

Разработка системы управления квадрокоптером в SimMechanics

П. Р. Соловьев, Л. Ю. Савина, С. В. Захаркина, Д. В. Масанов,

Е. А. Рыжкова

Российский государственный университет им. А.Н. Косыгина (Технологии. Дизайн. Искусство), Москва

Аннотация: В этой статье рассматривается автоматический подбор коэффициентов ПИД-регуляторов двигателей квадрокоптера на основе 3D-модели. Данный подбор стал возможен благодаря рассмотренному в статье методу экспорта 3D-модели квадрокоптера, созданной в САД-системе Solidworks, в среду Matlab/Simulink, а также с дальнейшим применением библиотеки SimMechanics. Для подбора коэффициентов в проекте Simulink также была реализована система управления квадрокоптером, сигналы с которой поступали на двигатели в соответствии с их физическим расположением. Результатом статьи является визуализация полета 3D-модели квадрокоптера с системой управления, реализованной в Simulink.

Ключевые слова: solidworks, matlab, simulink, квадрокоптер, бпла, пид, simmechanicsю

В настоящее время беспилотные летательные аппараты (БПЛА) находят применение в различных сферах и отраслях, становятся более востребованными: начиная от создания карт области, инвентаризации складских помещений, контроля безопасности, заканчивая медиа и киноиндустрией. Модели БПЛА адаптируют под заданные условия: небольшие дроны для мобильности и лёгкой маневренности, дроны с большой рамой для переноски более тяжёлого груза.

Для упрощения создания реальной модели необходимо провести специальные исследования на ранних этапах проектирования по поведению данного устройства в реальной среде для исследования динамических свойств. Компьютерное моделирование позволяет испытать прототип модели, выявить недостатки конструкции, провести первичные настройки органов управления и избежать многих ошибок при пуско-наладке.

Целью данного проекта является симуляция модели квадрокоптера в среде SimMechanics путём переноса созданной 3D модели, проведение имитационного моделирования полета, получение передаточной функции.

Квадрокоптер проектируется в CAD-среде SolidWorks, с помощью которой для модели можно задать не только внешний вид, но и физические свойства объекта, такие как центр тяжести, массу и инерцию. Готовый проект экспортируется в Simulink в виде блоков библиотеки SimScape SimMechanics, где происходит настройка системы управления полётом для имитации полета модели.

SimMechanics является частью библиотеки SimScape среды Matlab/Simulink и позволяет провести математическое имитационное моделирование с использованием 3D-визуализации [1]. В результате использования данной библиотеки, на выходе мы получаем не только расчёты, но и визуализацию работы квадрокоптера в реальной среде, учитывая такие физические параметры, как гравитация.

Для того, чтобы провести подобное необходимо синтезировать 3D-модель в среде Matlab/Simulink, для чего её сначала потребуется построить в CAD-системе. В данной статье рассмотрена работа с системой SolidWorks. Преимущество данной CAD-системы в том, что, помимо проектирования точной копии реального беспилотного летательного аппарата в 3D-среде, мы можем настроить физические параметры создаваемого объекта, такие, как вес и центр масс [2]. В 3D-модели можно пренебречь различными мелкими деталями, такими, как, например, винты. Данные детали слабо влияют на аэродинамические свойства, а также сильно нагружают систему, в итоге рендер небольшого куса полёта может занять очень долгое время [3].

Также, прежде чем начинать создание 3D-модели в системе SolidWorks, нужно учитывать тот факт, что среде MatLab/Simulink при визуальной симуляции осью, направленной вверх относительно горизонта, выступает ось Z, тогда как аналогичной осью в системе SolidWorks является ось Y. Если не учитывать данный факт при проектировании, то экспортированная в Simulink

модель будет перевернута, соответственно, не будет работоспособна и не пригодна к дальнейшей настройке [4].

Система SolidWorks изначально не позволяет производить экспорт моделей в среду Matlab/Simulink, поэтому для синтеза модели потребуются установка вспомогательного программного комплекса SimScape Multibody, разработанного специалистами MathWorks. С помощью этого дополнения реализуется возможность экспорта модели из CAD-системы SolidWorks в среду Matlab/Simulink, и весь процесс делится на 2 этапа.

Для активации описанного выше плагина сначала необходимо в самой среде Matlab прописать команду `smlink_linksw`. Эта команда добавляет плагин SimScape Multibody в реестр Windows, и делает его обнаружение доступным для CAD-системы. Теперь необходимо открыть систему SolidWorks, зайти в меню «Инструменты» - «Добавления», и там нажать галочку рядом с появившимся пунктом «SimMechanics Link». Таким образом плагин активирован.

Для того, чтобы произвести первый этап синтеза модели – экспорт модели из системы SolidWorks – необходимо открыть файл модели, затем открыть пункт меню «Инструменты», где надо найти пункт «SimMechanics Link». В данном пункте меню нужно выбрать пункт «Export» - «SimMechanics Second Generation». После нажатия появится окно выбора папки для сохранения экспортируемой модели. Выбрав место для экспорта, запустится и сам процесс, результатом которого станет папка с несколькими файлами формата STL, содержащими данные о геометрии составляющих 3D-модели, а также XML-файл, содержащий информацию о физических параметрах модели и положении объектов, находящихся в файлах, описанных выше.

Во втором этапе, импорте модели, сначала необходимо произвести последовательность действий в среде Matlab. В элементе Current folder

необходимо найти сохраненную папку с экспортированной из системы SolidWorks моделью, нажать правой кнопкой мыши по папке и выбрать действие «Add to path» - «Selected folders and subfolders». Данное действие требуется для того, чтобы среда Matlab понимала, где искать экспортированные файлы. После данной операции необходимо в командной строке среды Matlab прописать команду `smimport('name')`, где `name` – название xml-файла без указания его расширения. Данная команда приведет к началу процесса импорта экспортированной и системы SolidWorks модели в среду Matlab/Simulink, окончанием которого станет открытие окна Simulink с непрезентабельным, но готовым к работе кодом. Часть данного кода приведена на рисунке 1.

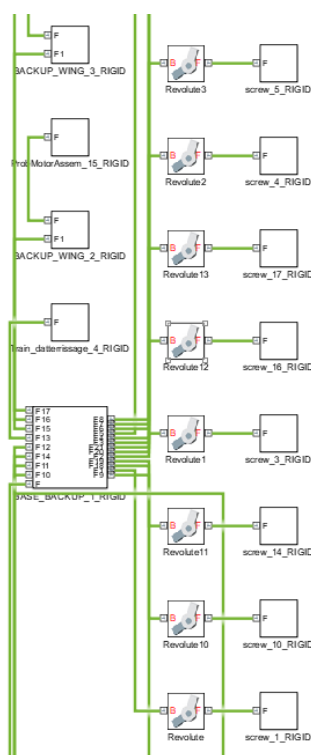


Рис. 1. – Экспортированная система

Таким образом, нам удалось произвести синтез модели из CAD-системы SolidWorks в среду Matlab/Simulink. В текущем виде для данной модели уже можно запустить моделирование процесса, однако ничего происходить не будет – мы получим лишь простой статичный объект, так как

в файле Simulink сейчас находится лишь информация о самом объекте, но не о работе двигателей и системе управления.

Прежде чем продолжить работу с данной моделью – изменим оригинальную структуру файла. Изначально блоки расположены в хаотичном порядке. Вручную структурируем блоки, пряча всё лишнее в одну подсистему, оставляя снаружи служебные блоки SimMechanics и несколько других подсистем, отвечающих за движение пропеллеров. Новое расположение блоков отображено на рисунке 2. В служебных блоках задается информация о мире, в том числе можно задать ускорение свободного падения. Установим этому параметру значение 9.7.

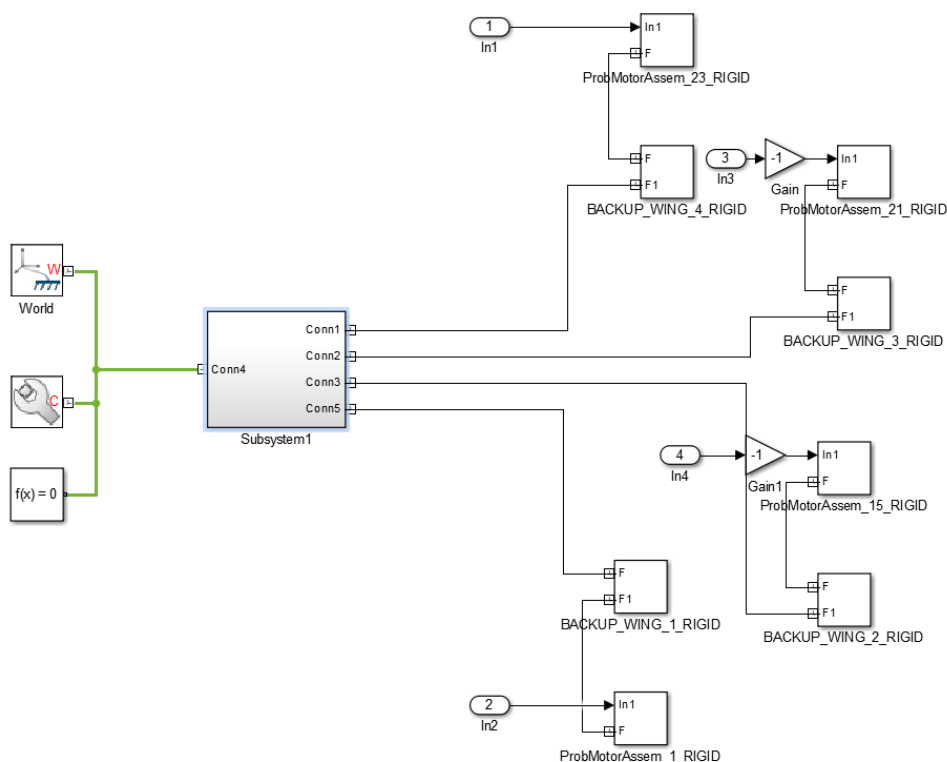


Рис. 2. – Отредактированное расположение блоков 3D-модели в Simulink

В результате, процесс синтезирования модели окончен, далее переходим к моделированию полёта. Благодаря нему мы сможем правильно настроить параметры регуляторов системы управления, не прибегая к экспериментам с реальным оборудованием, что снизит затраты времени на настройку и риски порчи оборудования.

Для перемещения беспилотного летательного аппарата используются четыре степени свободы – высота, на которую взлетает устройство; крен, продольная ось; рыскание, вертикальная ось; тангаж, поперечная ось.

Управляющим воздействием является пропорционально-интегрально-дифференцирующий регулятор (ПИД), подающий значения на двигатели, работающий в связке обратной связи, регулируя значения в зависимости от поступающей информации с датчиков, которые передают данные о погодных условиях. Каждый из составляющих регулятора оказывает непосредственное влияние на качество полёта квадрокоптера. В зависимости от поставленных задач, таких, как приоритет скорости или грузоподъёмности, выбираются различные настройки. Пропорциональный коэффициент является статическим множителем, отвечающим за силу исправления погрешностей, высокие коэффициенты подходят для скоростных аппаратов, низкие же необходимы для плавности хода. В свою очередь, пропорциональный коэффициент несёт в себе статическую ошибку и сам по себе не является достаточным для управления. Интегральный коэффициент, по большей мере, отвечает за скорость реакции на воздействия окружающей среды, в особенности на ветер. Дифференциальная часть корректирует пропорциональную составляющую, при больших значениях появляются вибрации, при низких происходит быстрое переключение двигателей, что влияет на срок эксплуатации устройства [5]. Регулировка параметров ПИД производится с помощью средств Matlab, с применением функции Pidtool.

Для того, чтобы заставить элементы квадрокоптера двигаться, необходимо между соединёнными объектами (материнским и дочерним) установить специальные блоки, позволяющие объекту двигаться с заданной степенью свободы относительно материнского объекта, к которому он привязан. В случае с пропеллерами подойдут шарнирные блоки, описывающие работу шарнира в качестве степени свободы пропеллера

относительно квадрокоптера. Помимо задания степени свободы, на эти блоки можно подать управляющий сигнал, который заставит шарнир двигаться, также можно снять информацию о положении шарнира, скорости и так далее [6].

Установим шарнирный блок на линию, соединяющую пропеллер с дроном. Он будет отвечать за вращение пропеллера. В настройках этого блока добавим управляющее входное воздействие, которое будет устанавливать скорость вращения. На данный вход через инвертирующий блок-сигнала Simulink-SimScare подадим управляющий сигнал, и выведем в виде входа на верхний уровень подсистемы. Такая же процедура проделывается с каждым пропеллером.

Также добавим некоторое количество шарниров, дополнительно переворачиваемых до необходимого состояния, на линию, соединяющую квадрокоптер с окружающим миром. Данные шарниры нужны для получения обратной связи от модели. Помимо шарниров добавляем блоки Rectangular и Prismatic joint, позволяющие модели свободно перемещаться в 3 плоскостях [7]. Таким образом, завершена подготовка модели к симуляции, и можно создавать систему управления.

Для каждой степени создается свой регулятор, в итоге получается четыре регулятора. На входы регуляторов через сумматор подаются два сигнала – команда управления (уставка), и сигнал с датчика о текущем положении (обратная связь). С каждого регулятора сигнал поступает на четыре вычислительных блока, результат которого является мощностью подаваемого на винт воздействия. Порядок вычислений каждого из блоков отличается в зависимости от расположения пропеллера относительно корпуса квадрокоптера [8].

При управлении квадрокоптером используются пропеллера, попарно-диагонально отличающиеся углом лопастей и направлением вращения, что

необходимо для правильного управления рысканием, так как все пропеллеры направлены вверх. Для управления высотой, для взлёта, одинаково прибавляется мощность всех двигателей. Для управления тангажом, для наклона вперёд, прибавляется мощность задних двигателей, и убавляется мощность передних, для наклона назад соответственно обратное действие. Для управления креном, для наклона вправо, прибавляется мощность левых двигателей, и убавляется мощность правых, противоположные действия для наклона влево. Для управления рысканием, при развороте по часовой стрелке, увеличивается мощность правого переднего и заднего левого двигателей, а передний левый и задний правый получают меньшее управляющее воздействие [9].

Основываясь на описанной выше информации об управлении двигателями, сигнал управления высотой для всех двигателей подается положительным. Для переднего правого двигателя к команде управления высотой прибавляются команда рыскания и тангажа, а команда крена вычитается. Для переднего левого двигателя – вычитаются команды рыскания и крена, и прибавляется тангаж. Для заднего правого вычитаются рыскание и тангаж, а прибавляется крен. И, наконец, для заднего левого двигателя прибавляется сигнал рыскания, и вычитаются крен и тангаж [10]. Схема управления изображена на рисунке 3.

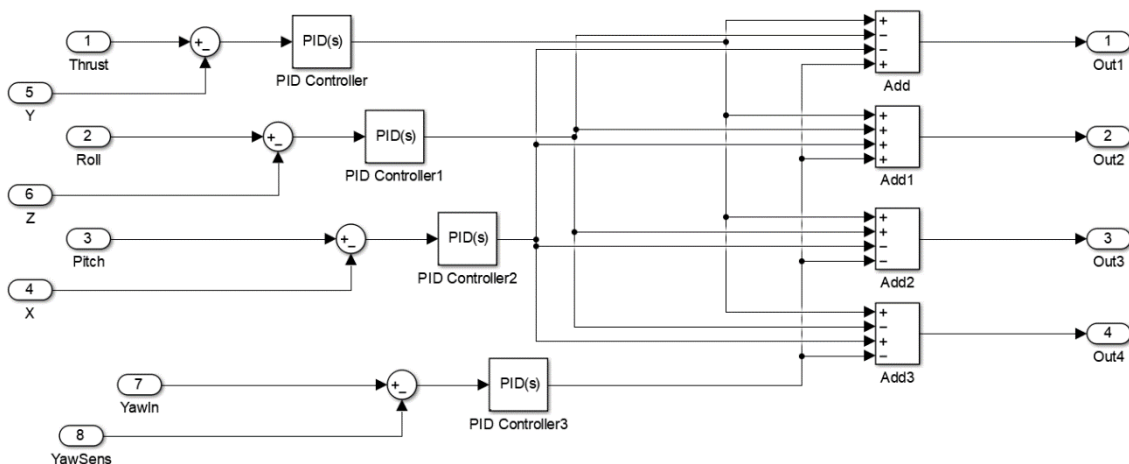


Рис. 3 – Схема управления квадрокоптером в Simulink

Таким образом у нас реализована система управления полётом. Всё, что осталось сделать – это правильно соединить блок управления и блок квадрокоптера: управляющие сигналы, обратная связь. Также добавим входные управляющие воздействия на блок управления и выведем информацию о сигналах с датчиков для наблюдения за графиком передаточной функции. Схема итогового подключения блоков изображена на рисунке 4.

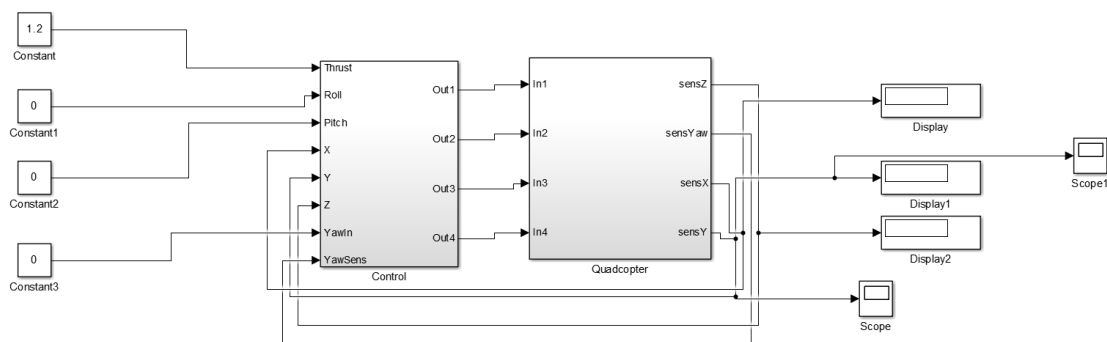


Рис. 4. – Общая схема симуляции и управления квадрокоптером в Simulink

В результате запуска этого проекта в окне Matlab можно увидеть визуализацию полета 3D-модели в заданное в Simulink время. На данной визуализации наблюдается взлёт и стабильное удержание квадрокоптера на

определённой высоте. Скриншот данной визуализации изображен на рисунке 5.

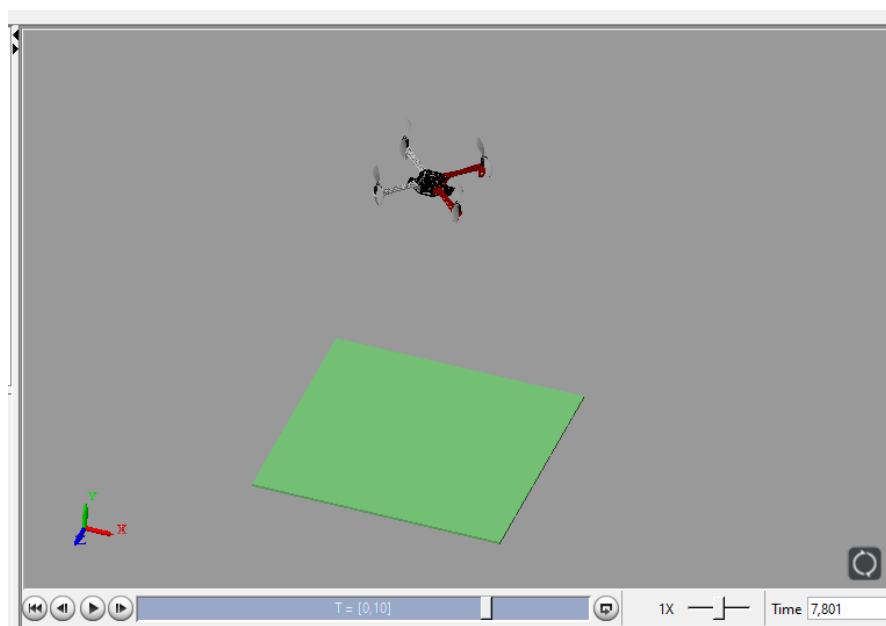


Рис. 5. – 3D-симуляция полёта квадрокоптера

Откроем график Scope, подключенный к датчику высоты беспилотного летательного аппарата. На нём также наблюдается плавный набор высоты и её дальнейшее удержание, что говорит о подходящей настройке ПИД-регулятора. График взлёта изображен на рисунке 6.

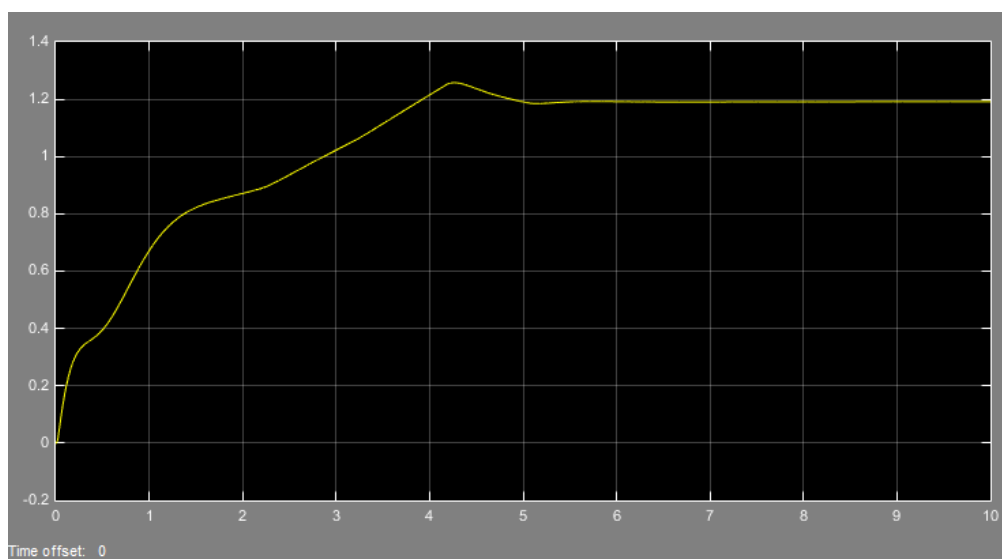


Рис. 6. – График контроля набора и удержания высоты квадрокоптера

В результате нам удалось синтезировать в среде Matlab/Simulink модель, построенную в САД-системе SolidWorks и разработать систему управления полетом для данной модели, и убедиться в её работоспособности.

Литература

1. Сайфеддин Д., Булгаков А.Г., Кругловаж Т.Н. Нейросетевая система отслеживания местоположения динамического агента на базе квадрокоптера. Инженерный вестник Дона. 2014. №1. URL: ivdon.ru/magazine/archive/n1y2014/2293/.

2. Лазарев В.С., Лащев А.А. Разработка математической модели БПЛА на базе квадрокоптера с рамой DJI F-450. Инженерный вестник Дона. 2018. №2. URL: ivdon.ru/ru/magazine/archive/N2y2018/5001/.

3. Quadcopter Flight Control Using a Non-invasive Multi-Modal Brain Computer Interface. Frontiers. URL: frontiersin.org/articles/10.3389/fnbot.2019.00023/full (дата обращения: 13.11.2023).

4. Белоглазов, Гайдук, Косенко, Медведев, Пшихопов, Соловьев, Титов, Финаев, Шаповалов, под ред. Пшихопова В.Х. Групповое управление подвижными объектами в неопределенных средах. М.: Физматлит, 2015. 305 с.

5. Кульченко А.Е., Лазарев В.С., Медведев М.Ю. Метод управления движением гексакоптера в трехмерной среде с препятствиями на базе динамических отталкивающих сил. Инженерный вестник Дона. 2016. №4. URL: ivdon.ru/ru/magazine/archive/n4y2016/3828/.

6. Madani T., Benallegue A. Backstepping control for a quadrotor helicopter. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. 2006. URL: ieeexplore.ieee.org/document/4058900.

7. Castillo P., Dzul A., Lozano R. Real-time stabilization and tracking of a four-rotor mini rotorcraft. IEEE Transactions on Control Systems Technology. 2004. №12 (4). URL: ieeexplore.ieee.org/document/1308180.
8. Saripalli S. Landing a Helicopter on a Moving Target. Robotics and Automation. IEEE International Conference. 2007. URL: ieeexplore.ieee.org/document/4209384.
9. Gavrilets V., Martinos I., Mettler B., Feron E. Control logic for automated aerobatic flight of miniature helicopter. Navigation and Control Conference. AIAA Guidance. 2002. 8 с.
10. Kyaw Myat Thu. Designing and Modeling of Quadcopter Control System Using L1 Adaptive Control. Procedia Computer Science. 2017. №103. С. 528-535.

References

1. Sayfeddin D., Bulgakov A.G., Kruglova T.N. Inzhenernyj vestnik Dona. 2014. No. 1. URL: ivdon.ru/magazine/archive/n1y2014/2293/.
2. Lazarev V.S., Lashchev A.A. Inzhenernyj vestnik Dona. 2018. No. 2. URL: ivdon.ru/ru/magazine/archive/N2y2018/5001/.
3. Quadcopter Flight Control Using a Non-invasive Multi-Modal Brain Computer Interface. Frontiers URL: frontiersin.org/articles/10.3389/fnbot.2019.00023/full (date of the application: 13.11.2023).
4. Beloglazov, Gaiduk, Kosenko, Medvedev, Pshikhopov, Soloviev, Titov, Finaev, Shapovalov, Ed. Pshikhopov V.Kh. Group control of moving objects in uncertain environments. M.: Fizmatlit [Physical Education], 2015. 305 p.
5. Kulchenko A.E., Lazarev V.S., Medvedev M.Yu. Inzhenernyj vestnik Dona. 2016. № 4. URL: ivdon.ru/ru/magazine/archive/n4y2016/3828/.



6. Madani T., Benallegue A. Backstepping control for a quadrotor helicopter. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. 2006. URL: ieeexplore.ieee.org/document/4058900.

7. Castillo P., Dzul A., Lozano R. Real-time stabilization and tracking of a four-rotor mini rotorcraft. IEEE Transactions on Control Systems Technology. 2004. №12 (4). URL: ieeexplore.ieee.org/document/1308180.

8. Saripalli S. Landing a Helicopter on a Moving Target. Robotics and Automation. IEEE International Conference. 2007. URL: ieeexplore.ieee.org/document/4209384.

9. Gavrilets V., Martinos I., Mettler B., Feron E. Control logic for automated aerobatic flight of miniature helicopter. Navigation and Control Conference. AIAA Guidance. 2002. 8 p.

10. Kyaw Myat Thu. Designing and Modeling of Quadcopter Control System Using L1 Adaptive Control. Procedia Computer Science. 2017. №103. P. 528-535.

Дата поступления: 24.10.2023

Дата публикации: 29.12.2023