

# ОПИСАНИЕ УСТРОЙСТВА ПОЛИГОНА ДЛЯ ИССЛЕДОВАНИЯ АЛГОРИТМОВ УПРАВЛЕНИЯ ГРУППАМИ РОБОТОВ

*В работе описана концепция устройства испытательного полигона для тестирования алгоритмов группового управления мобильными роботами. Испытательный полигон является работой в процессе и будет дорабатываться и изменяться.*

## ВВЕДЕНИЕ

Наличие собственного испытательного полигона позволяет значительно ускорить разработку и проверку алгоритмов управления групп мобильных роботов. Непосредственный функционал испытательных полигонов зависит от типа испытываемых на нем роботов. В нашем случае полигон должен соответствовать следующим требованиям:

- возможность работать с несколькими роботами одновременно;
- возможность автоматического дистанционного обновления программного обеспечения роботов;
- возможность агрегации телеметрии роботов в течение эксперимента для последующего анализа;
- возможность ручного и автоматического проведения экспериментов;
- возможность автоматического проведения серии последовательных экспериментов;

## I. КОНЦЕПЦИЯ

Испытательный полигон будет состоять из двух уровней:

- верхнего или системного;
- нижнего или локального;

Верхний уровень состоит из обязательного сервера и возможных датчиков окружения. В качестве сервера может выступать любой компьютер с операционной системой Linux. Датчики могут иметь разные предназначения, например, камера, подвешенная над полигоном для упрощения локализации и навигации[1], датчики измерения расстояния расположенные, в разных точках полигона и так далее. Необходимость использования датчиков зависит от конкретного проводимого эксперимента. Датчики относятся к верхнему уровню, потому что непосредственно участвуют в оценке эксперимента и обмениваются информацией напрямую с сервером.

Нижний уровень состоит из полигона с препятствиями, роботов и других объектов, с которыми роботы взаимодействуют в процессе тестирования. В нашем случае полигон представляет из себя набор стенок и соединительных элементов, изготовленных из фанеры лазерной резкой. Большое количество элементов позволяет составлять различные конфигурации полигона для проведения различных тестов. Примерами дополнительных объектов могут служить цветные коробки или шары, часто используемые в соревнованиях роботов.

## II. ПРИНЦИПЫ РАБОТЫ

Сервер системы работает на операционной системе Linux. На сервере работают узлы robot operating system (ROS) и набор скриптов, помогающих в автоматизации процессов. Основным языком разработки тестов и автоматизации в системе является python. При необходимости сервер опрашивает внешние датчики и анализирует их показания.

Основным интерфейсом связи в системе является Wi-fi. Сервер и тестируемые роботы должны быть объединены в единую локальную сеть. К этой же сети могут быть подключены внешние датчики. Для функционирования тестов не требуется интернет, за исключением экспериментов, в которых роботы или сервер должны напрямую взаимодействовать с интернетом.

Все тестируемые роботы должны иметь возможность подключения к точке доступа Wi-fi. Кроме этого, в зависимости от проводимого эксперимента, по Wi-fi могут передаваться команды управления. Система позволяет проводить тесты распределенных и независимых групп роботов, то есть не использовать сервер как пункт управления. В таких случаях на сервер передается только телеметрия, а с сервера поступают только команды о начале и завершении эксперимента.

В разрабатываемой системе предусмотрено два возможных способа обмена данными между роботами и сервером:

1. с использованием ROS;
2. с использованием WebSocket;

Для использования подключения через ROS необходимо, чтобы на работе присутствовал

любой компьютер, способный работать на Linux с установленным ROS. Такой подход позволяет тестировать на испытательном полигоне широко распространенных роботов, таких как TurtleBot, JetBot и других.

Подключение с использованием WebSocket предназначено специально для микроконтроллеров с доступом к Wi-fi, например ESP32. На микроконтроллерах невозможен запуск полноценной операционной системы, а тем более ROS. WebSocket позволяет обеспечить большую скорость передачи данных между роботами и сервером, что позволяет собирать данные о телеметрии даже при централизованном управлении.

### III. ПОДГОТОВКА ЭКСПЕРИМЕНТА

Все программное обеспечение полигона можно разделить на две группы:

1. системное;
2. пользовательское;

Системное программное обеспечение не должно изменяться при подготовке и проведении эксперимента. К системному программному обеспечению относятся: логирование данных с роботов, отправка, прием и обработка сообщений между роботами и сервером.

Пользовательское программное обеспечение должно разрабатываться отдельно для каждого проводимого эксперимента. К пользовательскому программному обеспечению относятся: системы управления роботами, обработка и анализ результатов эксперимента, формирование сообщений и так далее.

Минимальный эксперимент в системе выглядит следующим образом:

- робот подключается к серверу;
- пользователь или сервер запускает эксперимент;
- робот получает команду о начале эксперимента и выполняет запрограммированные в нем действия;
- во время эксперимента робот отправляет сообщение о том, что эксперимент в процессе и телеметрию на сервер как минимум 2 раза в секунду;
- робот отправляет на сервер команду о завершении эксперимента;

- сервер сохраняет все собранные в процессе эксперимента данные в отдельную папку;

Все сообщения в системе передаются в формате JSON. Самым простым сообщением о том, что эксперимент в процессе, является отправка одного единственного поля "expStatus" равным "1". Как видно из описания минимального эксперимента, робот не обязан отправлять какую-либо телеметрию на сервер. Точно также, сервер не обязан выбирать момент завершения эксперимента.

В текущей реализации сервер сохраняет все полученные в сообщении JSON поля в бинарный файл вне зависимости от пользовательского программного обеспечения. Пользовательское ПО может обрабатывать эти же сообщения в реальном времени.

Сервер перестает сохранять новые сообщения в файл после получения кода завершения эксперимента - поле "expStatus" равно "0". Сервер создает новый файл эксперимента после отправки или получения кода старта эксперимента: поля "expStatus" равного "1".

### IV. ВЫВОДЫ

Описанные в статье принципы организации тестового полигона для роботов могут быть полезны для разработки других подобных систем. Используемый метод обмена данными очень прост, но позволяет использовать большое количество методов анализа и управления в реальном времени без необходимости в изменении системного ПО и ПО самих роботов, что делает систему гибкой и удобной в использовании. Простая методология запуска и остановки эксперимента позволяет легко автоматизировать процесс тестирования, а наличие логов позволяет удобно анализировать данные экспериментов и сравнивать их между собой.

Проблемой предложенной реализации является необходимость разработки большого количества пользовательского программного обеспечения, что может быть частично исправлено в будущем посредством новых модулей.

Описанный в статье функционал является очень простым, но будет значительно расширен в будущем.

1. Т. Бройнль. Встраиваемые роботизированные системы. Институт компьютерных исследований. Москва 2012.

*Довнар Андрей Дмитриевич*, аспирант кафедры систем управления БГУИР, dovnar@bsuir.by.  
*Научный руководитель: Захарьев Вадим Анатольевич*, доцент кафедры систем управления, zahariev@bsuir.by.