

оригиналом, будут выделяться жёлтым, а части идентичные оригиналу зелёным. Если найденные структурные элементы не совпадают с блоком шаблона, то они выделяются красным.

Так как проектируется надстройка для текстовых процессоров, шапка текстовых процессоров останется, а вот рабочее поле будет разделено на две части в виде раскрытой тетради. Слева будет отображаться шаблон с возможностью прокрутки, справа текст с пометками. Также будут иметься возможности: быстрого поиска, по средствам клика на блок или на элемент блока; редактирования документа с метками; возможность сохранения отредактированного документа как с метками, так и без них и сохранения документа с метками без редактирования.

Основные функции текстового процессора будут сохранены, насколько это возможно.

Блоки документов будут сохраняться в папке с шаблоном и удаляться по завершению работы. Это делается для безопасности и возможности повторного анализа уже исправленного документа, так как будут сохраняться выделенные блоки шаблона это позволит не проводить повторно анализ шаблона.

Сравнивать приложение будет только данные шаблона, то есть текст, идущий после структурного элемента рассматриваемого блока не будет рассматриваться при анализе, так как в данной работе рассматривается структурный анализ.

При сравнение структурных элементов будет рассматриваться не только содержание, а также стили и расположение объектов. Приложение будет показывать специальным знаком (стрелка вверх или вниз) позицию откуда был перемещён блок.

Данное приложение имеет широкий круг применения, от учебных заведений до корпоративного сектора, например, может применяться при оформлении договоров, технических заданий, рабочих и учебных планов.

Приложение позволит ускорить процесс создания документов за счёт быстрой проверки документа на ошибки и наличие нужных структурных элементов, а также даёт возможности редактирования и внесения исправлений без использования нескольких программ. Также позволяет создавать новые шаблоны на основании имеющегося шаблона и документа.

Библиографический список

1. MSDN – сеть разработчиков Microsoft [Электронный ресурс] URL: <https://msdn.microsoft.com/ru-ru> (дата обращения: 19.12.2017).
2. Магомедов И.Н. Сопоставление структур форматов документов odt и docx // Старт-2017: Тезисы докладов III Общероссийской молодёжной науч.-техн. конф. СПб: Балт. гос. техн. ун-т., 2017. – С. 48.

УДК 004.4

ПОСТРОЕНИЕ СИСТЕМ УПРАВЛЕНИЯ РОБОТИЗИРОВАННЫМИ УСТРОЙСТВАМИ С ИСПОЛЬЗОВАНИЕМ ВЕБ-ТЕХНОЛОГИЙ И ROS

А. С. Мальцев, К. Е. Мамонова, Т. П. Щекочихин

Институт автоматики и электрометрии СО РАН

Введение

Роботизированные устройства находят применение в промышленности, транспорте и повседневной жизни, что ставит необходимость в создании специализированных систем управления. В последнее время широкое внимание уделяется беспилотному транспорту и шагающим механизмам, способным оперировать на сложной местности. Примером использования транспортных роботов в производстве является склад компании Amazon. В сентябре 2017 года численность роботов, использующихся компанией достигла 100 тысяч единиц [1]. На предприятии роботы действуют совместно с людьми, под их контролем. Беспилотным наземным транспортом занимаются такие крупные компании, как Renault-Nissan, Tesla, Alphabet и многие другие [2]. Грузы могут перемещаться и группой роботов. Распространен подход под названием «saging». Группа автономных транспортных средств окружает объект и двигает его вместе, контролируя местоположение каждого представителя группы [3]. Специальные алгоритмы управления отвечают за сохранение конфигурации. Для этого применяются методы, основанные на схеме «лидер-ведомые». В этом случае лидер движется по заданной траектории, а ведомые поддерживают требуемое местоположение.

В тоже время авария на АЭС Фукусима-1 показала, что колесные и гусеничные роботы плохо подходят для аварийных работ. Для таких случаев в Italian Institute of Technology и в University of Pisa

разрабатываются шагающие роботы. Также компания Boston Dynamics предлагает использовать шагающих роботов для доставки грузов на поле боя и для работ на складах. Среди отечественных шагающих роботов можно выделить AR-600 и робота «Федора», предназначенного для использования на МКС.

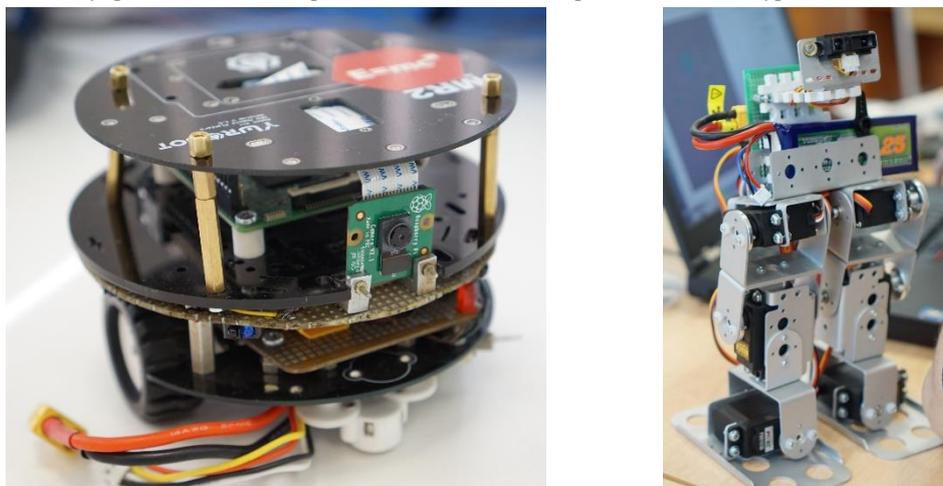
Для проведения практических экспериментов по управлению роботами и их группами нами созданы прототипы колесного и шагающего роботов. Колесный робот оснащен двумя моторами, набором инфракрасных датчиков и бортовой камерой. Он представляет собой прототип беспилотного транспортного средства, обладающего улучшенным функционалом по сравнению с популярным швейцарским роботом e-puck. Шагающий робот оснащен шесть сервомоторами и дальномером. Цель представленной работы состоит в создании системы управления обоими роботами по единому принципу, основанному на разделении функций между отдельными задачами и состоящую из верхнего и нижнего уровней управления.

1. Конструкция роботов

Конструкция прототипа колесного робота (рис. 1а) представляет собой платформу из двух уровней, оснащенную по периметру шестью инфракрасными датчиками, двумя моторами постоянного тока, платой управления с микроконтроллером STM32 и драйвером двигателей, бортовым компьютером Raspberry Pi 3 с подключенной к нему камерой Raspberry Pi Camera Board. Питание осуществляется от аккумуляторной батареи, закрепленной на дне корпуса.

Конструкция шагающего робота (рис. 1б) представляет собой человекоподобный скелет нижних конечностей, сделанный из алюминия. На двух ногах расположены по три сервомотора (на бедренной, коленной и ступичной частях), каждый из которых отвечает за поворот своей части. На верхней части робота расположена плата управления с микроконтроллером, к которой подсоединены моторы и дальномер, прикрепленный с помощью конструкционной детали. Питание осуществляется от аккумуляторной батареи напряжением 7В, закрепленной на верхней части конструкции. Батарея подключается к плате управления.

Система управления обоими прототипами состоит из верхнего и нижнего уровней.



а б
Рис. 1. Внешний вид прототипов колесного и шагающего роботов

2. Система управления нижнего уровня

На нижнем уровне микроконтроллер управляет механикой робота и опрашивает датчики. Этот контроллер принимает команды по последовательному порту, которые включают требуемую скорость вращения моторов колесного робота либо требуемый угол поворота указанного мотора и необходимое время вращения на этот угол для плавного движения шагающего робота.

Конструкция роботов непрерывно модернизируется, вследствие этого разрабатываемое программное обеспечение должно обладать свойствами расширяемости и гибкости под изменения установленного оборудования, условий и задач. Поэтому решено разрабатывать программное обеспечение для микроконтроллера с использованием операционной системы реального времени

FreeRTOS [4]. Это позволяет разбить программу на относительно простые независимые задачи, а в перспективе добавлять новые по мере необходимости, не затрагивая логику работы других. На рис. 2 показан пример взаимодействия таких задач для колесного робота, обмен данными между которыми реализован через очередь сообщений [5].

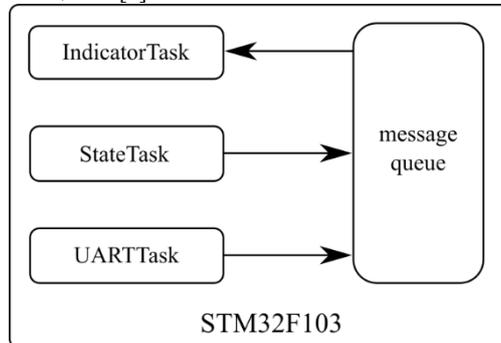


Рис. 2. Схема задач, исполняемых на микроконтроллере

Здесь представлено три задачи:

- индикация статуса программы и заряда аккумуляторной батареи;
- связь с бортовым компьютером и обмен данными по последовательному порту;
- контроль параметров платформы.

3 Система управления верхнего уровня

Система верхнего уровня исполняется на бортовом компьютере с операционной системой Linux и обеспечивает функции веб-сервера, сервера трансляции видео, автономной навигации, законов управления и драйвера к нижнему уровню. Каждая из функций реализуется в виде независимой программы, которые взаимодействуют между собой путем получения и отправки сообщений через механизм коммуникации Robot Operating System (ROS) [6]. Вариант структуры такой системы для колесного робота приведен на рис 3.

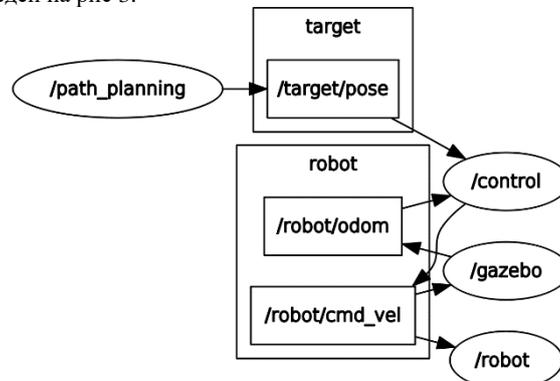


Рис. 3. Структура системы управления

На схеме овальной рамкой обозначены программы, прямоугольной - сообщения. Стрелки показывают получает программа сообщение или посылает его.

В состав системы входят следующие программы:

- планирование маршрута (path_planning);
- моделирование движения робота, симулятор (gazebo);
- сервер управления (control);
- драйвер к нижнему уровню управления (robot).

Механизм работы данной системы характеризуется асинхронной передачей сообщений между указанными программами путем публикации сообщений и подписки на них. Координаты очередной точки на заданной для робота траектории непрерывно публикуются программой планирования маршрута path_planning. Данные о точке назначения, текущих координатах, командах оператора

передаются в control, где на основе метода управления движением [7] вычисляется вектор скоростей и посылаются сообщения /robot/cmd_vel, которые представляют собой команды управления роботом. Симулятор gazebo получает сообщения с векторами скоростей и рассчитывает координаты и угол ориентации робота на плоскости. Драйвер robot транслирует команды вектора скоростей в команды управления моторами робота. Также сервер управления control обеспечивает следующие функции:

- загрузка на автоматизированное рабочее место оператора (АРМ) базовой части клиентского программного обеспечения;
- обработка запросов и ввод команд.

Взаимодействие оператора с системой управления осуществляется через веб-интерфейс автоматизированного рабочего места [8]. Он включает поле для трансляции видео с бортовой камеры робота и элементы ручного и автоматического управления движением, индикацию показаний бортовых датчиков. Для его создания использованы языки JavaScript, Python, векторная графика в формате SVG для элементов управления и библиотека JQueryUI. Использование формата SVG позволяет при проектировании интерфейса пользователя использовать как специализированные графические приложения (например, редактор векторной графики Inkscape), так и любой текстовый редактор. По нажатию кнопок управления в окне веб-браузера на сервер посылаются запросы с использованием асинхронного механизма на базе XMLHttpRequest. В этих запросах содержатся команды для нижнего уровня, управляющего механикой робота. Связь с оператором или другими роботами осуществляется по Wi-Fi, связь бортового компьютера с микроконтроллером - через serial port.

Данные о координатах поступают от симулятора в режиме моделирования либо от системы технического зрения с использованием такого модуля ROS, как ar_track_alvar, обрабатывающего кадры с закрепленной сверху видекамеры. При этом на роботе устанавливается специальный маркер. При использовании роботов вне помещения информация о координатах будет браться от GPS либо от навигационной системы, которая объединяет данные от разных источников на основе комплиментарного фильтра либо фильтра Калмана.

Заключение

Таким образом, в работе представлена архитектура управляющих программных систем, основанная на разделении функций между отдельными программами и использующая ROS в качестве механизма их взаимодействия. Такая архитектура позволяет легко наращивать либо изменять функциональность системы путем добавления новых компонент и использовать большое число готовых модулей из состава ROS, включая навигацию, построение карты местности, техническое зрение. Система может использоваться не только для управления реальными роботизированными устройствами, но и для моделирования различных ситуаций в интегрированном с ROS симуляторе Gazebo с целью отладки компонент системы управления.

Благодарности

Работа выполнена за счет средств субсидии на финансовое обеспечение выполнения государственного задания (№ АААА-А17-117060610006-6) в Институте автоматизации и электротехники СО РАН.

Библиографический список

1. Число роботов на складах Amazon достигло 100000 [Электронный ресурс]. URL: https://hightech.fm/2017/09/15/amazon_robot_workers (дата обращения: 12.01.18).
2. 36 проектов беспилотных автомобилей [Электронный ресурс]. URL: <https://geektimes.ru/post/277788/> (дата обращения: 12.01.18).
3. Rubenstein, M., Cabrera, A., Werfel, J. Collective transport of complex objects by simple robots: theory and experiments // Proc. of the 12th Intern. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2013). Saint Paul, MN, USA. 6-10 May 2013. P. 47–54.
4. Carmine Noviello. Mastering STM32. A step-by-step guide to the most complete ARM Cortex-M platform, using a free and powerful development environment based on Eclipse and GCC. Leadpub. 2017. 792p
5. Канина К.В. Система управления мобильной платформой на базе микроконтроллера STM32 и операционной системы реального времени FreeRTOS // Сборник трудов 55-й Международной студенческой научной конференции Новосибирск. 2017. С. 26.

6. Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng. ROS: an open-source Robot Operating System // In Proc. of ICRA workshop on open source software. 2009. Vol. 3. No 2. 5p.

7. Канина К.В., Мальцев А.С., Цупа А.Е. Создание экспериментальных роботов и алгоритмов управления в условиях возмущений // Труды XIX Международной конференции "Проблемы управления и моделирования в сложных системах" (12-15 сентября 2017 г., Самара, Россия). Самара: ООО "Офорт". 2017. С. 135-140.

8. Мамонова К.Е., Щекочихин Т.П. Создание системы удаленного управления и автономного движения мобильным роботом // Сборник трудов 55-й Международной студенческой научной конференции. Новосибирск. 2017. С. 37.

УДК 62.529

УСТРОЙСТВО ИСПРАВЛЕНИЯ ОШИБОК ПРИ ОДНОКРАТНОЙ ПЕРЕДАЧЕ ДАННЫХ

А. И. Митюшов, К. А. Островский

Балтийский государственный технический университет «ВОЕНМЕХ» имени Д.Ф. Устинова

Введение: На вооружение ведущих стран мира поступают новейшие системы оружия, главным отличительным признаком которых является реализованный принцип "выстрел - поражение", т.е. они обладают способностью гарантированно поразить цель одним выстрелом, в любое время суток, в сложных метеорологических условиях и интенсивном противодействии со стороны противника. Такие системы называются высокоточным оружием (ВТО). Наиболее общими чертами ВТО являются: резкое увеличение дальности стрельбы; широкая унификация оружия независимо от его базирования; исключение человека из процесса разведка - целеуказание - поражение; увеличение эффективности поражения за счет высокоточной навигации и повышения мощности взрывчатого вещества на головной части.

Развитие ВТО привело к необходимости интеграции средств разведки, управления, обеспечения и поражения в единые разведывательно-ударные комплексы (РУК). Необходимым условием функционирования таких комплексов является обмен информацией между средствами разведки и поражения. Как правило, передача информации происходит неконтактным способом при помощи радиоволн. Радиоканал подвержен воздействию различного рода помех как естественного, так и искусственного происхождения. Помехи искажают передаваемый сигнал. Это может привести к искажению содержащейся в нём информации и, в конечном счёте, к снижению эффективности комплекса ВТО. Поэтому задача исправления ошибок и повышения достоверности передачи информации является актуальной.

Разрабатываемые средства поражения, в частности артиллерийские системы, должны удовлетворять требованиям к высокоточному оружию. Поскольку типы целей и расстояния до них постоянно изменяются в ходе боя, возникает необходимость ввода в боеприпасы информации о времени и режиме срабатывания, времени замедления и т.д. непосредственно во время стрельбы, причем способ ввода информации не должен ограничивать скорострельность.

В настоящее время наибольшее распространение получили проводные, индукционные и радиолинии ввода информации.

Проводная линия реализует гальваническую связь электроники взрывателя с внешними электрическими цепями при помощи разъёмов или врезных контактов на корпусе боеприпаса. Её основное достоинство – простота.

Недостатки связаны со способом соединения. Ввод информации через разъёмы, соединяемые вручную перед выстрелом, требует времени и ограничивает скорострельность.

Кроме того, поскольку ввод информации осуществляется до выстрела, невозможно учесть реальную скорость снаряда после выстрела.

Радиолиния позволяет вводить информацию до выстрела (на линии заряжания) и после выстрела («вдогон» снаряду). Ввод информации до выстрела не позволяет учесть реальную скорость снаряда. Ввод информации по радиолинии «вдогон» позволяет определить скорость каждого снаряда при помощи доплеровского измерителя, скорректировать данные для управления подрывом и передать их во взрыватель, однако имеет три основных недостатка.