

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
(Университет ИТМО)

Факультет систем управления и робототехники

Лабораторная работа №1  
по дисциплине  
«ПРОСТО ТИТУЛЬНИК»

по теме:  
«ПРОСТО ТИТУЛЬНИК»

Студент:  
Группа № R34372

Предподаватель:

Санкт-Петербург 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 ОБЗОР МАТЕМАТИЧЕСКОЙ МОДЕЛИ КВАДРОКОПТЕРА ...	5
1.1 Описание модели .....	5
1.2 Локальная и глобальная системы координат .....	5
1.3 Поступательное и вращательное движение .....	8
1.3.1 Поступательное движение и силы сопротивления .....	8
1.3.2 Вращательное движение .....	11
1.3.3 Крутящие моменты .....	12
1.4 Итоговая система уравнений динамики квадрокоптера .....	13
2 АНАЛИЗ АЛГОРИТМОВ УПРАВЛЕНИЯ	
КВАДРОКОПТЕРАМИ .....	15
2.1 Классические методы управления .....	15
2.2 Нелинейные методы управления .....	16
2.2.1 Метод обратной связи (Feedback Linearization) .....	17
2.2.2 Метод обратного пошагового управления	
(Backstepping) .....	18
2.2.3 Метод скользящего режима .....	18
2.2.4 Model Predictive Control (MPC) .....	19
3 РАЗРАБОТКА АЛГОРИТМОВ УПРАВЛЕНИЯ .....	21
3.1 LQR регулятор по линеаризованной модели .....	21
3.1.1 Линеаризация у точки равновесия .....	23
3.2 LQR регулятор с линеаризацией обратной связью .....	26
3.3 Nonlinear MPC .....	28
4 МОДЕЛИРОВАНИЕ В СРЕДЕ MATLAB/SIMULINK .....	30

## ВВЕДЕНИЕ

В современном мире мультикоптерные летательные аппараты, в частности квадрокоптеры, находят применение во всем большем количестве сфер. Они используются для сбора информации об окружающей среде, мониторинга территорий, доставки грузов, исследований и даже съемок кино. Широкое распространение квадрокоптеров обусловлено их маневренностью, относительной простотой конструкции и возможностью выполнения задач в условиях, недоступных для других типов летательных аппаратов. Также растет и доступность многороторных летательных аппаратов за счет удешевления компонентов.

Квадрокоптер представляет собой сложную механическую систему, состоящую из четырех электродвигателей расположенными симметрично по углам рамы. Винты, раскручиваемые электродвигателями, создают тягу, которая поднимает квадрокоптер в воздух, позволяя ему удерживаться и маневрировать. В современных квадрокоптерах используются бесколлекторные электродвигатели, получившие широкое распространение в последние десятилетия за счет их эффективности, простой и надежной конструкции. Также одними из наиболее важных компонентов является микроконтроллер, называемый полетным контроллером - он отвечает за управление напряжением на моторах. Помимо этого квадрокоптеры оснащены драйверами двигателей, средствами коммуникации и различными датчиками: акселерометрами, gps-датчиками, камерами и в некоторых случаях лидарами.

Для создания законов управления и компьютерного моделирования необходима математическая модель динамики летательного аппарата. Поведение мультироторного устройства описывается нелинейными дифференциальными уравнениями, их количество может разниться в зависимости от количества электродвигателей и учета внешних возмущений. Математическая модель может быть синтезирована с использованием уравнений Эйлера-Лагранжа или Ньютона-Эйлера, учитывающими поступательное и вращательное движение аппарата. Квадрокоптер обладает шестью степенями свободы: тремя поступательными - движение вдоль осей  $X$ ,  $Y$ ,  $Z$ , и тремя вращательными - углы крена, тангажа, рыскания. Для упрощения

задачи синтеза законов управления используется линеаризованная модель квадрокоптера, полученная путем аппроксимации нелинейных уравнений около состояния равновесия, что позволяет применять классические методы теории управления для анализа системы и синтеза законов управления.

В настоящей работе рассматривается исследование и разработка алгоритмов управления квадрокоптером для решения задачи слежения за траекторией. В рамках работы планируется решить следующие задачи:

1. Создание математической модели квадрокоптера, учитывающей его динамику и возможные внешние возмущения.
2. Синтез алгоритмов управления обеспечивающих слежение за заданной траекторией
3. Компьютерное моделирование в среде MATLAB/Simulink
4. Экспериментальная проверка алгоритмов на реальном оборудовании
5. Сравнение качественных характеристик переходных процессов разных систем управления

Целью данной работы является синтез системы управления квадрокоптером в задаче слежения за траекторией и тестирование на реальном оборудовании.

# 1 ОБЗОР МАТЕМАТИЧЕСКОЙ МОДЕЛИ КВАДРОКОПТЕРА

## 1.1 Описание модели

Квадрокоптер обладает шестью степенями свободы:

- Линейные координаты  $X, Y, Z$ ,
- Угол крена  $\phi$  соответствует углу вращения вокруг оси  $X$ .
- Угол тангажа  $\theta$  соответствует углу вращения вокруг оси  $Y$ .
- Угол рыскания  $\psi$  соответствует углу вращения вокруг оси  $Z$ .

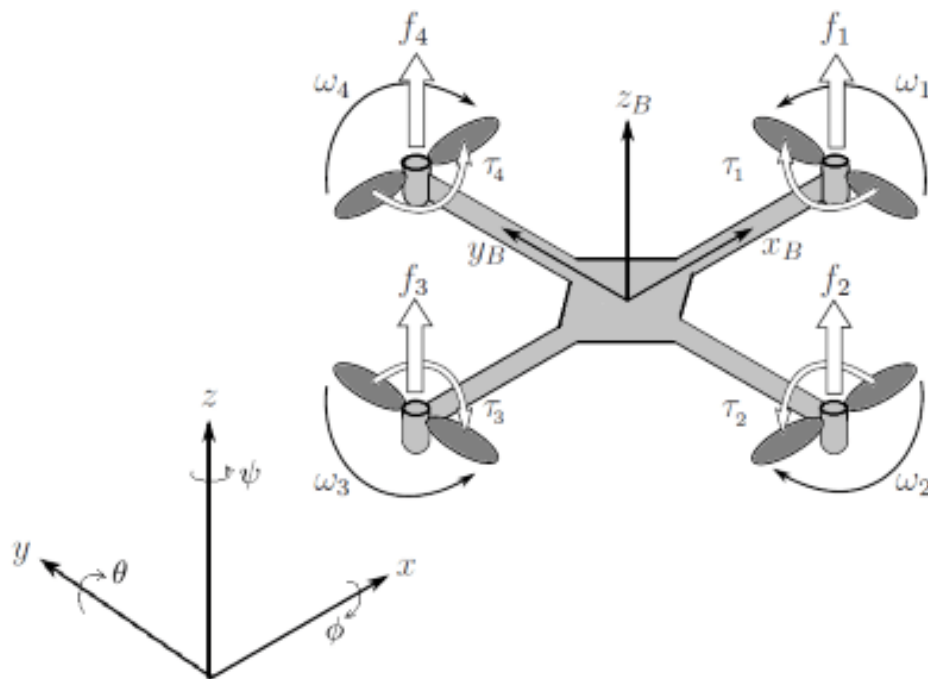


Рисунок 1 — Схематичное изображение квадрокоптера

## 1.2 Локальная и глобальная системы координат

Для линейных координат рассмотрим переход системы координат от глобальной  $X, Y, Z$  к системе координат связанной с квадрокоптером  $X_B, Y_B, Z_B$ .

Чтобы перейти к системе координат, связанной с центром масс квадрокоптера, необходимо использовать матрицу поворота или, как её ещё называют матрицу косинусов.

Эта матрица представляет собой перемноженные три матрицы, которые осуществляют поворот вокруг оси  $X$ ,  $Y$  и  $Z$  соответственно по углам эйлера:  $\phi$ ,  $\theta$  и  $\psi$ .

Рассмотрим пример получения матрицы поворота вокруг оси  $X$ . При повороте вокруг оси  $X$  координата  $x$  не меняется. Рассмотрим проекцию на плоскость  $OYZ$ , представленную на [рисунке 2б](#).

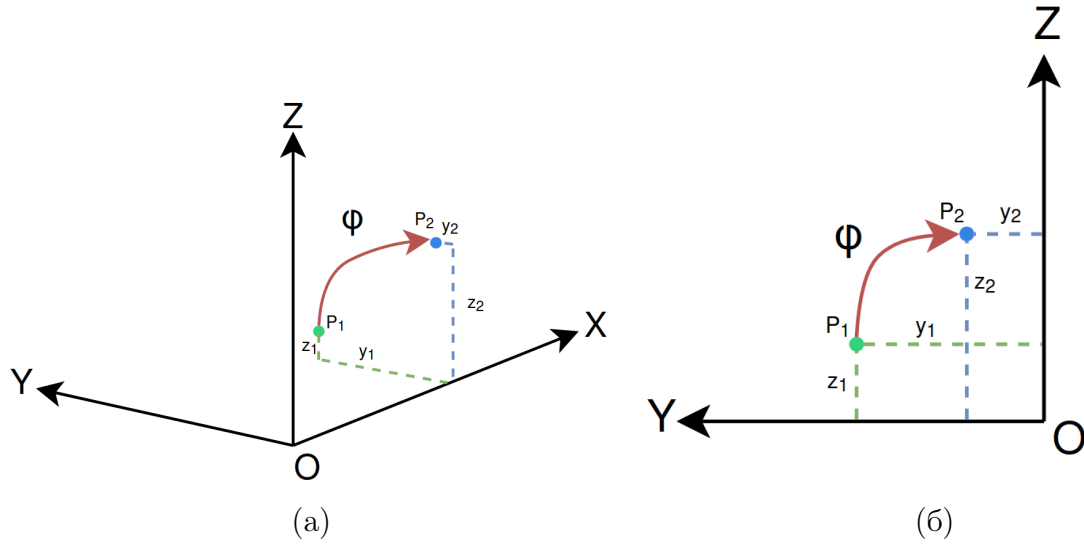


Рисунок 2 — Поворот точки вокруг оси. На рисунке (а) поворот точки в трёхмерном пространстве. На рисунке (б) поворот точки в плоскости  $OYZ$ .

В проекции  $OYZ$  перейдем к полярным координатам: расстояние от начала координат  $r = \sqrt{y_1^2 + z_1^2}$ , начальный угол будет  $\alpha_1 = \arctan(\frac{y_1}{z_1})$ .

После поворота на угол  $\phi$  получим новый угол  $\alpha_2 = \alpha_1 + \phi$ , а расстояние от начала координат не изменилось.

Выразим новые координаты  $y_2, z_2$  через расстояние от начала координат и угол  $\alpha_2$ . Получим:  $y_2 = r \cos(\alpha_2)$ ,  $z_2 = r \sin(\alpha_2)$ . Где  $\cos(\alpha_2) = \cos(\alpha_1 + \phi) = \cos(\alpha_1) \cos(\phi) - \sin(\alpha_1) \sin(\phi)$  и  $\sin(\alpha_2) = \sin(\alpha_1) \cos(\phi) + \cos(\alpha_1) \sin(\phi)$ .

В итоге получаем следующую систему для перехода при повороте на угол  $\phi$  вокруг оси  $X$ :

$$\begin{cases} x_2 = x_1 \\ y_2 = r(\cos(\alpha_1) \cos(\phi) - \sin(\alpha_1) \sin(\phi)) \\ z_2 = r(\sin(\alpha_1) \cos(\phi) + \cos(\alpha_1) \sin(\phi)) \end{cases} \quad (1)$$

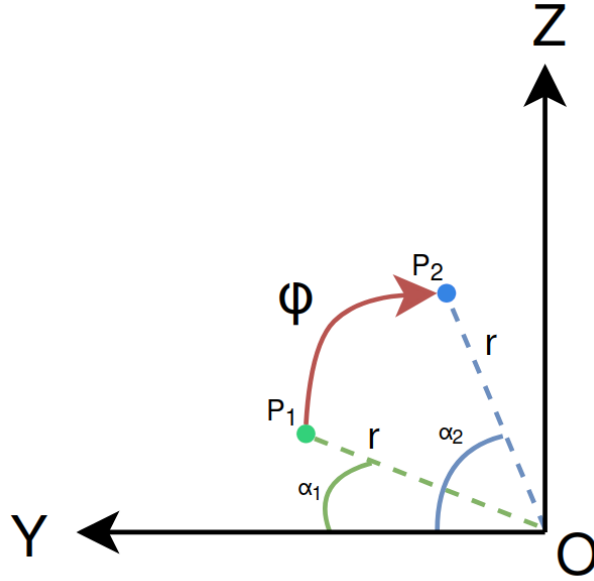


Рисунок 3 — Проекция на плоскость OYZ в полярных координатах

Заменяем  $\cos(\alpha_1) = \frac{y_1}{r}$  и  $\sin(\alpha_1) = \frac{z_1}{r}$ :

$$\begin{cases} x_2 = x_1 \\ y_2 = r \left( \frac{y_1}{r} \cos(\phi) - \frac{z_1}{r} \sin(\phi) \right) \\ z_2 = r \left( \frac{z_1}{r} \cos(\phi) + \frac{y_1}{r} \sin(\phi) \right) \end{cases} \quad (2)$$

Упростим выражения:

$$\begin{cases} x_2 = x_1 \\ y_2 = y_1 \cos(\phi) - z_1 \sin(\phi) \\ z_2 = y_1 \sin(\phi) + z_1 \cos(\phi) \end{cases} \quad (3)$$

В матричном виде:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (4)$$

Аналогичные матрицы поворота используются для поворота вокруг осей  $Y$  и  $Z$ .

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (5)$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Матрица направляющих косинусов  $R$  получается путем перемножения матриц поворота вокруг осей  $Z, Y, X$ :

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}, \quad (7)$$

где  $C = \cos$ ,  $S = \sin$ . Матрица  $R$  позволяет перейти от глобальной к локальной системы координат, а чтобы осуществить обратное преобразование необходимо взять обратную матрицу  $R^{-1}$ , но так как матрица  $R$  - ортогональная, то  $R^{-1} = R^T$ .

### 1.3 Поступательное и вращательное движение

В глобальной системе координат вводятся векторы линейных и угловых скоростей.

$$v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (8)$$

где  $v, \omega$  — векторы линейных и угловых скоростей соответственно в локальной системе координат, а  $v_x, v_y, v_z$  — проекции линейных скоростей на соответствующие оси,  $\omega_x, \omega_y, \omega_z$  — угловые скорости вокруг соответствующих осей.

#### 1.3.1 Поступательное движение и силы сопротивления

Поступательное движение квадрокоптера описывается с помощью уравнения Ньютона-Эйлера.

$$m\ddot{r} = F_\Sigma, \quad (9)$$



где  $m$  — масса квадрокоптера,  $r$  — вектор положения центра масс,  $F_{\Sigma}$  — сумма сил, действующих на квадрокоптер.

Квадрокоптер движется за счет силы тяги, которая создается при помощи электродвигателей, расположенных по углам рамы. Силу тяги можно представить в следующем виде:

$$T = T_1 + T_2 + T_3 + T_4 \quad (10)$$

$$T_i = k_i \omega_i^2, \quad (11)$$

где  $T$  — суммарная сила тяги, которая направлена вдоль оси  $OZ$ , поэтому представляет собой вектор  $\begin{bmatrix} 0 & 0 & T \end{bmatrix}^T$ .  $T_i$  — сила тяги отдельного электродвигателя с пропеллером.  $k$  — коэффициент тяги, который можно определить экспериментально, из данных об электродвигателе и пропеллере или рассчитать приблизительное значение по формуле, предложенной в [Lysukho]  $k_i = \frac{1}{2} \rho C_{\gamma} S_{prop_i} r_{prop_i}^2$ , где  $\rho$  — плотность воздуха,  $C_{\gamma}$  — коэффициент подъемной силы,  $S_{prop_i}$  — площадь окружности, которую описывает винт радиусом  $r_{prop_i}$ .

Также на квадрокоптер действует сила тяжести и аэродинамические силы. Аэродинамические силы включают в себя флаттер лопастей, индуктивное сопротивление и лобовое сопротивление.

Флаттер лопастей — это явление, возникающее из-за прогиба лопастей пропеллера при движения квадрокоптера вперед. Лопасти, движущиеся вперед, создают больше подъемной силы, чем лопасти, движущиеся назад, что приводит к наклону подъемной силы и созданию силы сопротивления. В статье [1] представлено моделирование лопастного флаттера следующим образом:

$$F_{flap} = -T A_{flap} v, \quad (12)$$

где  $A_{flap}$  — матрица флаттера, равная:

$$A_{flap} = \begin{bmatrix} c_a & -c_b & 0 \\ c_b & c_a & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (13)$$

где  $c_a$  и  $c_b$  — коэффициенты флаттера. Однако коэффициент  $c_b$  связан с поперечным флаттером лопастей, поэтому им можно пренебречь, так как продольный флаттер намного больше, чем поперечный.

Индуктивное сопротивление — это сопротивление, вызванное вихрями на торцах пропеллеров. Вихри образуются перетеканием воздуха из области под лопастью в область над лопастью.

$$F_{ind} = -T A_{ind} v, \quad (14)$$

где  $A_{ind} = \text{diag}(c_{dx}, c_{dy}, 0)$ ,  $c_{dx}, c_{dy}$  — коэффициенты индуктивного сопротивления. Однако в силу симметричности квадрокоптера можно принять, что  $c_{dx} = c_{dy} = c_d$

Лобовое сопротивление — это сопротивление, вызванное трением воздуха о поверхность квадрокоптера. Лобовое сопротивление будем учитывать, используя стандартное выражение для расчета сопротивления

$$F_{drag} = -\frac{1}{2} \rho C_{drag} S \|v\| v, \quad (15)$$

$C_{drag}$  — коэффициент сопротивления,  $S$  — характерная площадь поверхности квадрокоптера.  $\|v\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$

Сумма всех сил действующих на квадрокоптер в глобальной системе координат с учетом силы тяжести, которая действует вдоль оси  $OZ$ , будет выглядеть следующим образом:

$$m\ddot{r} = TR - m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + F_{flap} + F_{ind} + F_{drag} \quad (16)$$

$$m\ddot{r} = TR - m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - TRA_{ind}R_T\dot{r} - TRA_{flap}R_T\dot{r} - \frac{1}{2}\rho C_{drag}S\sqrt{v_x^2 + v_y^2 + v_z^2}\dot{r} \quad (17)$$

Из этого уравнения можно получить систему уравнений для выражения положения квадрокоптера в глобальной системе координат:

$$\begin{cases} \ddot{r}_x = \frac{T(\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) - T(RA_{ind}R^T\dot{r})_x - T(RA_{flap}R^T\dot{r})_x - \frac{1}{2}\rho C_{drag}S\|\dot{r}\|\dot{r}_x}{m} \\ \ddot{r}_y = \frac{T(\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) - T(RA_{ind}R^T\dot{r})_y - T(RA_{flap}R^T\dot{r})_y - \frac{1}{2}\rho C_{drag}S\|\dot{r}\|\dot{r}_y}{m} \\ \ddot{r}_z = \frac{T \cos \theta \cos \phi - T(RA_{ind}R^T\dot{r})_z - T(RA_{flap}R^T\dot{r})_z - \frac{1}{2}\rho C_{drag}S\|\dot{r}\|\dot{r}_z}{m} - g \end{cases} \quad (18)$$

### 1.3.2 Вращательное движение

Для перевода угловых скоростей из глобальной системы координат в локальной нужно ввести матрицу  $W$ . Для вывода этой матрицы последовательно рассмотрим поворот вдоль каждой оси. Также, как и в случае с матрицей поворота используется стандартная последовательность вращений: сначала вокруг оси  $Z$ , затем вокруг оси  $Y$  и затем вокруг оси  $X$ .

Пусть исходная система координат, совпадающая с глобальной будет  $X_0, Y_0, Z_0$ , соответственно, после первого поворота на угол  $\psi$  система координат будет  $X_1, Y_1, Z_1$ , после второго поворота на угол  $\theta$  система координат будет  $X_2, Y_2, Z_2$  и после третьего поворота на угол  $\phi$  система координат будет  $X_3, Y_3, Z_3$ .

Локальной системой координат является  $X_3, Y_3, Z_3$ , в которой угловые скорости будут производными по соответствующим углам Эйлера, однако угловые скорости  $\dot{\theta}$  и  $\dot{\psi}$  необходимо привести к локальной системе координат, так как они выполняются в других системах координат:  $X_2, Y_2, Z_2$  и  $X_1, Y_1, Z_1$ . Чтобы получить угловую скорость вокруг оси  $Y_3$  нужно привести систему координат в состояние до поворота на угол  $\phi$ , соответственно умножить вектор угловой скорости на матрицу поворота  $R_x(\phi)^{-1} = R_x(-\phi)$ , аналогично для получения угловой скорости вокруг оси  $Z_3$  необходимо также привести систему координат до поворота на угол  $\theta$  умножив на матрицу  $R_y(-\theta)$ .

Угловая скорость  $\omega$  в локальной системе по компонентам:

$$\omega_\phi = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

$$\omega_\theta = R_x(-\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} = \dot{\theta} \begin{bmatrix} 0 \\ \cos \phi \\ -\sin \phi \end{bmatrix} \quad (20)$$

$$\omega_\psi = R_x(-\phi)R_y(-\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \dot{\psi} \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} \quad (21)$$

Вектор угловой скорости в локальной системе координат  $\omega = \omega_x + \omega_y + \omega_z$ :

$$\omega = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (22)$$

Для перевода из локальной в глобальную  $W^{-1}$

$$W^{-1} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix}. \quad (23)$$

### 1.3.3 Крутящие моменты

Маневрирование квадрокоптера осуществляется за счет создания крутящих моментов. Крутящие моменты можно вычислить следующим образом:

$$\begin{cases} \tau_\phi = \tau_{q\phi} + \tau_{m\phi} + \tau_{p\phi} \\ \tau_\theta = \tau_{q\theta} + \tau_{m\theta} + \tau_{p\theta} \\ \tau_\psi = \tau_{q\psi}, \end{cases} \quad (24)$$

где  $\tau_{q\phi}, \tau_{q\theta}, \tau_{q\psi}$  - моменты, создаваемые разницей сил тяги, которые создаются парами винтов. Момент  $\tau_{m\phi}, \tau_{m\theta}$  - гироскопические моменты двигателей,  $\tau_{p\phi}, \tau_{p\theta}$  - гироскопические моменты винтов. Эти моменты могут быть выражены следующим образом:

$$\tau_{q\phi} = (T_3 - T_1)l \quad (25)$$

$$\tau_{q\theta} = (T_2 - T_4)l \quad (26)$$

$$\tau_{q\psi} = (T_2 + T_4 - T_1 - T_3) \quad (27)$$

$$\tau_{m\phi} = I_m\omega_\theta(\omega_1 + \omega_3 - \omega_2 - \omega_4) \quad (28)$$

$$\tau_{m\theta} = I_m\omega_\phi(\omega_2 + \omega_4 - \omega_1 - \omega_3) \quad (29)$$

$$\tau_{p\phi} = I_p\omega_\theta(\omega_1 + \omega_3 - \omega_2 - \omega_4) \quad (30)$$

$$\tau_{p\theta} = I_p\omega_\phi(\omega_2 + \omega_4 - \omega_1 - \omega_3) \quad (31)$$

Однако значения гироскопических моментов мало, поэтому ими можно пренебречь. Для вращательного движения второй закон Ньютона будет выглядеть следующим образом:

$$I\dot{\omega} + \omega \times (I\omega) = \tau, \quad (32)$$

где  $I = \text{diag}(I_x, I_y, I_z)$  - это матрица инерции квадрокоптера,  $\omega \times (I\omega)$  - описывает гироскопические силы, которые возникают при вращении квадрокоптера.

$$\begin{cases} \tau_\phi = I_x\dot{\omega}_\phi + (I_z - I_y)\omega_\theta\omega_\psi \\ \tau_\theta = I_y\dot{\omega}_\theta + (I_x - I_z)\omega_\phi\omega_\psi \\ \tau_\psi = I_z\dot{\omega}_\psi + (I_y - I_x)\omega_\phi\omega_\theta \end{cases} \quad (33)$$

Из этой системы уравнений можно выразить угловые ускорения в локальной системе координат:

$$\begin{cases} \dot{\omega}_\phi = \frac{\tau_\phi + (I_z - I_y)\omega_\theta\omega_\psi}{I_x} \\ \dot{\omega}_\theta = \frac{\tau_\theta + (I_x - I_z)\omega_\phi\omega_\psi}{I_y} \\ \dot{\omega}_\psi = \frac{\tau_\psi + (I_y - I_x)\omega_\phi\omega_\theta}{I_z} \end{cases} \quad (34)$$

#### 1.4 Итоговая система уравнений динамики квадрокоптера

Таким образом, итоговая система уравнений динамики квадрокоптера будет выглядеть следующим образом:

$$\left\{ \begin{aligned}
\ddot{r}_x &= \frac{T(\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) - T(RA_{ind} R^T \dot{r})_x - T(RA_{flap} R^T \dot{r})_x - \frac{1}{2} \rho C_{drag} S \|\dot{r}\| \dot{r}_x}{m} \\
\ddot{r}_y &= \frac{T(\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) - T(RA_{ind} R^T \dot{r})_y - T(RA_{flap} R^T \dot{r})_y - \frac{1}{2} \rho C_{drag} S \|\dot{r}\| \dot{r}_y}{m} \\
\ddot{r}_z &= \frac{T \cos \theta \cos \phi - T(RA_{ind} R^T \dot{r})_z - T(RA_{flap} R^T \dot{r})_z - \frac{1}{2} \rho C_{drag} S \|\dot{r}\| \dot{r}_z}{m} - g \\
\dot{\phi} &= \omega_\phi + \tan \theta (\omega_\theta \sin \phi + \omega_\psi \cos \phi) \\
\dot{\theta} &= \omega_\theta \cos \phi - \omega_\psi \sin \phi \\
\dot{\psi} &= \frac{\sin \phi}{\cos \theta} \omega_\theta + \frac{\cos \phi}{\cos \theta} \omega_\psi \\
\dot{\omega}_\phi &= \frac{\tau_\phi + (I_z - I_y) \omega_\theta \omega_\psi}{I_x} \\
\dot{\omega}_\theta &= \frac{\tau_\theta + (I_x - I_z) \omega_\phi \omega_\psi}{I_y} \\
\dot{\omega}_\psi &= \frac{\tau_\psi + (I_y - I_x) \omega_\phi \omega_\theta}{I_z}
\end{aligned} \right. \quad (35)$$

## 2 АНАЛИЗ АЛГОРИТМОВ УПРАВЛЕНИЯ КВАДРОКОПТЕРАМИ

В предыдущей главе была выведена математическая модель квадрокоптера. В данной модели представлены динамические уравнения описывающие поступательное и вращательное движение квадрокоптера. Управление квадрокоптером осуществляется за счет изменения скорости вращения четырех электродвигателей, которые создают тягу. Меняя соотношения скоростей вращения электродвигателей, можно менять положение квадрокоптера в пространстве по шести степеням свободы.

В уравнениях квадрокоптера присутствует множество нелинейностей, которые осложняют синтез алгоритмов управления, в добавок к сложной модели - на квадрокоптер действуют внешние сопротивления. Поэтому для корректного управления с качественно хорошими переходными процессами необходимо синтезировать сложные системы управления.

По решению задачи слежения за траекторией написано множество статей. Для управления квадрокоптером используются классические, адаптивные и робастные регуляторы разных видов. Разные методы управления имеют свои достоинства и недостатки.

### 2.1 Классические методы управления

Классические методы управления основаны на линейных моделях системы, что позволяет использовать простые и эффективные регуляторы. Такие методы управления просты в реализации и не требуют больших вычислительных ресурсов.

В качестве одного из базовых и эффективных алгоритмов управления можно назвать LQR регулятор по линеаризованной модели квадрокоптера. Данная система управления простая в реализации, не требует тонких настроек и больших вычислительных ресурсов, однако она не всегда может обеспечить необходимую точность управления, особенно в условиях сильных внешних воздействий. Также для реализации классических методов управления необходимо линеаризовать систему, что может привести к потере точности управления в некоторых случаях.

Линеаризация системы производится, в основном двумя способами:

- в точке равновесия;
- по методу обратной связи — однако этот метод уже относится к нелинейному управлению.

Линеаризация в точке равновесия позволяет получить линейную модель системы, которая может быть использована для синтеза регуляторов. Однако такая модель имеет недостаток — она не учитывает динамику системы в других точках пространства состояний, что ограничивает ее применение, например для агрессивных маневров. В основе метода линеаризации лежит приближение нелинейных уравнений системы с помощью разложения в ряд Тейлора.

Во многих статьях посвященных решению задачи слежения за траекторией для квадрокоптера - система управления разделяется на два контура: внутренний и внешний. Внутренний контур отвечает за управление углом наклона квадрокоптера, а внешний контур отвечает за управление положением квадрокоптера в пространстве. Таким образом, при сначала рассчитывается необходимое состояние по положению, а далее следующий регулятор определяет необходимые углы крена и тангажа. Положение по рысканию может включается как во внешний контур[article], так и во внутренний[article]. Далее возможны различные комбинации алгоритмов управления для каждого из контуров.

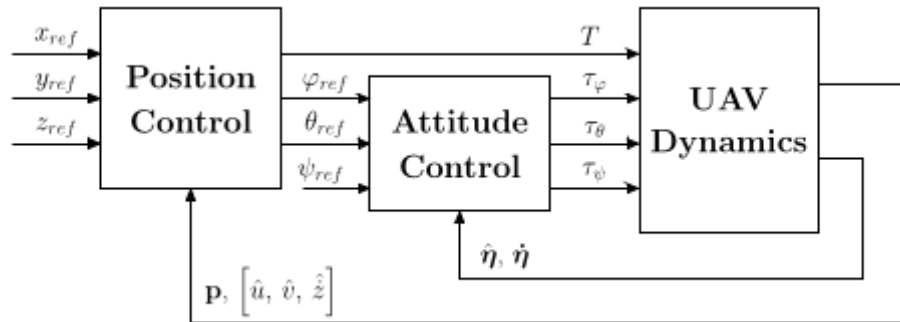


Рисунок 4 — Схема управления квадрокоптером с двумя контурами управления и линеаризацией по точке равновесия.

## 2.2 Нелинейные методы управления

В отличие от классических линейных методов управления, имеющих ограниченные возможности, системы управления, основанные на полной



нелинейной модели квадрокоптера, обеспечивают более высокую точность и гибкость. Они способны адаптироваться к быстро меняющимся условиям окружающей среды и учитывать сложную динамику системы. Однако такие регуляторы сложнее в реализации и требуют значительно больших вычислительных ресурсов.

К числу популярных нелинейных методов управления относятся:

- нелинейное управление с обратной связью (Feedback Linearization),
- пошаговое обратное управление (Backstepping),
- управление методом скользящего режима (Sliding Mode Control),
- управление на основе предсказательной модели (Model Predictive Control, MPC).

### 2.2.1 Метод обратной связи (Feedback Linearization)

Линеаризация по методу обратной связи позволяет получить более точную модель системы, которая учитывает динамику системы в разных точках пространства состояний. Основная идея метода заключается в замене управления в системе на виртуальное, которое делает систему линейной, а далее виртуальное управление обратно преобразовывается в реальное.

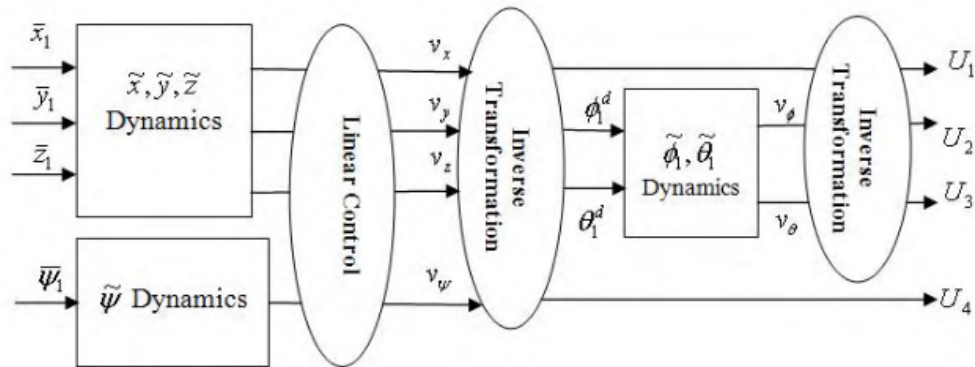


Рисунок 5 — Схема управления квадрокоптером с двумя контурами управления и линеаризацией по методу обратной связи.

В статье ‘Different linearization control techniques for a quadrotor system’ описаны алгоритмы управления квадрокоптером на основе LQR регулятора с линеаризацией сразу двумя способами — в первом варианте используется линеаризация по точке равновесия, во втором — по методу

обратной связи по схеме приведенной на [рисунке 5](#)[article]. Полученные результаты показали, что линеаризация методом обратной связи обеспечивает более высокую точность управления, преимущественно лучше справляется в задаче стабилизации квадрокоптера по углам крена и тангажа.

### 2.2.2 Метод обратного пошагового управления (Backstepping)

Backstepping — это подход к синтезу нелинейных регуляторов, который основан на пошаговом регулировании вложенных подсистем. Данный метод хорошо применим к системам с иерархической структурой, как в случае квадрокоптера, где положение зависит от ориентации, а ориентация — от управляющих моментов.

В этом методе так же, как и в рассмотренном ранее управлении в системе заменяются на виртуальные, но это управления синтезируются последовательно и как правило делается с помощью кандидата функции Ляпунова, таким образом доказывается устойчивость каждого уровня управления.

Backstepping один из самых распространенных способов управления квадрокоптером, который комбинируется с различными модификациями.

В статье ‘Robust Backstepping Sliding Mode Control for a Quadrotor Trajectory Tracking Application’ предложен двухконтурный регулятор для квадрокоптера, в котором используются Backstepping и интегральное скользящее управление [article]. Внутренний контур по ориентации реализован как робастный backstepping контроллер на скользящих режимах, а внешний контур по позиции — с помощью интегральных скользящих режимов. Основная сложность — в настройке большого числа параметров и учёте всех возмущений. Несмотря на это, система была устойчивой, даже при действии сильных помех.

### 2.2.3 Метод скользящего режима

Одним из широко используемых методов робастного управления в задачах управления квадрокоптером является метод скользящего режима (Sliding Mode Control). Данный метод обеспечивает устойчивость системы

даже при наличии существенных внешних возмущений и неопределённостей в модели.

Суть метода заключается в проектировании скользящей поверхности в пространстве состояний системы. Управляющее воздействие формируется таким образом, чтобы система как можно быстрее вышла на эту поверхность и затем оставалась на ней, двигаясь к заданному положению равновесия. Таким образом система борется с разными возмущениями, оставаясь на скользящей поверхности. Но этот метод нуждается в модификации и доработках, чтобы избежать частых переключений управления.

В статье [Labbadi et al., 2019] представлен продвинутый подход к управлению квадрокоптером на основе скользящих режимов [article]. В алгоритме есть несколько модификаций: адаптивная оценка верхних границ неопределенностей, терминальная модификация, которая позволяет системе быстро сходиться за конечное время. Система управления сложная, но результаты моделирования, приведенные в работе, впечатляют своей эффективностью. Также в работе приведено сравнение с подходами на основе Backstepping и Feedback Linearization.

#### 2.2.4 Model Predictive Control (MPC)

Одним из самых универсальных и современных алгоритмов является MPC (Model Predictive Control), который использует дискретную модель квадрокоптера для предсказания его поведения на некотором горизонте планирования. Управляющее воздействие определяется как решение задачи оптимизации. Алгоритм способен учитывать внешние возмущения, а также может быть адаптирован для агрессивного маневрирования. MPC может быть легко модифицирован, например, можно ввести ограничения на отдельные компоненты вектора состояния и управления. Основными недостатками MPC являются очень высокая вычислительная нагрузка и сложность реализации в режиме реального времени. MPC регулятор также обладает большим количеством настраиваемых параметров, такие как — веса стоимости управления, веса стоимости состояния, горизонт планирования, количество итераций для оптимизации, алгоритм оптимизации и другое. Такое обилие настраиваемых параметров с одной стороны позво-

ляет разработать наиболее подходящий регулятор для конкретных задач, а с другой стороны увеличивает сложность разработки.

В работе представлен сравнительный анализ линейного и нелинейного MPC-регуляторов для управления квадрокоптером с учётом ограничений на входы и состояния[<empty citation>]. Линейная модель получена путём линеаризации в окрестности точки зависания, а нелинейная модель построена в виде SDC-представления, где матрицы системы зависят от текущего состояния. Для оптимизации управления используется последовательный квадратичный программный решатель (SQP). По полученным результатам симуляций видно, что нелинейный MPC был гораздо эффективнее — управление меньше, а точность лучше. Но нужно опять же понимать, что регулятор по нелинейной модели требует гораздо больше вычислительной мощности.

### 3 РАЗРАБОТКА АЛГОРИТМОВ УПРАВЛЕНИЯ

#### 3.1 LQR регулятор по линеаризованной модели

В качестве одного из базовых регуляторов можно использовать LQR регулятор по линеаризованной модели. Для синтеза регулятора определим модель системы в форме вход-состояние-выход.

Необходимо осуществлять управление по координатам  $r_x$ ,  $r_y$ ,  $r_z$  в соответствии с этим вектор выхода системы будет выглядеть следующим образом:

$$Y = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \quad (36)$$

Система (??) в матричном виде вход-состояние-выход будет выглядеть следующим образом:

$$\begin{cases} \dot{X} = AX + BU + D \\ Y = CX, \end{cases} \quad (37)$$

Вектор состояния расширен, добавлением положением по координатам  $r_x$ ,  $r_y$  и  $r_z$ .

$$X = \begin{bmatrix} r_x & r_y & r_z & \dot{r}_x & \dot{r}_y & \dot{r}_z & \phi & \theta & \psi & \omega_\phi & \omega_\theta & \omega_\psi \end{bmatrix}^T \quad (38)$$

Вектор управления:

$$U = \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (39)$$

Матрица  $A$  будет выглядеть следующим образом:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{drag1:1} & A_{drag1:2} & A_{drag1:3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{drag2:1} & A_{drag2:2} & A_{drag2:3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{drag3:1} & A_{drag3:2} & A_{drag3:3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T_\theta S_\phi & T_\theta C_\phi & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_\phi & -S_\phi & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{S_\phi}{C_\theta} & \frac{C_\phi}{C_\theta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(I_z - I_y)\omega_\theta}{I_x} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(I_x - I_z)\omega_\phi}{I_y} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(I_y - I_x)\omega_\phi}{I_z} & 0 & 0 \end{bmatrix}, \quad (40)$$

где  $C_\phi = \cos \phi$ ,  $S_\phi = \sin \phi$ ,  $C_\theta = \cos \theta$ ,  $S_\theta = \sin \theta$ ,  $T_\theta = \tan \theta$ , матрица  $A_{drag}$

$$A_{drag} = -\frac{T}{m}R(A_{ind} + A_{flap})R^T - \frac{\rho C_{drag} S}{2m} \|\dot{r}\| I \quad (41)$$

Матрица  $B$ :

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi}{m} & 0 & 0 & 0 \\ \frac{\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi}{m} & 0 & 0 & 0 \\ \frac{\cos \theta \cos \phi}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \quad (42)$$

Матрица  $D$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (43)$$

Матрица  $C$ :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (44)$$

### 3.1.1 Линеаризация у точки равновесия

Для синтеза LQR регулятора необходимо сначала линеаризовать квадрокоптер.

Производится линеаризация около точки равновесия, в которой угловые скорости и ускорения равны нулю. Состояние системы, которое будет соответствовать точке равновесия будет определяться следующим образом:

$$\bar{X} = \begin{bmatrix} \bar{r}_x & \bar{r}_y & \bar{r}_z & 0 & 0 & 0 & 0 & 0 & \bar{\psi} & 0 & 0 & 0 \end{bmatrix} \quad (45)$$

Линеаризация системы у точки (45):

$$\bar{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{A}_{drag1:1} & \bar{A}_{drag1:2} & \bar{A}_{drag1:3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{A}_{drag2:1} & \bar{A}_{drag2:2} & \bar{A}_{drag2:3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{A}_{drag3:1} & \bar{A}_{drag3:2} & \bar{A}_{drag3:3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (46)$$

где

$$\bar{A}_{drag} = -\frac{T}{m}(A_{ind} + A_{flap}) \quad (47)$$

$$\bar{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \quad (48)$$

Матрицы  $C$  и  $D$  остаются без изменений.

LQR регулятор минимизирует критерий качества, который определяется как:

$$J = \int_0^\infty (X^T Q X + U^T R U) dt, \quad (49)$$

Для синтеза LQR регулятора необходимо определить матрицы  $Q$  и  $R$ , которые определяют весовые коэффициенты для состояния и управления соответственно. Пусть  $Q$  и  $R$  будут определены следующим образом:



$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (50)$$

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (51)$$

Для обеспечения минимизации критерия качества  $J$  используется решение уравнения Риккати:

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (52)$$

Из этого уравнения можно получить  $P$  — матрица, с помощью которой можно найти матрицу  $K$  для LQR регулятора:

$$K = R^{-1} B^T P \quad (53)$$

Управление в системе будет осуществляться по следующему закону:

$$U = -KX \quad (54)$$

Рассчитанные коэффициенты:

$$K = \begin{bmatrix} 0 & 0 & 5.48 & 0 & 0 & 2.67 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3.16 & 0 & 0 & -1.54 & 0 & 3.69 & 0 & 0 & 0.13 & 0 & 0 \\ 3.16 & 0 & 0 & 1.54 & 0 & 0 & 0 & 3.69 & 0 & 0 & 0.13 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.16 & 0 & 0 & 0.17 \end{bmatrix}$$

### 3.2 LQR регулятор с линеаризацией обратной связью

В линеаризации обратной связью необходимо упростить уравнения квадрокоптера вводом новых виртуальных управлений. Управления будут выглядеть следующим образом:

$$\begin{cases} \nu_x = \frac{1}{m}(\cos \phi \sin \theta + \sin \psi \sin \phi)U_1 \\ \nu_y = \frac{1}{m}(\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi)U_1 \\ \nu_z = -g + \frac{1}{m} \cos \theta \cos \phi U_1 \\ \nu_\phi = \frac{I_z - I_y}{I_x} \omega_\psi \omega_\theta + \frac{1}{I_x} U_2 \\ \nu_\theta = \frac{I_x - I_z}{I_y} \omega_\phi \omega_\psi + \frac{1}{I_y} U_2 \\ \nu_\psi = \frac{I_y - I_x}{I_z} \omega_\phi \omega_\theta + \frac{1}{I_z} U_2 \end{cases} \quad (54)$$

Введение ошибок состояния:

$$\begin{aligned} \tilde{r}_x &= r_x - r_x^d, & \tilde{v}_x &= v_x - v_x^d, \\ \tilde{r}_y &= r_y - r_y^d, & \tilde{v}_y &= v_y - v_y^d, \\ \tilde{r}_z &= r_z - r_z^d, & \tilde{v}_z &= v_z - v_z^d, \\ \tilde{\phi} &= \phi - \phi^d, & \tilde{\omega}_\phi &= \omega_\phi - \omega_\phi^d, \\ \tilde{\theta} &= \theta - \theta^d, & \tilde{\omega}_\theta &= \omega_\theta - \omega_\theta^d, \\ \tilde{\psi} &= \psi - \psi^d, & \tilde{\omega}_\psi &= \omega_\psi - \omega_\psi^d. \end{aligned} \quad (55)$$

Динамика ошибок преобразуется в линейные подсистемы:

$$\begin{cases} \dot{\tilde{r}}_x = \tilde{v}_x, & \dot{\tilde{r}}_y = \tilde{v}_y, & \dot{\tilde{r}}_z = \tilde{v}_z, \\ \dot{\tilde{v}}_x = \nu_x, & \dot{\tilde{v}}_y = \nu_y, & \dot{\tilde{v}}_z = \nu_z, \end{cases} \quad (56)$$

$$\begin{cases} \dot{\tilde{\phi}} = \tilde{\omega}_\phi, & \dot{\tilde{\theta}} = \tilde{\omega}_\theta, & \dot{\tilde{\psi}} = \tilde{\omega}_\psi, \\ \dot{\tilde{\omega}}_\phi = \nu_\phi, & \dot{\tilde{\omega}}_\theta = \nu_\theta, & \dot{\tilde{\omega}}_\psi = \nu_\psi. \end{cases} \quad (57)$$

Для каждой подсистемы применяется LQR регулятор:

$$\begin{cases} \nu_x = -k_1^x \tilde{r}_x - k_2^x \tilde{v}_x, \\ \nu_y = -k_1^y \tilde{r}_y - k_2^y \tilde{v}_y, \\ \nu_z = -k_1^z \tilde{r}_z - k_2^z \tilde{v}_z, \\ \nu_\phi = -k_1^\phi \tilde{\phi} - k_2^\phi \tilde{\omega}_\phi, \\ \nu_\theta = -k_1^\theta \tilde{\theta} - k_2^\theta \tilde{\omega}_\theta, \\ \nu_\psi = -k_1^\psi \tilde{\psi} - k_2^\psi \tilde{\omega}_\psi. \end{cases} \quad (58)$$

Для подсистеме по координате  $r_x$  синтез LQR регулятор выглядит следующим образом: Матрицы состояния и управления:

$$A_x = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_x = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (59)$$

Осуществляется выбор матриц  $Q$  и  $R$  для критерия качества. И решается уравнение Риккати:

$$A_x^T P + P A_x - P B_x R^{-1} B_x^T P + Q = 0 \quad (60)$$

Откуда получаем коэффициенты  $k_1^x$  и  $k_2^x$ :

$$k_x = R^{-1} B_x^T P \quad (61)$$

Аналогично синтезируются регуляторы для остальных подсистем.

Полученные виртуальные управления обратно преобразуются в реальные управления:

$$\begin{cases} U_1 = \frac{m(\nu_z + g)}{\cos \phi \cos \theta}, \\ U_2 = I_x \nu_\phi - (I_y - I_z) \omega_\theta \omega_\psi, \\ U_3 = I_y \nu_\theta - (I_x - I_z) \omega_\phi \omega_\psi, \\ U_4 = I_z \nu_\psi - (I_x - I_y) \omega_\phi \omega_\theta. \end{cases} \quad (62)$$

Для управлений были выбраны различные матрицы  $Q$  и  $R$ , полученные коэффициенты регулятора:

### 3.3 Nonlinear MPC

Nonlinear MPC — один из самых эффективных регуляторов с точки зрения оптимизации управления и точности. Суть регулятора довольно простая, на вход поступает вектор состояний квадрокоптера и желаемое состояние на некоторое количество шагов вперед. Количество шагов называется горизонтом предсказания. Далее внутри регулятора происходит моделирование поведения квадрокоптера с управлением, которое может быть изначально проинициализировано или получено с предыдущего шага. Затем происходит минимизация заданного критерия качества, который обычно имеет следующий вид:

$$J = \sum_{k=0}^{N-1} (\|X_k - X_k^d\|_Q^2 + \|U_k\|_R^2), \quad (63)$$

где  $X_k$  — состояние системы на  $k$ -м шаге,  $X_k^d$  — желаемое состояние на  $k$ -м шаге,  $U_k$  — управление на  $k$ -м шаге,  $Q$  и  $R$  — весовые матрицы, определяющие важность отклонения состояния и управления соответственно,  $N$  — горизонт предсказания.

Минимизация критерия качества осуществляется с учетом ограничений на динамику системы, которые задаются в виде:

$$X_{k+1} = f(X_k, U_k), \quad (64)$$

где  $f(X_k, U_k)$  — нелинейная модель квадрокоптера, а также ограничений на управление и состояния, например:

$$U_{\min} \leq U_k \leq U_{\max}, \quad X_{\min} \leq X_k \leq X_{\max}. \quad (65)$$

Решение задачи оптимизации дает последовательность управлений  $\{U_0, U_1, \dots, U_{N-1}\}$ , из которых на выходе регулятора используется только первое управление  $U_0$ . Это управление подается на квадрокоптер, после чего процесс повторяется на следующем временном шаге с обновленным состоянием системы.

Сам регулятор можно очень расширяемый и модифицируемый можно вводить ограничения, штрафные стоимости за быстрые переключения

управления, модифицировать модель объекта, например учитывать характер внешних возмущений.

В среде Matlab/Simulink есть реализация Nonlinear MPC регулятора, но в текущей работе было принято решение полностью написать функцию Nonlinear MPC регулятора с введением некоторых модификаций, и чтобы в дальнейшем было проще перенести реализацию на реальную систему.

Система управления не разделена на контуры, и реализует вышеописанную логику. Схема регулятора представлена на рисунке 6. Для минимизации критерия  $J$  используется градиентный спуск, в математическом представлении:

$$U_{k+1} = U_k - \nabla J(U_k) \quad (66)$$

Также введены дополнительные мягкие ограничения на углы  $\phi$  и  $\theta$ , чтобы квадрокоптер не переворачивался в воздухе.

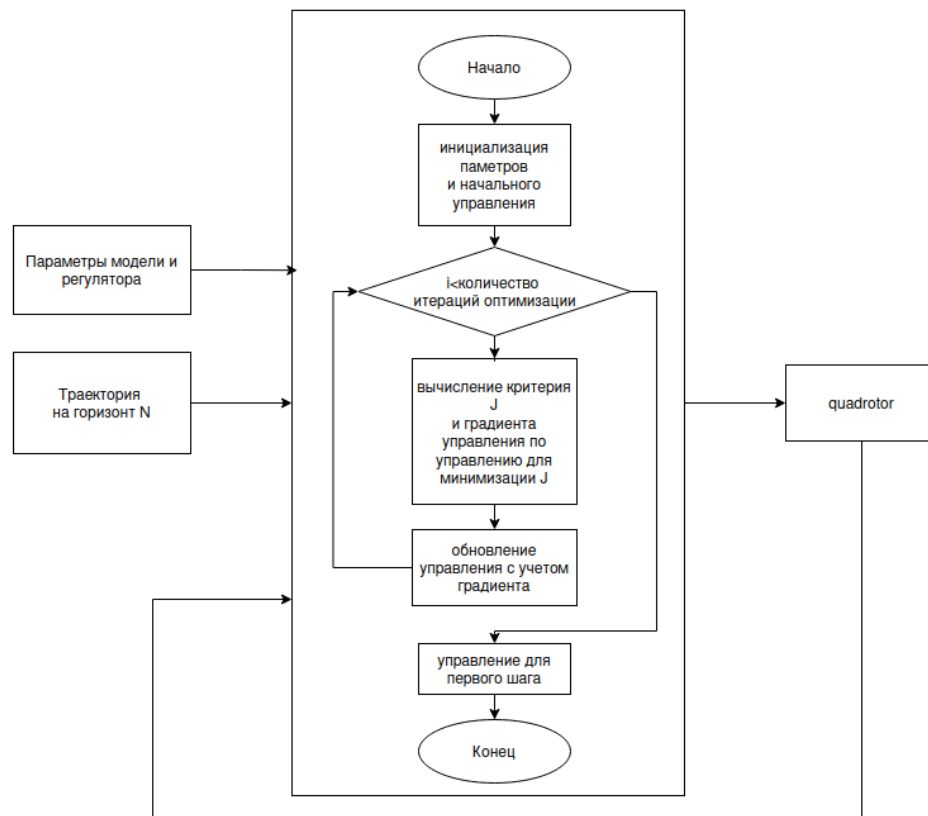


Рисунок 6 — Схема регулятора Nonlinear MPC

## 4 МОДЕЛИРОВАНИЕ В СРЕДЕ MATLAB/SIMULINK