электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

#09, сентябрь 2015

УДК 621.865.8

Обзор и сравнение коммерческих и открытых программных комплексов для моделирования робототехнических систем

Гайнетдинов А.Ф., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Робототехнические системы и мехатроника»

Научный руководитель: Романова И.К., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Робототехнические системы и мехатроника»
sm7@sm.bmstu.ru

1. Введение

Симуляторы играют важную роль в робототехнических исследованиях как инструменты для испытаний эффективности, безопасности, и надёжности новых алгоритмов. В последние годы мы стали свидетелями потребности в развитии точных, надёжных, и простых в использовании программных средств для моделирования робототехнических моделей, датчиков и управления в виртуальных средах. Эта потребность вызвана увеличением областей применения современных роботов, в которых они должны тесно взаимодействовать с людьми. Это имеет место, например в медицинской робототехнике, в роботах для обеспечения безопасности, бытовых роботах. В этих случаях, в связи с изменчивостью окружающей обстановки, крайне важно проверить безопасность и надёжность нового алгоритма с помощью реалистичного и надёжного симулятора. Испытания надёжности робота путём изменения только определённых условий моделирования является одним из преимуществ симуляторов. Кроме того, одно из требований, предъявляемых к современным симуляторам это переносимость управляющего кода из симулятора на реальную платформу.

Вследствие этих требований, растёт количество различных коммерческих и открытых программных робототехнических симуляторов. Несмотря на большое количество коммерческих и открытых робототехнических исследовательских инструментов, насколько известно, нет комплексного обзора и сравнения их

особенностей. В результате этой статьи, исследователи смогут принять решение, какое программное обеспечение лучше всего подходит для их целей.

Эта статья представляет собой подробный обзор и сравнение последних распространённых коммерческих и открытых робототехнических программных комплексов для моделирования робототехнических систем.

2. Обзор робототехнических симуляторов

Здесь представлен обзор последних и широко используемых программных комплексов для моделирования и программирования роботов. Он разделён на две категории: коммерческие и с открытым исходным кодом. Краткая информация о них изложена в таблице.

2.1. Коммерческие пакеты программного обеспечения

Webot – это профессиональный 3D симулятор для робототехники, широко используемый для учебных целей. Модели имеют большое количество различных настраиваемых параметров, таких как структура, масса, трение, форма и т.д. Эти параметры предоставлены динамически подключаемой библиотекой открытого физического движка (Open Dynamics Engine), который используется для имитирования динамики твёрдого тела. Он может моделировать различные типы робототехнических платформ, такие как мобильные роботы (Epuck, Pioneer, LegoMindstorm), четвероногие роботы (Sony’s AIBO), и человекоподобные роботы (Fujitsu’s HOAP-2). Webot может моделировать большое количество сенсоров, широко используемые в робототехнике, такие как датчики расстояния, датчики света, датчики прикосновения, GPS, акселерометры, лазеры, камеры. Webot поддерживает графический интерфейс для взаимодействия пользователя с окружающей средой, роботами или другими объектами. Встроенный редактор движения используется для создания и хранения последовательности перемещений сочленений робота, которые могут быть воспроизведены с помощью контроллера робота. Webot может быть использован для создания AVI или MPEG видео моделирования. Пользователь может моделировать одного или нескольких роботов одновременно со средствами связи. Webot для создания программы управления использует следующие языки программирования: C, C++, Java, Python и Matlab. Webot может взаимодействовать с другими программными обеспечениями через TCP/IP, например с LabView. Ещё одна особенность Webot это возможность передачи управляющего кода через Bluetooth. Также пользователи могут

использовать в управляющем коде алгоритмы высокого уровня, такие как SLAM и планирование траектории. Webot имеет профессиональную и учебную версию для Linux, MAC OSX и Windows.

Easy-Rob – это программное обеспечение для планирования и моделирования робототехнических платформ, работающих на автоматизированных участках производственных предприятий. Easy-Rob позволяет пользователю программировать и визуализировать в 3D различные процессы, такие как обработка, сборка, покраска, запечатывание. Также Easy-Rob имеет возможность создания AVI видео моделирования. Пользователи могут использовать API, предоставленную Easy-Rob, внешние робототехнические библиотеки, встроенные инструменты для импорта и экспорта AutoCAD моделей. Easy-Rob имеет версии для Microsoft Windows XP, Vista и Windows 7.

MATLAB с Robotics System Toolbox обеспечивает алгоритмы и аппаратные средства связи для разработки приложений для автономных мобильных роботов. Алгоритмы Toolbox включают в себя построение карты, планирование траектории и следование траектории роботов с дифференциальным приводом. Можно проектировать и исследовать контроллеры двигателей, компьютерное зрение, конечные автоматы. Система обеспечивает интерфейс между MATLAB Simulink и Robot Operating System (ROS), который позволяет протестировать и проверить управляющий код роботов, работающих под управлением ROS. Он поддерживает генерацию C++ кода, позволяя создать узел ROS из модели Simulink, и применить его в сети ROS. Robotics System Toolbox включает в себя примеры, показывающие, как работать с виртуальными роботами в Gazebo и с реальными роботами под управлением ROS.

2.2. Пакеты программного обеспечения с открытым исходным кодом

Player/Stage – проект свободного программного комплекса, распространяемого по лицензии GNU, который предоставляет набор современных инструментов для взаимодействия и управления одним или несколькими роботами. Player - это аппаратно-независимый сервер, обеспечивающий управление роботом через сеть. Player поддерживает подключение нескольких клиентов (например, программ управления) к одному или нескольким устройствам (например, платформы для робота, лазеры, камеры) через подключение к сети, используя протокол TCP/IP. Это позволяет клиентам быть написанным на любом языке программирования, который обеспечивает поддержку сокета TCP. Клиент и API были написаны так, что пользователь может написать код управления

на следующих языках: C, C++, Tcl, Python, Java, CommonLisp, и Matlab. Player поддерживает большое количество платформ для роботов и сенсоров.

Stage – это 2D симулятор платформ роботов и сенсоров. Stage предоставляет моделирование роботов, не требующего много вычислительных способностей, которое позволяет моделировать одновременно большое количество роботов (сотни и тысячи), он также предоставляет планирование траектории и построение карты. Stage работает в режиме реального времени по умолчанию, но он будет работать медленнее, если модель робота требует большего времени для вычислений. Stage доступен на Linux и Mac OSX.

Gazebo – это 3D симулятор, для использования с Player. Все функции используемые в Player/Stage могут использованы в Gazebo без каких-либо модификаций. Gazebo использует движок объектно-ориентированного рендеринга (Object-Oriented Graphics Rendering Engine) и открытый физический движок (Open Dynamics Engine) для рендеринга 3D объектов. Эти библиотеки позволяют точно воспроизводить динамические 3D среды, с которыми робот может столкнуться, все моделируемые объекты имеют массу, трение, и большое число других параметров, которые позволяют им вести себя реально, когда их толкают, тянут, опрокидывают или несут. Моделируемые роботы могут быть получены, используя примитивы с различными соединениями между ними; это позволяем пользователю моделировать широкий круг платформ для роботов. Моделирование окружающей среды является также сложной задачей, как и моделирование самого робота, и оно может быть построено с помощью имитации света, статических объектов и т.д. Gazebo доступен на Mac OSX и Linux.

Robot Operating System (ROS) - это операционная система с открытым исходным кодом, которая предоставляет аппаратную абстракцию, управление устройствами низкого уровня, передачу сообщений между процессами, и управление пакетами. Одна из наиболее интересных черт ROS является широкое и быстро растущее сообщество исследователей, способствующее его расширению. ROS поддерживает четыре различных языка программирования: C++, Python, Octave, и LISP. ROS имеет большое количество инструментов, которые выполняют широкий спектр задач. При реализации управляющего кода в ROS, пользователи используют узлы, сообщения, темы и сервисы для выполнения задачи. ROS доступен на Linux, MacOSX и Windows (с ограничениями).

Simbad – это Java 3D симулятор для одного или нескольких роботов, разработанный для обучения искусственному интеллекту и машинному обучению в робототехнике. Simbad поддерживает видеокамеры, датчики расстояния и контактные датчики. Simbad предоставляет библиотеку нейронных сетей и библиотеку эволюционных

алгоритмов. Поддерживаемые языки программирования Java и Python. Simbad доступен на Mac OSX, Windows и некоторых версиях Linux.

Carnegie Mellon Robot Navigation Toolkit (CARMEN) – это набор ПО для управления мобильными роботами. CARMEN предназначен для того, чтобы предоставить пользователю лёгкий доступ к алгоритмам робототехники, таких как управление платформой и сенсорами, уклонение от препятствий, навигация, планирование траектории и построение карты. CARMEN представляет собой программное обеспечение для управления, 2D симулятор и поддерживает взаимодействие нескольких реальных роботов и сенсоров (сенсоры расстояния, бесконтактные датчики, GPS). CARMEN поддерживает C и Java. На данный момент Linux единственная поддерживая ОС.

Unified System for Automation and Robot Simulation (USARSim)– это 3D симулятор на основе игрового движка Unreal Tournament. USARSim была создана для моделирования городских поисково-спасательных роботов и окружения, предназначенных для изучения взаимодействия человека и роботов и координации нескольких роботов. Код управления может быть написан, используя интерфейс GameBot, запатентованный движком Unreal, открытую мобильную архитектуру моделирования и систему инструментов (MOAST), интерфейс Player или ToolboxUSARSim в Matlab. Контроллер (например, Player, MOAST) подключается к серверу Unreal, который отправляет команды USARSim, чтобы создать модель робота. Контроллер получает данные от сенсоров и отправляет команды, для управления роботом. Наиболее важной частью USARSim является среда, в которой передвигается робот; есть несколько готовых городских карт для поисково-спасательных целей или они могут быть созданы пользователем. Поддерживаются различные датчики, такие как датчики прикосновения, звуковые датчики, камеры и лазеры. USARSim доступен на Windows и Linux.

Microsoft Robotics Developer Studio (MRDS)– это 3D симулятор на базе Windows для создания приложений для роботов с различными платформами и сенсорами. MRDS технически не является симулятором с открытым исходным кодом, но она доступна публике бесплатно. MRDS использует визуальный язык программирования и подключаемые наборы блоков, чтобы создавать приложения для роботов. Блоки могут быть изменены вручную. MRDS может подключиться как к виртуальной среде моделирования так и к реальному роботу. Визуальная среда моделирования является инструментом 3D моделирования, который использует NVIDIA PhysX и Microsoft XNA Framework, чтобы обеспечить физическое поведение и 3D рендеринг в режиме реального

времени. Различные датчики поддерживаются и доступны с помощью сервисов, которые предоставляет MRDS.

Существует ещё много другого ПО для моделирования робототехнических систем. Но, как было сказано в начале, рассмотрены только те симуляторы, которые широко используются исследователями и студентами.

2.3. Основные критерии оценки программных комплексов

В таблице представлено резюме о главных особенностях коммерческих и открытых симуляторов. Программные комплексы сравниваются по следующим критериям. Операционная система; он показывает, какие ОС поддерживает симулятор. Тип симулятора; он показывает, поддерживает симулятор 2D или 3D моделирование. Язык программирования; он показывает, какой язык программирования поддерживается симулятором. Документация; он показывает уровень документации доступной вместе с симулятором, которая может быть высокоуровневой или низкоуровневой. Высокоуровневая документация предоставляет только описание функций библиотек робототехники. Низкоуровневая документация предоставляет программный код библиотек для робототехники. Учебное руководство; он показывает, предоставлены ли примеры и пошаговое руководство. Переносимость; «Да» означает, что код, написанный для моделирования, может быть использован на реальном роботе. Датчики; он показывает поддерживаемые датчики. Написаны только самые востребованные, из-за ограниченности размера таблицы. Отладка/Логирование; он показывает, есть ли отладка, пошаговое выполнение программы и логирование. Графический пользовательский интерфейс; он показывает, возможно ли изменять объекты и окружающую среду во время выполнения программы в среде разработки. Графический пользовательский интерфейс не включает в себя окно, которое открыто для изображения моделирования.

	Player/Stage	**Gazebo**	**ROS**	**Simbad**	**CARMEN**
Операционная Система	Linux, Mac, Windows	Linux	Linux, Mac, Windows	Linux, Mac, Windows	Linux
Тип симулятора	2D	3D	2D, 3D	3D	2D
Язык Программи-рования	Player(any), Stage(C, C++, Python, Java)	C, C++, Python, Java	C++, Python, Octave, LISP, Java, Lua	Java	C, Java
Документация	Низко-	Низко-	Высоко-	Высоко-	Низко-

	уровневая	уровневая	уровневая	уровневая	Уровневая
Учебное руководство	Да	Да	Да	Ограниченная	Ограниченная
Переносимость	Да	Да	Да	Ограниченная	Да
Датчики	Одометрия, датчики расстояния	Одометрия, камеры, датчики расстояния	Одометрия, камеры, датчики расстояния	Камеры, датчики расстояния, датчики касания	Одометрия, датчики расстояния, GPS
Отладка/ Логирование	Да	Да	Нет	Да	Да
Графический пользовательский интерфейс	Нет	Нет	Нет	Нет	Нет
Тип симулятора	3D	3D	3D	3D	2D, 3D
Язык Программи-рования	C, C++, Java	VPL, C#, Visual Basic, JScript, IronPython	C, C++, Java, Matlab, Python	C++	C
Документация	Высоко-уровневая	Высоко-уровневая	Высоко-уровневая	Высоко-уровневая	Высоко-уровневая
Учебное руководство	Ограниченная	Да	Да	Да	Ограниченная
Переносимость	Да (используя Player)	Да	Да	Нет	Нет
Датчики	Одометрия, датчики расстояния, камеры, датчики касания	Одометрия, датчики расстояния, камеры	Одометрия, Датчики расстояния, камеры, GPS	Одометрия	Одометрия, датчики расстояния
Отладка/ Логирование	Да	Да	Да	Да	Да
Графический пользовательский интерфейс	Да	Да	Да	Да	Да

3. Архитектура

В этом разделе, представлено подробное описание архитектуры системы и главных особенностей двух наиболее популярных программных симуляторов: Player /Stage /Gazebo и ROS. Эти два симулятора фактически демонстрируют уникальные особенности, такие как точное моделирование 3D пространства окружающей среды, а также взаимодействие с реальными роботами и датчиками. Реализованы пошаговые инструкции, как пользоваться Player /Stage /Gazebo и ROS, которые будут полезны студентам и исследователям.

3.1. Player/Stage/Gazebo

Как было упомянуто в разделе 2.2 Player это аппаратно-независимый сервер, обеспечивающий управление роботом через сеть по протоколу TCP/IP. Клиенты (например, программа управления) могут быть написаны на любом языке программирования, который поддерживает TCP/IP, и могут подключаться к любому устройству (дальномер, лазер) через Player.

Player создаёт соединения между клиентами и устройствами, основанное на конфигурационном файле, которое содержит информацию, какие драйверы должны быть созданы и как связать их с устройствами. Несмотря на то что, клиенты получают данные по умолчанию с частотой 10Hz, скорость передачи данных можно увеличить или уменьшить для соответствия устройству. Несколько клиентов могут подписываться к одному и тому же устройству и могут посылать одновременно команды (рис. 1). Однако, в связи отсутствия очереди, старые команды будут пренебрегаться. Клиенты должны иметь соответствующее разрешение на отправку или получение информации, так как Player относится к устройствам как к файлам.

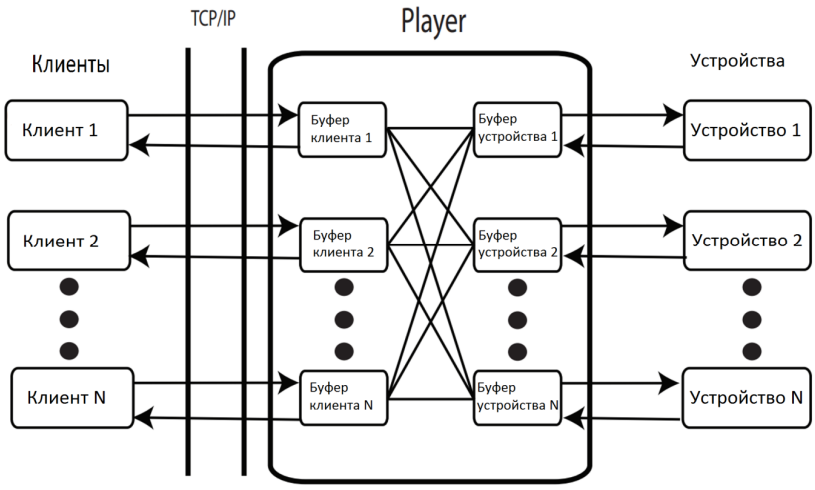

Рис. 1. Диаграмма показывает связь между клиентами и устройствами в Player

Здесь Stage будет описан, как плагин для Player, несмотря на то, что он может быть использован двумя другими способами; как самостоятельная программа (в которой пользователь предоставляет библиотеку) и как часть собственной программы пользователя. Stage осуществляет алгоритм детектора блобов и поддерживает различные типы сенсоров (лазеры, сонары, датчики положения). Stage также позволяет моделировать несуществующие датчики, такие как лазеры, измеряющие сквозь стену.

Параметры окружающей среды, платформа робота и сенсоры, моделируемые в Stage, определены в файле worldfile. Среда визуализации при моделировании использует изображения формата PNG в качестве входных данных, которые изображают препятствия в двухцветном (чёрном, белом) виде сверху.

Моделирование с помощью Gazebo определено в файле worldfile типа XML. Как упомянуто в пункте 2.2, Gazebo рендерит 3D модели с помощью библиотек Object-Oriented Graphics Rendering Engine (OGRE) и Open Dynamics Engine (ODE). ODE предназначен для моделирования динамики твёрдого тела и включает в себя такие функции как соединения, обнаружение столкновений, инерционность. OGRE рендерит модели роботов, окружающую среду и объекты в моделировании. Модели роботов сделаны из твёрдых тел и соединений. Жёсткие тела создаются с помощью комбинирования примитивов, таких как куб, сфера или цилиндр, каждый из которых имеет массу, трение, цвет и текстуру. Соединения соединяют тела вместе и образуют связь между ними. Существует несколько типов соединения, таких как жёсткое соединение, универсальный шарнир, шаровой шарнир, цилиндрическийшарнир. Рис. 2 показывает отношения между ODE, OGRE 3D, примитивами, соединениями и окружающей средой.

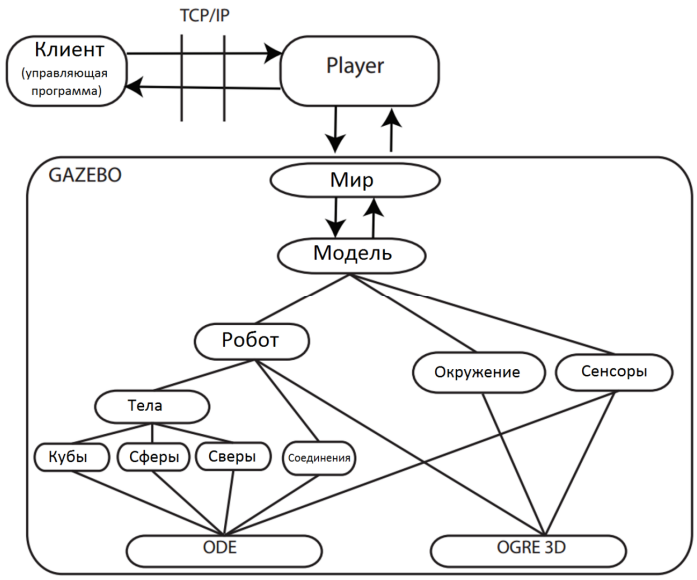

Рис. 2. Диаграмма показывает связь между Клиентом, Player и Gazebo

Player, Stage и Gazebo поддерживают алгоритмы, такие как детектор блобов, методы локализации и построения карты. Благодаря абстракции, используемой в Player, управляющий код может быть использован как на симуляторе, так и на реальном роботе. Вычислительная сложность Gazebo во время моделирования динамики твёрдого тела и 3Dсреды может потребовать больших вычислительных ресурсов, поэтому это ограничивает количество роботов, моделируемых одновременно.

3.2. ROS

Robot Operating System (ROS) обеспечивает стандартные службы операционной системы: аппаратную абстракцию, низкоуровневый контроль устройств, реализацию часто используемых функций, передачу сообщений между процессами, управление пакетами. ROS основан на узлах, сообщениях, темах и сервисах. Узлы - это процессы или программные модули в управляющем коде(например, узел камеры может обрабатывать все визуальные данные). Узлы могут связываться с другими узлами с помощью передачи сообщений. Узлы публикуют или подписываются к одному или нескольким топикам. ROS поддерживает TCP/IP и UDP для передачи сообщений; специальный тип сообщений, называемый сервисом, состоит из пары сообщений, один для запроса, и другой для ответа. Мастер ROS отслеживает все сервисы и темы; он также обеспечивает регистрацию узла и сервер параметров, который позволяет узлам хранить и извлекать параметры.

ROS имеет несколько клиентских библиотек, таких как rosccp (библиотека C++), rospy (библиотека Python), rosoct (библиотека Octave), roslisp (библиотека LISP), rosjava (библиотека Java), и roslua (библиотека Lua). Клиентские библиотеки объединены в пакеты. Пакеты (организованные программные модули) содержат узлы ROS, независимые от ROS библиотеки, наборы данных, файлы конфигурации, стороннее программное обеспечение или другие полезные модули.

Инструменты ROS позволяют легко использовать пакеты и стеки, и разделены на три категории: файловая систем, командная строка, и логирование. Rospack, roscd, rosmake, androscreate - это общие инструменты файловой системы, которые позволяют пользователю выполнять задачив файловой системе ROS. Инструменты командной строки, такие как roscore, rosrun, roslaunch и rosservice позволяют пользователю выполнять, инициализировать или получать подробную информацию о узлах и сервисах. Инструменты логирования позволяют пользователю отлаживать пакеты и узлы ROS, например rosbag записывает и воспроизводит сообщения из темы.Также имеются несколько команд для визуализации контента, хранимого с помощью команды rosbag,

такие как rxbag и rxbagplugins. Rviz - это среда 3D визуализации для роботов с использованием ROS. Rviz имеет пользовательский графический интерфейс, который позволяет пользователю настраивать и изменять объекты и окружающую среду.

Преимущество ROS в том, что программный код может быть использован повторно, и им можно поделиться с другими исследователями. Обмен программным кодом позволяет всемиметь одну общую базу и помогает распространять и тестировать её. С помощью использования инструментов ROS, реализация кода управления становиться эффективной, и упрощаются некоторые сложные задачи, необходимые для выполнения кода, но это создаёт зависимость от ROS и уменьшает мобильность кода. ROS может подключиться к реальным роботам и использовать смоделированные роботы в Gazebo, Stage или Rviz. Однако, программный код, написанный дляStage и Gazebo под управление Player, необходимо будет модифицировать для работы в Gazebo/Stage под управлением ROS.

4. Выводы

В этой статье был представлен подробный обзор и сравнение десяти коммерческих и открытых программных комплексов для моделирования робототехнических систем. Вследствие ограниченности объёма статьи, было представлено углубленное описание архитектуры только двух программных комплексов. Эта статья позволит начинающим исследователям получить всесторонние знания о доступных программных комплексах для моделирования.

Список литературы

1. Bruyninckx H. Robotics software: The future should be open // IEEE Robotics Automation Magazine. 2008. № 15. P. 9 –11.
2. Webots: 3D simulation. Available at: http://www.cyberbotics.com/overview, accessed 16.05.2015.
3. MathWorks. Available at: http://www.mathworks.com/products/robotics, accessed 16.05.2015
4. Player/Stage/Gazebo. Available at: http://playerstage.sourceforge.net, accessed 16.05.2015.
5. Robot Operating System. Available at: http://www.ros.org/wiki, accessed 16.05.2015.
6. Simbad: Java 3D simulator. Available at: http://simbad.sourceforge.net, accessed 16.05.2015.

7. Carnegie Mellon Robot Navigation Toolkit. Available at: http://carmen.sourceforge.net, accessed 16.05.2015.
8. Unified System for Automation and Robot Simulation. Available at: http://usarsim.sourceforge.net/wiki/index.php, accessed 16.05.2015.
9. Microsoft Robotics Developer Studio. Available at: http://www.mrds.ru, accessed 16.05.2015.