

*С. К. Пелагей, О. В. Толстель, Д. В. Михеенко*

## ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ УПРАВЛЕНИЯ ДВИЖЕНИЕМ АВТОМАТИЧЕСКОГО ДВИЖУЩЕГОСЯ СРЕДСТВА НА БАЗЕ ТЕЛЕЖКИ AGV

Поступила в редакцию 12.08.2021 г.

Рецензия от 06.09.2021 г.

14

*Для модернизации производственных линий по сборке автомобилей рассматривается возможность применения автоматических транспортных средств. В данной статье описывается процесс разработки программно-аппаратного комплекса для управления движением такого транспортного средства.*

*To modernize car assembly lines, the possibility of using automatic vehicles is considered. This article describes the process of developing a software and hardware complex for controlling the movement of such a vehicle.*

**Ключевые слова:** автоматически управляемое транспортное средство, AGV, математическая модель, ROS, Gazebo, Raspberry Pi

**Keywords:** automated guided vehicle, AGV, mathematical model, ROS, Gazebo, Raspberry Pi

ООО «Автотор Холдинг» является одним из крупнейших российских частных предприятий, специализирующихся на сборке автомобилей. В настоящее время предприятие рассматривает возможность модернизации своих производственных линий за счет введения автоматически управляемых тележек, выступающих в роли мобильного конвейера.

Автоматически управляемые транспортные средства, или AGV (от англ. Automated Guided Vehicles), представляют собой широкий спектр мобильных роботов, предназначенных для транспортировки грузов в условиях производства, в складских помещениях, в медицинской сфере и пр. Эти роботы оснащены электронными органами чувств, позволяющими им ориентироваться в пространстве, следовать назначенному маршруту, а также избегать препятствий [1].

Наиболее распространенными способами навигации AGV являются:

– навигация с использованием магнитной или цветной линии (рис. 1),

– навигация с использованием отражающих точек,

– лазерная навигация, например с использованием технологии

Лидар,

– навигация с использованием камер,

– навигация по проводу, передающему радиосигнал.



Рис. 1. AGV на магнитной линии

15

По способам управления чаще всего встречаются следующие конфигурации (рис. 2):

- дифференциальный привод, представляющий собой два фиксированных двигателя, соединенных с колесами. Поворот платформы определяется разностью скоростей обоих колес;
- трехколесный привод, в котором одно управляемое колесо отвечает за линейную и угловую скорости, а остальные (минимум два) пассивных колеса выступают в роли поддержки;
- четырехколесный привод, в котором несколько управляемых колес позволяют платформе совершать поворот на месте.

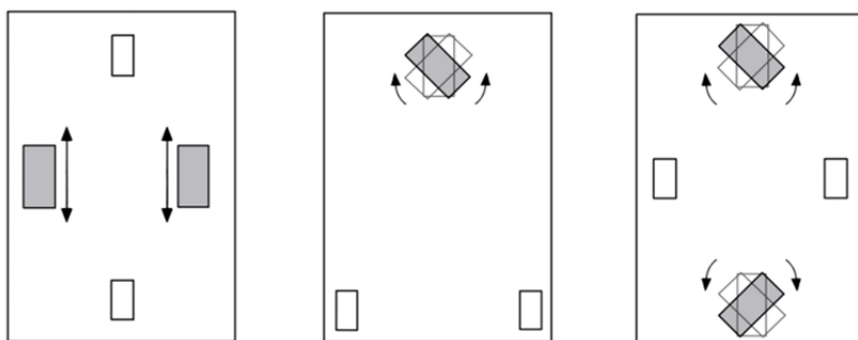


Рис. 2. Конфигурации AGV

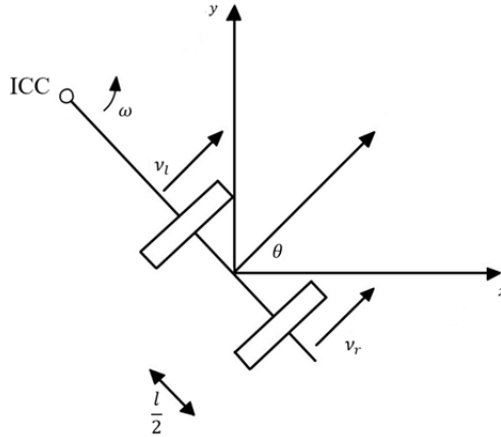
В данной статье описывается процесс разработки программно-аппаратного комплекса управления движением автоматически управляемого транспортного средства, представленного в дифференциальной конфигурации и управляемого посредством направляющей линии.

Для достижения этой цели необходимо решить следующие задачи:

- 1) рассмотреть математическую модель транспортного средства;
- 2) разработать структурную схему транспортного средства;
- 3) построить модель транспортного средства;
- 4) разработать программное обеспечение для осуществления движения транспортного средства.



Кинематика дифференциального привода строится вокруг мгновенного центра кривизны (ICC, instantaneous center of curvature) мобильного робота. Это точка в двумерном пространстве, вокруг которой осуществляется поворот всей платформы (рис. 3).



16

Рис. 3. Положение робота  $x, y, \theta$  в глобальной системе координат

Линейная скорость мобильного робота описывается как

$$v = \omega R,$$

где  $\omega$  — угловая скорость платформы,  $R$  — расстояние от точки ICC до точки между колесами транспортного средства.

Поскольку угловая скорость вокруг точки ICC должна быть одинаковой для обоих колес, мы можем записать уравнение линейной скорости для левого и правого колес:

$$v_r = \omega \left( R + \frac{l}{2} \right); \quad v_l = \omega \left( R - \frac{l}{2} \right),$$

где  $l$  — расстояние между колесами.

Тогда

$$R = \frac{l (v_r + v_l)}{2 (v_r - v_l)},$$

$$\omega = \frac{v_r - v_l}{l}.$$

Зная конечные линейную и угловую скорости платформы, можем вычислить соответствующие линейные скорости обоих колес:

$$v_r = \frac{2v + \omega l}{2}, \quad v_l = \frac{2v - \omega l}{2}.$$

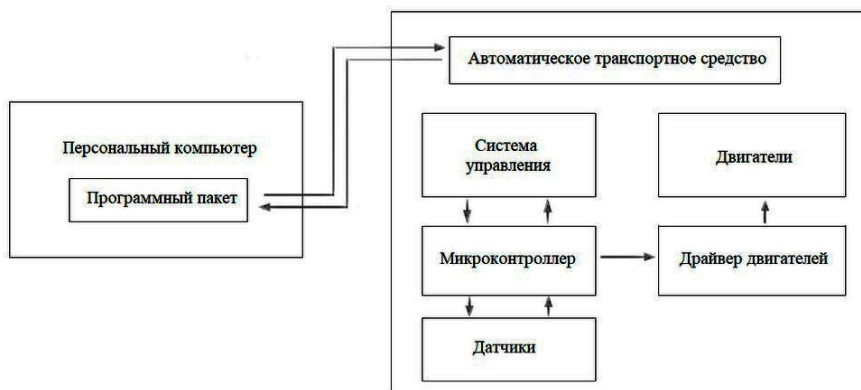


Угловые скорости колес будут равны

$$\omega_r = \frac{V_r}{R}, \quad \omega_l = \frac{V_l}{R}.$$

В данном случае  $R$  — радиус колеса.

Структурная схема (рис. 4) предполагает разделение системы управления на отдельный компьютер, выполняющий требуемые вычисления, и само транспортное средство. В случае необходимости эти части могут быть объединены в автономную единицу.



17

Рис. 4. Структурная схема

Для обеспечения контроля за транспортными средствами центральный компьютер должен получать информацию о состоянии транспортного средства, о пройденном им пути и о прохождении узловой точки.

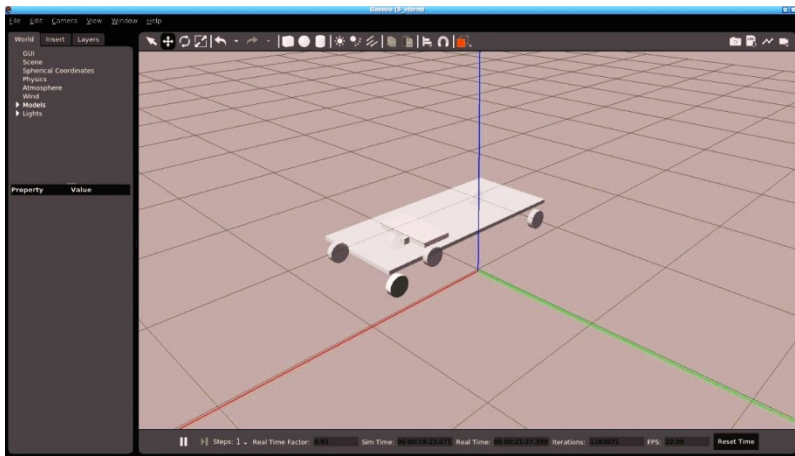
В качестве основы системы управления был выбран фреймворк ROS — Robot Operating System, предназначенный для написания программного обеспечения роботов [2].

Симуляция робота производилась в программе Gazebo, тесно работающей с ROS.

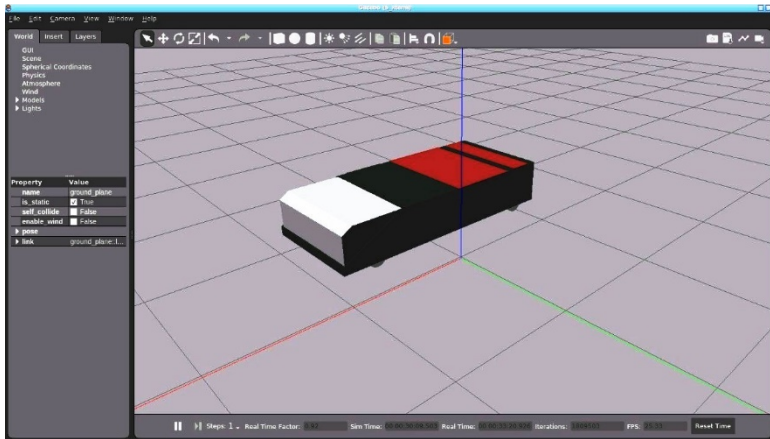
На основе структурной схемы и математической модели была создана трехмерная модель мобильного робота при помощи универсального формата описания роботов URDF (Universal Robot Description Format). Он позволяет описывать кинематику, некоторые части динамики и другие части данных, связанных с роботом, в формате, понятном программой симуляции (рис. 5, а).

Визуально формат URDF позволяет представлять робота в виде геометрических примитивов, а также подключать трехмерные модели в формате .dae (рис. 5, б).

Кроме симуляции робота, Gazebo представляет возможность симулировать окружение, что было использовано для создания короткого трека с направляющей линией, поворачивающейся под прямым углом в нескольких местах.



а



б

Рис. 5. Модель AGV в Gazebo:  
а – рама с шасси; б – полная трехмерная модель

Для обеспечения обратной связи о местоположении транспортного средства на треке были размещены метки, информация о достижении которых передается на центральный компьютер. Зная расположение этих меток и получая информацию о пройденном пути с энкодеров, можно точно вычислить положение аппарата на карте.

После создания модели транспортного средства были рассмотрены различные способы следования направляющей линии.

Первым способом было выбрано движение по оптическим датчикам (рис. 6). Обычно эту задачу осуществляют при помощи отдельных инфракрасных датчиков или массивов из них. Для симуляции работы такого датчика к модели был подключен плагин камеры. Затем с помощью библиотеки компьютерного зрения OpenCV получалась информация о нескольких точках в зоне видимости камеры. Для наглядности



точки выделяются окружностями. Gazebo позволяет симулировать непостоянство освещения, поэтому значения цвета точек постоянно колеблются. Для коррекции снятия данных с датчиков в программе была учтена ошибка, которая в среднем составляет 20 %.

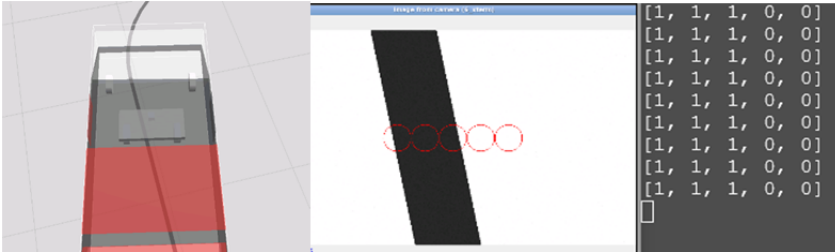


Рис. 6. Поиск направляющей линии по оптическим датчикам

После получения данных с датчиков вычисляется ошибка отклонения от центра.

$$error = \frac{\sum_{i=0}^{n-1} val[i] \left( i - \frac{n-1}{2} \right)}{\sum_{i=0}^{n-1} val[i]}.$$

Она выбрана таким образом, чтобы при нахождении сенсора строго над линией выдавать значение 0. Значение *error* выступает затем в роли ошибки отклонения ПИД-регулятора, который участвует в стабилизации движения. Далее в зависимости от полученных данных программа отправляет линейную и угловую скорость в тему ROS, ответственную за контроль скоростей, что заставляет колесного робота повернуться в нужном направлении, следуя заданной линии.

Другим способом поиска направляющей линии был выбран поиск цветового пятна на основании моментов изображения (рис. 7). Здесь в качестве ошибки выступает отклонение центроида от середины изображения. Последующие шаги совпадают с первым способом следования линии после вычисления ошибки.

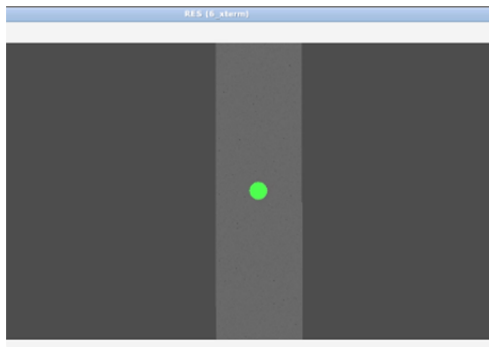


Рис. 7. Поиск направляющей линии по цветовому пятну



Еще одной важной составляющей любого автономного робота является обеспечение безопасности. Для этого в систему управления был добавлен датчик, который позволяет определить расстояние до объекта. Расстояние перед платформой делится на зоны, в которых в зависимости от расстояния до объекта происходит плавное замедление или экстренная остановка (рис. 8).

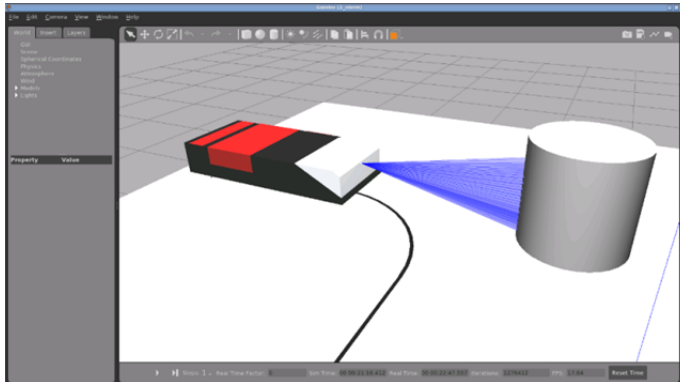


Рис. 8. Работа системы безопасности

После завершения этапа моделирования был произведен выбор оборудования для создания реального образца в миниатюре. Основой системы стал одноплатный компьютер Raspberry Pi 2B, контроллером выступил ATmega 328p на базе платы Arduino Uno. Raspberry Pi связывается с персональным компьютером посредством сетевого протокола SSH через Wi-Fi по принципу master-slave, предоставляемому ROS. Это позволило переложить сложные для относительно слабого Raspberry Pi 2B вычисления на более производительный компьютер. Использование последних моделей Raspberry Pi даст возможность производить все вычисления на самом мобильном роботе.

Связь между компьютером робота и контроллером осуществлялась посредством последовательного порта с использованием библиотеки `rosserial_arduino`, позволяющей реальным устройствам робота публиковать и подписываться на темы ROS.

В качестве инструмента поиска направляющей линии была взята USB-камера. После создания модели мы протестировали все алгоритмы следования линией.

Для определения положения платформы в пространстве использовались RFID-модуль RC522, передающий информацию о достижении RFID-метки, и оптический энкодер, установленный на колесо робота и позволяющий оценить путь, пройденный от метки.

В целях предотвращения коллизий был использован инфракрасный дальномер Sharp 2Y0A21 (рис. 9). Проблема ограниченности угла обзора датчика была решена его установкой на сервопривод SG90. Для получения данных об угле поворота была произведена модификация



сервопривода, благодаря чему появилась возможность использовать данные встроенного в SG90 потенциометра. Среди полученных данных о расстоянии до объекта выбиралось наименьшее.

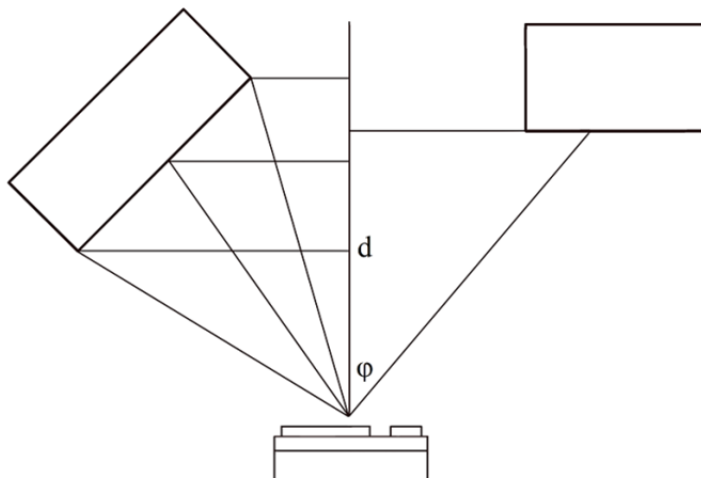


Рис. 9. Схема работы инфракрасного дальномера

Использование реальной модели позволило найти недочеты в работе системы безопасности, которые нельзя было отловить в симуляции. Так, в работе контроллера применялись функции, блокирующие работу программы на время их исполнения. Код таких функций был изменен с использованием прерываний и внутренних таймеров ATmega 328p.

#### Список литературы

1. Dudek G., Jenkin M. Computational principles of mobile robotics. Cambridge, 2010.
2. Quigley M., Gerkey B., Smart W.D. Programming robots with ROS: A practical introduction to the robot operating system. Sebastopol, 2015.

#### Об авторах

Сергей Константинович Пелагей — магистрант, Балтийский федеральный университет им. И. Канта, Россия.

E-mail: pelagey.sergey@outlook.com

Олег Владимирович Толстель — канд. техн. наук, доц., Балтийский федеральный университет им. И. Канта, Россия.

E-mail: tolstel.oleg@mail.ru

Дмитрий Викторович Михеенко — начальник отдела промышленной электроники ООО «Автотор Холдинг»

E-mail: tolstel.oleg@mail.ru





### **The authors**

Sergey K. Pelagey, Master's Student, Immanuel Kant Baltic Federal University, Russia.

E-mail: pelagey.sergey@outlook.com

Dr Oleg V. Tolstel, Associate Professor, Immanuel Kant Baltic Federal University, Russia.

E-mail: tolstel.oleg@mail.ru

Dmitriy V. Mikheenko, Head of the Industrial Electronics Department, Avtotor Holding.

E-mail: tolstel.oleg@mail.ru