

К. С. Амелин

Директор НОЦ СПбГУ «Математическая робототехника и искусственный интеллект»
Санкт-Петербургский государственный университет

Н.А. Житнухин

Выпускник кафедры информатики Математико-механического факультета
Санкт-Петербургский государственный университет

В. И. Кияев

Доцент кафедры информатики
Санкт-Петербургский государственный экономический университет

СИСТЕМА АВТОНОМНОЙ НАВИГАЦИИ ДЛЯ БЕСПИЛОТНЫХ РОБОТОТЕХНИЧЕСКИХ УСТРОЙСТВ

Аннотация. Современное состояние цифровых технологий позволяет в значительной мере использовать технологии Интернета вещей (IoT) в производственно-экономической и социальной сферах. Разнообразные робототехнические с исполнительными элементами в виде колёсных, гусеничных, подводных и воздушных аппаратов широко используются в промышленных и научных проектах, в мониторинговых и поисково-спасательных-работах, при обследовании больших незаселённых территорий, береговых линий и шельфов морей и крупных озёр. В случаях такого применения открытым остается вопрос об эффективной системе бортовой автономной навигации робототехнического устройства, которая должна обеспечить его надёжную работу в условиях динамической неопределённости. В статье предлагается решение в виде прототипа системы автономной навигации для беспилотного летательного аппарата (БПЛА).

Ключевые слова: IoT-технологии, управляющие и мониторинговые системы, робототехнические устройства, беспилотный летательный аппарат.

K. S. Amelin, N. A. Zhitnuhin, V. I. Kiyayev

AUTONOMOUS NAVIGATION SYSTEM FOR UNMANNED ROBOTIC DEVICES

Abstract. The current state of digital technologies makes it possible to largely use the technologies of the Internet of Things (IoT) in the production, economic and social spheres. A variety of robotic ones with actuators in the form of wheeled, tracked, underwater and air vehicles are widely used in industrial and scientific projects, in monitoring and search and rescue operations, when examining large uninhabited areas, coastlines and shelves of seas and large lakes. In cases of such an application, the question of an effective on-board autonomous navigation system of a robotic device remains open, which should ensure its reliable operation under conditions of dynamic uncertainty. The article proposes a solution in the form of a prototype of an autonomous navigation system for an unmanned aerial vehicle (UAV).

Keywords: IoT technologies, control and monitoring systems, robotic devices, unmanned aerial vehicles.

Введение

За последнее десятилетие технологии, системы и инструменты «Интернета вещей» (Internet of Things, IoT) перестали быть чем-то экзотическим и во многих областях промышленного производства и социальной жизни превратились в повседневную действительность. Понятия «умный дом», «умный город», «цифровые технологии» и «цифровые отношения» перешли из области онтологий в понятия, описывающие реальную жизнь и реальные события. Британский агрегатор и поставщик информации IHS Markit (Information Handling Services) несколько лет назад прогнозировал, что к 2025 году количество используемых устройств «Интернета вещей» вырастет с 17 до 80 миллиардов [4]. Благодаря этому производители электронных и цифровых устройств стали активно пересматривать подходы к планированию деятельности, управлению предприятиями, управлению эффективностью производственных и финансовых активов в режиме

реального времени и производству «умных» и синхронизированных с требованиями пользователей цифровых продуктов и услуг.

Интернет вещей, искусственный интеллект (Artificial Intelligence, AI) современная робототехника (Robotic Devices, RD) входят в пятерку наиболее интересных и востребованных технологий первой четверти XXI века. Отметим, что использование этих технологий в различных сочетаниях при решении сложных научно-практических, производственных и управленческих задач часто приводят к синергетическому эффекту. Примером могут служить многофункциональные мультиагентные системы, использующие интеллектуальных агентов и робототехнические устройства широкого профиля в качестве исполнительных элементов [1, 5].

В настоящее время в качестве окончательных исполнительных устройств робототехнических и мультиагентных систем в IoT-комплексах наиболее часто используются колёсные и гусеничные роверы, миниатюрные подводные лодки и беспилотные летательные аппараты (далее – БПЛА). Диапазон их применения очень широк – от научных исследований до производственного и бытового применения. Ярким примером первого аспекта является совместная работа колёсного ровера-марсохода и БПЛА (марсолёт Ingenuity) на Марсе. Второй пример – успешная посадка космического корабля «Starship» компании Илона Маска. Для исключения риска на осмотр места посадки были отправлены две роботизированные платформы, оснащенные видеокамерами, от инженерной компании Boston Dynamics, специализирующейся в робототехнике. Эти платформы действовали автономно на основе «разумного консенсуса», но в случае необходимости получали управляющие сигналы непосредственно из командного центра. Такая ситуация наглядно иллюстрирует потребность в устройствах, которые при необходимости берут на себя некоторые функции человека.

Примеры второго аспекта – робототехнические устройства различного назначения, в том числе и БПЛА, которые могут эффективно действовать для реализации оперативного мониторинга состояния нефтегазовых транспортных сетей, плавучих буровых и добывающих платформ, для исследования береговых линий и шельфов крупных рек и озер, морей и океанов [2]. При использовании БПЛА в таких случаях возникает существенная проблема группового управления беспилотниками – часто в децентрализованном (автономном) варианте управления [5].

В отсутствие оперативно-управленческого диспетчерского звена задачи управления действиями БПЛА сводятся к задачам частичной координатной синхронизации с помощью метода виртуального лидера. При большом числе агентов – а в ряде задач число агентов может достигать нескольких сотен – требование заданного (запланированного) поведения всех без исключения агентов оказывается трудновыполнимым. В таких случаях выделяется характерная точка во множестве состояний агентов (центр геометрического абриса, центр тяжести, лидер), а желательным поведением является заданное поведение центра при условии ограниченности отклонений от него состояний всех агентов. При этом достижение консенсуса или согласования характеристик достигается при условии, в котором каждый агент стремится уменьшить отклонение своей целевой переменной от соответствующих переменных своих соседей.

Вторая проблема – автономное поведение БПЛА в условиях динамической неопределенности, когда невозможно заранее определить стратегию обнаружения и мониторинга объекта. Это, например, поисково-спасательные работы в условиях непогоды и в ночное время, обследование крупных промышленных и инженерных сооружений, обследование пещер и т. д.

Гипотеза

Указанные выше проблемы можно решить с помощью автоматизированной системы автономной навигации и принятия решений для БПЛА, что позволит существенно повысить

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

вероятность самостоятельного обнаружения и мониторинга заданного объекта в условиях динамической неопределенности.

Методы

Для решения поставленной задачи предлагается разработать прототип программно-аппаратного комплекса автономной навигации 4-го уровня для беспилотного летательного аппарата на примере квадрокоптера и при этом сделать его достаточно универсальным, чтобы предложенное решение было многоцелевым. Настройку и испытательную прогонку системы можно осуществить с помощью специального программно-аппаратного симулятора [3].

Квадрокоптер – многомоторный или мультироторный летательный аппарат, несущие винты которого осуществляют вращение диагонально в противоположных направлениях. Маневренность обеспечивается за счет изменения скорости вращения на соответствующих винтах. Внутренняя схема квадрокоптера (рис. 1) довольно проста: полетный контроллер получает управляющий сигнал с пульта или генерирует его самостоятельно из записанной в памяти бортового компьютера полётной миссии, после чего отправляет сигналы широтно-импульсной модуляции (ШИМ-сигналы) на моторы. За питание системы отвечает бортовой аккумулятор.

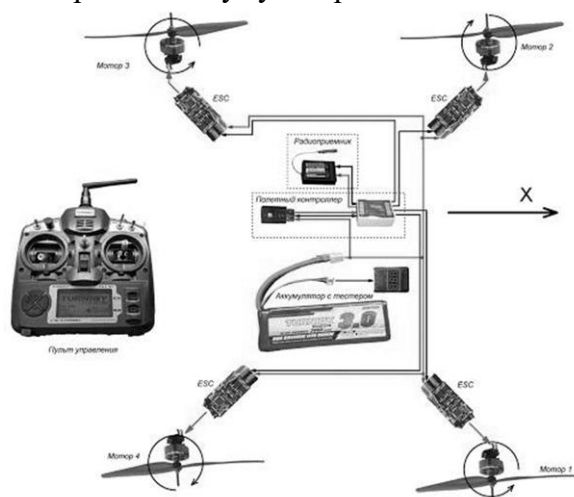


Рисунок 1. Структурная схема квадрокоптера

Для улучшения наглядности и повышения отказоустойчивости система управления квадрокоптером разбита на 3 части: наземная станция, бортовой компьютер, автопилот (рис. 2):

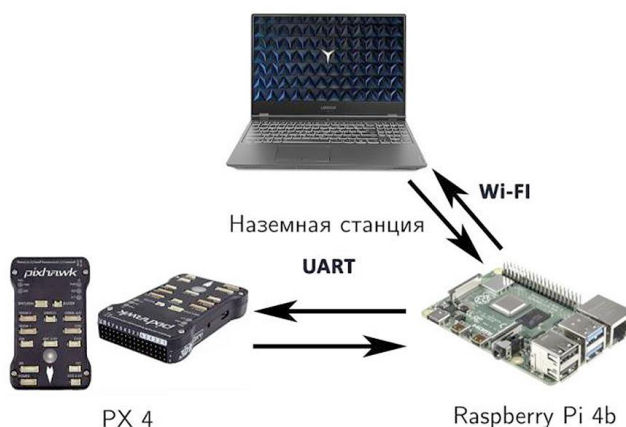


Рисунок 2. Состав управляющего программно-аппаратного комплекса

Исходя из условий поставленной задачи, соединение с квадрокоптером может быть нестабильным – следовательно, в качестве основного необходимо использовать соединение по протоколу User Datagram Protocol (UDP). Но поскольку полётная миссия БПЛА может включать несколько заданий, требующих передачи большого объёма данных, дополнительно было бы очень удобно получать крупные пакеты данных, например, изображение с бортовой камеры. Для чего наилучшим образом подходит протокол TCP (Transmission Control Protocol). В итоге оптимальным вариантом является пара соединений TCP/UDP, где «легковесная» информация передается посредством UDP-соединения.

Одним из ключевых элементов управляющей системы является видеочамера, так как от качества изображения напрямую зависит результат работы алгоритма автономной навигации. Особенно важны следующие параметры: количество кадров в секунду – минимум 60 кадров в секунду; разрешение изображения – минимум HD, желательнее FHD; выдержка – чем меньше, тем лучше; контрастность – чем больше, тем лучше. На момент проведения исследования единственные камеры, удовлетворяющие требованиям – это серия видеочамер Intel RealSense, конкретно была выбрана стереоскопическая видеочамера Intel RealSense D435 с двумя объективами.

Алгоритм автономной навигации строился с использованием принципов визуальной одометрии. Это метод оценки положения и ориентации робототехнического или иного устройства с помощью анализа последовательности изображений, полученных установленной на нём видеочамерой. Ниже перечислены основные части, на которые можно разбить алгоритм:

- получение входного изображения и его коррекция по яркости и четкости;
- обнаружение и отслеживание характерных признаков в разных кадрах;
- выявление выбивающихся значений векторов поля, оптического потока и их коррекция;
- оценка движения камеры (квадрокоптера) по скорректированному оптическому потоку.

Калибровка камеры осуществлялась по стандартному алгоритму – после этого можно работать с изображением, не опасаясь погрешности в результатах, вызванной искажениями поля изображения – такими как радиальная дисторсия или астигматизм.

Для оценки изменения положения видеочамеры в течение полёта квадрокоптера нужна точка отсчета, смещение относительно которой и будет искомой величиной её смещения по координатным осям. В таком случае логично зафиксировать место начала работы алгоритма одометрии при старте квадрокоптера как нулевую точку, однако она далеко не всегда попадает в область видимости камеры. Решить проблему помогают особые точки.

Особые точки – набор пикселей, окрестность которых уникальна и существенным образом выделяется на изображении. Ниже перечислены свойства особых точек, обуславливающие их применимость как ориентиров, относительно которых рассчитывается перемещение:

- *определенность* – особенность должна выделяться на фоне среди соседних точек;
- *устойчивость* – изменение яркости, контрастности и цветовой гаммы не должны влиять на место особой точки на объекте или сцене;
- *инвариантность* – особые точки должны обладать устойчивостью к повороту, изменению масштаба изображения и смене ракурса съемки;
- *стабильность* – зашумленность изображения, не превышающая определенный порог, не должна влиять на работу детектора;
- *интерпретируемость* – особые точки должны быть представлены в формате, пригодном для дальнейшей работы.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Существует достаточно много методов детекции особых точек. Мы выбрали для исследования три самых популярных, так как остальные уступают им по качеству или производительности – это алгоритмы SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Feature) и FAST (Features from accelerated segment test). Анализ возможностей, достоинств и недостатков каждого алгоритма позволил обоснованно выбрать алгоритм FAST с механизмом идентификатора (дескриптора) особой точки BRIEF, который делает эту точку уникальной относительно остальных особых точек.

После получения отсортированных пар особых точек в проекции на плоскость камеры, необходимо определить их трёхмерные координаты относительно точки центра тяжести квадрокоптера. Это делается с помощью карты глубин – массива чисел, сопоставляющего каждому пикселю расстояние от матрицы камеры до изображенной в этой точке части объекта. В основе стандартного алгоритма построения карты глубин по стереопаре лежит идея поиска для каждого пикселя левого изображения смещения соответствующего ему пикселя на соответствующий эпиполярной линии правого кадра. При условии известного расстояния между видеокамерами по смещению можно определить расстояние до частей объекта в каждой точке снимка. Таким образом, построив систему особых точек на основе стереоснимков, система управления квадрокоптером получит подробную карту расстояний до отождествляемого объекта. Эта карта будет динамично изменяться по мере перемещения квадрокоптера относительно объекта.

Следующий шаг алгоритма – локализация местоположения, ключевая задача системы. Задача оценки перемещения заключается в нахождении вектора перемещения \mathbf{t} и матрицы поворота \mathbf{R} . На рисунке 3 показано отображение точки в мировых координатах на проективную плоскость камеры.

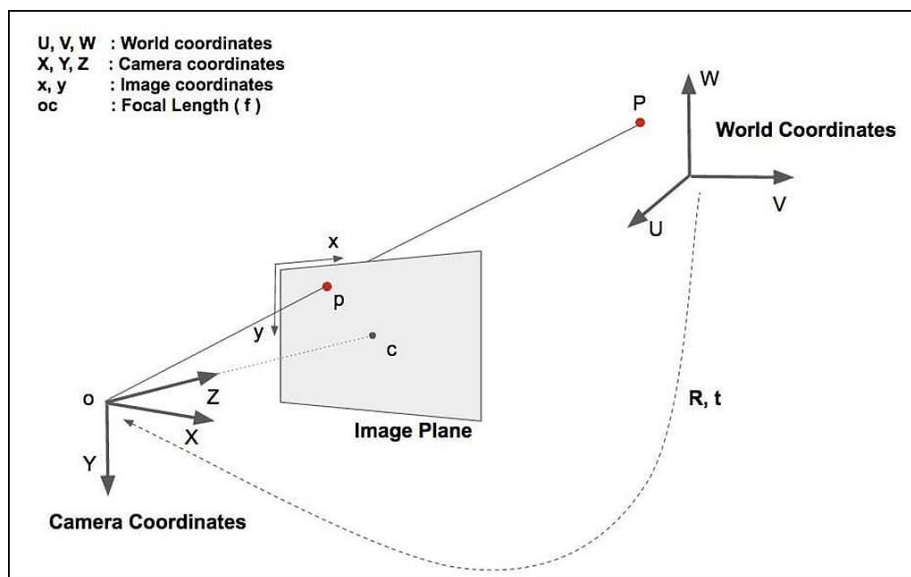


Рисунок 3. Связь мировой системы координат, координат камеры и плоскости изображения

Перевод из одной системы координат в другую производится по следующим формулам:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R * \begin{bmatrix} U \\ V \\ W \end{bmatrix} + t$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [R|t] * \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}$$

где X, Y, Z – координаты в системе отсчета камеры; U, V, W – представление в мировой системе координат; **R** – матрица вращения, представленная в виде кватерниона, **t** – вектор смещения.

После преобразования формула приводится к следующему виду:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} * \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}$$

где известны координаты X, Y, Z и U, V, W; r_{ij} – элементы матрицы вращения.

Задача нахождения **t** и **R**, решается с помощью алгоритма DLT (Direct Linear Transformation) – алгоритма, разрешающего систему линейных уравнений вида $x_k = Ay_k$ для $k = 1, \dots, N$, где x_k и y_k – вектора, A – матрица трансформации. С помощью итеративного метода Левенберга-Марквардта в результате получается матрица, отличающаяся от необходимой не более чем на заданную величину.

Следующая проблема в построении маршрута – выделение препятствий, без них можно было бы просто пролететь по прямой к целевой точке. Однако из постановки задачи следует, что система должна функционировать, в том числе и в объектах со сложной архитектурой (здания, мосты, естественные пещеры и т. д.). Решением проблемы является применение видоизмененного алгоритма Ли (так называемый волновой алгоритм) [6]. Его суть заключается в прохождении по графу в ширину и поиске минимального пути до искомой точки. Пример его работы показан на рисунке 4.

7	7		6	6	6	6	6	6	6	7	8
6	6		5	5	5	5	5	5	6	7	8
5	5	5	4	4	4	4	4	5	6	7	8
4	4	5	4	3	3	3	4			7	8
3					2	3	4	5	6	7	8
3	2	1	1	1	2	3	4	5			8
3	2	1	0	1	2	3	4	5			8
3	2	1	1	1	2	3	4	5	6	7	8

Рисунок 4. Пример работы алгоритма Ли на двухмерной сетке

После выполнения указанных выше шагов алгоритма в памяти бортового компьютера сформировано облако особых точек с координатами относительно точки взлета квадрокоптера, далее они используются для реконструкции препятствий следующим образом:

- трехмерное пространство разбивается на сетку кубов размером $n*n$ (n задается заранее в настройках алгоритма);
- куб считается пустым, если в нем не содержится ни одна обнаруженная особая точка;
- строится маршрут в соответствии с алгоритмом Ли;
- квадрокоптер совершает переход по прямой в центр следующего по маршруту квадрата.

Путем циклического повторения данного алгоритма достигается избегание препятствий в динамически меняющейся среде.

Результаты и обсуждение

Рассмотрим общий принцип работы разработанного программно-аппаратного комплекса, представленного выше на рисунке 2. На квадрокоптере установлен микрокомпьютер Raspberry Pi4B, который отвечает за необходимые навигационные расчеты, описанные выше. Питание бортового компьютера осуществляется от аккумулятора БПЛА.

Наличие интерфейса USB 3.0 и серьезно переработанный по сравнению с 3-й версией USB-контроллер позволяют подключить камеру, обеспечивающую передачу 60 кадров в секунду сразу по нескольким каналам. Соединение с автопилотом PX4 осуществляется через узел вычислительных устройств Universal Asynchronous Receiver-Transmitter (UART).

Соединение управляющей системы квадрокоптера с наземной станцией происходит по сети Wi-Fi – «легковесная», но важная информация относительно текущего статуса системы передается несколько раз по упомянутому выше протоколу UDP, что позволит в будущем перенести это соединение на радиоантенну. Объемная информация, такая как, например, изображения с видеокamеры, передается по отдельному TCP-соединению. После подачи питания в навигационную систему включаются автопилот и бортовой компьютер – последний после инициализации становится доступен для подключения со стороны бортовой станции. После настройки полётной миссии он может начинать ее непосредственное выполнение.

Для обеспечения передачи данных между квадрокоптером и наземной станцией использована клиент-серверная архитектура и универсальная система легковесных сериализуемых пакетов, аналогично передача данных реализована в протоколе mavlink [7]. Станция обеспечивает возможность производить первичную настройку БПЛА, отслеживать работу квадрокоптера в реальном времени, задавать координаты назначения.

Управляющая система в виде программного приложения реализована нами на языке C++ с использованием фреймворка QT на платформе Linux. За обработку изображения отвечает библиотека компьютерного зрения OpenCV, визуализация траектории полета дрона производится с использованием графической библиотеки OpenGL.

После инициализации системы запускается серверная часть для связи с наземной станцией. Когда соединение установлено и получена команда на старт полётной миссии, запускается выделенный в отдельную программу скрипт, осуществляющий связь с автопилотом посредством специализированной библиотеки DroneKit [8]. Соединение между основной программой и скриптом осуществляется посредством локального сетевого соединения. Архитектура программной части, решающая задачи одометрии и навигации, построена на основе архитектурного шаблона Sense-Compute-Control (рис. 5). Он предполагает разделение действий системы на три фазы – снятие показаний с датчиков, вычисление управляющего воздействия, посылка управляющего сигнала на исполнительные элементы (актуаторы).

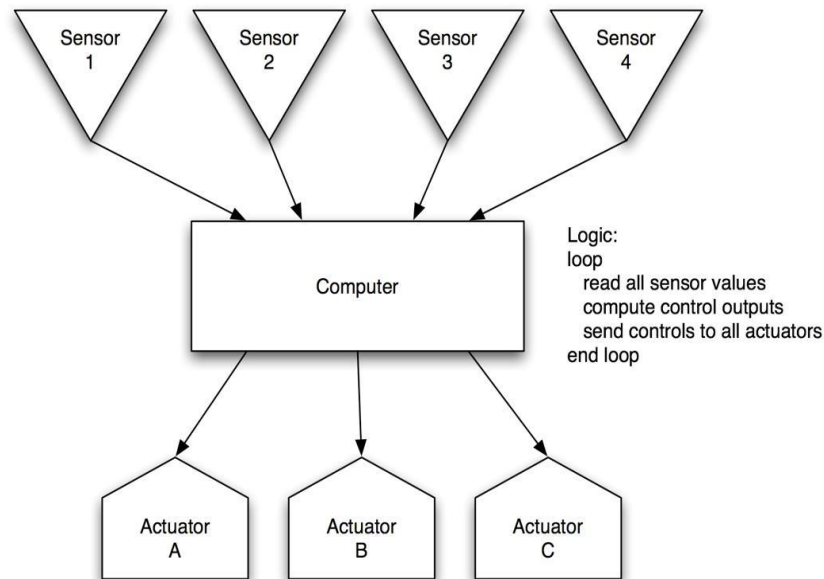


Рисунок 5. Схема работы системы, основанной на шаблоне Sense-Compute-Control

Вычислительный модуль, работая асинхронно, получает на вход изображения, данные с лидара, гироскопа и акселерометра. После обработки данных модуль формирует и отправляет управляющий сигнал в автопилот, который преобразует данные в ШИМ-сигналы и подаёт их на двигатели квадрокоптера.

Подключение к видеокамере осуществляется через поставляемый компанией инструментарий Intel SDK. Несмотря на наличие уже готового решения по калибровке камеры в используемой в проекте библиотеке компьютерного зрения OpenCV, испытательные тесты показали, что предлагаемый API видеокамеры метод позволяет получить более качественную картинку и разгрузить процессор бортового компьютера за счет использования встроенного в камеру микропроцессора. Также для улучшения работы вычислительных алгоритмов у изображения можно увеличивать контрастность – это позволяет лучше детектировать объекты и выделять особые точки.

Поиск особых точек, по которым строится объемная картина расстояний до объекта, является самым ресурсоемким местом в работе разработанной программы, поэтому в рамках оптимизации на него было обращено особое внимание. Как было указано выше, оптимальным методом детектирования был признан метод ORB, объединяющий алгоритмы FAST и BRIEF в более эффективный гибридный алгоритм.

В ходе тестовых испытаний было выявлено, что при получении последовательных кадров необязательно каждый раз искать особые точки заново, часть из них можно получить с помощью трекинга – отслеживания точки в окрестности того пикселя, в котором точка была замечена на предыдущем кадре. Это позволяет найти и сразу сопоставить часть пикселей, что уменьшает сложность и время перебора при стандартном сопоставлении кадров.

Заметим, что для триангуляции в идеальных условиях хватает трёх наблюдаемых точек. Разумеется, для реальных условий эта оценка неприменима, однако с учетом возможного наличия в поле зрения камеры движущихся объектов, которые наверняка внесут помехи в облако точек, и того, что особые точки впоследствии применяются для реконструкции карты местности при построении маршрута, может показаться, что чем больше особых точек, тем лучше. Однако следует учитывать, что в основе метода сличения особых точек на стереоснимках лежит методика использования брутфорса, сложность которого растет по экспоненте. По результатам тестов было определено, что для текущей конфигурации системы оптимальное число особых точек равняется 500 и следует настроить систему так, чтобы она стремилась к этому количеству.

Карту глубин, о которой говорилось выше, можно получить по 2-м кадрам, сделанным левым и правым объективами видеокамеры при известном расстоянии между ними, однако это дополнительная нагрузка на бортовой компьютер – поэтому было решено использовать встроенный в камеру лазерный дальномер (лидар) и готовую карту глубин, которую можно получить из API вместе с новым изображением. Даже с учетом того, что созданная лидаром карта глубин имеет разрешение 640*480 пикселей, после её масштабирования до размеров цветного изображения, погрешность при вычислении расстояния в 3 метра составляет всего 1 сантиметр против 15 сантиметров у карты глубин, полученной на основе стереоизображения. А это, в свою очередь, способствует более успешному выполнению полётной миссии.

Заключение

Разработанная система автономной навигации для управления квадрокоптером (рис. 6) в условиях динамической неопределенности, связанной с невозможностью заранее полностью спланировать полёт и построить исчерпывающий маршрут к объектам поиска, была протестирована и настроена на симуляторе физического полёта квадрокоптера [3].

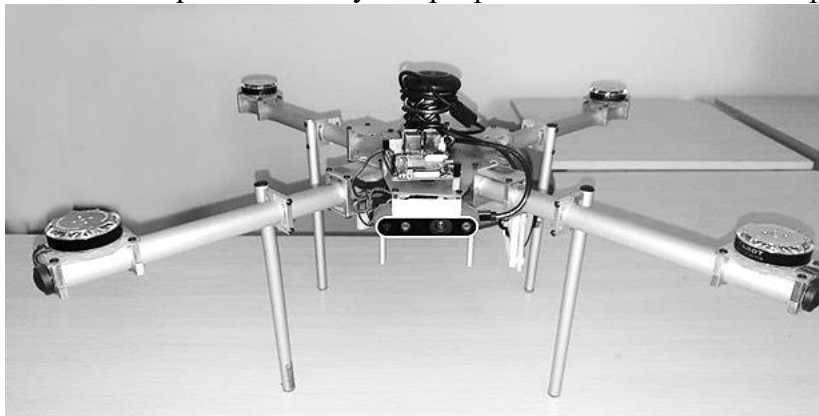


Рисунок 6. Прототип квадрокоптера с установленной на него системой автономной навигации

Проведённые тестовые и натурные испытания подтвердили высокую степень готовности разработанной системы автономной навигации 4-го класса автономности к практическому использованию. Такую систему можно применить не только на квадрокоптерах, но и на других видах беспилотных летательных аппаратов.

Разработанная система автономной навигации может быть использована не только в случае управления беспилотным летательным аппаратом (БПЛА), но и для любого робототехнического устройства, выполняющего разнообразные задачи в трёх средах в условиях динамической неопределённости, когда невозможно заранее наметить объекты для поиска и мониторинга и точно рассчитать маршрут.

Список литературы

1. Калитеевский В.Н. Метод коммуникации в децентрализованной сети автономной группы мобильных роботов. 2016. – URL: https://www.math.spbu.ru/user/gran/soi11_1/kalit.pdf (дата обращения: 12.07.2021).
2. Амелина Н.О., Амелин К.С., Граничин О.Н., Кияев В.И. Развитие нефтегазовых комплексов и сетей: мониторинг и мультиагентное управление // В сб. материалов V научно-практической конференции "Суперкомпьютерные технологии в нефтегазовой отрасли. Математические методы, программное и аппаратное обеспечение", Москва, февраль 2015. – С. 17- 21.
3. Кияев В. И., Дятлов К. А. Использование IoT-технологий для мониторинговых систем в хозяйственной деятельности // Научно-практический журнал «Гипотеза», №2 (11). – СПб.: Изд-во СПбГЭУ. 2020. С. 17-25.
4. IoT platforms: enabling the Internet of Things. – IHS TECHNOLOGY, March 2016. URL: <https://ihs.com> (дата обращения: 12.07.2021).

5. Erofeeva V., Granichin O., Kiyayev V. Multi-agent based adaptive swarm robotics control in dynamically changing and noisy environments // Proceedings of the International Conference. «Quantum Informatics and Applications. SPb, 21-th of December, 2015». 2016. – p. 808-813.
6. Hoel Jeffrey H. Some Variations of Lee's Algorithm // IEEE Transactions on Computers. – 1976. – Vol. C-25, no. 1. – p. 19-24.
7. Mavlink. Introduction // Mavlink. – 2021. – ERL: <https://mavlink.io/en/> (дата обращения: 08.07.2021)
8. Robotics 3D. Welcome to DroneKit-Python documentation // 3D Robotics. – 2021. – URL: <https://dronekit.netlify.app/> (дата обращения: 09.07.2021).