

численные библиотеки компонентов, в том числе, для управления роботами и системами машинного зрения. Например, на LabView основана среда программирования Lego NXT.

В состав LabVIEW Robotics входит:

1. Pyro – инструментарий управления роботом, написанный на языке Python.
2. OpenRAVE – трёхмерный симулятор.
3. YARP – фреймворк с открытым исходным кодом, предназначенный для работы с оборудованием робота и написанный на языке C++.
4. CLARAty – программная платформа, распространяемая в виде open source-проекта. Реализует широкий спектр алгоритмов.

В результате проведенного анализа существующих решений программного обеспечения для разработки архитектуры системы управления роботом, можно подвести следующие общие итоги:

1. Все решения представлены в виде промежуточного слоя между обычной ОС и программами/скриптами для управления роботом.
2. Все решения имеют модульную структуру, которая работает поверх базовой прослойки (фреймворка).
3. Все решения имеют распределённую клиент-серверную структуру.

Список литературы

1. ИНЖЕНЕР инфо. Статьи: Контроллеры. Программирование и составление программы PLC на ПЛК. – Режим доступа: <http://www.ingener.info/pages-page-27-2.html> (дата обращения: 27.11.13).

УДК 621.382

ИСПОЛЬЗОВАНИЕ ФРЕЙМВОРКА ROS ДЛЯ РАЗРАБОТКИ АРХИТЕКТУРЫ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ

*магистрант У.В. Михайлова, магистрант Е.А. Михайлов,
руководитель – А.С. Сарваров*

*ФГБОУ ВПО «Магнитогорский государственный технический
университет им. Г.И. Носова», Россия, г. Магнитогорск
ylianapost@gmail.com*

Аннотация

В данной статье рассмотрена структура и принципы работы фреймворка ROS для разработки архитектуры системы управления роботом. Разобраны особенности данного фреймворка.

Ключевые слова: робот, системы управления, фреймворк ROS.

THE USE OF ROS FRAMEWORK FOR ROBOT CONTROL SYSTEM ARCHITECTURE DEVELOPMENT

undergraduate U.V. Mikhailova, undergraduate E.A. Mikhailov, supervisor – A.S. Sarvarov

*Nosov Magnitogorsk State Technical University
Russia, Magnitogorsk
ylianapost@gmail.com*

Abstract

This article examines the structure and principles of the ROS framework for the development of robot control system architecture. The features of this framework are collated.

Key words: robot, control system, the framework ROS

Наиболее важными факторами успеха при разработке архитектуры системы управления роботом являются использование распространенной элементной базы и программного обеспечения с открытым кодом (по возможности, свободно распространяемого). В свою очередь, дизайн программного обеспечения должен быть сфокусирован на максимальной портативности и возможности реконфигурации. Программное обеспечение открытых управляющих систем переносимо на различные элементные базы и операционные системы. Открытая управляющая система должна иметь модульную структуру с возможностью легкой замены модулей на более новые, что означает расширяемость рабочих характеристик и функциональности. Открытая управляющая система позволяет осуществить быструю интеграцию новых функциональных характеристик. Это означает, во-первых, наличие стандартного программного обеспечения и элементной базы, что позволяет привлекать к разработкам третьих лиц; во-вторых, у открытых управляющих систем должен быть механизм plug-and-play для быстрого внедрения этих новых компонент. Открытая управляющая система должна быть легко реконфигурируемой, т.е. обладать механизмом легкой адаптации своих параметров и настроек для решения конкретных практических задач [1].

Другой важный фактор при создании современной открытой архитектуры управляющей системы робота - это использование программного обеспечения с открытым кодом (ПООК), который привел к применению числового программного управления в автоматизированных системах. Одним из наиболее многообещающих применений программного обеспечения с открытым кодом в открытых управляющих системах является использование их на уровне операционной системы. Во-первых, программное обеспечение с открытым кодом позволяет уменьшить стоимость всей управляющей системы, так как не нужно оплачивать лицензию за коммерческую операционную систему, работающую в режиме реального времени. Во-вторых, преимущества открытого кода позволяют разработчикам провести полную настройку для своих нужд.

Всем указанным выше требованиям удовлетворяет программное решение, разработанное в 2007 году совместными усилиями исследовательской лаборатории Willow Garage и лаборатории искусственного интеллекта Стэнфордского университета. Это операционная система для роботов с открытым исходным кодом ROS (Robot Operating System). ROS выпускается в соответствии с условиями BSD-лицензии, которая является одной из самых популярных лицензий для свободного программного обеспечения. ROS - это фреймворк для программирования роботов, предоставляющий возможность построения распределенных систем с использованием языков программирования C++ и Python. Фреймворк ROS обеспечивает основные стандартные службы операционной системы, а именно:

- 1) аппаратную абстракцию;
- 2) низкоуровневый контроль устройств;
- 3) передачу сообщений между процессами;
- 4) реализацию часто используемых функций;
- 5) управление пакетами данных.

Архитектура ROS является клиент-серверной и основана на архитектуре графов, в которой обработка данных происходит в узлах, которые могут получать и передавать сообщения между собой. Библиотека ориентирована на Unix-подобные системы (Ubuntu Linux включена в список «поддерживаемых» в то время как другие варианты, такие как Fedora и Mac OS X считаются «экспериментальными»). Фреймворк ROS является бесплатным для использования, как в исследовательских, так и в коммерческих целях.

Фреймворк ROS имеет две основных составляющих: операционная система - `ros`, и `ros-pkg` - набор поддерживаемых пользователями пакетов (организованных в наборы, которые называются стек). Пакеты из `ros-pkg` распространяются на условиях различных открытых лицен-

зий. Эти пакеты реализуют различные функции робототехники: метод одновременной навигации и построения карты (SLAM), планирование, восприятие, моделирование и др.

В рамках фреймворка ROS, реализовано несколько алгоритмов SLAM, таких как реализация метода SLAM на основе:

1) данных от лазерного дальномера и одометрии робота - GMapping;

2) методов компьютерного зрения - VSLAM (Visual SLAM) основывается на отслеживании ключевых точек на изображении, поступающем с камер робота. Реализация данного метода еще имеет недостатки, так как в случаях, когда камера движется быстро (например, при разворотах), возникает размытие изображения и отслеживание ключевых точек теряется. В результате чего алгоритм перестаёт работать;

3) облака точек с RGBD датчиков (Kinect или стерео-камеры rgbdslam) - пакет, для регистрации rgbdslam;

4) данных от LIDAR-ов - hector_mapping. Применяют для платформ без одометрии. Использует высокую скорость обновления систем лазерного сканирования, и обеспечивает данные оценки 2D-положения со скоростью сканирования датчиков (40Гц для UTM-30LX);

5) одометрии робота и данных лазерного дальномера - karto;

Еще несколько алгоритмов SLAM находятся в процессе разработки.

Операционная система ROS поддерживает роботов на базе: PR2, TurtleBot, PR1, HERB, STAIR I и II, Nao: Nao, Husky A200, iRobot Create, Lego Mindstorms NXT. В состав ROS входит набор из различных библиотек, таких как:

- OpenCV — библиотека, содержащая алгоритмы компьютерного зрения и обработки изображений;

- PCL- библиотека для работы с облаками 3D-точек;

- OGRE — объектно-ориентированный графический движок с открытым исходным кодом;

- Orocos — библиотека для управления роботами (например, расчет кинематики).

В качестве базовых концепций файловой системы ROS используются [2]:

1. Пакет (package) - наименьшая единица файловой системы. Представляет собой директорию, содержащую в себе какие-либо данные, библиотеки, исполняемые и конфигурационные файлы и т.д. и т.п., логически объединенные в какой-то полезный модуль.

2. Стэк – объединение пакетов.

3. Узел - это запущенный процесс, который умеет общаться с

другими процессами.

4. Шина - именованный канал, соединяющая различные узлы.

Узлы и шины формируют асинхронный механизм обмена данными (рис. 1).

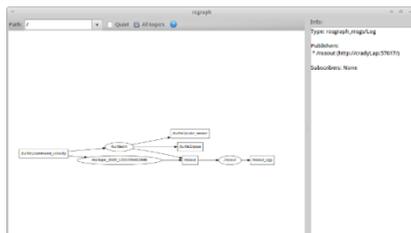


Рис. 1. Окно ОС ROS, в котором отображены активные узлы, и шины через которые они между собой взаимодействуют

Основными достоинствами фреймворка ROS по сравнению с другими предлагаемыми решениями в области разработок архитектуры систем управления роботами является поддержка повторного использования программного кода. Но это не единственное его достоинство, так как фреймворк является платформой с большим количеством функций. Обеспечение распределенной сети процессов (узлов) позволяет разрабатывать каждый узел индивидуально. Разработанная структура и дизайн ROS предоставляет возможность разработчику архитектуры систем управления роботами независимо принимать решения о разработке и реализации. После чего все это может быть объединено вместе, средствами инфраструктуры ROS.

Список литературы

1. ИНЖЕНЕР инфо. Статьи: Контроллеры. Программирование и составление программы PLC на ПЛК. – Режим доступа: <http://www.ingener.info/pages-page-27-2.html> (дата обращения: 27.11.13).

2. Основы работы с Robotic Operating System. – Режим доступа: <http://habrahabr.ru/post/128024/> (дата обращения: 27.11.13).