Санкт-Петербургский государственный университет

Математико-механический факультет

Кафедра теоретической кибернетики

Никитин Денис Александрович

Идентификация параметров квадрокоптера

Бакалаврская работа

Научный руководитель:

д. т. н., проф. А. Л. Фрадков

Рецензент:

д. ф.-м. н., проф. А. С. Матвеев

Санкт-Петербург

2016

SAINT-PETERSBURG STATE UNIVERCITY

Mathematics and Mechanics

Theoretical Cybernetics

Denis Nikitin

Identification of Quadrotor Parameters

Bachelor's Thesis

Scientific supervisor:

Dr. Sc. Eng., Prof. Alexander L. Fradkov

Reviewer:

Dr. Sc. Phys.Math., Prof. Alexey S. Matveev

Saint-Petersburg

2016

Оглавление

Введение

В последнее время в обществе всё большую роль начинают играть беспилотные летательные аппараты, в особенности самые доступные из них — квадрокоптеры. Их популярность в последние годы обеспечена простотой конструкции и появлением большого количества различных систем управления, многие из которых являются open-source программами.

Существует множество задач, которые можно решать с помощью квадрокоптеров, в том числе задач научных. К примеру, работы [1], [2], [3] демонстрируют обучение роботов эффективным траекториям полета, работы [4] и [5] — кооперативному поведению, а [6], [7] и [8] — навигации при помощи камер и RGBD-сенсоров. Оценка ориентации подобных машин обычно осуществляется модификациями комплементарного фильтра ([9] и [10]) или же расширенным фильтром Калмана ([11]).

Для управления квадрокоптером обзор основных существующих решений можно найти в статье [14]. Однако, можно выделить два общих свойства, присущие этим системам: они все работают на PID-регуляторах, и они все используют углы Эйлера в качестве ошибки, которую стремится минимизировать регулятор.

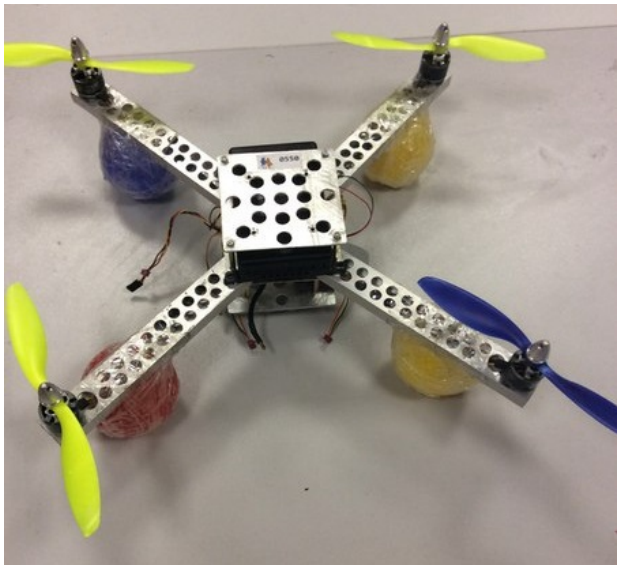
Проблема использования углов Эйлера заключается в том, что этот способ имеет сингулярность, а значит, в некоторых областях пространства ориентаций не будет работать. К примеру, ни одна из общедоступных систем не способна сделать полностью контролируемый flip (сальто, переворот), не отключая при этом основной регулятор. В отличие от углов Эйлера, кватернионы не имеют подобных проблем, хотя их редко используют для создания систем управления квадрокоптерами. Примером системы, полностью основанной на кватернионах, может являться [15].

Проблема же PID-регулятора в том, что для каждого конкретного робота нужно заново подбирать коэффициенты. К open-source продуктам прилагаются специальные инструкции о том, как это делать, однако всё равно, такой подход снижает эффективность системы, так как не позволяет достичь оптимальных параметров. Примеры синтеза PID-регулятора для квадрокоптера можно найти в работах [16], [17], [18].

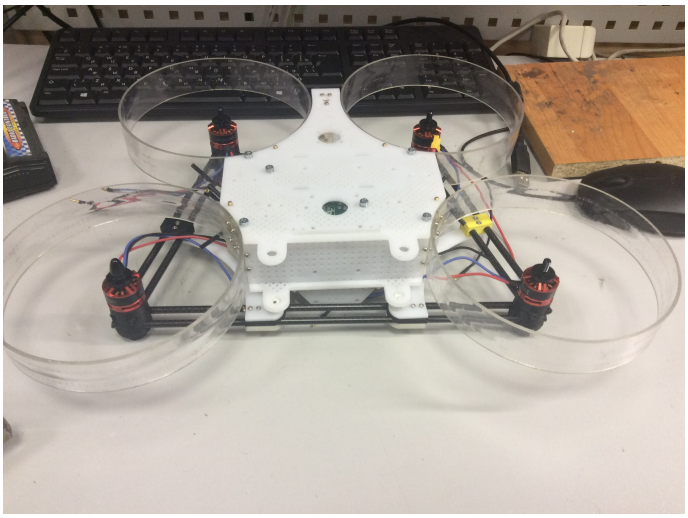
В связи с этими проблемами была создана своя система стабилизации, использующая кватернионы в качестве ошибки регулятора, и идею, похожую на линеаризацию обратной связью, для синтеза самого регулятора. В результате получился регулятор, большинство параметров которого являются в точности физическими параметрами конкретного квадрокоптера, а оставшиеся коэффициенты не зависят от робота и могут быть подобраны единожды для всего класса машин.

Также, некоторые из физических параметров конкретного робота часто бывает сложно измерить, или же они могут меняться со временем прямо в полете. Поэтому особенно большую ценность приобретает система, позволяющяя идентифицировать неизвестные параметры и использовать их в системе стабилизации. Существующие же системы идентификации, к примеру [12] и [13], полагаются на стандартные PID-регуляторы и углы Эйлера.

Данная работа состоит из четырёх частей. В первой описан вывод уравнений динамики квадрокоптера. Вторая часть посвящена системе стабилизации и доказательству её устойчивости в малом. В третьей части приводится система адаптации и идентификации некоторых параметров квадрокоптера, ключевых для устойчивого полета. Последняя часть описывает проведенное моделирование общей системы.



Робот Квадракон, контроллер FlyMaple



Робот Бонд, контроллер ТРИК

Рис. 1. Тестовые самодельные квадрокоптеры

Надо также сказать, что система стабилизации была проверена на двух настоящих, самодельных коптерах, один из которых работал на контроллере FlyMaple, второй — на контроллере ТРИК. Разработанная в данной работе система адаптации призвана решить некоторые проблемы, имевшиеся у данных роботов.

1. Вывод модели

1.1. Ориентация квадрокоптера

В трехмерном пространстве ориентация может быть описана несколькими способами, но одним из самых удобных является запись через кватернионы. Пусть у нас есть абсолютная система координат XYZ, тогда для системы $X'Y'Z'$, связанной с квадрокоптером, существует единственная ось u и единственный угол ϕ такие, что, если повернуть штрихованную систему вокруг оси u на угол ϕ, получится исходная система. Тогда кватернионом, описывающим поворот $X'Y'Z'$ относительно XYZ, называется

$$q = \left(\cos\frac{\phi}{2},\ u_x \sin\frac{\phi}{2},\ u_y \sin\frac{\phi}{2},\ u_z \sin\frac{\phi}{2}\right).$$

Заметим, что $\|q\| = 1$ для любых u и ϕ. Пусть теперь нам задан вектор v' в системе $X'Y'Z'$. Для его представления v в системе XYZ будет выполнено равенство

$$V = q * V' * q^*,$$

где V и V' – кватернионные представления векторов v и v', звездочка обозначает кватернионное произведение, а q^* – сопряженный к q кватернион.

Если система $X'Y'Z'$ вращается относительно XYZ с угловой скоростью ω, то закон изменения кватерниона поворота q будет следующим:

$$\dot{q} = \frac{1}{2} q * \Omega, \tag{1}$$

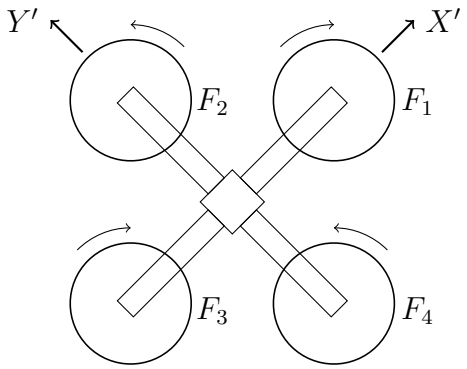
где Ω – кватернионное представление ω. В данном уравнении ω – это координаты угловой скорости в системе, связанной с квадрокоптером. Таким образом, q и ω могут стать частью вектора состояния динамической системы, описывающей квадрокоптер. Кроме того, именно эту угловую скорость мы можем измерять при помощи гироскопов, установленных на робота, и использовать её для оценки q. Вывод уравнения (1), а также дополнительные сведения о кватернионной арифметике можно посмотреть в [19].

1.2. Модель динамики квадрокоптера

Через уравнение Ньютона-Эйлера можно описать динамику вращательного движения квадрокоптера в его собственной системе отсчета. Уравнение в общем виде выглядит так:

$$I\dot{\omega} = M - \omega \times I\omega,$$

где I – тензор моментов инерции, а M – момент сил. Вывод уравнения можно посмотреть в [20].



Перейдем к описанию моментов сил, действующих на квадрокоптер. Предположим, что робот имеет форму креста, оси X' и Y' направлены вдоль лучей, а ось Z' – наверх. Расстояние от центра масс до винтов обозначим за L. Тогда моменты сил вдоль осей X' и Y' будут равны $L(F_2 - F_4)$ и $L(F_3 - F_1)$ соответственно. Здесь F_i – сила тяги, создаваемая i-м винтом. Сила тяги винта квадратично зависит от скорости вращения винта v_i. Обозначим коэффициент пропорциональности C. Тогда $F_i = Cv_i^2$. Вращающий момент Q_i также пропорционален квадрату скорости винта, а значит, пропорционален силе тяги F_i. Назовем коэффициент пропорциональности ξ. Он может быть приближенно вычислен по формуле

$$\xi = k\sqrt{\frac{C}{2\rho A}} = \sqrt{\frac{F^{max}}{2\rho A}}\frac{k}{v^{max}},$$

где ρ – плотность воздуха, A – площадь диска вращения винта, k – коэффициент "потерь мощности", приблизительно $k \approx 1.15$, F^{max} и v^{max} – максимальная тяга и максимальная скорость вращения винта. Данная формула носит приближенный характер, и полученный коэффициент требует уточнения. Для более детального изучения аэродинамики винтов см. [21].

Теперь мы можем записать полный момент сил, действующий на квадрокоптер. Согласно рис.1, винты 1 и 3 вращаются по часовой стрелке, винты 2 и 4 – против, поэтому соответству-

ющие вращающие моменты ξF_i будут входить в итоговый момент сил вдоль оси Z' с разными знаками:

$$M = \begin{pmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ \xi(F_1 + F_3 - F_4 - F_2) \end{pmatrix}$$

Запишем далее уравнения динамики поступательного движения робота. На него действуют две силы – сила тяжести mg, направленная вниз против оси Z, и суммарная сила тяги $\sum_{i=1}^{4} F_i$, направленная вдоль оси Z'. Определим вектор P положения робота в абсолютной системе отсчета. Тогда после поворота суммарной силы тяги при помощи кватерниона q, второй закон Ньютона дает уравнение

$$\ddot{P} = q * \begin{pmatrix} 0 \\ 0 \\ \frac{1}{m}(F_1 + F_2 + F_3 + F_4) \end{pmatrix} * q^* - \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix}$$

Осталось еще написать уравнения динамики двигателей. Пусть управляющее напряжение на моторе i равно u_i, коэффициент связи напряжения и скорости – k_v, а характеристика задержки – k_t. Тогда уравнение, описывающее разгон двигателя, выглядит так:

$$\dot{v}_i = \frac{k_v u_i - v_i}{k_t}$$

Этим описание модели квадрокоптера завершено, мы можем выписать полную систему уравнений:

$$\begin{aligned} \dot{q} &= \frac{1}{2} q * \omega \\ I\dot{\omega} &= \begin{pmatrix} LC(v_2^2 - v_4^2) \\ LC(v_3^2 - v_1^2) \\ \xi C(v_1^2 + v_3^2 - v_4^2 - v_2^2) \end{pmatrix} - \omega \times I\omega \\ \ddot{P} &= q * \begin{pmatrix} 0 \\ 0 \\ \frac{C}{m}(v_1^2 + v_2^2 + v_3^2 + v_4^2) \end{pmatrix} * q^* - \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \\ \dot{v}_i &= \frac{k_v u_i - v_i}{k_t}, \qquad i \in \{1..4\} \end{aligned}$$

1.3. Упрощение модели

Полученная модель существенно нелинейна, поэтому далее будет сделано несколько упрощающих предположений. В первую очередь, скорость реакции моторов обычно гораздо выше, чем скорость движения самого робота. Поэтому динамикой разгона моторов можно пренебречь. Это влечет за собой равенство $v_i = k_v u_i$. Подставив v_i в выражение для F_i, получим $F_i = Ck_v^2 u_i^2$. Назовем новый коэффициент пропорциональности $K = Ck_v^2$. Замечая, далее, что управления входят во все выражения только с квадратом, сделаем замену $U_i = u_i^2$. Таким образом, $F_i = KU_i$.

Второе упрощение заключается в том, что мы не наблюдаем и не пытаемся управлять находящимися в горизонтальной плоскости компонентами вектора P. Обозначим $H = P_z$. Если расписать кватернионное произведение в уравнении динамики поступательного движения, то видно, что Z-компонента повернутого вектора суммарной тяги, имеющего в свою очередь единственную ненулевую компоненту вдоль оси Z', может быть вычислена умножением суммарной тяги на коэффициент $k_{mod} = 1 - 2q_x^2 - 2q_y^2$.

Упрощенная модель тогда есть

$$\begin{aligned} \dot{q} &= \frac{1}{2} q * \omega \\ I\dot{\omega} &= \begin{pmatrix} LK(U_2 - U_4) \\ LK(U_3 - U_1) \\ \xi K(U_1 + U_3 - U_4 - U_2) \end{pmatrix} - \omega \times I\omega \\ \ddot{H} &= \frac{K}{m}\left(1 - 2q_x^2 - 2q_y^2\right)(U_1 + U_2 + U_3 + U_4) - g \end{aligned} \tag{2}$$

2. Система стабилизации

2.1. Контроллер угловой стабилизации

Основная идея закона управления лежит в инверсии уравнений модели. Висение робота на месте означает совпадение систем отсчета XYZ и $X'Y'Z'$, что соответствует значению кватерниона q, равному $(1, 0, 0, 0)$. Определим тогда целевой кватернион q_d, который будет целью управления. Для висения его значение будет $(1, 0, 0, 0)$, также его можно использовать для движения в горизонтальной плоскости.

Далее, определим желаемую производную кватерниона q как τ_d. Это значение должно обеспечивать цель $q \to q_d$. Естественным было бы определить его как $\tau_d = k_P(q_d - q)$, где $k_P > 0$ – коэффициент пропорционального регулятора. Однако, значение реальной производной кватерниона обязано быть ортогонально кватерниону, так как тот подчиняется закону $\|q\| = 1$. Поэтому доопределим τ_d как проекцию: $\tau_d = k_P(q_d - < q_d, q > q)$, где $< \cdot, \cdot >$ означает скалярное произведение в $\mathbb{R}^4$. Теперь из уравнения $\tau_d = \frac{1}{2} q * \omega_d$ можно определить целевую угловую скорость $\omega_d = 2\,\tau_d * q^*$. Благодаря тому, что τ_d ортогонален q, полученный кватернион будет иметь нулевую скалярную часть, то есть быть вектором. Теперь определим ρ_d как целевую производную угловой скорости: $\rho_d = k_D(\omega_d - \omega)$. Данное уравнение соответствует дифференциальному регулятору, а $k_D > 0$.

Теперь мы можем рассчитать целевой момент сил $M_d = I\rho_d + \omega \times I\omega$. Если у нас есть точные значения констант I, L, ξ и K, то правильной установкой команд U_i мы можем обеспечить воздействие момента M_d на робота. Это даст нам точное совпадение $\dot{\omega} \equiv \rho_d$. Таким образом, система управления выглядит так:

$$\begin{aligned} \tau_d &= k_P(q_d - < q_d, q > q) \\ \omega_d &= 2\,\tau_d * q^* \\ \rho_d &= k_D(\omega_d - \omega) \\ M_d &= I\rho_d + \omega \times I\omega \\ \dot{\omega} &\equiv \rho_d \end{aligned} \tag{3}$$

Заметим, что единственными параметрами регулятора тогда являются коэффициенты k_P и k_D, которые не зависят от физических параметров системы. То есть однажды найденные коэффициенты смогут стабилизировать любой квадрокоптер.

Данный алгоритм дает нам 3 уравнения на управляющие воздействия U_i, и нужно четвертое уравнение, чтобы полностью определить управления.

2.2. Линеаризация и доказательство устойчивости

Для простоты проведем линеаризацию системы только вокруг точки висения, $q_d = (1, 0, 0, 0)$. Аналогичные уравнения можно получить и для любой другой точки.

Кватернион поворота $q = \left(\cos\frac{\phi}{2},\ u_x \sin\frac{\phi}{2},\ u_y \sin\frac{\phi}{2},\ u_z \sin\frac{\phi}{2}\right)$ после линеаризации приводится к виду $q \approx (1, q_x, q_y, q_z)$. Далее:

$$
\begin{aligned}
\dot{q} &\approx \frac{1}{2}(0, \omega_x, \omega_y, \omega_z) \\
\tau_d &\approx (0, -k_P q_x, -k_P q_y, -k_P q_z) \\
\omega_d &\approx (-2k_P q_x, -2k_P q_y, -2k_P q_z) \\
\rho_d &\approx (-k_D(\omega_x + 2k_P q_x), -k_D(\omega_y + 2k_P q_y), -k_D(\omega_z + 2k_P q_z)) \\
\dot{\omega} &\equiv \rho_d
\end{aligned}
$$

Оставим из всей системы только переменные q_i и ω_i, так как именно их устойчивость нас интересует. Кроме того, заметим, что, к примеру, $\dot{q}_x$ зависит только от ω_x, а $\dot{\omega}_x$ зависит только от ω_x и q_x. То же самое верно и для других координат. Таким образом, можно сосредоточиться только на одной координате:

$$
\begin{pmatrix} \dot{q}_x \\ \dot{\omega}_x \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} \\ -2k_P k_D & -k_D \end{pmatrix} \begin{pmatrix} q_x \\ \omega_x \end{pmatrix}
$$

Собственные числа матрицы этой системы выражаются таким образом: $\lambda_{1,2} = \frac{-k_D \pm \sqrt{k_D^2 - 4k_P k_D}}{2}$. Тогда достаточным условием устойчивости здесь будет положительность коэффициентов k_P и k_D, что, по предположению, выполнено. Этим доказана устойчивость в малом.

2.3. Контроллер стабилизации высоты

Пусть задана целевая высота H_d, которую необходимо выдержать коптеру. Тогда применим PD-регулятор для стабилизации высоты, задав целевое ускорение:

$$\Upsilon_d = A_P(H_d - H) - A_D\dot{H}, \tag{4}$$

где A_P и A_D положительные. Из системы (2) получаем четвертое уравнение на управляющие воздействия:

$$\sum_{i=1}^{4} U_i = \frac{m}{K}\frac{(\Upsilon_d + g)}{k_{mod}} \tag{5}$$

2.4. Финальные формулы контроллера

Из (3) мы получаем целевой момент M_d, определяющий три уравнения в контроллере. Четвертое уравнение определяется регулятором высоты (4), и целью там является обеспечение ускорения в системе коптера, равного $\frac{\Upsilon_d+g}{k_{mod}}$. Введем матрицу G, связывающую целевые моменты и ускорения с управляющими воздействиями U_i:

$$G = \begin{pmatrix} 0 & KL & 0 & -KL \\ -KL & 0 & KL & 0 \\ K\xi & -K\xi & K\xi & -K\xi \\ \frac{K}{m} & \frac{K}{m} & \frac{K}{m} & \frac{K}{m} \end{pmatrix}^{-1} = \begin{pmatrix} 0 & -\frac{1}{2KL} & \frac{1}{4K\xi} & \frac{m}{4K} \\ \frac{1}{2KL} & 0 & -\frac{1}{4K\xi} & \frac{m}{4K} \\ 0 & \frac{1}{2KL} & \frac{1}{4K\xi} & \frac{m}{4K} \\ -\frac{1}{2KL} & 0 & -\frac{1}{4K\xi} & \frac{m}{4K} \end{pmatrix}$$

И тогда управления рассчитываются просто из матричного умножения:

$$\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2KL} & \frac{1}{4K\xi} & \frac{m}{4K} \\ \frac{1}{2KL} & 0 & -\frac{1}{4K\xi} & \frac{m}{4K} \\ 0 & \frac{1}{2KL} & \frac{1}{4K\xi} & \frac{m}{4K} \\ -\frac{1}{2KL} & 0 & -\frac{1}{4K\xi} & \frac{m}{4K} \end{pmatrix} \begin{pmatrix} M_{dx} \\ M_{dy} \\ M_{dz} \\ \dfrac{\Upsilon_d + g}{k_{mod}} \end{pmatrix} \tag{6}$$

3. Система адаптации

3.1. Оценка коэффициента тяги K

В уравнениях модели неизвестными параметрами, зависящими от физической модели коптера, являются расстояние от центра масс до винтов L, масса коптера m, три момента инерции I_x, I_y и I_z (мы считаем, что коптер собран так, что оси винтов совпадают с главными осями симметрии, поэтому тензор инерции имеет диагональный вид), коэффициент связи тяги и вращательного момента ξ, коэффициент связи тяги и управления K. Заметим, что независимое определение всех параметров невозможно: мы можем, к примеру, увеличить в 2 раза L и ξ, затем уменьшить в 2 раза m и K, и динамика коптера не изменится.

Два из неизвестных параметров, m и L, очень легко могут быть достаточно точно измерены на физической модели. Поэтому, чтобы избавиться от сложностей переопределенности, зафиксируем m и L и в дальнейшем будем считать их известными.

Из оставшихся параметров особое внимание следует обратить на коэффициент K, так как различие его значений в реальной модели и в системе стабилизации приведет, как дальше будет показано, к невыполнению цели управления $H \to H_d$, тогда как незнание точных значений других параметров приведет нас лишь к уменьшению эффективности регулятора. Поэтому сосредоточимся на задаче определения коэффициента K. В системе уравнений регулятора есть уравнение, зависящее только от него, и не зависящее от I и ξ. Это уравнение (5).

Пусть в каждый момент времени у нас будет оценка коэффициента K, которую мы обозначим за $\hat{K}$. Тогда управление высоты определяется двумя уравнениями:

$$\sum_{i=1}^{4} U_i = \frac{m}{\hat{K}} \frac{\left(A_P(H_d - H) - A_D\dot{H} + g\right)}{k_{mod}} \tag{7}$$

$$\ddot{H} = k_{mod}\frac{K}{m}\sum_{i=1}^{4} U_i - g \tag{8}$$

Подставив (7) в (8) и обозначив новую переменную $\Psi = \frac{1}{\hat{K}}$, получим

$$\ddot{H} = K\Psi\left(A_P(H_d - H) - A_D\dot{H} + g\right) - g = K\Psi\left(A_P(H_d - H) - A_D\dot{H}\right) + g\left(K\Psi - 1\right) \tag{9}$$

Видно, что при любом фиксированном положительном $\hat{K}$ уравнение устойчиво, но при $\hat{K} \neq K$

не выполнено $H \to H_d$. Вместо этого появляется добавка:

$$H \to H_d + \frac{g}{A_P}\left(1 - K\Psi\right)$$

Теперь можно сформулировать и доказать теорему о законе адаптации для коэффициента тяги K.

Теорема. Пусть выполнено соотношение положительных коэффициентов регулятора высоты

$$A_D^2 > A_P. \tag{10}$$

Тогда, при помощи закона адаптации

$$\dot{\hat{K}} = -\gamma \hat{K}^2 \Upsilon_d \left(\Upsilon_d + g\right), \tag{11}$$

где Υ_d определена по (4), оценка коэффициента тяги $\hat{K}$ сходится к истинному значению, а координата по высоте квадрокоптера — к цели H_d.

Доказательство. Рассмотрим функцию

$$Q(H) = \frac{1}{2}\left(A_P\left(H_d - H\right) - A_D\dot{H}\right)^2 + \frac{1}{2}\Delta\left(H - H_d\right)^2, \tag{12}$$

где $\Delta = 2A_P A_D^2 - A_P^2$. По условию (10) эта функция положительно определена и обращается в 0 только при $H \equiv H_d$ и $\dot{H} \equiv 0$. Кроме того, из (9) очевидно, что её стремление к нулю будет равносильно стремлению $\hat{K}$ к истинному значению K. Производная этой функции в силу системы равна

$$\dot{Q}(H) = \left(A_P\left(H_d - H\right) - A_D\dot{H}\right)\left(-A_P\dot{H} - A_D\ddot{H}\right) + \Delta(H - H_d)\dot{H}$$

Подставив сюда выражение для $\ddot{H}$ из (9), получим

$$\dot{Q}(H,\Psi) = -A_D K\Psi\left(A_P\left(H_d - H\right) - A_D\dot{H}\right)^2 - \left(A_P\dot{H} + gA_D\left(K\Psi - 1\right)\right)\left(A_P\left(H_d - H\right) - A_D\dot{H}\right) + \Delta(H - H_d)\dot{H} \tag{13}$$

Теперь для оценки Ψ мы можем применить метод скоростного градиента в дифференциальной форме (см. [22]). Для этого достаточно лишь проверить условия. Условие регулярности, очевидно, выполнено, как и условие роста $Q(H) \to +\infty$ при $H \to \infty$. Условие выпуклости выполнено, так как функция $\dot{Q}(H,\Psi)$ линейна по Ψ. Условие достижимости цели следует из того, что при

$\Psi \equiv \frac{1}{K}$, то есть при идеально подобранной оценке тяги выполнено

$$\dot{Q}(H) = -A_D \left(A_P \left(H_d - H \right) - A_D \dot{H} \right)^2 - A_P \dot{H} \left(A_P \left(H_d - H \right) - A_D \dot{H} \right) + \Delta (H - H_d) \dot{H}$$

Раскроем скобки и подставим Δ, получим

$$\dot{Q}(H) = -A_D A_P^2 (H_d - H)^2 - (A_D^3 - A_P A_D) \dot{H}^2$$

Из положительности коэффициентов регулятора высоты и условия (10) следует отрицательная определенность $\dot{Q}(H)$, откуда, для квадратичных форм, следует

$$\dot{Q}(H) \leqslant -\rho Q(H)$$

для некоторого ρ, что и является условием достижимости цели.

В итоге, беря градиент в (13), мы можем применить метод скоростного градиента для адаптации Ψ:

$$\dot{\Psi} = -\frac{\dot{\hat{K}}}{\hat{K}^2} = \gamma \left(A_D K \left(A_P \left(H_d - H \right) - A_D \dot{H} \right)^2 + g A_D K \left(A_P \left(H_d - H \right) - A_D \dot{H} \right) \right)$$

Вспомним, что выражение $A_P \left(H_d - H \right) - A_D \dot{H}$ в системе управления (4) обозначалось как Υ_d. Также, уберем в коэффициент γ постоянные величины A_D и K. Тогда, в итоге, получаем закон адаптации для $\hat{K}$:

$$\dot{\hat{K}} = -\gamma \hat{K}^2 \Upsilon_d \left(\Upsilon_d + g \right),$$

3.2. Случай разных коэффициентов K на разных моторах

В реальной машине, к сожалению, из-за небольших погрешностей в изготовлении винтов или двигателей, а также из-за возможных повреждений, коэффициенты K для каждого винта будут немного отличаться. И эта разница оказывает существенное влияние, в первую очередь, на угловую стабилизацию, так как робота начинает кренить в одну из сторон, и не выполняется условие на кватернионы $q \to q_d$. Далее рассматривается случай, когда у каждого двигателя есть свой коэффициент тяги K_i, а также своя оценка этого коэффициента, $\hat{K}_i$. Рассмотрим замкнутую систему, которая получится в таком случае.

Уравнения модели можно записать так:

$$\begin{pmatrix} I\dot{\omega} + \omega \times I\omega \\ \\ \dfrac{\ddot{H} + g}{k_{mod}} \end{pmatrix} = \begin{pmatrix} 0 & LK_2 & 0 & -LK_4 \\ -LK_1 & 0 & LK_3 & 0 \\ \xi K_1 & -\xi K_2 & \xi K_3 & -\xi K_4 \\ \dfrac{K_1}{m} & \dfrac{K_2}{m} & \dfrac{K_3}{m} & \dfrac{K_4}{m} \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix}, \tag{14}$$

а уравнения регулятора (6) — так:

$$\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} 0 & -\dfrac{1}{2\hat{K}_1 L} & \dfrac{1}{4\hat{K}_1 \xi} & \dfrac{m}{4\hat{K}_1} \\ \dfrac{1}{2\hat{K}_2 L} & 0 & -\dfrac{1}{4\hat{K}_2 \xi} & \dfrac{m}{4\hat{K}_2} \\ 0 & \dfrac{1}{2\hat{K}_3 L} & \dfrac{1}{4\hat{K}_3 \xi} & \dfrac{m}{4\hat{K}_3} \\ -\dfrac{1}{2\hat{K}_4 L} & 0 & -\dfrac{1}{4\hat{K}_4 \xi} & \dfrac{m}{4\hat{K}_4} \end{pmatrix} \begin{pmatrix} M_{dx} \\ M_{dy} \\ M_{dz} \\ \dfrac{\Upsilon_d + g}{k_{mod}} \end{pmatrix} \tag{15}$$

Как и в предыдущем параграфе, для удобства обозначим $\dfrac{1}{\hat{K}_i}$ через Ψ_i. Подставив (15) в (14) и выразив M_d по алгоритму (3), получим

$$\begin{pmatrix} I\dot{\omega} + \omega \times I\omega \\ \\ \dfrac{\ddot{H} + g}{k_{mod}} \end{pmatrix} = B \begin{pmatrix} I\rho_d + \omega \times I\omega \\ \\ \dfrac{\Upsilon_d + g}{k_{mod}} \end{pmatrix} \tag{16}$$

где матрица B определена как

$$\begin{pmatrix} \frac{1}{2}(K_2\Psi_2 + K_4\Psi_4) & 0 & -\frac{L}{4\xi}(K_2\Psi_2 - K_4\Psi_4) & \frac{Lm}{4}(K_2\Psi_2 - K_4\Psi_4) \\ 0 & \frac{1}{2}(K_1\Psi_1 + K_3\Psi_3) & -\frac{L}{4\xi}(K_1\Psi_1 - K_3\Psi_3) & -\frac{Lm}{4}(K_1\Psi_1 - K_3\Psi_3) \\ -\frac{\xi}{2L}(K_2\Psi_2 - K_4\Psi_4) & -\frac{\xi}{2L}(K_1\Psi_1 - K_3\Psi_3) & \frac{1}{4}(K_1\Psi_1 + K_2\Psi_2 + K_3\Psi_3 + K_4\Psi_4) & \frac{\xi m}{4}(K_1\Psi_1 - K_2\Psi_2 + K_3\Psi_3 - K_4\Psi_4) \\ \frac{1}{2Lm}(K_2\Psi_2 - K_4\Psi_4) & -\frac{1}{2Lm}(K_1\Psi_1 - K_3\Psi_3) & \frac{1}{4\xi m}(K_1\Psi_1 - K_2\Psi_2 + K_3\Psi_3 - K_4\Psi_4) & \frac{1}{4}(K_1\Psi_1 + K_2\Psi_2 + K_3\Psi_3 + K_4\Psi_4) \end{pmatrix}$$

Как можно было ожидать, в случае точного совпадения оценок коэффициентов с их реальными значениями, то есть в случае $K_i\Psi_i \equiv 1$, матрица B совпадает с единичной.

В (16) перейдем к линейной системе, отбросив члены старших порядков, расписав ρ_d так,

как это было сделано в параграфе про линеаризацию, и раскрыв Υ_d:

$$\begin{pmatrix} I_x\dot{\omega}_x \\ I_y\dot{\omega}_y \\ I_z\dot{\omega}_z \\ \ddot{H}+g \end{pmatrix} = B \begin{pmatrix} -I_x k_D\omega_x - 2I_x k_D k_P q_x \\ -I_y k_D\omega_y - 2I_y k_D k_P q_y \\ -I_z k_D\omega_z - 2I_z k_D k_P q_z \\ A_P(H_d - H) - A_D\dot{H} + g \end{pmatrix} \tag{17}$$

Теперь введем новые обозначения, чтобы перейти к полностью матричной записи задачи. Определим новый вектор состояния $X = \left(X_1^T, X_2^T\right)^T$, где

$$X_1 = \left(2I_x q_x,\ 2I_y q_y,\ 2I_z q_z,\ H - H_d\right)^T, \qquad X_2 = \left(I_x\omega_x,\ I_y\omega_y,\ I_z\omega_z,\ \dot{H}\right)^T,$$

и еще обозначим через вектор $\tilde{g}$ общее воздействие гравитации на систему:

$$\tilde{g} = (0,\ 0,\ 0,\ g)^T$$

Тогда система (17) может быть переписана как

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= B(-T_1X_1 - T_2X_2 + \tilde{g}) - \tilde{g}, \end{aligned} \tag{18}$$

где матрицы T_1 и T_2 — матрицы коэффициентов регуляторов:

$$T_1 = \begin{pmatrix} k_D k_P & 0 & 0 & 0 \\ 0 & k_D k_P & 0 & 0 \\ 0 & 0 & k_D k_P & 0 \\ 0 & 0 & 0 & A_P \end{pmatrix}, \qquad T_2 = \begin{pmatrix} k_D & 0 & 0 & 0 \\ 0 & k_D & 0 & 0 \\ 0 & 0 & k_D & 0 \\ 0 & 0 & 0 & A_D \end{pmatrix}$$

Фактически, они диагональны, но ниже предполагается лишь их симметричность, положительная определенность и, ради удобства вычислений, перестановочность между собой.

Теперь, чтобы применить метод скоростного градиента, определим функцию

$$\tilde{Q}(X) = \|T_1X_1 + T_2X_2\|^2. \tag{19}$$

Фактически, значение этой функции равно норме вектора управления. Её стремление к нулю

равносильно стремлению $X_2 \to -T_2^{-1}T_1X_1$, что, в свою очередь, благодаря первому уравнению системы (18) и положительной определенности матриц T_1 и T_2, равносильно устойчивости системы (18).

К сожалению, функция (19) сама по себе не подходит для применения метода скоростного градиента, так как не обладает свойством $\tilde{Q} \to \infty$ при $X \to \infty$: достаточно рассмотреть случай $X_2 = -T_2^{-1}T_1X_1$, при котором $\tilde{Q} \equiv 0$ при любом X_1.

Чтобы функция (19) удовлетворяла условию роста, её можно модифицировать добавлением еще одного квадратичного члена:

$$Q(X) = \|T_1X_1 + T_2X_2\|^2 + X_1^T\Gamma X_1 = \begin{pmatrix} X_1^T & X_2^T \end{pmatrix} \begin{pmatrix} T_1^2 + \Gamma & T_1T_2 \\ T_1T_2 & T_2^2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \tag{20}$$

где Γ — некоторая симметричная положительно определенная матрица.

Очевидно, что условие регулярности также выполнено.

Возьмем производную в силу системы от функции $Q(X)$:

$$\dot{Q}(X,B) = 2\begin{pmatrix} X_2^T & -X_1^TT_1B^T - X_2^TT_2B^T + g^T(B^T - I) \end{pmatrix} \begin{pmatrix} T_1^2 + \Gamma & T_1T_2 \\ T_1T_2 & T_2^2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \tag{21}$$

Условие выпуклости функции $\dot{Q}$ выполнено, так как она линейна по матрице B, которая в свою очередь линейна по оценкам Ψ_i. Теперь для удобства разложим на слагаемые уравнение (21), для этого введем матрицы W_{ij} — матрицы квадратичной формы между X_i и X_j, и матрицы W_{gi} — матрицы квадратичной формы между $\tilde{g}$ и X_i:

$$\begin{aligned} W_{11} &= -2T_1T_2BT_1, \\ W_{12} &= 2T_1^2 + 2\Gamma - 2T_1T_2BT_2 - 2T_1B^TT_2^2, \\ W_{22} &= 2T_1T_2 - 2T_2^2BT_2, \\ W_{g1} &= 2(B^T - I)T_1T_2, \\ W_{g2} &= 2(B^T - I)T_2^2. \end{aligned} \tag{22}$$

Посмотрев на эти матрицы, можно заметить, что выбор вспомогательной матрицы Γ никак не повлияет на финальный алгоритм, так как нет членов, в которых матрицы B и Γ встречались бы вместе.

Осталось проверить последнее условие — условие достижимости цели. Для этого подставим

в функцию (21) и в уравнения (22) $B \equiv I$, соответствующее идеально найденным оценкам Ψ_i. Получим

$$\dot{Q}_I(X) = -\begin{pmatrix} X_1^T & X_2^T \end{pmatrix} \begin{pmatrix} 2T_1^2T_2 & 2T_1T_2^2 - T_1^2 - \Gamma \\ 2T_1T_2^2 - T_1^2 - \Gamma & 2T_2^3 - 2T_1T_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = -X^T R X$$

Достаточно показать, что матрица этой квадратичной формы R положительно определена хоть для какого-либо значения Γ. Определим $\Gamma = T_1^2$ и попробуем найти неособое преобразование координат, превращающее эту матрицу в блочно-диагональную. Одним из таких преобразований может быть

$$R = \begin{pmatrix} 2T_1^2T_2 & 2T_1T_2^2 - 2T_1^2 \\ 2T_1T_2^2 - 2T_1^2 & 2T_2^3 - 2T_1T_2 \end{pmatrix} = \begin{pmatrix} T_1 & T_1T_2^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} 2T_1T_2^{-1} & 0 \\ 0 & 2T_2^3 - 2T_1T_2 \end{pmatrix} \begin{pmatrix} T_1 & 0 \\ T_1T_2^{-1} & I \end{pmatrix}$$

откуда следует критерий положительной определенности матрицы R:

$$T_1T_2^{-1} > 0,$$
$$T_2^3 - T_1T_2 > 0,$$

который, учитывая положительность матриц T_1 и T_2, преобразуется в условие

$$T_2^2 > T_1. \tag{23}$$

Итак, если для матриц коэффициентов выполнено условие (23), то адаптация коэффициентов Ψ_i по методу скоростного градиента приведет к пределу $Q(X, B) \to 0$ и, следовательно, $X \to 0$. Означает ли это, что оценки Ψ_i сойдутся к своим истинным значениям $\frac{1}{K_i}$? Для доказательства этого факта рассмотрим систему (18). Так как точка $X = 0$ является устойчивой точкой равновесия для этой системы, должно быть выполнено равенство

$$\tilde{g} = B\tilde{g},$$

которое означает, что матрица B имеет собственный вектор $(0, 0, 0, 1)^T$ с собственным числом 1.

Раскрыв это равенство как систему уравнений, получим

$$\begin{aligned} K_2\Psi_2 - K_4\Psi_4 &= 0, \\ K_1\Psi_1 - K_3\Psi_3 &= 0, \\ K_1\Psi_1 - K_2\Psi_2 + K_3\Psi_3 - K_4\Psi_4 &= 0, \\ K_1\Psi_1 + K_2\Psi_2 + K_3\Psi_3 + K_4\Psi_4 &= 4, \end{aligned}$$

откуда немедленно следует набор равенств $K_i\Psi_i = 1$ и, соответственно, $\Psi_i = \frac{1}{K_i}$ для всех $i \in \{1..4\}$.

Введем вспомогательную функцию $Z(X, J)$, которая будет описывать производную функции $\dot{Q}$ при $J = \frac{\partial B}{\partial \Psi_i}$, взятую с обратным знаком:

$$Z(X,J) = \begin{pmatrix} X_1^T & X_2^T & \tilde{g}^T \end{pmatrix} \begin{pmatrix} 2T_1T_2JT_1 & T_1T_2JT_2 + T_1J^TT_2^2 & -J^TT_1T_2 \\ T_2J^TT_1T_2 + T_2^2JT_1 & 2T_2^2JT_2 & -J^TT_2^2 \\ -T_1T_2J & -T_2^2J & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \tilde{g} \end{pmatrix} \tag{24}$$

Адаптация по методу скоростного градиента может быть тогда записана так:

$$\dot{\Psi}_i = -\frac{\dot{\hat{K}}_i}{\hat{K}_i^2} = \gamma Z\left(X, \frac{\partial B}{\partial \Psi_i}\right),$$

Неизвестные, но постоянные значения настоящих коэффициентов тяги K_i входят в соответствующую частную производную $\frac{\partial B}{\partial \Psi_i}$ линейно, и потому могут быть вынесены из функции Z в множитель γ.

В итоге, запишем финальный закон адаптации для коэффициентов тяги $\hat{K}_i$:

$$\dot{\hat{K}}_i = -\gamma \hat{K}_i^2 Z\left(X, \frac{\partial B}{\partial \Psi_i}\right). \tag{25}$$

Для полноты алгоритма приведем конкретные значения используемых производных $\frac{\partial B}{\partial \Psi_i}$,

считая при этом коэффициенты K_i убранными в множитель γ.

$$\frac{\partial B}{\partial \Psi_1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{L}{4\xi} & -\frac{Lm}{4} \\ 0 & -\frac{\xi}{2L} & \frac{1}{4} & \frac{\xi m}{4} \\ 0 & -\frac{1}{2Lm} & \frac{1}{4\xi m} & \frac{1}{4} \end{pmatrix} \qquad \frac{\partial B}{\partial \Psi_2} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{L}{4\xi} & \frac{Lm}{4} \\ 0 & 0 & 0 & 0 \\ -\frac{\xi}{2L} & 0 & \frac{1}{4} & -\frac{\xi m}{4} \\ \frac{1}{2Lm} & 0 & -\frac{1}{4\xi m} & \frac{1}{4} \end{pmatrix}$$

$$\frac{\partial B}{\partial \Psi_3} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{L}{4\xi} & \frac{Lm}{4} \\ 0 & \frac{\xi}{2L} & \frac{1}{4} & \frac{\xi m}{4} \\ 0 & \frac{1}{2Lm} & \frac{1}{4\xi m} & \frac{1}{4} \end{pmatrix} \qquad \frac{\partial B}{\partial \Psi_4} = \begin{pmatrix} \frac{1}{2} & 0 & \frac{L}{4\xi} & -\frac{Lm}{4} \\ 0 & 0 & 0 & 0 \\ \frac{\xi}{2L} & 0 & \frac{1}{4} & -\frac{\xi m}{4} \\ -\frac{1}{2Lm} & 0 & -\frac{1}{4\xi m} & \frac{1}{4} \end{pmatrix} \tag{26}$$

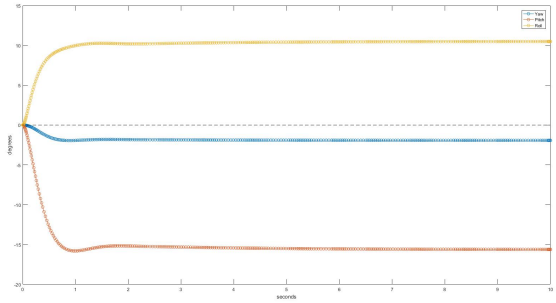
4. Моделирование

Моделирование алгоритма, представленного в прошлом параграфе, было проведено в системе MATLAB R2016a. Для этого была реализована модель коптера, система управления им и система адаптации. Физические параметры были взяты от одного из реальных изготовленных роботов, и они перечислены в следующей таблице:

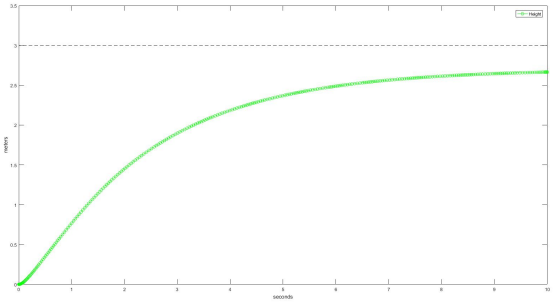
Физические параметры	Параметры регуляторов	Коэффициенты тяги
$m = 1.3$ kg		
$L = 0.22$ m	$K_P = 2.6$	$K_1 = 12.6$ H
$\xi = 0.017$ m	$K_D = 10.0$	$K_2 = 18.6$ H
$I_x = 0.12$ kg*m^2	$A_P = 2.0$	$K_3 = 4.6$ H
$I_y = 0.12$ kg*m^2	$A_D = 5.0$	$K_4 = 7.0$ H
$I_z = 0.22$ kg*m^2	$H_d = 3$ m	
$g = 9.8$ m/sec^2		

Здесь же указаны примененные коэффициенты регуляторов, которые удовлетворяют условию (23), и истинные коэффициенты тяги, являющиеся физическими, но неизвестными нам параметрами коптера. В начальном положении робот считается покоящимся на горизонтальной поверхности, его координата по высоте принята за 0.

Для проверки алгоритма было проведено 3 эксперимента, в первом из них адаптации не было ($\gamma = 0$), во втором $\gamma = 0.0005$, в третьем $\gamma = 0.005$. Начальные оценки коэффициентов $\hat{K}_i$ равны 8.6 H, время симуляции — 10 секунд.



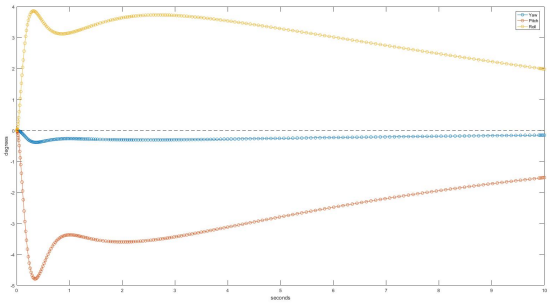
Ориентация, углы Эйлера



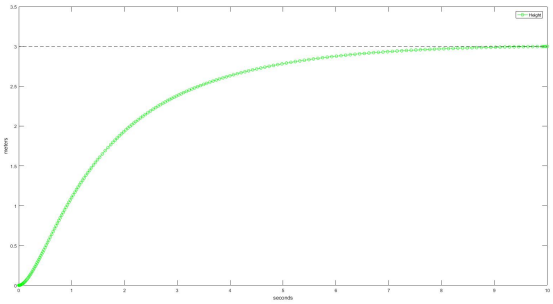
Координата по высоте H

Рис. 2. Отсутствие адаптации, $\gamma = 0$

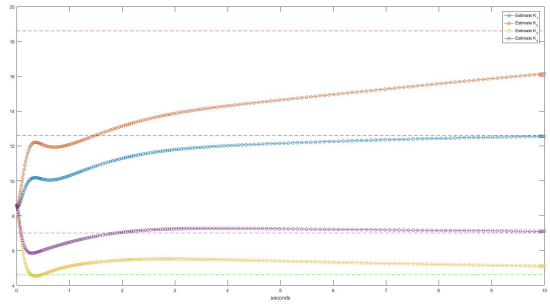
В первом случае (Рис. 2), в отсутствие адаптации, робот входит в устойчивое движение, но не достигает цели управления, в частности, у него есть наклон по крену и тангажу, что в реальной системе привело бы к появлению ускорения в плоскости XY.



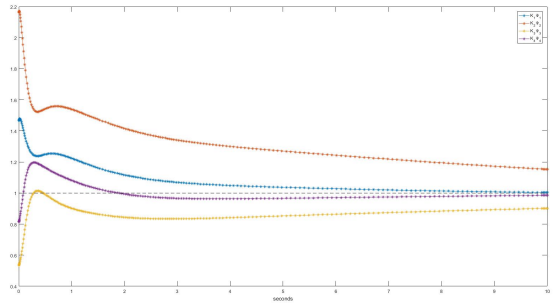
Ориентация, углы Эйлера



Координата по высоте H



Оценки коэффициентов $\hat{K}_i$

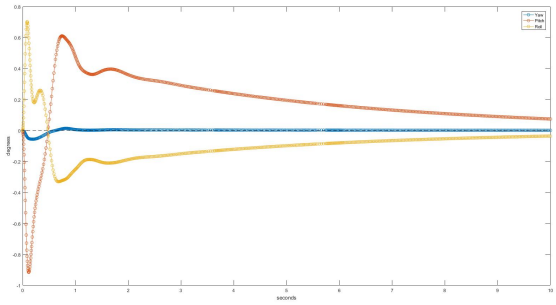


Величины $K_i\Psi_i$

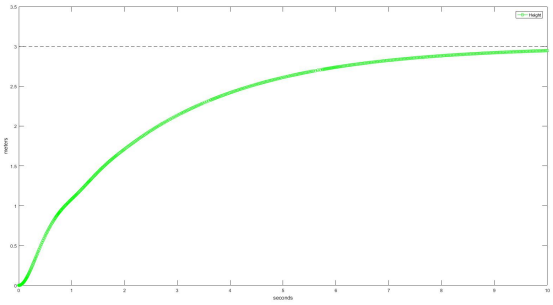
Рис. 3. Медленная адаптация, $\gamma = 0.0005$

В случае медленной адаптации (Рис. 3) состояние робота асимптотически достигает своих целевых значений, как и коэффициенты тяги. Также можно заметить, что при отсутствии адаптации, ошибка по крену и тангажу составила 10-15 градусов, тогда как при её наличии она не превысила 5, а через 10 секунд составляла уже 2 градуса. Ошибки такого порядка уже могут быть приемлемы в реальной системе.

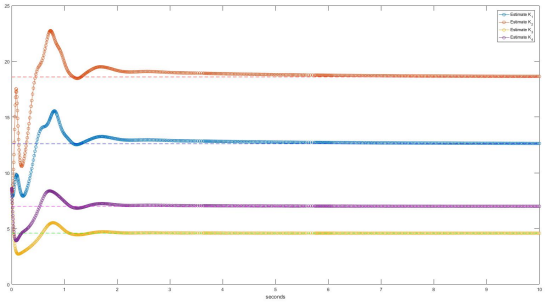
При быстрой адаптации (Рис. 4) коэффициенты сходятся к своим истинным значениям гораздо быстрее, а ошибка по крену и тангажу не превышает 1 градус. Однако можно заметить, что в начальный период коэффициенты, а вместе с ними и ориентация испытывают сильные колебания. Дальнейшее увеличение коэффициента γ приводит к тому, что вычисления становятся неустойчивыми.



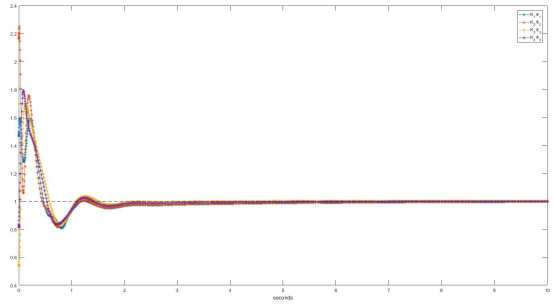
Ориентация, углы Эйлера



Координата по высоте H



Оценки коэффициентов $\hat{K}_i$



Величины $K_i\Psi_i$

Рис. 4. Быстрая адаптация, $\gamma = 0.005$

Заключение

В данной работе был описан основанный на кватернионах параметрический регулятор для стабилизации квадрокоптера, а также система идентификации коэффициентов тяги на каждом винте. Также было проведено численное моделирование, подтверждающее работоспособность предложенных алгоритмов.

Направлением дальнейшей работы, в первую очередь, является оценка остальных параметров модели, в особенности моментов инерций I_x, I_y и I_z и коэффициента ξ. Также необходимо изучить, как будут вести себя предложенные алгоритмы при дискретизации.

Список литературы

1. Markus Hehn, Raffaello D'Andrea. Quadrocopter trajectory generation and control, IFAC world congress, Vol. 18. No. 1. 2011.
2. Markus Hehn, Raffaello D'Andrea. A frequency domain iterative learning algorithm for high-performance, periodic quadrocopter maneuvers, in Mechatronics 24.8 (2014): 954-965.
3. Angela P. Schoellig, Clemens Wiltsche, and Raffaello D'Andrea. Feed-forward parameter identification for precise periodic quadrocopter motions. American Control Conference (ACC), 2012. IEEE, 2012.
4. Federico Augugliaro, Angela P. Schoellig, and Raffaello D'Andrea. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.
5. Robin Ritz, Mark W. Muller, Markus Hehn, and Raffaello D'Andrea. Cooperative quadrocopter ball throwing and catching. Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.
6. Glenn P. Tournier, Mario Valenti, and Jonathan P. How. Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. AIAA Guidance, Navigation and Control Conference and Exhibit. 2006.
7. Shaojie Shen, Yash Mulgaonkar, Nathan Michael, Vijay Kumar. Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor. Robotics: Science and Systems. Vol. 1. 2013.
8. Paolo Stegagno, Massimo Basile, Heinrich H. Bulthoff, Antonio Franchi. Vision-based Autonomous Control of a Quadrotor UAV using an Onboard RGB-D Camera and its Application to Haptic Teleoperation. 2nd IFAC Work. on Research, Education and Development of Unmanned Aerial Systems, Compiegne, France (2013).
9. Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. Automatic Control, IEEE Transactions on 53.5 (2008): 1203-1218.
10. Sebastian O.H. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK) (2010).
11. Ern J. Lefferts, F. Landis Markley, and Malcolm D. Shuster. Kalman filtering for spacecraft attitude estimation. Journal of Guidance, Control, and Dynamics 5.5 (1982): 417-429.
12. I. Carro Perez, D. Flores-Araiza: J. A. Fortoul-D?az, R. Maximo and H. G. Gonzalez-Hernandez.

Identification and PID control for a quadrocopter. Electronics, Communications and Computers (CONIELECOMP), 2014 International Conference on. IEEE, 2014.

13. Jiangcheng Zhu, Endong Liu, Shan Guo, Chao Xu. A gradient optimization based PID tuning approach on quadrotor. Control and Decision Conference (CCDC), 2015 27th Chinese. IEEE, 2015.
14. Hyon Lim, Jaemann Park, Daewon Lee, and H.J. Kim. Build your own quadrotor. Open-Source Projects on Unmanned Aerial Vehicles, in IEEE Robotics and Automation Magazine, Septermer 2012.
15. Emil Fresk and George Nikolakopoulos. Full Quaternion Based Attitude Control for a Quadrotor, in 2013 European Control Conference (ECC), July 17-19, 2013, Zurich, Switzerland.
16. Paul Pounds, Robert Mahony, Peter Corke. Modelling and Control of a Quad-Rotor Robot, Proceedings Australasian Conference on Robotics and Automation 2006. Australian Robotics and Automation Association Inc., 2006.
17. Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles. Modelling, Estimation, and Control of Quadrotor, in IEEE Robotics and Automation Magazine, Septermer 2012.
18. Shahida Khatoon, Muhammad Shahid, and Himanshu Chaudhary. Dynamic modeling and stabilization of quadrotor using PID controller. Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on. IEEE, 2014.
19. В.В. Матвеев, В.Я. Распопов. Основы построения бесплатформенных инерциальных навигационных систем.: ЦНИИ "Электроприбор", 2009. — 278 с.
20. Л.Д. Ландау, Е.М. Лифшиц. Теоретическая физика, том 1. Механика. 5-е издание. М.:ФИЗМАТЛИТ, 2013. — 224 с.
21. J.G.Leishman. Principles of Helicopter Aerodynamics, Second Edition. 2006. Cambridge Aerospace Series.
22. Мирошник И.В., Никифоров В.О., Фрадков А.Л. Нелинейное и адаптивное управление сложными динамическими системами. СПб.: Наука, 2000. — 549 с.

Приложение

Код программы моделирования

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----COPTER_CONTROL_FUNCTION-----%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ control ] = copter_control( state, InertiaMatrix, H_desired, L, m, xi, ...
                                       K1, K2, K3, K4, K_P, K_D, A_P, A_D );

g = 9.8;
q_des = [1 0 0 0];
%state = [qw, qx, qy, qz, wx, wy, wz, h, dot_h, hK1, hK2, hK3, hK4]

ControlMatrix = [ ...
    0 -1/(2*L*K1) 1/(4*xi*K1) m/(4*K1); ...
    1/(2*L*K2) 0 -1/(4*xi*K2) m/(4*K2); ...
    0 1/(2*L*K3) 1/(4*xi*K3) m/(4*K3); ...
    -1/(2*L*K4) 0 -1/(4*xi*K4) m/(4*K4)];


q = [state(1) state(2) state(3) state(4)];
w = [state(5) state(6) state(7)];

tau_des = K_P * (q_des - (q_des*q')*q);
w_des = 2 * quatmultiply(tau_des, quatconj(q));
w_des = w_des(2:4);
rho_des = K_D * (w_des - w);
M_des = InertiaMatrix * rho_des' + cross(w', InertiaMatrix * w');

Yota_des = A_P * (H_desired - state(8)) - A_D * state(9);
control = ControlMatrix * [M_des; (Yota_des+g)/(1 - q(2)^2 - q(3)^2)];

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----COPTER_MODEL_FUNCTION-----%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ dot_state ] = copter_model( state, ModelMatrix, InertiaMatrix, L, m, xi )

g = 9.8;
H_desired = 3;
K_P = 2.6;
K_D = 10;
A_P = 2;
A_D = 5;
gamma = 0.005;
%state = [qw, qx, qy, qz, wx, wy, wz, h, dot_h, hK1, hK2, hK3, hK4]
```

```
%%%----------CONTROL----------

hK1 = state(10);
hK2 = state(11);
hK3 = state(12);
hK4 = state(13);
control = copter_control(state, InertiaMatrix, H_desired, L, m, xi, ...
                         hK1, hK2, hK3, hK4, K_P, K_D, A_P, A_D);

%%%----------MODEL----------

q = [state(1) state(2) state(3) state(4)];
w = [state(5) state(6) state(7)];

dq = 0.5 * quatmultiply(q, [0 w]) - (norm(q) - 1) * q;

res_vector = ModelMatrix * control;
dw = InertiaMatrix \ (res_vector(1:3) - cross(w', InertiaMatrix * w'));

ddot_H = (1 - q(2)^2 - q(3)^2) * res_vector(4) - g;

%%%----------ADAPTATION----------

T1 = [K_D*K_P 0 0 0; 0 K_D*K_P 0 0; 0 0 K_D*K_P 0; 0 0 0 A_P];
T2 = [K_D 0 0 0; 0 K_D 0 0; 0 0 K_D 0; 0 0 0 A_D];

X = [q(2:4)'; state(8) - H_desired; w'; state(9); 0; 0; 0; g];

Z = @(J) X' * [2*T1*T2*J*T1, T1*T2*J*T2+T1*J'*T2^2, -J'*T1*T2; ...
        T2*J'*T1*T2+T2^2*J*T1, 2*T2^2*J*T2, -J'*T2^2; ...
        -T1*T2*J, -T2^2*J, zeros(4)] * X;

dB_dPsi1 = [0 0 0 0; 0 0.5 -L/(4*xi) -L*m/4; ...
    0 -xi/(2*L) 0.25 xi*m/4; 0 -1/(2*L*m) 1/(4*xi*m) 0.25];

dB_dPsi2 = [0.5 0 -L/(4*xi) L*m/4; 0 0 0 0; ...
    -xi/(2*L) 0 0.25 -xi*m/4; 1/(2*L*m) 0 -1/(4*xi*m) 0.25];

dB_dPsi3 = [0 0 0 0; 0 0.5 L/(4*xi) L*m/4; ...
    0 xi/(2*L) 0.25 xi*m/4; 0 1/(2*L*m) 1/(4*xi*m) 0.25];

dB_dPsi4 = [0.5 0 L/(4*xi) -L*m/4; 0 0 0 0; ...
    xi/(2*L) 0 0.25 -xi*m/4; -1/(2*L*m) 0 -1/(4*xi*m) 0.25];

dhK1 = -gamma * hK1^2 * Z(dB_dPsi1);
dhK2 = -gamma * hK2^2 * Z(dB_dPsi2);
dhK3 = -gamma * hK3^2 * Z(dB_dPsi3);
dhK4 = -gamma * hK4^2 * Z(dB_dPsi4);

%%%----------RESULT----------

dot_state = [dq'; dw; state(9); ddot_H; dhK1; dhK2; dhK3; dhK4];

end
```

```
%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
%-----MAIN--SCRIPT-----%
%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;

L = 0.22;
m = 1.3;
xi = 0.017;

Ix = 0.12;
Iy = 0.12;
Iz = 0.22;
InertiaMatrix = [Ix 0 0; 0 Iy 0; 0 0 Iz];

K1 = 12.6;
K2 = 18.6;
K3 = 4.6;
K4 = 7;

ModelMatrix = [ ...
    0 L*K2 0 -L*K4; ...
    -L*K1 0 L*K3 0; ...
    xi*K1 -xi*K2 xi*K3 -xi*K4; ...
    K1/m K2/m K3/m K4/m];

H_desired = 3;

%state = [qw, qx, qy, qz, wx, wy, wz, h, dot_h, hK1, hK2, hK3, hK4]

f = @(t, state) (copter_model(state, ModelMatrix, InertiaMatrix, L, m, xi));
state_0 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 8.6, 8.6, 8.6, 8.6];

[t, state] = ode45(f, [0 10], state_0);

ang = quat2eul(state(:,1:4), 'ZYX') * 180 / pi;

figure;
plot(t, ang(:,1), 'o-');
hold on;
plot(t, ang(:,2), 'o-');
plot(t, ang(:,3), 'o-');
plot(t, zeros(size(t)), 'k--');
xlabel('seconds');
ylabel('degrees');
legend('Yaw', 'Pitch', 'Roll');

figure;
plot(t, state(:,8), 'go-');
hold on;
plot(t, H_desired * ones(size(t)), 'k--');
ylim([0 3.5]);
xlabel('seconds');
ylabel('meters');
legend('Height');
```

```
figure;
plot(t, state(:,10),'o-');
hold on;
plot(t, state(:,11),'o-');
plot(t, state(:,12),'o-');
plot(t, state(:,13),'o-');
legend('Estimate K_1', 'Estimate K_2', 'Estimate K_3', 'Estimate K_4');
plot(t, K1 * ones(size(t)), 'b--');
plot(t, K2 * ones(size(t)), 'r--');
plot(t, K3 * ones(size(t)), 'g--');
plot(t, K4 * ones(size(t)), 'm--');
xlabel('seconds');

figure;
plot(t, K1 ./ state(:,10),'*-');
hold on;
plot(t, K2 ./ state(:,11),'*-');
plot(t, K3 ./ state(:,12),'*-');
plot(t, K4 ./ state(:,13),'*-');
legend('K_1\Psi_1', 'K_2\Psi_2', 'K_3\Psi_3', 'K_4\Psi_4');
plot(t, ones(size(t)), 'k--');
xlabel('seconds');
```