

РЕАЛИЗАЦИЯ АЛГОРИТМА АВТОНОМНОЙ НАВИГАЦИИ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

Викторов Александр Сергеевич

аспирант, Костромской государственной университет, Россия, г. Кострома

Autonomous navigation algorithm implementation for unmanned aerial vehicle

Alexander Victorov

postgraduate of Kostroma State University, Russia, Kostroma

Аннотация. В статье предлагается к рассмотрению алгоритм, предназначенный для реализации системы планирования траектории полета и автономной навигации беспилотного летательного аппарата. Для реализации подобной навигационной системы способной работать автономно, анализируя поток данных, генерируемый сенсорной системой аппарата, используются программные компоненты фреймворка для программирования роботов ROS.

Abstract. In this paper is considered a algorithm, designed for flight trajectory planning and autonomous navigation of unmanned aerial vehicle. For such system realization capable work autonomously, analyzing input data stream generated by vehicle sensor system, is used a programme components of ROS framework which designed for robot software development.

Ключевые слова: беспилотный летательный аппарат БПЛА; планирование траектории; навигация; уклонение от столкновения.

Keywords: unmanned aerial vehicle UAV; flight trajectory planning; navigation; obstacle avoidance.

В данное время происходит бурное развитие разнообразных специализированных геоинформационных систем, которое связано с ростом востребованности актуальных геопространственных данных. Геопространственные данные широко используются для прогнозирования и планирования производственно-хозяйственной деятельности. Например, при проектировании различных сооружений и эксплуатации построенных объектов, что требует разработки проектной документации различного уровня: проектов обоснования инвестиций, инженерных проектов, рабочей документации и сметно-финансовых расчетов. Грамотно оформленная документация обеспечивает оптимизацию временных и материальных затрат, предотвращает возникновение ущерба окружающей среде в ходе строительства и эксплуатации объекта. Подготовка проектной документации связана с выполнением инженерных изысканий и проектированием сооружений. Качество и скорость выполнения данных работ напрямую зависит от качества и актуальности геопространственных данных, представляемых геоинформационными сервисами. Проведенные исследования показали, что

на сегодняшний день существуют разнообразные специализированные геоинформационные системы, которые позволяют производить анализ и различные операции с геопространственными данными, в частности производить построение цифровых моделей местности. Но существующие системы обладают существенным недостатком, заключающимся в том, что эти системы не позволяют оперативно и в автоматическом режиме, без участия оператора, производить обновление цифровых моделей местности, что важно для поддержания пространственных баз данных в актуальном состоянии. Реализация автоматического процесса обновления моделей местности является одной из первостепенных задач при проектировании геоинформационных систем. Наиболее перспективное решение рассматриваемой проблемы использование данных дистанционного зондирования земли ДДЗ, полученных с беспилотного летательного аппарата БПЛА. Для сбора данных, соответствующих определенному участку местности, БПЛА производит облет местности по заданной траектории, например, имеющей форму змейки.

Во время осуществления полета БПЛА его траектория контролируется удаленным сервером, на котором установлен фреймворк ROS (robot operating system) и программное обеспечение для анализа данных телеметрии. Фреймворк ROS [3; 5] предоставляет разнообразный функционал для распределенной обработки и анализа данных, поступающих от сенсорной системы роботов, механизмы планирования и принятия решений. ROS также предоставляет обширную коллекцию инструментов, библиотек и структурно состоит из двух компонентов: ядра и пользовательского стека. Ядро ROS представлено минимальным необходимым для работы набором инструментов и библиотек. Пользовательский стек представлен набором программных пакетов, развиваемых сообществом независимых разработчиков, которые реализуют различный функционал необходимый для управления роботом: драйвера для работы с робототехническим оборудованием, модули обработки данных сенсорной системы, модули планирования и принятия решений и т.д. Ядро ROS разработано на основе концепции независимости программного обеспечения ядра от структуры конечной управляемой робототехнической системы. Основная функция ядра ROS заключается в обеспечении обмена сообщениями между вычислительными узлами ROS, которые образуют вычислительный граф, являющийся по сути одноранговой сетью вычислительных узлов, производящих совместную обработку и обмен данными. Узлы вычислительной сети ROS могут быть запущены на разных ЭВМ, при этом связь между узлами сети осуществляется через сети передачи данных с использованием протокола передачи данных TCP/IP. Каждый вычислительный узел представлен исполняемым файлом пакета ROS, который использует клиентские библиотеки ROS для связи с другими узлами. Пакет ROS включает в себя файл манифест, который содержит описание зависимостей пакета и также может содержать исходные коды, библиотеки, конфигурационные файлы и другие файлы, обеспечивающие предусмотренный для узла функционал. При этом ROS предоставляет необходимые инструменты для эффективного управления пакетами. Существуют различные клиентские библиотеки ROS, благодаря чему узлы ROS могут быть реализованы на различных языках программирования: на Python с использованием клиентской библиотеки Rospy; на C++ с использованием библиотеки Roscpp; на Java с использованием библиотеки Rosjava. При запуске новый вычислительный узел регистрируется на мастер-узле, благодаря чему уже зарегистрированные узлы могут узнать об изменении в структуре вычислительной сети и начать обмен сообщениями с вновь зарегистрированным узлом. Мастер-узел также информирует все остальные узлы об изменениях в структуре вычислительной сети, благодаря чему возможно динамическое изменение ее конфигурации и функционала в режиме реального времени. Обмен сообщениями между узлами осуществляется при помощи следующих механизмов: механизма топиков, основанного на парадигме издатель/подписчик; механизма сервисов, основанного на технологии удаленного вызова процедур RPC; сервера параметров, основанного на технологии XMLRPC и предназначенного для хранения конфигурационных параметров узлов и обмена параметрами между узлами. Алгоритмы планирования траектории условно можно разделить на алгоритмы глобального планирования траектории GPP (global path planning) [1] и алгоритмы локального планирования траектории LPP (local path planning) [2]. Алгоритмы GPP применяются в том случае, если пространство в котором перемещается робот детерминировано и полностью наблюдаемо. Алгоритмы LPP применяются в том случае, если пространство не детерминировано полностью. Для функционирования алгоритма глобального планирования траектории необходимо наличие следующих исходных данных: карты пространства, в котором перемещается робот; начальное пространственное положение робота; целевое пространственное положение робота; модель

геометрии робота. После осуществления процедуры планирования траектории для контроля перемещения робота по запланированной траектории необходимо производит оценку и коррекцию пространственного положения робота, для реализации чего используется специальный модуль определения положения. Рассматриваемое программное обеспечение и исходные данные достаточны для навигации робота, если карта пространства достаточно полно и точно отображает геометрию пространства, в котором будет перемещаться робот, пространство статично и модуль определения положения робота производит оценку пространственного положения робота без погрешностей. Так как на практике выполнение всех описанных выше условий невозможно из-за того, что в реальности пространство не статично, необходима реализация алгоритма локальной навигации, который позволил бы обнаруживать препятствия, избегать столкновения с ними и обновлять карту пространства локально на основании данных сенсорной системы. То есть на практике алгоритм глобальной навигации используется только на начальном этапе планирования траектории, при этом результатом его работы является траектория, сгенерированная перед началом полета и представленная набором путевых точек (точки пространства, используемые для навигации). Алгоритм планирования траектории осуществляет ее построение, производя поиск такой траектории, которая бы минимизировала целевую функцию, которая учитывает множество критериев качества траектории таких, как: длина траектории; затраченное количество топлива или энергии аккумулятора; безопасность траектории, например, определяемая таким критерием, как минимальное расстояние робота от препятствий при прохождении траектории; отклонение пространственного положения робота от заданных контрольных пространственных точек при движении по сгенерированной траектории. В качестве конкретной реализации алгоритма планирования траектории используется алгоритм RRT* библиотеки OMPL. Так как геометрия пространства, в котором перемещается робот может не соответствовать ранее построенной трехмерной карте пространства или данная карта отсутствует, то необходимо для осуществления перемещения между заданными путевыми точками реализовывать алгоритм локального планирования. Алгоритм локального планирования осуществляет планирование траектории [4], анализируя локальную карту местности, построенную на основании данных полученных с сенсорной системы робота при помощи алгоритма SLAM (simultaneous localization and mapping) [6] и других вспомогательных алгоритмов. При планировании траектории необходимо также наличие модели робота. Для данной цели ROS поддерживает следующие форматы для описания робота URDF (unified robot description format), который представляет из себя xml список, содержащий описание робота и его составных узлов, и SRDF (semantic robot description format), который представляет из себя xml список, содержащий описание взаимосвязей между узлами робота. Так как в реальности пространство, в котором перемещается робот не статично, то для реализации системы автономного управления движением робота, схема которой представлена на рисунке 1, также требуется реализация алгоритма обнаружения препятствий и алгоритма уклонения от представляющих опасность препятствий и движущихся объектов (obstacle avoidance).

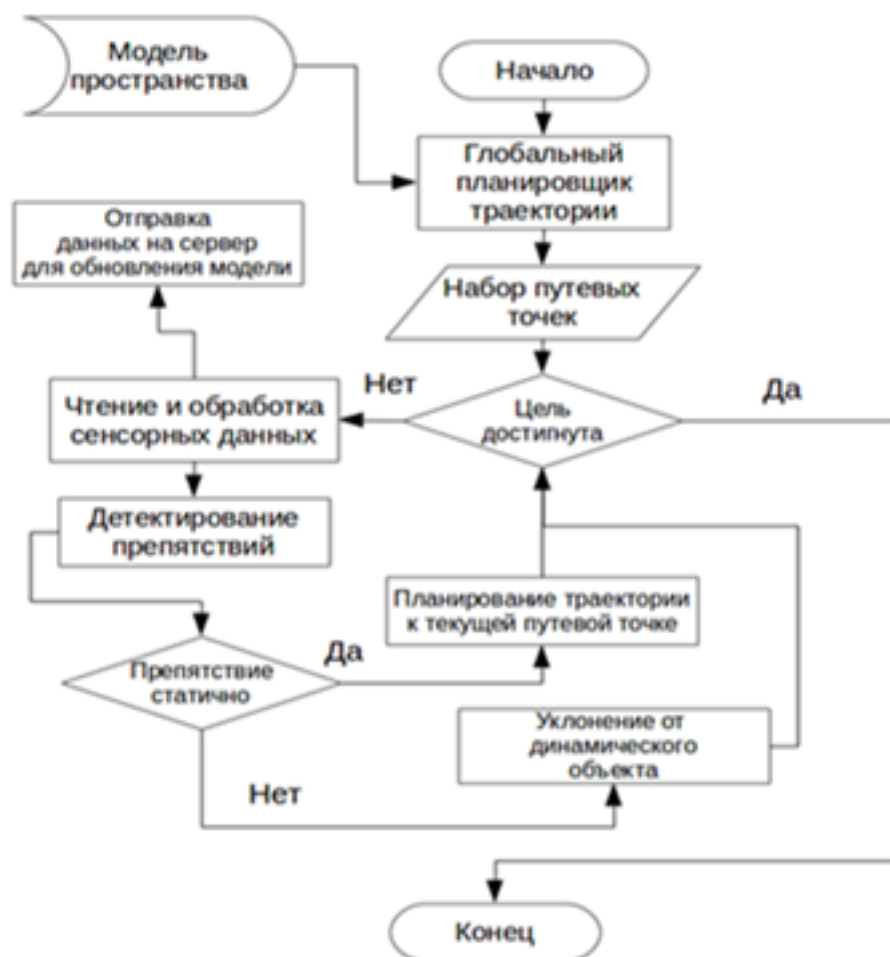


Рисунок 1. Схема алгоритма автономной навигации

Так как данные измерений сенсорной системы зашумлены в той или иной мере, то для обнаружения препятствий и навигации используется представление локальной карты местности в виде карты препятствий (occuancy grid mapping). Так как рабочий диапазон сенсоров, используемых для обнаружения препятствий ограничен, то алгоритм уклонения от столкновения с высокодинамичными объектами или от своевременно не обнаруженных из-за плохой видимости препятствий должен обладать достаточно высоким быстродействием так, как время для принятия решения от момента обнаружения препятствия до столкновения с ним, как правило ограничено.

Список литературы:

1. Alajlan M.A., Koubâa A.R. Writing Global Path Planners Plugins in ROS: A Tutorial // Robot Operating System (ROS). Switzerland: Springer International Publishing, 2016. С. 73–97.
2. Cobano J.N., Ollero A.S., Rodriguez L.P. Efficient local path planning for UAVs in unknown environments // XV workshop of physical agents: book of proceedings, WAF 2014. 2014. №1. С. 125–134.
3. Fernndez E.P., Martinez A.R. Learning ROS for Robotics Programming. Packt Publishing изд. BIRMINGHAM: Packt Publishing, 2013.
4. Honegger D.W., Meier L.S., Pollefeys M.A. PX4: A Node-Based Multithreaded Open Source Robotics Framework for Deeply Embedded Platforms // Robotics and Automation (ICRA), 2015 IEEE International Conference on. 2015. №1. С. 6235–6240.

5. Klingauf U.R., Kohlbrecher S.A., Meyer J.P., Sendobry A.S., Stryk O.V. Comprehensive simulation of quadrotor UAVs using ROS and gazebo // SIMPAR'12 Proceedings of the Third international conference on Simulation, Modeling, and Programming for Autonomous Robots. Heidelberg: Springer-Verlag Berlin, Heidelberg, 2012. C. 400–411.
6. Shvedenko V.N., Victorov A.S. Improved visual odometry method for simultaneous unmanned aerial vehicle navigation and earth surface mapping. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2017, vol. 17, no. 3, pp. 475–482 (in Russian). doi: 10.17586/2226-1494-2017-17-3-475-482.