



СИСТЕМНЫЙ АНАЛИЗ

УДК 656.7

DOI: 10.31799/2077-5687-2021-4-3-10

АЛГОРИТМ ТОЧНОЙ ПОСАДКИ МУЛЬТИРОТОРНЫХ БЕСПИЛОТНЫХ АВИАЦИОННЫХ СИСТЕМ НА СТАНЦИЮ АВТОНОМНОЙ ЗАРЯДКИ

Е. А. Вознесенский

Санкт-Петербургский государственный университет аэрокосмического приборостроения

В данной статье предложен алгоритм точной посадки мультироторных (квадрокоптеры, гексакоптеры и т. д.) беспилотных летательных аппаратов (БПЛА) на станцию автономной зарядки. Также в этой статье представлены методы локализации зарядной станции и посадки БПЛА в ночное время. В разделе 1 описаны общие последовательные процедуры посадки. В разделе 2 рассказывается о методах обнаружения маркера ArUco и оценки его положения и ориентации с помощью библиотеки компьютерного зрения OpenCV, показан результат распознавания. В разделе 3 подробно разобран алгоритм точной посадки, приведена структурная блок-схема алгоритма. В разделе 4 обсуждается интеграция камеры ночного видения в алгоритм посадки.

Ключевые слова: квадрокоптеры, дрон, БПЛА, точная посадка, зарядная станция.

Для цитирования:

Вознесенский Е. А. Алгоритм точной посадки мультироторных БПЛА на станцию автономной зарядки // Системный анализ и логистика: журнал.: выпуск №4(30), ISSN 2077-5687. – СПб.: ГУАП., 2021 – с. 3-10. РИНЦ. DOI: 10.31799/2077-5687-2021-4-3-10.

ALGORITHM FOR PRECISE LANDING OF MULTIROTOR UNMANNED AIRCRAFT SYSTEMS AT AN AUTONOMOUS CHARGING STATION

E. A. Voznesenskii

State University of Aerospace Instrumentation

In this article, we propose an algorithm for accurately landing multirotor (quadcopters, hexacopters, etc.) unmanned aerial vehicles (UAVs) at an autonomous charging station. This article also presents methods for locating the charging station and landing the UAV at night. Section 1 describes the general sequential landing procedures. Section 2 describes methods for detecting the ArUco marker and evaluating its position and orientation using the OpenCV computer vision library and shows the recognition result. In section 3, the precise landing algorithm is analyzed in detail, and a block diagram of the algorithm is given. Section 4 discusses the integration of the night vision camera into the landing algorithm.

Key words: quadcopters, drone, UAV, precise landing, charging station.

For citation:

Voznesenskii E. A. Algorithm for precise landing of multirotor UAVs at the autonomous charging station // System analysis and logistics.: №4(30), ISSN 2077-5687. – Russia, Saint-Petersburg.: SUAI., 2021 – p. 3-10. DOI: 10.31799/2077-5687-2021-4-3-10.

Введение

С начала 21 века наблюдается рост популярности дистанционно управляемых беспилотных летательных аппаратов (БПЛА) и, в частности, мультикоптеров (квадрокоптеры, трикоптеры, октокоптеры и т. д.). Квадрокоптеры теперь повсеместно распространены и используются для решения различных задач: от развлекательных полетов до военных миссий. На текущий момент при помощи беспилотников выполняются такие задачи, как анализ окружающей среды, картографирование, поиск и спасение людей в сложной местности и так далее. Подобные задачи могут решать беспилотники в любое время, в сложных погодных условиях без риска для персонала [1,2,3].

Но беспилотники имеют на борту аккумуляторы с относительно малой удельной энергоемкостью, что не позволяет им совершать длительные автономные полетные миссии. Для решения этой проблемы в инфраструктуру беспилотников внедряются различные концепты автономной подзарядки дронов в полете или на наземных станциях [4-9].

Но для реализации системы зарядки дрона, которая не требует вмешательства человека,



также необходимо автоматически направлять дрона прямо к зарядной станции и управлять им таким образом, чтобы приземляться точно на зарядную платформу в правильном положении и ориентации. Для достижения этой цели было решено объединить систему GPS позиционирования и алгоритмы, основанные на компьютерном зрении.

Фактическое местоположение, в которое летит беспилотник, имеет определенные смещения по сравнению с целевым местоположением из-за систематических ошибок со стороны контроллера полета и модуля GPS. Смещения обычно находятся в пределах полуметра. Смещения не являются существенными для миссии, которая требует, чтобы беспилотник летел в определенное место для выполнения поставленной задачи. Однако для точной посадки, требующей точных движений, погрешности такого масштаба значительно снижают точность посадки.

1. Обзор алгоритма точной посадки

Для автономной посадки БПЛА на зарядную платформу решено использовать GPS-навигацию совместно с системой нахождения маркеров ArUco (рис. 1) [10] для контроля положения дрона в пространстве. Бортовой компьютер беспилотника, который имеет связь с сервером, инициирует алгоритм подзарядки если у квадрокоптера достаточно разряжен аккумулятор. Бортовой компьютер с сервера получает информацию о ближайшей свободной зарядной платформе и командует дрону приземлиться на назначенную станцию. Расположение зарядных платформ хранится в статической базе данных, к которой бортовой компьютер может получить доступ.

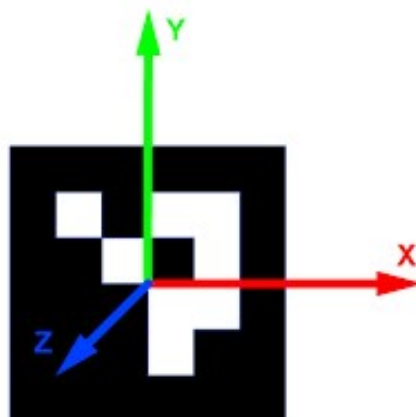


Рис. 1. ArUco-маркер

Процедуры точной посадки имеют четыре этапа:

1. Когда беспилотник инициирует миссию подзарядки, бортовой компьютер командует дрону лететь к целевому GPS-местоположению зарядной станции.
2. Поскольку бортовой GPS-приемник имеет относительно большую погрешность, изначальная навигация по направлению к зарядной платформе недостаточно точна, чтобы точно посадить беспилотник на зарядную платформу. На втором этапе точной посадки используется маркер ArUco, расположенный на зарядной платформе, а также камера, направленная строго вниз, нужная для отслеживания положения маркера. При обнаружении ArUco метки, алгоритм управления начинает позиционировать беспилотник в центр над маркером, повторяя подстройку местоположения дрона. Регулировка положения продолжается до тех пор, пока беспилотник не достигнет центра прямо над маркером или не произойдет ошибка по времени выполнения выравнивания.
3. После завершения цикла позиционирования на определенной высоте беспилотник опускается на более низкий уровень, чтобы повторить еще один цикл



позиционирования, чтобы беспилотник все время оставался над маркером во время спуска. Также возможно использовать несколько ArUco маркеров разных размеров чтобы, когда больший маркер выйдет за поле зрения камеры, квадрокоптер сможет распознать меньший маркер, таким образом расстояние от платформы до беспилотника будет сведено к минимуму. И окончательная посадка, производимая по инерциальным датчикам, не успеет накопить достаточную погрешность, чтобы повлиять на точность посадки. Ошибки, вызванные ветром и дрейфом, также сведены к минимуму.

4. Бортовой компьютер сообщает о состоянии посадки, когда двигатели дрона выключены. Автономный процесс заряда бортового аккумулятора начинается после завершения посадки.

2. Обнаружение маркеров и оценка их положения

Для обнаружения маркеров ArUco и оценки позиции используется библиотека маркеров OpenCV ArUco [11] Оценка включает в себя нахождение положения маркера относительно камеры.

Как только маркер будет сгенерирован с помощью предопределенного словаря DICT_4x4_250, затем напечатан, он размещается поверх посадочной платформы.

Для обнаружения маркера анализируется каждый кадр изображения, захваченный камерой, и используется встроенная функция обнаружения OpenCV detectMarkers() для обнаружения маркера [12, 13]. Обработывая изображение, процесс обнаружения должен возвращать список обнаруженных маркеров. Каждый обнаруженный маркер включает в себя:

- Идентификатор/номер/id маркера (По своей сути, маркер – закодированное число. Если маркер имеет размер 4×4, то кодировка занимает 16 бит соответственно)
- Положение 4-х углов маркера на изображении в определенном порядке. Таким образом определяется относительное положение маркера в трехмерном пространстве.

Алгоритм обнаружения состоит из двух частей [13]:

Полученный кадр анализируется, производится поиск квадратных фигур, которые станут кандидатами на роль маркеров. Анализ начинается с адаптивного порогового преобразования для сегментации маркеров, затем из порогового изображения извлекаются контуры. Те контуры, которые не являются выпуклыми или не похожи на квадратную форму - отсеиваются. В дополнение имеет место быть дополнительные процедуры фильтрации, такие как удаление контуров со слишком малыми или большими геометрическими размерами, отбрасывание слишком близко расположенных друг к другу контуров и так далее. На рисунке 2 представлен пример порогового преобразования для определения прямоугольного контура.

Затем анализируются уже отобранные кандидаты и определяется их принадлежность к ArUco - маркерам путем анализа их внутреннего кодирования. Этот шаг начинается с извлечения битов в каждом маркере. Для этого получается каноническая форма маркера путем перспективного преобразования. Полученное каноническое изображение обрабатывается с помощью метода Оцу, который вычисляет порог бинаризации, чтобы отделить белые и черные биты. Исходя из размеров маркера и его границ, изображение делится на различные секции. Далее, чтобы выяснить является данная секция белым или черным, считается количество черных или белых пикселей в каждой секции. В итоге по полученным битам происходит сопоставление маркера со словарем, определяется его принадлежность.

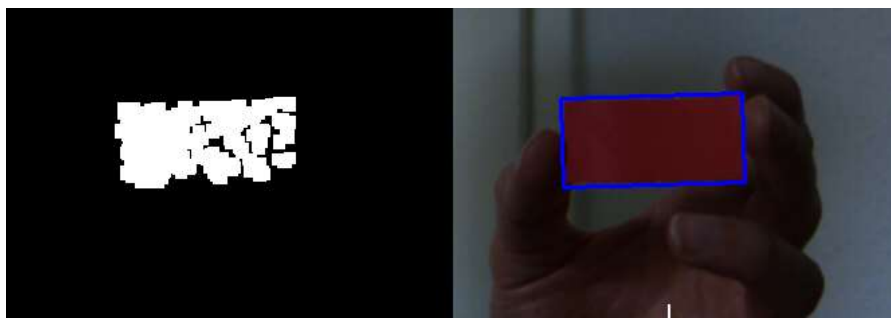


Рис. 2. - Пороговое преобразование изображения с меткой прямоугольной фигурой

Функция `estimatePoseSingleMarkers()` [12] из OpenCV используется для оценки положения маркера относительно кадра камеры. Функция принимает матрицу камеры, коэффициенты искажения, координаты углов маркера и реальный размер маркера в качестве входных данных и выводит вектор поворота и вектор перемещения, которые преобразуют точки из системы координат маркера ArUco в систему координат камеры. Вектор перемещения содержит положение маркера относительно камеры. Чтобы получить положение маркера относительно камеры, используется другая встроенная функция OpenCV `Rodrigues()` [10] для преобразования вектора поворота в матрицу поворота, которая содержит угол крена, тангажа и рыскания маркера относительно камеры в системе координат маркера.

В реальном сценарии маркер статичен на Земле, когда как камера, закрепленная на беспилотнике, и движется. Поэтому вычисляется положение камеры относительно маркера.

На следующем изображении показано определение положения маркеров. Цветовые соответствия оси X: красный, Y: зеленый, Z: синий. Обратите внимание на направления осей повернутых маркеров на этом изображении. Также можно заметить, что похожие на маркеры изображения на коробке отсеялись в ходе распознавания.

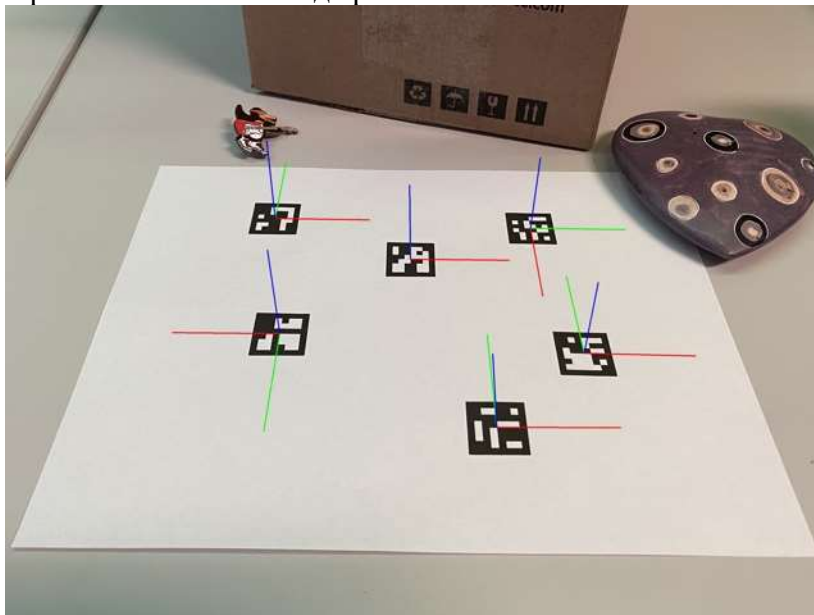


Рис. 3. Маркеры с обозначенными осями собственных координат

3. Алгоритм посадки

Алгоритм управления посадкой использует положение маркера на зарядной платформе в качестве обратной связи для позиционирования дрона так, чтобы он мог зависать непосредственно над центром маркера перед посадкой.

Когда нужно расположить беспилотник над маркером перед посадкой, на точность влияет систематическая ошибка и условия окружающей среды, такие как, например, ветер.



Чтобы компенсировать помехи, существует как минимум два метода управления: повторяющееся командное управление и PID-управление. Также лучше постепенно опускаться беспилотник и центрировать его на нескольких высотных уровнях перед окончательной посадкой. На рисунке 4 показана общая структурная схема алгоритма точной посадки.

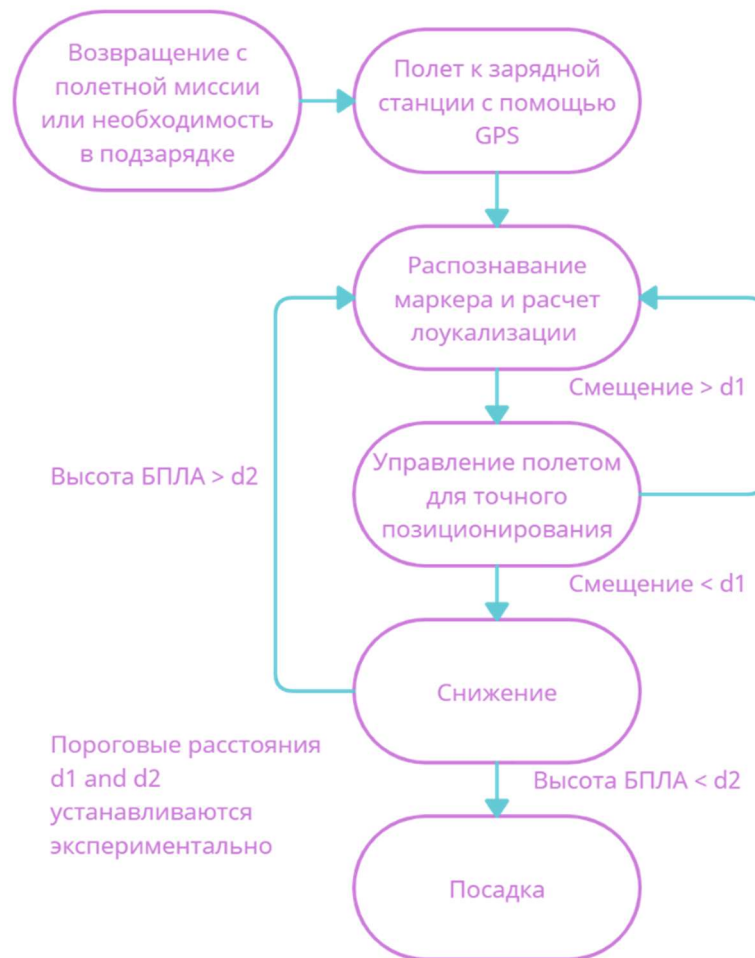


Рис. 4. Блок-схема алгоритма механизма управления точной посадкой

Как только беспилотник зависает над зарядной платформой примерно в 10 метрах, начинается процедура обнаружения маркера и оценки положения дрона относительно него. После того, как вычислено смещение между местоположением мультикоптера и местоположением маркера, алгоритм управления полетом выравнивает беспилотник над маркером. В конце каждого цикла позиционирования выполняется еще один этап обнаружения маркера ArUco и оценки положения, чтобы проверить относительное расстояние между дроном и маркером. Если расстояния по осям x и y мировой системы координат находятся в пределах определенного экспериментально установленного порога смещения d_1 , беспилотник опускается на меньшую высоту. Если расстояние по оси x или y превышает пороговое значение, то цикл позиционирования повторяется до тех пор, пока не закончится зарезервированное время для позиционирования на каждом уровне высоты или же относительное расстояние не опустится ниже порогового значения. Ошибка “тайм-аута” записывается, если беспилотник не может переместиться к месту в пределах лимита по времени при выполнении процесса позиционирования. Важно, чтобы мультикоптер окончательно садился на относительно низкой высоте, чтобы ветер имел меньше времени на помеху местоположению дрона, а инерциальные датчики не успели накопить существенную ошибку. Если высота дрона все еще выше порога d_2 , высота дрона плавно понижается,



передавая те же координаты x и y текущего положения дрона, а также более низкие координаты z в функцию управления в ROS [14]. Величина декрементов по оси z корректируется в ходе фактических испытаний. Если высота дрона над платформой становится меньше расстояния d_2 , а смещение между дроном и маркером также меньше d_1 , то беспилотник окончательно садится на платформу, используя инерциальные датчики или опираясь на систему позиционирования optical flow.

Optical flow – технология позиционирования, так же как и позиционирование по ArUco – маркерам, основана на компьютерном зрении. Для ее работы необходима камера, направленная вниз и дальномер, измеряющий высоту над рельефом. Кадры с камеры обрабатываются на предмет смещения относительно раннего положения. Таким образом регистрируемое движение можно компенсировать и достаточно точно зависнуть в плоскости xy при окончательной посадке.

Для управления дроном в глобальной и локальной системах координат используются специальные функции ROS (Robot Operating System) фреймворка [14,15]. ROS – гибкая структура для написания программного обеспечения робота. Это набор инструментов, библиотек и соглашений, направленных на упрощение задачи создания сложного и надежного поведения роботов на самых разных роботизированных платформах.

4. Ночное видение

При выполнении ночных полетных заданий для реализации алгоритма точной посадки должна использоваться камера ночного видения с автоматическим ИК-датчиком, потому что автоматизированная система управления мультироторными БПЛА, а в частности зарядная платформа, должна работать в ночное время. С помощью модуля камеры ночного видения беспилотник может обнаружить маркер ArUco в темной время суток и выполнить те же процедуры посадки, что описаны выше. Конкретные методы управления настраиваются в соответствии с камерой ночного видения.

Ночью обычная камера типа "рыбий глаз" не может обнаружить маркер ArUco, в то время как камера ночного видения с мощными ИК-светодиодами и ИК-датчиками может посылать ИК-свет на маркер и фиксировать отраженный ИК-свет для обнаружения маркера.

Инфракрасный датчик на камере автоматически убирает инфракрасный фильтр при низком уровне видимого света, таким образом камера иницирует режим ночного видения. Камера ночного видения может также работать в течение дня с примененным ИК-фильтром. Точность обнаружения маркера ArUco камерой ночного видения в течение дня и ночи одинакова.

Заключение

Описанный алгоритм в данной статье – лишь один из шагов на пути к полной автоматизации и так огромной работы беспилотников.

Создание инфраструктуры беспроводной зарядки в сфере выполнения задач мультикоптерами может обеспечить возможность непрерывного выполнения полетной миссии без какого-либо вмешательства человека. Время полета современных квадрокоптеров сильно ограничено из-за емкости аккумулятора. Беспроводная зарядка позволяет беспилотникам заряжаться автономно во время полета к месту назначения, тем самым продолжая полет после подзарядки без какого-либо внешнего прерывания.

СПИСОК ЛИТЕРАТУРЫ

1. Вознесенский Е. А., Костин А. С., Майоров Н. Н. Анализ возможностей и ограничений при внедрении беспилотных авиационных систем для решения транспортных задач // Сборник трудов XIII Всероссийская научно-практическая конференция студентов и аспирантов «Актуальные проблемы развития авиационной техники и методов ее эксплуатации – 2020» 8-9 декабря 2020 г. Том 1. – Иркутск:



- Иркутский филиал МГТУ ГА, 2021.– 303 с. ISBN 978-5-9500497-7-4 – С. 265-270.
2. Kostin A. S., Bogatov N. V., Maiotov N. N. Control and analysis of quadcopter flight when setting a complex trajectory of motion. *Journal of Physics: Conference Series*. 1925 012043 1925 (2021) 012043. doi:10.1088/1742-6596/1925/1/012043
 3. Майоров Н. Н., Костин А. С., Богатов Н. В., Вознесенский Е. А. Исследование маршрутизации полетных заданий беспилотных авиационных систем для операционных складских задач / Н. Н. Майоров, А. С. Костин, Н. В. Богатов, Е. А. Вознесенский // *Аэрокосмическое приборостроение и эксплуатационные технологии. Сборник докладов Второй Международной научной конференции*. СПб.:ГУАП. 2021. С. 130-137. Doi: 10.31799/978-5-8088-1554-4-2021-2-130-137
 4. Контактная зарядная станция [Электронный ресурс] // Skysense - URL: <https://www.skysense.co/> (дата обращения: 15.05.2021).
 5. Станция с механизированной заменой аккумуляторов [Электронный ресурс] // Airobotics - URL: <https://www.airoboticsdrones.com/> (Дата обращения: 15.05.2021).
 6. Беспроводная зарядка дронов в полете [Электронный ресурс] // GET - URL: <http://getcorp.com/> (дата обращения: 15.05.2021).
 7. Лазерная зарядка БПЛА в полете [Электронный ресурс] // Сколково - URL: <https://old.sk.ru/news/b/press/archive/2018/12/10/v-rossii-razrabotali-metod-nepreryvnoy-zaryadki-dronov-s-pomoschyu-lazera.aspx> (дата обращения: 15.05.2021).
 8. “Летающие” батареи для квадрокоптеров [Электронный ресурс] // Cornell University - URL: <https://arxiv.org/abs/1909.10091> (дата обращения: 15.05.2021).
 9. Беспроводная зарядная станция [Электронный ресурс] // Webotic URL: <https://www.wibotic.com/> (дата обращения: 15.05.2021).
 10. Francisco J Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47, 2018.
 11. Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. <http://dx.doi.org/10.1016/j.patcog.2014.01.005>
 12. Распознавание ArUco - маркеров [Электронный ресурс] // OpenCV - URL: https://docs.opencv.org/master/d5/dae/tutorial_aruco_detection.html (дата обращения: 15.05.2021).
 13. Библиотека OpenCV [Электронный ресурс] // OpenCV - URL: <https://docs.opencv.org/4.5.2/> (дата обращения: 15.05.2021).
 14. Фреймворк ROS [Электронный ресурс] // ROS - URL: <https://www.ros.org/> (дата обращения: 15.05.2021).
 15. Свидетельство о государственной регистрации программы для ЭВМ 2021613689. Программа идентификации объектов при выполнении автономного полетного задания беспилотной авиационной системой с возможностью изменения маршрута / Н. Н. Майоров, А. С. Костин, Е. А. Вознесенский; заявитель и правообладатель Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет аэрокосмического приборостроения». – № 2021612780 ; заявл. 09.03.2021 ; опубл 12.03.2021 – 1 с.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Вознесенский Евгений Александрович —

магистрант кафедры системного анализа и логистики

Санкт-Петербургский государственный университет аэрокосмического приборостроения

190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А

E-mail: voznenskiy.evgeniy@mail.ru



INFORMATION ABOUT THE AUTHOR

Voznesenskii Evgenii Aleksandrovich —

master's degree student of the Department of System Analysis and Logistics

Saint-Petersburg State University of Aerospace Instrumentation

67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia

E-mail: voznensenskiy.evgeniy@mail.ru