

Черняев Максим Юрьевич

студент

ФГБОУ ВО «Белгородский государственный
технологический университет им. В.Г. Шухова»

г. Белгород, Белгородская область

**РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ СИСТЕМЫ УПРАВЛЕНИЯ
БЕСПИЛОТНЫМ ЛЕТАТЕЛЬНЫМ АППАРАТОМ
С ИСПОЛЬЗОВАНИЕМ ROS, GAZEBO**

***Аннотация:** в данной статье рассматривается вопрос осуществления имитационного моделирования объектов робототехники. Проводится сравнительный анализ существующих программных решений. Описывается процесс компьютерной симуляции полета беспилотного летательного аппарата с помощью программного комплекса Robotic Operation System и среды моделирования Gazebo. Приводятся результаты исследования.*

***Ключевые слова:** имитационная модель, беспилотный летательный аппарат, ROS, Gazebo.*

Будущее развития робототехники, в том числе беспилотных летательных аппаратов, это качественное моделирование [1], которое позволяет быстрее разрабатывать и повышать производительность за счет сокращения процесса проб и ошибок при постоянном тестировании на физических аппаратах.

Однако поскольку моделирование беспилотного летательного аппарата распространяется на моделирование всей роботизированной системы, включая роение, поиск путей, экологическую осведомленность, автономное картографирование и навигацию без GPS, необходимы как можно более мощные инструменты.

В ходе исследований было проанализировано три основных программных комплекса [2–4] для решения поставленной задачи. Стоит отметить, что приведены лишь некоторые системы, характерные для своего класса, интересные с точки зрения критериев, определенных спецификой задачи. Основные характеристики, рассмотренных систем, отображены в табл. 1.

В результате оценки и сравнения выбрана связка ROS и Gazebo. Для симуляции поведения группы роботов была использована модель робота в ROS: квадрокоптер Hector Quadrotor. Достоинством данной модели является её широкая популярность в исследовательских работах, наличие различных плагинов взаимодействия в симуляторе Gazebo и в робототехнической операционной системе ROS.

Таблица 1

Сравнительная характеристика рассмотренных систем

Название	MSRS	ROS + Gazebo	Webots
Параметры			
Гибкость физических настроек	Низкая	Высокая	Высокая
Возможность проектирования произвольных роботов	Да	Да	Да
Удобство импортирования моделей из средств твердотельного моделирования	Низкое	Низкое	Низкое
Открытый исходный код	Нет	Да	Нет
Наличие визуального языка программирования	Да	Нет	Нет
Возможность загрузки управляющей программы в реального робота	Нет	Да	Нет

Модель квадрокоптера Hector Quadrotor, поддерживаемая ROS и доступная в библиотеке моделей Gazebo, может включать в себя лазерный дальномер, камеру и другие датчики. Прототипом такого квадросоторного БПЛА являются квадрокоптеры, зачастую собираемые вручную из небольших, легких элементов конструкции и электромеханических компонентов с низким энергопотреблением, как, например, квадрокоптер, показанный на рис. 1а. Модель из Gazebo для квадрокоптера Hector Quadrotor представлена на рис. 1б.



Рис. 1. Четырехроторный БПЛА: фото (а) и модель из Gazebo (б)

Для управления моделью квадрокоптера использовался плагин `geometry_msgs` [5]. Модель принимает сообщения `geometry_msgs / Twist` по теме `cmd_vel`, как и многие другие роботы в ROS. Структура входных данных имеет следующий вид:

'[3.0, 0.0, 0.0]' '[0.0, 0.0, 2]' – сообщение `geometry_msgs/Twist`, имеющее два вектора: `linear` и `angular`, каждый из которых состоит из трех числовых элементов с плавающей точкой, '[3.0, 0.0, 0.0]' становится линейным значением с $x=3$, $y=0$, $z=0$ и '[0.0, 0.0, 2]' является угловым значением с $x=0$, $y=0$, $z=2$. Эти аргументы имеют YAML – синтаксис. Опубликовать сообщение в тему можно вручную:

```
rostopic pub -1 /cmd_vel geometry_msgs/Twist -- '[0.0, 0.0, 0.3]' '[0.0, 0.0, 0.0]'
```

На основе передачи команд плагину, была написана управляющая программа на языке Python, позволяющая осуществлять управление посредством нажатия клавиш. Скрипт запускается как узел ROS:

```
$ roscore
```

```
$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

В процессе работы данного скрипта на тему `/cmd_vel` узел `teleop_twist_keyboard` публикует сообщения каждый раз, когда была нажата клавиша на клавиатуре.

Для представления вычислительного графа ROS в графическом виде, предоставляется инструмент `rqt_graph`. Схема вычислительного графа показывает все активные узлы и темы, которые их связывают.

```
$ rosrun rqt_graph rqt_graph
```

Видно, что узел `teleop_twist_keyboard` публикует сообщения в тему `/cmd_vel`, а узел `rostopic` подписывается на эту тему (рис.2).

Для автоматического управления моделью квадрокоптера написан скрипт, использующий тот же принцип публикации сообщений в тему `/cmd_vel` через узел `teleop_twist_keyboard`. Это позволяет создавать заранее заготовленную последовательность команд управления и осуществлять автономный полёт. Команды хранятся в файле.csv, который читает скрипт.

- функциональность (способность подучать необходимый результат при моделировании);
- эффективность (способность обеспечивать работоспособность по отношению к выделяемым для этого ресурсам);
- сопровождаемость (минимизация усилий при внесении изменений для устранения ошибок и модификации).

Список литературы

1. Бондаревский А.С. Имитационное моделирование: определение, применимость и техническая реализация / А.С. Бондаревский, А.В. Лебедев // Фундаментальные Исследования. – 2011. – №12–3.
2. Использование симуляторов в робототехнике [Электронный ресурс]. – Режим доступа: <https://m.habr.com/company/robotgeeks/blog/2589/> (дата обращения: 18.12.2018).
3. Гай В. Microsoft Robotics Developer Studio. Программирование алгоритмов управления роботами. – М.: ЭКОМ Паблишерз. – 2012. – 184 с.
4. Gazebo [Электронный ресурс]. – Режим доступа: <http://gazebo.org/> (дата обращения: 20.12.2018).
5. ROS Documentation (geometry_msgs) [Электронный ресурс]. – Режим доступа: http://wiki.ros.org/geometry_msgs (дата обращения: 16.12.2018).