УДК 681.51

И.С. ТРЕНЕВ
(Институт проблем управления им. В. А. Трапезникова РАН, Москва)

СТАБИЛИЗАЦИЯ ДВИЖЕНИЯ КВАДРОКОПТЕРА PARROT MAMBO ВДОЛЬ ЗАДАННОЙ ТРАЕКТОРИИ НА ОСНОВЕ PID-РЕГУЛЯТОРОВ

В работе производится линеаризация модели квадрокоптера и строится программное управление на основе заданных траекторий для линеаризованной модели, а также осуществляется подбор коэффициентов ПИД-регуляторов. Устанавливается стабильная связь с квадрокоптером. Производится численное моделирование.

Введение. Одной из главных задач современной теории управления остается задача стабилизации динамики подвижного объекта. Существует множество вариантов выбора управления квадрокоптером как в линейной, так и в нелинейной теории управления. В разных подходах используется полный вектор состояния, наблюдения или данные оценивателя, в качестве цели управления может выбираться достижение определенного положения, следование по траектории, стабилизация в заданной точке при внешних возмущениях и т. д. В качестве управления можно выбрать: управление, основанное на теории Ляпунова; управление, основанное на линейном квадратичном регуляторе; бэкстэппинг и др. В данной работе в качестве объекта управления выбран квадрокоптер *Parrot Mambo*, для которого решается задача управления на основе ПИД-регулятора. Среда Simulink, с помощью дополнительных пакетов, предоставляет набор вспомогательных инструментов и дает возможность загрузки собственного программного обеспечения на реальный объект [1]. Целью работы является стабилизация движения квадрокоптера вдоль заданной программной траектории с помощью ПИД-регуляторов. Программная траектория – желаемая траектория движения, задаваемая в начальный момент времени, по которой должен перемещаться объект.

Постановка задачи. С помощью уравнения Ньютона-Эйлера получена нелинейная математическая модель квадрокоптера, характеризующая его движение, в зависимости от общей тяги U_1 и крутящих моментов U_2, U_3, U_4 [2]:

$$\begin{cases} \ddot{x} = -\dfrac{1}{m}\cdot(\sin\psi\cdot\sin\varphi + \cos\psi\cdot\sin\theta\cdot\cos\varphi)\cdot U_1, \\ \ddot{y} = -\dfrac{1}{m}\cdot(-\cos\psi\cdot\sin\varphi + \sin\psi\cdot\sin\theta\cdot\cos\varphi)\cdot U_1 \\ \ddot{z} = -\dfrac{1}{m}\cdot\cos\theta\cdot\cos\varphi\cdot U_1 + g, \\ \dot{\varphi} = p + \sin\varphi\cdot\tan\theta\cdot q + \cos\varphi\cdot\tan\theta\cdot r, \\ \dot{\theta} = q\cdot\cos\varphi - r\cdot\sin\varphi,\ \dot{\psi} = q\cdot\dfrac{\sin\varphi}{\cos\theta} + r\cdot\dfrac{\cos\varphi}{\cos\theta}, \\ \dot{p} = \dfrac{I_{yy}-I_{zz}}{I_{xx}}\cdot q\cdot r + \dfrac{U_2}{I_{xx}},\ \dot{q} = \dfrac{I_{zz}-I_{xx}}{I_{yy}}\cdot p\cdot r + \dfrac{U_3}{I_{yy}}, \\ \dot{r} = \dfrac{I_{xx}-I_{yy}}{I_{zz}}\cdot p\cdot q + \dfrac{U_4}{I_{zz}}, \end{cases} \tag{1}$$

где x, y, z – координаты положения центра масс квадрокоптера в географической системе координат (ИСК); φ, ψ, θ – углы Крылова, определяющие ориентацию квадрокоптера в ИСК; p, q, r – угловые скорости в системе координат, связанной с квадрокоптером; m – масса квадрокоптера; g – ускорение свободного падения; I_{xx}, I_{yy}, I_{zz} – диагональные элементы матрицы инерции.

Научный руководитель: к.ф.-м.н., старший научный сотрудник Юрченков А.В.

Общая программная траектория состоит из 3 участков: зависание (подъем и вращение вокруг собственной оси); смещение вдоль оси OX; смещение вдоль оси OY, которые можно комбинировать и получать различные вариации движения. Программное управление будем получать из исходной нелинейной системы и заданной программной траектории.

Требуется решить задачу управления − стабилизировать движение квадрокоптера вдоль заданной программной траектории с помощью ПИД-регуляторов, настроив их коэффициенты по критерию минимума ошибки между желаемой и получаемой траекториями (рис. 1) [3].

Схема используемых ПИД-регуляторов. Для решения указанной задачи линеаризуем модель (1) в окрестности программной траектории X_d и программного вектора управления U_d [4]:

$$\Delta\dot{X} = A\Delta X + B\Delta U, \tag{2}$$

где $\Delta X = X - X_d$, $\Delta U = U - U_d$, A и B – матрицы, состоящие из значений частных производных нелинейной системы.

Для стабилизации квадрокоптера необходимо найти численное значение тяги и крутящих моментов двигателей.

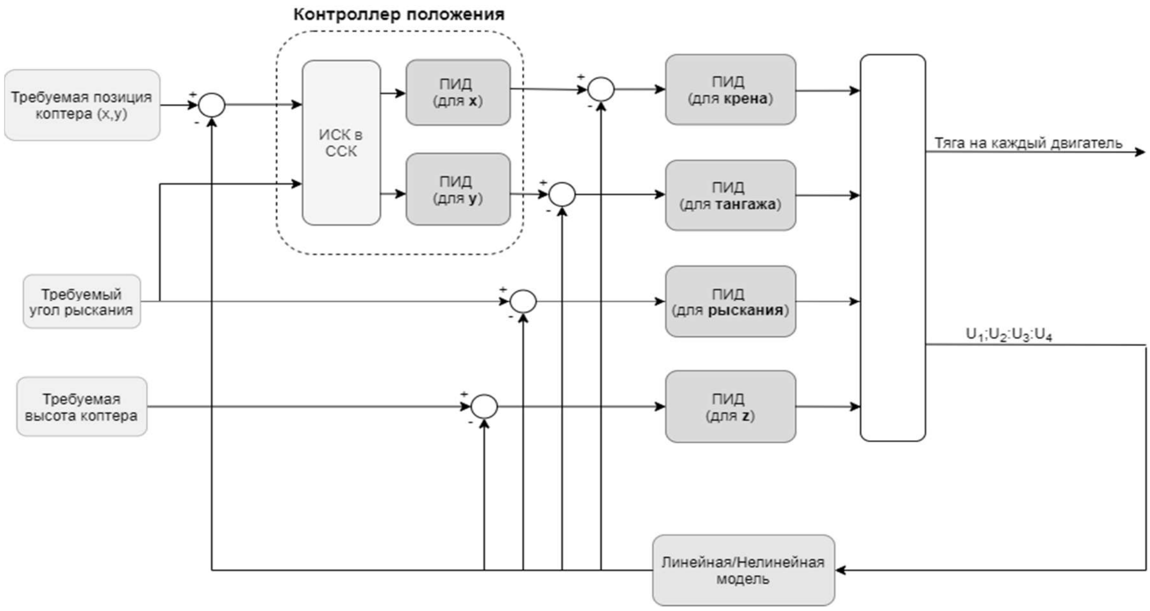

Рис. 1. Схема алгоритма управления для модели квадрокоптера

Для настройки коэффициентов ПИД-регуляторов используется линеаризованная математическая модель, а для реализации движения по программной траектории – нелинейная модель.

Моделирование. Для реализации требуемого алгоритма управления и для визуализации используется среда *Matlab/Simulink*. В *Simulink* было построено 3 линейные математические модели квадрокоптера и подобраны коэффициенты для каждого набора ПИД-регуляторов, которые позволяют реализовать движение по каждой из требуемых траекторий. Для наглядности, в данной работе используется визуализация из пакета *AerospaceBlockset*. На рис. 2–4 приведены графические иллюстрации работы ПИД-регуляторов. Исходя из полученных результатов, можно сделать вывод, что стабилизация по оси OZ происходит в течение 3 секунд: смещение вдоль оси OX и OY происходит за 4,5 секунды, поворот на 90 градусов вокруг собственной оси квадрокоптера происходит менее, чем за одну секунду.

Связь с реальным объектом. Для установления связи с квадрокоптером *Parrot Mambo* через *Bluetooth* используется дополнительный пакет *Simulink Support Package for Parrot Minidrones*. Данный пакет позволяет загружать собственное программное обеспечение на реальный объект.

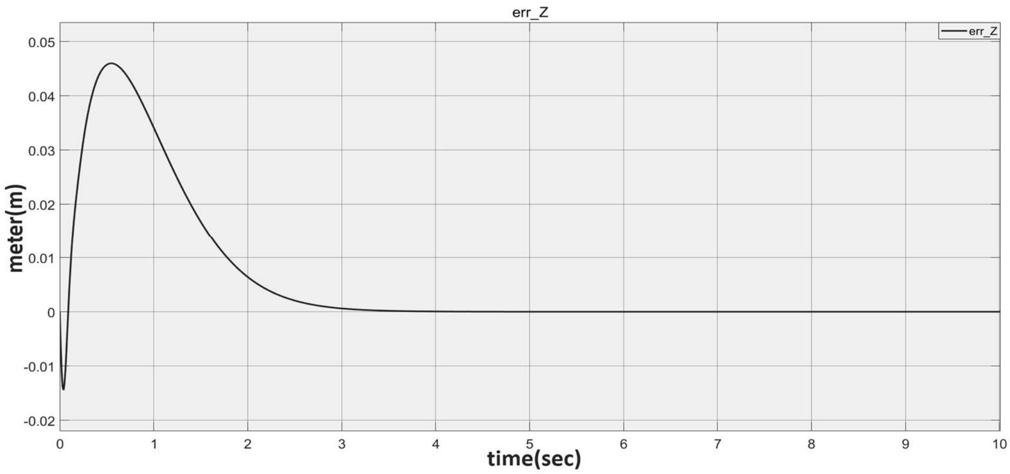

Рис. 2. Невязка координаты Z (высоты)

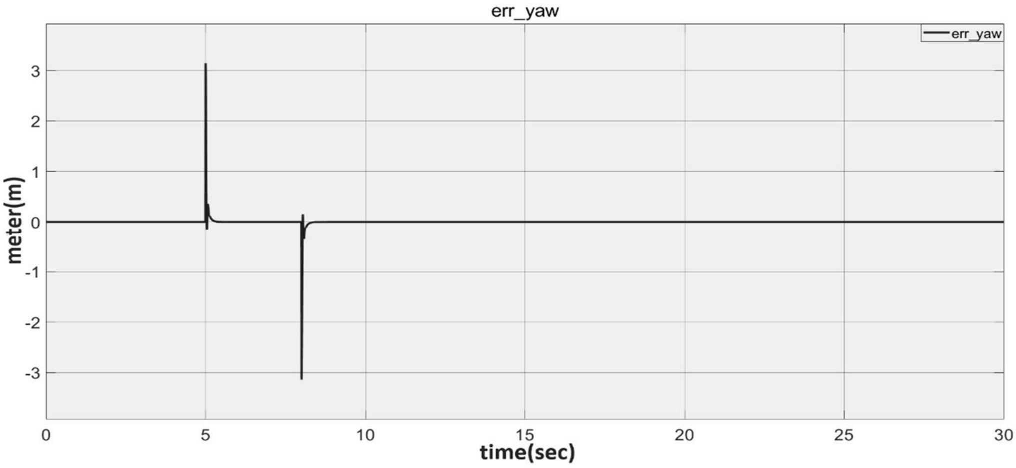

Рис. 3. Невязка угла крена

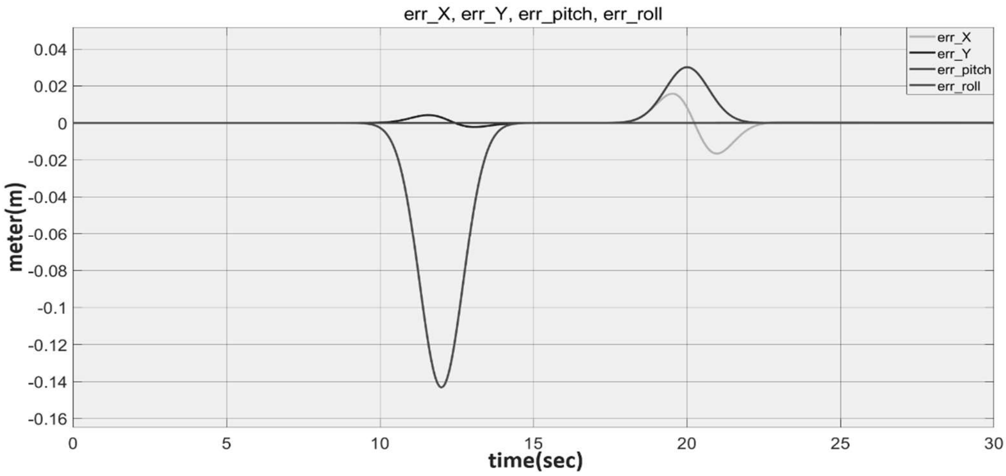

Рис. 4. Невязки координат x, y, и углов крена и тангажа

Заключение. В работе решена задача стабилизации движения квадрокоптера вдоль заданных программных траекторий с помощью построенного закона управления на основе ПИД регуляторов. Путем линеаризации исходной модели, была получена модель в отклонениях, после чего произведено численное моделирование в среде *Simulink*. Программа для обработки данных, получаемых с датчиков квадрокоптера, позволит загрузить разработанный проект с полученной моделью движения и блоком управления на основе ПИД-регуляторов из *Matlab/Simulink* на реальный объект *Parrot Mambo*.

ЛИТЕРАТУРА

1. **G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin**, "Quadrotor helicopter flight dynamics and control: Theory and experiment", Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Aug. 2007. https://www.researchgate.net/publication/228808757_Quadrotor_Helicopter_Flight_Dynamics_and_Control_Theory_and_Experiment
2. **T. Bresciani**, Modelling, Identification and Control of a Quadrotor Helicopter (2008). https://lup.lub.lu.se/student-papers/search/publication/8847641
3. MathWorks | Drone Simulation and Control [Электронный ресурс]. https://www.mathworks.com/videos/series/drone-simulation-and-control.html
4. **T. Luukkonen**, School of Science (2011). Available at: http://sal.aalto.fi/publications/pdf-files/eluu11_public.pdf

---

I.S. Trenev (V.A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow). **Movement stabilization of the parrot mambo quadcopter along a given trajectory based on pid-controllers.**

Abstract. In this work, the linearization of the quadcopter model is carried out and the program control is built based on the specified trajectories for this model, as well as the selection of the coefficients of the PID-controllers is carried out. Numerical simulation is performed and a stable connection with the quadcopter is established.

В АВТОРСКОЙ РЕДАКЦИИ
УДК 681.51

А.А. ТКАЧЕНКО
(Институт проблем управления им. В. А. Трапезникова Российской академии наук, Москва)

РЕШЕНИЕ ЗАДАЧИ СТАБИЛИЗАЦИИ КВАДРОКОПТЕРА ПРИ ПОМОЩИ ЛИНЕЙНО-КВАДРАТИЧНОГО РЕГУЛЯТОРА: МОДЕЛИРОВАНИЕ И ПРАКТИКА

В работе рассматривается применение LQR-регулятора к задаче управления полетом квадрокоптера, реализованного в среде Matlab. Для модели в отклонениях построен регулятор, обеспечивающий стабилизацию в точке. Работоспособность алгоритма продемонстрирована путем имитационного моделирования.

Введение. Квадрокоптер — это беспилотный летательный аппарат с четырьмя несущими винтами, расположенными на концах двух, пересекающихся в центре лучей. На борту такого аппарата устанавливается полетный контроллер, отвечающий за своевременное распределение сигналов по регуляторам оборотов, которые в свою очередь отправляют их на моторы, заставляя дрон менять свое положение в пространстве. Для того, чтобы аппарат передвигался в пространстве согласно нашим ожиданиям, необходимо спроектировать систему управления. Основным звеном такой системы является регулятор. Он позволяет следить за изменением состояния объекта управления и с помощью определенных алгоритмов синтезировать управляющие сигналы. В данной работе будет рассмотрен синтез системы управления, позволяющей стабилизировать квадрокоптер в пространстве. В качестве регулятора выбран LQR. Целью работы является моделирование полета квадрокоптера в среде Matlab с применением линейно-квадратичного регулятора и проверка правильности выведенной математической модели для дальнейшего тестирования на реальной физической модели аппарата.

Математическая модель. Математическая модель, описывающая динамику движения квадрокоптера, необходима для разработки регуляторов и для тестирования в имитационных полетах. Необходимо рассмотреть две системы координат: связанную с Землей ортогональную систему координат и связанную с корпусом аппарата ортогональную систему координат. Переход от системы координат, связанной с корпусом, к системе координат, связанной с Землей, осуществляется матрицей поворота R [1]:

$$R=\begin{bmatrix} C_\psi C_\theta & S_\psi C_\theta & -S_\theta \\ C_\psi S_\theta S_\phi - S_\psi C_\phi & S_\psi S_\theta S_\phi + C_\psi C_\phi & C_\theta S_\phi \\ C_\psi S_\theta C_\phi + S_\psi S_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi & C_\theta C_\phi \end{bmatrix}, \quad (1)$$

где $C_\alpha = \cos(\alpha)$; $S_\alpha = \sin(\alpha)$; θ, ϕ, ψ – углы Эйлера.

Уравнения поступательного движения квадрокоптера могут быть получены из второго закона Ньютона

$$\frac{d}{dt}(m\bar{V}) = \sum \overline{F_i}, \Rightarrow m\dot{\bar{V}} = mg \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + R \cdot \sum T_i \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad (2)$$

где: m – масса аппарата; g – ускорение свободного падения; $\bar{V}$ – вектор линейных скоростей; $\sum \bar{F_i}$ – сумма внешних сил; $\sum \bar{T}_i$ – суммарная тяга моторов.

Из второго закона Ньютона для вращательного движения получим

$$\sum \overline{M}_i = \frac{d\bar{L}}{dt}, \Rightarrow I\dot{\bar{\omega}} = -\bar{\omega} \times I\bar{\omega} + \bar{M}, \quad (3)$$

Научный руководитель: к.ф.-м.н., старший научный сотрудник Юрченков А.В.

где: $\overline{M}$ – момент вращающей силы; $\overline{L}$ – момент импульса; I – момент инерции; $\overline{\omega}=[p,q,r]^T$ – вектор угловых скоростей.

Запишем кинематические дифференциальные уравнения для углов Эйлера [1]

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta)\\ 0 & \cos(\phi) & -\sin(\phi)\\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta)\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}. \quad (4)$$

Из уравнения (3) получим следующие соотношения для угловых скоростей и ускорений:

$$\dot{p}=\frac{I_{yy}-I_{zz}}{I_{xx}}\cdot qr+\frac{1}{I_{xx}}M_1, \quad (5)$$

$$\dot{q}=\frac{I_{zz}-I_{xx}}{I_{yy}}\cdot rp+\frac{1}{I_{yy}}M_2, \quad (6)$$

$$\dot{r}=\frac{I_{xx}-I_{yy}}{I_{zz}}\cdot pq+\frac{1}{I_{zz}}M_3. \quad (7)$$

Таким образом, получена система уравнений, описывающая динамику движения квадрокоптера.

Линеаризация. Для построения системы управления используется приближенная замена уравнений — линеаризация в окрестности желаемого положения. Определим положение равновесия следующим образом

$$\overline{x}_e=\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \psi & 0 & 0 & 0\end{pmatrix}^T,$$

$$\overline{u}_e=\begin{pmatrix}mg & 0 & 0 & 0\end{pmatrix}^T. \quad (8)$$

Тогда линейная система в отклонениях будет иметь вид

$$\Delta\dot{\overline{x}}=A\Delta\overline{x}+B\Delta\overline{u}, \quad (9)$$

где $\Delta\overline{x}=\overline{x}-\overline{x}_e,\ \Delta\overline{u}=\overline{u}-\overline{u}_e,\ A=\left.\frac{\partial f(\overline{x},\overline{u})}{\partial\overline{x}}\right|_{\substack{\overline{x}=\overline{x}_e\\ \overline{u}=\overline{u}_e}},\ B=\left.\frac{\partial f(\overline{x},\overline{u})}{\partial\overline{u}}\right|_{\substack{\overline{x}=\overline{x}_e\\ \overline{u}=\overline{u}_e}}.$

LQR-синтез. Найдем закон управления в форме линейной обратной связи по состоянию

$$u=Kx,\ K\in\mathbb{R}^{m\times n}, \quad (10)$$

минимизирующий квадратичный критерий качества [2]

$$J=\int_0^\infty[x^T(\tau)Qx(\tau)+u^T(\tau)Ru(\tau)]d\tau, \quad (11)$$

где Q и R – положительно-определенные матрицы, которые, обычно, задаются таким образом, чтобы регулятор удовлетворял желаемой динамике. Чем больше значения коэффициентов Q относительно коэффициентов R, тем интенсивнее будет управляющий сигнал. Коэффициенты R и Q подбирались эмпирически. Матрица коэффициентов K, в таком случае, имеет вид

$$K=-R^{-1}B^TP, \quad (12)$$

где матрица P находится из матричного уравнения Риккати [3]

$$A^TP+PA-PBR^{-1}B^TP+Q=0. \quad (13)$$

Моделирование. С помощью инструментов в среде Matlab был минимизирован функционал по вектору состояний x, определены матрицы P и матрица коэффициентов K. Для реализации имитационного моделирования модель была дополнена недостающими параметрами, такими как: масса, ускорение свободного падения и значения для тензора инерции. Результаты тестирования приведены на графиках (рис. 1.).

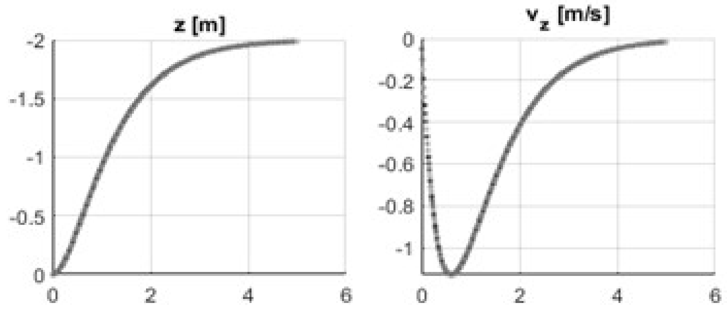

Рис. 1. Результаты моделирования

Практическая часть. Для реализации лабораторных испытаний и демонстрации алгоритмов управления была собрана модель летательного аппарата на базе полетного контроллера Pixhawk.

Рис.2. Полетный контроллер

Для управления настройками летательного аппарата, каналами связи и параметрами обратной связи системы стабилизации необходимо скачать и установить вспомогательное программное обеспечение. Воспользуемся программой *QGroundControl* [4]. Она позволяет отображать важную информацию, полученную с полетного контроллера: текущие координаты, состояние датчиков и их показания. Доступ к подобным данным необходим во время выполнения полетных задач.

После установки необходимо выполнить первоначальные настройки: загрузить прошивку, выбрать тип используемой рамы, откалибровать датчики акселерометра, гироскопа, а также регуляторы оборотов.

Заключение. В работе было реализовано моделирование полета квадрокоптера в среде Matlab с применением линейно-квадратичного регулятора и установлена корректность полученной математической модели. Полученный алгоритм управления может быть использован на собранном квадрокоптере на базе полетного контроллера Pixhawk. Для использования данного полетного контроллера с Matlab используется дополнительный вспомогательный пакет UAV Toolbox Support Package for PX4 Autopilots, который позволяет генерировать C++ код и использовать среду PX4 toolchain для создания и загрузки собственных алгоритмов, совместимых с Pixhawk, на борт летательного аппарата.

ЛИТЕРАТУРА

1. **Tommaso Bresciani** «Modelling, Identification and Control of a Quadrotor Helicopter», Department of Automatic Control, Lund University, October 2008.
2. **Joao P. Hespanha** «Undergraduate Lecture Notes on LQG/LQR controller design», April 1, 2007.
3. **Mendes, A. S.** “Vision-based automatic landing of a quadrotor UAV on a floating platform”, Master of Science thesis, Faculty of Aerospace Engineering, Delft University of Technology, 2012.
4. https://docs.qgroundcontrol.com/master/en/getting_started/download_and_install.html

---

A.A. Tkachenko (Institute of Control Sciences V.A. Trapeznikov Academy of Sciences, Moscow). **Solving the problem of stabilizing a quadrocopter using a linear-quadratic controller: modeling and practice**

Abstract. The paper considers the application of LQR-synthesis to the simulation of a quadcopter flight, implemented in the Matlab environment. This simulation is necessary to establish the correctness of the derived mathematical model and the operability of the designed regulator before using it on the physical model of the device based on the Pixhawk flight controller.