

УДК 004.4, 004.89, 004.932

РАСПРЕДЕЛЕННАЯ СИСТЕМА УПРАВЛЕНИЯ КОАЛИЦИЯМИ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ, ФУНКЦИОНИРУЮЩИХ В УСЛОВИЯХ ОТСУТСТВИЯ СИГНАЛОВ ГЛОБАЛЬНЫХ НАВИГАЦИОННЫХ СИСТЕМ¹

К.С. Яковлев (*yakovlev@isa.ru*)

Институт системного анализа Российской академии
наук, Москва

В.В. Хитков (*vskhitkov@gmail.com*)

М.И. Логинов (*lunarstrainut@gmail.com*)

А.В. Петров (*gmdidro@gmail.com*)

Рыбинский государственный авиационный технический
университет имени П.А. Соловьева, Рыбинск

В работе описывается распределенная система управления группой беспилотных летательных аппаратов в среде без доступа к GPS/ГЛОНАСС на примере коллективной навигации, локализации маркеров и построения карты маркеров в помещении группой квадрокоптеров AR.Drone 2.0. Разработанная интеллектуальная система управления основана на модели 4D/RCS и реализована и применением платформы ROS. Задача картирования решается с помощью реализации алгоритма EKF-SLAM.

Введение

Современные интеллектуальные системы характеризуются повышением степени автономности и распределенности. С одной стороны, автономность повышается с точки зрения независимости от оператора, а с другой стороны, в случае многоагентных систем, речь идет об автономности как степени автоматизации агентов, выполняющих единую задачу. Распределенность современных систем понимается и как обеспечение коллективного взаимодействия между несколькими

¹ Работа выполнена при финансовой поддержке ОНИТ РАН, проект «Развитие методов интеллектуального анализа данных и управления робототехническими системами с применением бесконтактных человеко-машинных интерфейсов».

интеллектуальными агентами (распределение знаний и информации между агентами), и с точки зрения технологий распределенного программного обеспечения.

В статье представлена распределенная система управления и навигации беспилотных летательных аппаратов (БЛА), в частности, квадрокоптеров, функционирующих в условиях отсутствия сигналов глобальных навигационных систем (GPS/ГЛОНАСС). Реализация представленной системы демонстрируется на примере задачи построения карты, решаемой группой БЛА. N квадрокоптеров (использовались AR.Drone 2.0) функционируют в закрытом помещении и управляются удаленно (по Wi-Fi) программной системой управления с наземной станции (ноутбука с ОС Linux). В помещении расположены несколько маркеров – кубиков с нанесенными на стороны QR-кодами. Каждый БЛА ищет маркеры и вычисляет их положения относительно собственной стартовой позиции (строит локальную карту), при этом параллельно этому локальные карты от различных БЛА интегрируются в глобальную и уточняются с течением времени.

В статье предлагается концептуальная структура архитектуры системы управления, описываются реализованные алгоритмы распознавания маркеров, вычисления координат маркеров и построения карты. Также рассматривается реализация системы управления на базе платформы ROS (Robotic Operation System). В заключение приводятся результаты эксперимента и дальнейшие направления работы.

1. Многоуровневая архитектура распределённой системы управления беспилотными летательными аппаратами

К настоящему моменту известно множество подходов к построению систем управления сложными техническими объектами, в том числе беспилотной техникой и, в частности, беспилотными летательными аппаратами [Осипов и др., 2009], [Яковлев и др., 2013]. Одним из наиболее распространенных является подход, основанный на функциональной декомпозиции, когда система управления представляет собой совокупность модулей, каждый из которых предназначен для решения определенного типа задач. Обычно выделяют следующие задачи и соответствующие им модули системы управления: построение модели (картины) мира, концептуальное планирование (планирование миссии), планирование траектории и навигация, обеспечение взаимодействия, парирование нестандартных ситуаций, управление коммуникацией, управление датчиками и актуаторами БЛА [Яковлев и др., 2013]. Пример системы управления беспилотными летательными аппаратами, содержащей модули для решения вышеперечисленных задач и состоящей из одного уровня приведен, например, в [Jameson et al., 2005].

Стоит заметить, что наиболее распространено разбиение функциональности системы управления не только по отдельным модулям, но и по отдельным уровням. Обычно, когда речь идёт о создании интеллектуальных беспилотных транспортных средств, выделяют три уровня управления: стратегический (верхний), тактический (средний) и реактивный (нижний). К примерам таких архитектур можно отнести 3T [Bonasso et al., 1995], ATLANTIS [Gat, 1992] и др. Основным отличием модулей расположенных на разных уровнях является степень обработки входящей информации и время отклика [Яковлев и др., 2013]. Фактически уровень в современных системах управления беспилотной техникой есть абстракция «интеллектуальности»: чем выше уровень, тем более «интеллектуальные» задачи на нём решаются.

Система управления для решения поставленной задачи имеет ряд особенностей, оказывающих непосредственное влияние на её архитектуру. Во-первых, отсутствует необходимость реализовывать стратегический уровень управления, т.к. рассматривается только одна высокоуровневая задача – «построение карты». Во-вторых, для решения задачи использовались БПЛА Parrot AR.Drone, каждый из которых оснащен бортовым вычислителем с предустановленным программным обеспечением, уже реализующим функционал модулей управления реактивного уровня. Методы и алгоритмы реактивного управления реализованы компанией Parrot и описаны в [Bristeau et al., 2011]. Таким образом, отсутствует необходимость реализации реактивного управления: имеющиеся ПО полностью автоматизируют управление актуаторами (двигателями) БЛА для обеспечения выполнения следующих элементарных манёвров: взлёт, посадка, поворот по/против часовой стрелки, полет вперёд/назад и вправо/влево. Команды на осуществление этих манёвров передаются на бортовой вычислитель по wi-fi каналу от управляющей станции (ноутбука). Таким образом, основное внимание было уделено разработке модулей тактического уровня управления, ответственных за решения таких задач как распознавание маркеров по видеопотоку, построение локальных карт, интеграции локальных карт в глобальную карту (модель) местности и ряда других.

При реализации тактического уровня управления за основу были взяты формализм и модель архитектуры системы управления беспилотными транспортными средствами и их коалициями 4D/RCS [Albus et al., 2002], разработанные группой профессора Албуса в национальном институте стандартов США (NIST). Особенность этой модели – абстрактно-функциональная декомпозиция, а именно: согласно 4D/RCS каждый уровень СУ состоит из совокупности узлов, каждый из которых в свою очередь есть совокупность четырёх процессов:

- обработка сенсорной информации (ОСИ);

- построение модели мира (ПММ);
- оценивание свойств (ОС);
- выбор управляющих воздействий (генерация поведения) (ГП).

Состав компонент узлов 4D/RCS не меняется от уровня к уровню, но меняется семантика протекающих в узлах процессов – например, на более высоком уровне управления под генерацией поведения понимается планирование поведения (действий), а на более низком – планирование траектории (перемещений).

В рассматриваемом случае 4D/RCS узел – абстракция отдельного объекта управления (квадрокоптера Parrot AR.Drone), а соответствующие процессы имеют следующую интерпретацию:

ОСИ – получение (по wi-fi каналу) и транскодирование видеопотоков от двух видеокамер, установленных на БЛА; получение (так же по wi-fi) данных, касающихся состояния и ориентации БЛА (АТ-пакетов, содержащих результаты измерений датчиков БЛА); интеграция данных.

ОС – определение ориентации БЛА в глобальной системе координат; распознавание и идентификация маркеров; определение расстояния до маркеров (как в локальной, так и в глобальной системе координат).

ПММ – построение локальной карты, интеграция множества локальных карт (карт построенных различными БЛА) в единую, глобальную модель (карту) местности.

ГП – выбор очередного промежуточного пункта маршрута полёта с помощью системы правил.

Как было сказано выше, каждый узел 4D/RCS отвечает за управление одним БЛА. Функциональные процессы на программном уровне реализованы как модули Robotic operating system [Quigley et al., 2009] (подробное описание см. ниже), исполняемые параллельно в операционной системе Linux. Система управления обеспечивает механизмы поддержки глобальной базы данных, т.е. локальные карты, построенные отдельными БЛА, хранятся на управляющей станции и обрабатываются с целью их дальнейшей интеграции в единую глобальную карту местности (доступную всем БЛА).

Ниже будут более подробно описаны подходы, методы и алгоритмы, используемые для определения ориентации БЛА в пространстве, идентификации и распознавании маркеров, картирования. Также будут описаны существенные детали программной реализации системы управления.

2. Алгоритм построения карты расположения маркеров

В ходе работы реализован алгоритм построения карты расположения маркеров который может применяться в практических задачах автономной навигации при возможности установки визуальных маркеров. Карта

строится одновременно несколькими БЛА, каждый из которых наблюдает некоторое подмножество маркеров, расположенных в общей среде функционирования, с последующими их объединением в общее множество (глобальную карту маркеров). Это происходит следующим образом.

Каждый БЛА (имеется в виду узел программы управления конкретным БЛА на наземной станции – см. раздел 3) во время полета использует видеопоток и информацию инерциальных датчиков для одновременного вычисления своего положения и построения карты с помощью EKF-SLAM-алгоритма одновременной навигации и построения карты с использованием расширенного фильтра Калмана). Каждый БЛА независимо локализует видимые маркеры на кадрах видеопотока, идентифицирует их (они полагаются уникальными) и вычисляет их относительные координаты по 6 степеням свободы (перемещение и поворот) с помощью компьютерного зрения. Для каждого найденного маркера проверяется его наличие на глобальной карте, копия которой доступна узлу управления БЛА в любой момент. При этом возможны следующие ситуации.

- 1) Маркер не присутствует на глобальной карте, а БЛА не знает своего расположения в глобальной системе координат (т.е. он еще не наблюдал тех же маркеров, что и другие БЛА). В этом случае он наносится на индивидуальную карту БЛА с координатами, усредненными за несколько наблюдений в разных кадрах.
- 2) Маркер присутствует на глобальной карте, но БЛА еще не знает своих координат в глобальной системе. В этом случае маркер используется как связующее звено между локальной системой координат данного БЛА и глобальной системой координат. Осуществляется единовременная трансляция локальной системы координат БЛА к глобальной с пересчетом координат всех объектов на индивидуальной карте БЛА, которая в дальнейшем совпадает с глобальной.
- 3) Маркер не присутствует на глобальной карте, а БЛА знает свое расположение в глобальной системе координат. В этом случае его карта совпадает с глобальной картой, и маркер вместе с его координатами наносится, таким образом, и на индивидуальную, и на общую карту. Координаты фильтруются за несколько наблюдений.
- 4) Маркер присутствует на глобальной карте, и БЛА знает свое местоположение. Этот случай используется для уточнения координат как БЛА, так и устранения «перекоса» карты, с помощью «блочного уравнивания» (bundle adjustment) по набору ключевых положений, подобно [Klein et al., 2007] (этот подход

применим ввиду полагаемого небольшого количества маркерокубиков).

В качестве системы маркеров и инструмента их локализации, идентификации и определения положения по 6 степеням свободы в работе используется библиотека ArUco [Munoz-Salinas et al., 2014], предназначенная для распознавания двумерных штрих-кодов и характеризующаяся надежностью распознавания за счет применения кода Хэмминга. При известных внутренних калибровочных параметрах камеры становится возможным построение карты в метрической системе координат.

Для оценки трехмерного положения БЛА применяется расширенный фильтр Калмана с вектором состояния $(x_t, y_t, z_t, \alpha_t, \beta_t, \gamma_t)^T$, элементы которого соответственно означают 3 пространственных координаты и 3 угла Эйлера в глобальной или индивидуальной системе координат. Предсказание положения на основе показаний инерциальных датчиков выполняется аналогично описанному в [Engel et al., 2012]. Для оценки координат по видеопотоку используются рассчитанные абсолютные координаты всех локализованных на кадре и уже присутствующих на карте маркеров в конкретный момент времени.

3. Базирующаяся на ROS программная реализация системы управления

Программная реализация описываемой системы основана на платформе ROS и архитектуре 4D/RCS. На рисунке 2 показана структура программы.

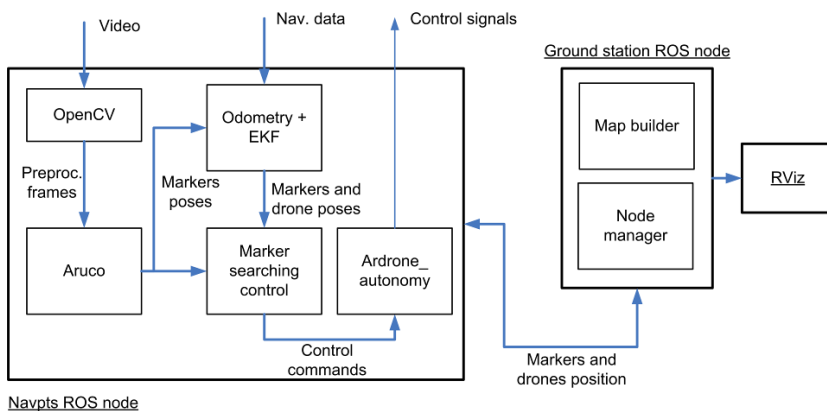


Рисунок 1 - Структура программной реализации системы распределенного управления, базирующейся на ROS

Пакеты, узлы ROS, а также внешние библиотеки показаны на рисунке 2. Существует два основных узла ROS: узел “Navpts” и “Ground station”. Некоторое количество узлов “Navpts” исполняются параллельно, при этом обеспечивая контроль за одним AR.Drone каждый. Узел “Ground Station”, запущенный в единственном экземпляре, реализует построение карты и алгоритм ее совместного использования. Пользуясь архитектурой 4D/RCS при описании узла “Navpts” можно отметить, что библиотека OpenCV выполняет обработку сенсорной информации, пакеты Aguco и EKF - оценку значений, пакет поиска маркера вырабатывает требуемое поведение. Процесс построения окружающего мира 4D/RCS распределен между представлениями карты узлами “Navpts” и “Ground Station”.

Узел “Ground Station” представляет собой модуль построения карты, реализуемый описанные алгоритмы, и менеджер узлов, запускающий и связывающий необходимые темы (топики) для узлов “Navpts”.

Для визуализации карты используется пакет ROS - RViz.

Было выполнено тестирование разработанной реализации системы управления в лабораторных условиях. На рисунке 3, справа, показан результат в форме совместной карты.

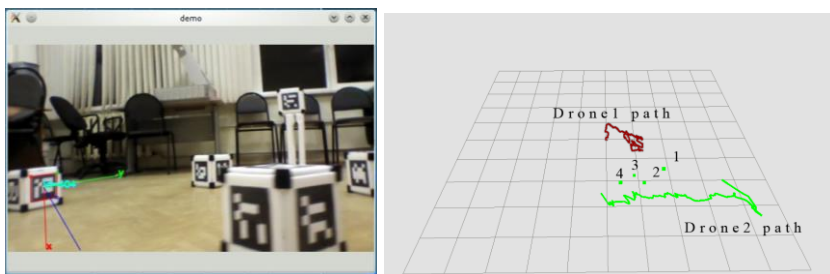


Рисунок 3 - Обнаружение маркера на изображении (слева) и совместная карта в RViz (справа)

На рисунке 3 (слева) показаны экспериментальная установка и обнаружение маркера. На рисунке 3 (справа) показаны совместная карта с траекториями БПЛА и позициями общих маркеров.

Заключение

В заключение отметим, что предложенная в работе реализация алгоритма построения карты не является оптимальным решением для задачи подобного рода по сравнению с аналогами и используется в качестве иллюстрации выбранной архитектуры и программной реализации системы распределенного управления и навигации БЛА в среде без

доступа к данным глобальных систем позиционирования (GPS, ГЛОНАСС). Планируется расширить имеющуюся систему в следующем направлении: интеграция с более эффективным SLAM алгоритмом, который может быть применен в среде с неизвестными ландшафтными метками; модификация процесса генерации поведения путем введения совместного планирования для того, чтобы обеспечить решение в ситуации, в которой один из БЛА не в состоянии обнаружить ни один маркер; повышение вычислительной эффективности системы.

Список литературы

- [Осипов и др., 2009] Осипов Г.С., Тихомиров И.А., Хачумов В.М., Яковлев К.С. Интеллектуальное управление транспортными средствами: стандарты, проекты, реализации // Авиакосмическое приборостроение, № 6, 2009. С. 34-43.
- [Jameson et al., 2005] Jameson S., Franke J., Szczerba R., Stockdale S. Collaborative Autonomy for Manned/Unmanned Teams. AHS International Forum 61. Grapevine. TX. 2005.
- [Яковлев и др., 2013] Яковлев К.С., Макаров Д.А., Панов А.И., Зубарев Д.В. Принципы построения многоуровневых архитектур систем управления беспилотными летательными аппаратами // Авиакосмическое приборостроение, 4, 2013. С.10-28
- [Bonasso et al., 1995] Bonasso, R.P., et al. Experiences with an Architecture for Intelligent, Reactive Agents. International Joint Conference on Artificial Intelligence. 1995.
- [Gat, 1992] Gat, E. Integrating planning and reacting in a heterogenous asynchronous architecture for controlling real-world mobile robots. National Conference for Artificial Intelligence. 1992
- [Bristeau et al., 2011] Bristeau, P. J., Callou, F., Vissière, D., & Petit, N. The navigation and control technology inside the AR.Drone micro UAV. In 18th IFAC World Congress (Vol. 18, No. 1, pp. 1477-1484). 2011.
- [Albus et al., 2002] Albus, J., Huang, H. M., Messina, E., Murphy, K., Juberts, M., Lacaze, A., & Finkelstein, R. 4D/RCS Version 2.0: A reference model architecture for unmanned vehicle systems. National Institute of Standards and Technology, Gaithersburg, MD, NISTIR, 6912. 2002.
- [Quigley et al., 2009] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., & Ng, A. Y. ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2). 2009.
- [Klein et al., 2007] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In International Symposium on Mixed and Augmented Reality, 2007.
- [Munoz-Salinas et al., 2014] R. Munoz-Salinas, S. Garrido-Jurado, Aruco library, <http://sourceforge.net/projects/aruco/>
- [Engel et al., 2012] J. Engel, J. Sturm, D. Cremers. Camera-Based Navigation of a Low-Cost Quadcopter. In Proc. of the International Conference on Intelligent Robot Systems (IROS), 2012.