sw_1)-2*(t>=t_sw_2));
fun = @(t,x) A*[x(1); x(2); x(3)]+B*u(t);
[t,x] = ode45(fun, 0:0.001:T_off, x0);
```

```

figure(2)
plot(t,x, '--', t, u(t)/35); grid on; hold on;
legend('x_1', 'x_2', 'x_3', 'u');
xlabel('t');

t_opt = [t_sw_1 t_sw_2 T_off];

[tt, fval] = fminsearch(@(tau) costfun(tau, A, B, Um, x0), t_opt)

u = @(t) Um*(-(t<tt(1))+(t>=tt(1))-2*(t>=tt(2)));
fun2 = @(t,x) A*[x(1); x(2); x(3)]+B*u(t);
[t,x] = ode45(fun, 0:0.001:T_off, x0);
[t_1,x_1] = ode45(fun2, 0:0.001:tt(3), x0);
figure(3)
plot(t_1,x_1, '--', t_1, u(t_1)/35); grid on; hold on;
legend('x_1', 'x_2', 'x_3', 'u');
xlabel('t');

function f = costfun(t_opt, A, B, um, x0)
    t1 = t_opt(1); t2 = t_opt(2); T = t_opt(3);
    u = @(t) um*(-(t<t1)+(t>=t1)-2*(t>=t2));
    fun = @(t,x) A*[x(1) x(2) x(3)]'+B*u(t);
    [~, x] = ode45(fun, [0 T], x0);
    f = x(end,:)*x(end,:);
end

```

## *Листинг 2 – Параметрическая оптимизация линейного закона управления:*

```

clear, clc, close all;
initial_data;

x0 = [0 0 10];
K0 = [100 100];
Um = 35;
T = 100;

[K, fval] = fminsearch(@(KK) costfun(KK, A,B,Um,x0), K0)
fun = @(t,x) A*[x(1);x(2);x(3)]+B*control_calc(x, Um, K);
[t,x] = ode45(fun, [0 T], x0);
figure(1)
plot(t,x); grid on; hold on;
alignment_angle = zeros(1,length(t));
for i = 1:length(t)
    alignment_angle(i) = control_calc(x(i,:), Um, K);
end
plot(t,alignment_angle, '--k'); grid on
legend('x_1', 'x_2', 'x_3', 'u');

function f = costfun(K, A, B, umax, x0)
    fun = @(t,x) A*[x(1);x(2);x(3)]+B*control_calc(x, umax, K);
    [t,x] = ode45(fun, [0 100], x0);
    f = t_per_proc(t, x, 3);
end

function f = t_per_proc(t, x, j)
    for i=length(x):-1:1
        if abs(x(i,j))>0.05*10
            f = t(i);
        end
    end
end

```

```

        break
    end
end
end
end

function u = control_calc(x, umax, K)
    u = -K(1)*x(3)-K(2)*x(1);
    if abs(u)>umax
        u = umax*sign(u);
    end
end
end

```

*Листинг 3 – Минимизация линейного квадратичного функционала:*

```

clear, clc, close all;
initial_data;
x0 = [0 0 10];
Um = 35;
lambda = 1;

[lam, fval] = fminsearch(@(l_) costfun(l_, A,B,Um,x0), lambda)
Q = zeros(3,3);
Q(3,3) = lam
K = lqr(A,B,Q,1)
fun = @(t,x) A*[x(1);x(2);x(3)]+B*control_calc(x, Um, K);
[t,x] = ode45(fun, [0 40], x0);
figure(1)
plot(t,x); grid on; hold on;
for i = 1:length(t)
    uu(i) = control_calc(x(i,:), Um, K);
end
plot(t,uu, '--k'); grid on
legend('x_1', 'x_2', 'x_3', 'u');
t_trans = transition_time(t, x, 3)

function f = costfun(lambda, A, B, umax, x0)
    Q = zeros(3,3);
    Q(3,3) = lambda;
    K = lqr(A,B,Q,1);
    fun = @(t,x) A*[x(1);x(2);x(3)]+B*control_calc(x, umax, K);
    [t,x] = ode45(fun, [0 200], x0);
    f = transition_time(t, x, 3);
end

function t_trans = transition_time(t, x, j)
    for i=length(x):-1:1
        if abs(x(i,j))>0.05*10
            t_trans = t(i);
            break
        end
    end
end

function u = control_calc(x, umax, K)
    u = -K(1)*x(1)-K(2)*x(2)-K(3)*x(3);
    if abs(u)>umax
        u = umax*sign(u);
    end
end
end

```