

компьютере специальной программы для работы с клиентом. Важной стороной компьютерного тестирования является возможность протоколирования для каждого клиента и каждого прохождения им теста с указанием подробной информации о пользователе, о качестве его ответов на заданные вопросы и итога тестирования.

Для реализации разрабатываемого веб-приложения мы воспользуемся средствами программной платформы Node.js, расширяющей узкоспециализированный язык JavaScript до событийно ориентированного языка общего назначения. Каркасом приложения избрано такое средство, как фреймворк Express.js. Планируется использование внешней базы данных для хранения тестов, реквизитов авторизации пользователей, результатов прохождения тестов и статистических данных.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Силаев, Н. В. О системе теоретического тестирования / Н. В. Силаев, А. А. Харитонюк, И. Г. Машлякевич // Материалы V международной научно-практической Интернет-конференции «Инновационные технологии обучения физико-математическим дисциплинам», Мозырь, 26–29 марта 2013 г. – Мозырь, 2013. – С. 69.

С.В. Слоневский, А.А. Козинский
Беларусь, Брест, БрГУ имени А.С. Пушкина

ВИЗУАЛИЗАЦИЯ ТРАССЫ ДЛЯ АВТОНОМНОГО РОБОТА СРЕДСТВАМИ БИБЛИОТЕК PYTHON

В процессе работы над проектом, целью которого является создание системы управления автономным роботом, возникла задача визуализации показаний датчиков и построения траектории трассы. После анализа возможных решений в качестве средства реализации определена библиотека языка python matplotlib. Данный выбор объясняется тем, что matplotlib является гибким, легко конфигурируемым пакетом. Пакет поддерживает построение графиков и диаграмм.

Результатом проектирования системы визуализации стало приложение, которое отображает показания всех датчиков подобных Sharp GP2Y0A02 в виде графиков зависимости их показаний от времени. Пример визуализации см. на рисунке 1.

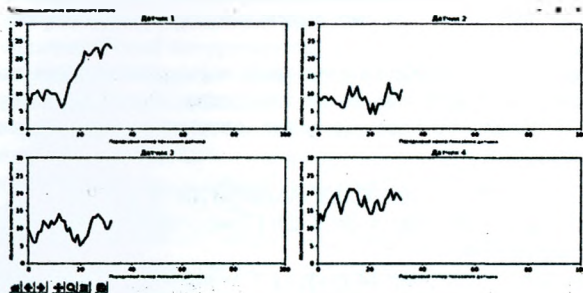


Рисунок 1 – Показания датчиков автономного робота

На указанном рисунке показаны четыре графика, каждый из которых соответствует показаниям соответствующего датчика расстояния. Для каждого графика на горизонтальной оси отмечены порядковые номера показаний датчика, на вертикальной оси – абсолютные значения показаний датчиков в условных единицах длины.

На рисунке 2 показано расположение препятствий (границ трассы) относительно автономного робота в текущий момент времени.

Каждый датчик определяет расстояние до физического объекта – препятствия, расположенного перед ним. По показаниям четырех датчиков расстояний можно определить точное расположение барьеров на трассе. Для решения задачи зафиксируем начало отсчета в некоторой точке на роботе, находящейся между датчиками. Для каждого датчика координаты точек x , y препятствия находят по формуле $(r \cos(a), r \sin(a))$, где r – расстояние, определенное по показаниям датчика, a – угол поворота датчика относительно оси OX . Соединив точки с вычисленными координатами x , y можем получить вид препятствий, показанный на рисунке 2.

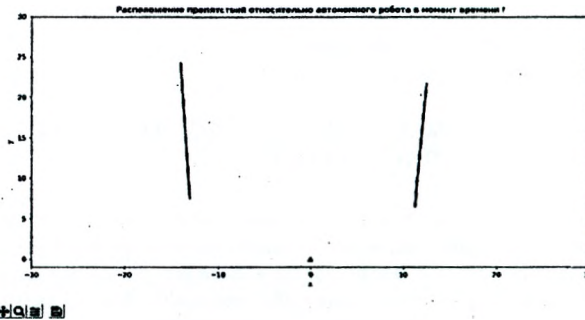


Рисунок 2 – Расположение препятствий относительно автономного робота в момент времени t

Следует заметить, что положения стенок требует динамического пересчета. Наличие ошибок в показаниях датчиков требует дополнительной фильтрации данных. Однако предложенный метод с высокой степенью точности решает задачу визуализации препятствий.

Результатом полной обработки полученной информации является план трассы для прохождения роботом. Полученные данные планируется использовать для решения задач машинного обучения, которое позволит оптимизировать движения робота на произвольной трассе.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Официальный сайт пакета matplotlib [Электронный ресурс]. – Режим доступа: <https://matplotlib.org/index.html>.
2. Matplotlib. [Электронный ресурс] // Wikipedia. – Режим доступа: <https://ru.wikipedia.org/wiki/Matplotlib>.