

Техника и технология транспорта: научный Интернет-журнал <http://www.transport-kgasu.ru>
2019. № S13 http://transport-kgasu.ru/index.php?option=com_content&view=article&id=10&Itemid=2
URL статьи: <http://transport-kgasu.ru/files/N13-58RTS19.pdf>
Статья опубликована 09.09.2019

Ссылка для цитирования этой статьи:

Имамеев Д.Т., Лавренов Р.О., Сагитов А.Г., Шабалина К.С., Магид Е.А., Мартинес-Гарсия Э.А., Су К.-Л. Разработка процедуры автономной параллельной парковки мобильного робота Unior // Техника и технология транспорта. 2019. № S13. С. 58. URL: <http://transport-kgasu.ru/files/N13-58RTS19.pdf>

УДК 007.52, 519.878, 519.1, 004.942

РАЗРАБОТКА ПРОЦЕДУРЫ АВТОНОМНОЙ ПАРАЛЛЕЛЬНОЙ ПАРКОВКИ МОБИЛЬНОГО РОБОТА UNIOR

Имамеев Д.Т.¹, Лавренов Р.О.¹, Сагитов А.Г., Шабалина К.С.¹, Магид Е.А.¹,
Мартинес-Гарсия Э.А.², Су К.-Л.³

¹ Казанский (Приволжский) Федеральный Университет, г. Казань, Россия

² Автономный университет Сьюдад-Хуарес, г. Сьюдад-Хуарес, Мексика

³ Национальный университет науки и техники, г. Юньлинь, Тайвань

Аннотация

Эта работа нацелена на реализацию определения траектории движения и управления для робомобиля Unior при решении задачи параллельной парковки. Определение траектории движения основано на геометрическом подходе. Мы реализовали существующий алгоритм на модели робота Unior для симуляции и провели валидацию в симуляторе Gazebo.

Ключевые слова: автономная параллельная парковка, робомобиль Unior, Gazebo.

Параллельная парковка является сложной задачей как и для начинающих, так и для опытных водителей. Процесс парковки включает в себя следующие этапы: поиск парковочного места, совершение требуемых маневров, выполнение которых может привести к повреждениям транспортного средства в случае узких парковочных мест. Многие производители автомобилей разработали собственные решения, которые не предоставляют полной автоматизации при парковке (парковочные ассистенты). К примеру, Toyota Prius (Intelligent Parking Assist), Ford (Active Park Assist), BMW, Mercedes и т.д.

Данная работа рассматривает применение существующего алгоритма [1] на роботе Unior. Рассмотренное определение траектории и управление протестированы лишь на симуляционной модели робота Unior, но могут быть применены и на реальном роботе или автомобиле.

Мобильный робот Unior (рис. 1) разработан российской компанией Aurora Robotics. Робот спроектирован для обучения студентов мобильной робототехнике. Unior обладает 8 датчиками для парковки: 4 спереди и 4 сзади корпуса робомобиля. Также камера Microsoft Kinect установлена в передней части робота и лазерный сенсор Нокуо на верхней части. Линейные измерения робота Unior представлены в табл. 1. Робомобиль обладает двумя колёсами заднего привода и двумя передними колёсами поворота, которые представляют собой систему управления Аккермана.



Рис. 1. Робомобиль Unior

Таблица 1

Наименование	Размерность в миллиметрах
Диаметр обода	205
Диаметр колеса	260
Продольное расстояние между колёсами	700
Поперечное расстояние между колёсами	540
Высота робомобиля	570
Продольная длина всего робомобиля	1120
Ширина покрышек	104
Радиус покрышек	130
Ширина робомобиля	650

Для предотвращения столкновений и поломок была использована модель робомобиля Uniog для симуляции Gazebo [2] (рис. 2). Модель обладает всеми датчиками реального робота Uniog и полностью повторяет размеры реального робота.

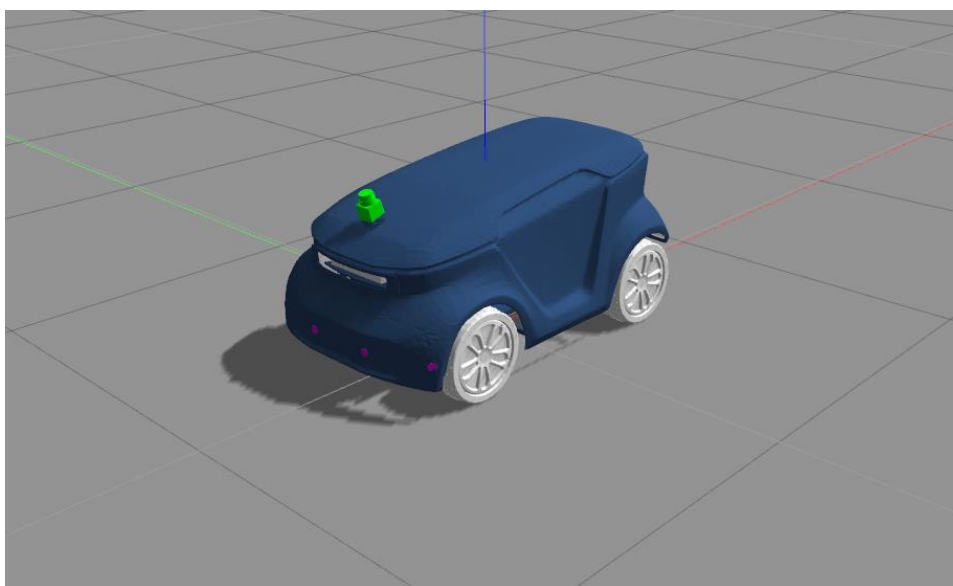


Рис. 1. Модель робота Uniog в симуляции Gazebo

Существует две проблемы относительно параллельной парковки робомобиля Uniog. Первая заключается в том, что робот Uniog является неголономным транспортным средством. Вторая – определение траектории движения.

Робот Uniog – четырёхколёсный мобильный робот с задними ведущими и передними для поворота колёсами. Робот обладает двумя переменными управления (θ – угол поворота рулевого колеса, v – продольная скорость), но его пространство конфигурации имеет размерность 3 (x, y, θ). Движение транспортного средства описывается следующими равенствами:

$$\begin{cases} \dot{x} = v \cos \phi \cos \theta, \\ \dot{y} = v \cos \phi \sin \theta, \\ \dot{\theta} = \frac{v}{L} \sin \phi, \end{cases} \quad (1)$$

Равенства (1) относятся к системе с неголономными ограничения, потому что они состоят из производных положения и ориентации транспортного средства на плоскости, которые являются неинтегрируемыми [1].

Для определения траектории был использован метод, основанный на геометрическом подходе [3]. Данный метод высчитывает всю траекторию. Хранение всех точек траектории является избыточным, поэтому модель робота Uniog хранит только ключевые точки (точки правого и левого поворотов). Вычисление ключевых точек траектории основывается на

геометрии транспортного средства: максимальный угол поворота (β), длина колёсной базы (e), длина переднего и заднего выступов (p), ширина (w) и длина (L) транспортного средства [3]. Траектория движения при параллельной парковке может быть определена, если длина парковочного места не меньше L_{min} [3], иначе потребуется более одной попытки (манёвр, состоящий из одного поворота в одну сторону и другого поворота в обратную) для парковки транспортного средства.

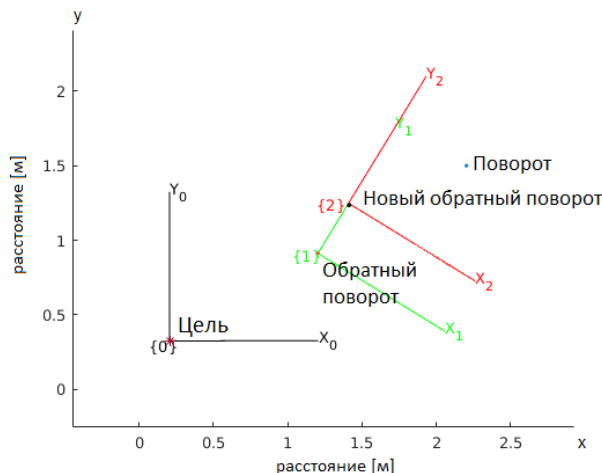


Рис. 3. Преобразование для вычисления новой точки второго поворота

В данной работе мы использовали реализацию алгоритма, которая была взята с репозитория [4]. Применение данной реализации к роботу Unioг имеет проблему из-за координат, отправляемых роботом. Робот отправляет координаты центра относительно всего робомобиля, когда алгоритм требует координаты середины задней оси.

Логика алгоритма не была изменена, но было изменено управление для симуляционной модели робомобиля Unior. Схема управления для параллельной парковки состоит из отслеживания достижения транспортным средством ключевых точек. Причиной изменения управления является отсутствие тормозной системы как у реального робота, так и у его модели. В этом случае, из-за инерции в симуляции Gazebo процесс остановки начинается заранее. Для создания новой схемы (см. приложение 1) управления требуется измерения времени, затрачиваемого на полную остановку и на полный поворот передних колёс. Все изменения в Python-скрипте затрагивают только вычисление ключевых точек. Для изменения координат первой точки поворота (поворот вправо) было добавлено смещение только по оси OX . Более сложные преобразования требуются для вычисления второй точки поворота (поворот влево), так как они включают в себя и поворот, и смещение.

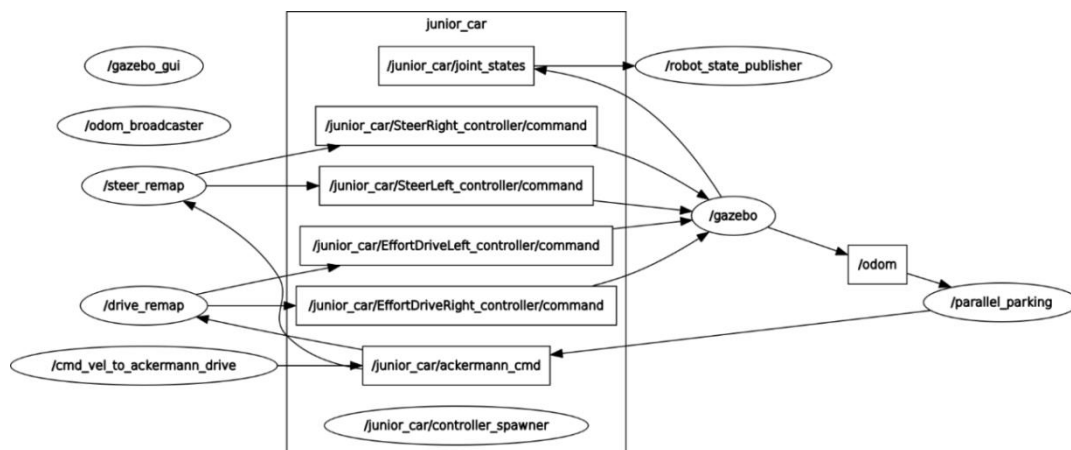


Рис. 4. Архитектура ROS модели робомобиля Unior
(из Робототехнической Операционной системы (ROS))

Для проверки верности математических выражений были созданы MATLAB-скрипты. Первый вычисляет точку второго поворота (обратного поворота) для центра относительно всего транспортного средства. Для вычисления новой точки второго поворота (обратного поворота) требуется система координат для уже существующей точки {1} (обратного поворота). Система координат с началом координат в точке изначальной точки второго поворота (обратного поворота) повернута на угол β (угол поворота транспортного средства) относительно глобальной системы координат {0} и смещена на $e/2$ (см. рис. 3). Второй скрипт вычисляет угол β на основе угла поворота.

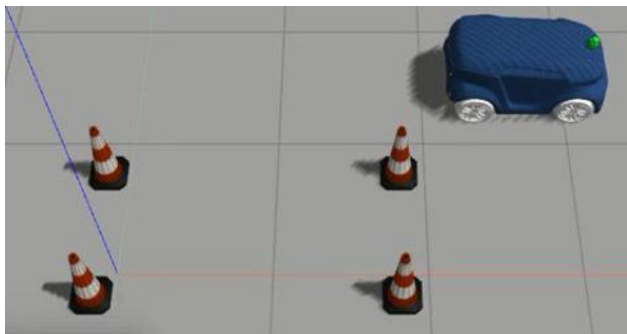


Рис. 5. Парковочное место и робот Uniор

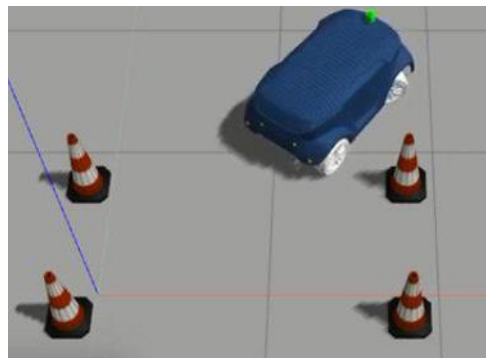


Рис. 6. Робот начинает движение к парковке

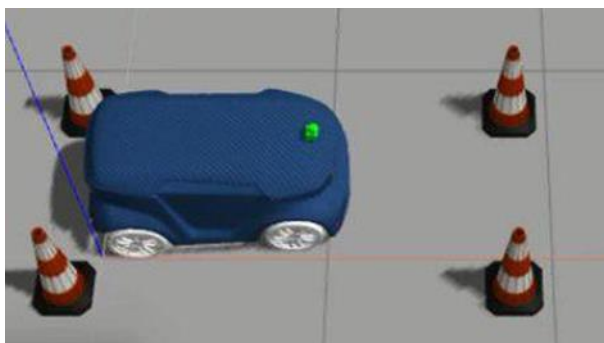


Рис. 7. Парковка завершена

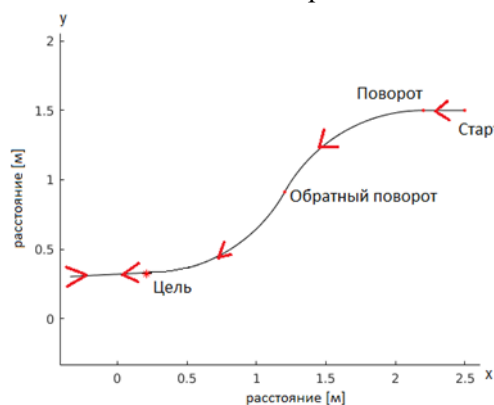


Рис. 8. Траектория движения модели робомобиля Uniор

Далее, Требовалось реализовать парковку робота в среде ROS\Gazebo [5]. Для вычисления времени требуемого для полной остановки был проделан эксперимент. ROS-сообщение типа *AckermannDriveStamped* с параметрами *speed* = 0 и *steering_angle* = 0 отправляются на топик '*junior_car/cmd*' (рис. 4) после движения на малой скорости. Во время остановки робомобиль Uniор сначала делает небольшой откат назад, а потом вперед. На полную остановку около 40 секунд. Для замера времени требуемого для полного поворота передних колёс модель робомобиля была перевернута. После этого ROS-сообщение типа *AckermannDriveStamped* с параметрами *speed* = 0 и *steering_angle* = β отправляются на топик '*junior_car/cmd*'. Для полного поворота требуется 3 секунды.

После внесённых изменений была проведена экспериментальная валидация. Валидация заключается в определении стартовой позиции транспортного средства и создания парковочного места вокруг точки парковки. В этом эксперименте точка парковки обладает координатами (0.206, 0.325) и стартовая позиция – (2.5, 1.5) (значения координат точек в метрах). Площадь парковочного места вычислена для точки парковки (относительно задней оси) с использованием $p = 0.206$ м, $w = 0.65$ м и $L_{min} = 1.7243$ м. После был создан мир с парковочным местом и роботом Uniор для симуляции Gazebo (см. рис. 5, 6, 7). Валидация обновлённой реализации заключается в проверке прохождения моделью робомобиля Uniор ключевых точек и точки парковки. Модель робота Uniор проходит через все нужные точки, но

траектория движения не идеальна (см. рисунок 8) из-за процесса остановки. Также наблюдается смещение вдоль оси OX на ~ 0.06 м и вдоль оси OY на ~ 0.013 м относительно точки парковки.

Выводы

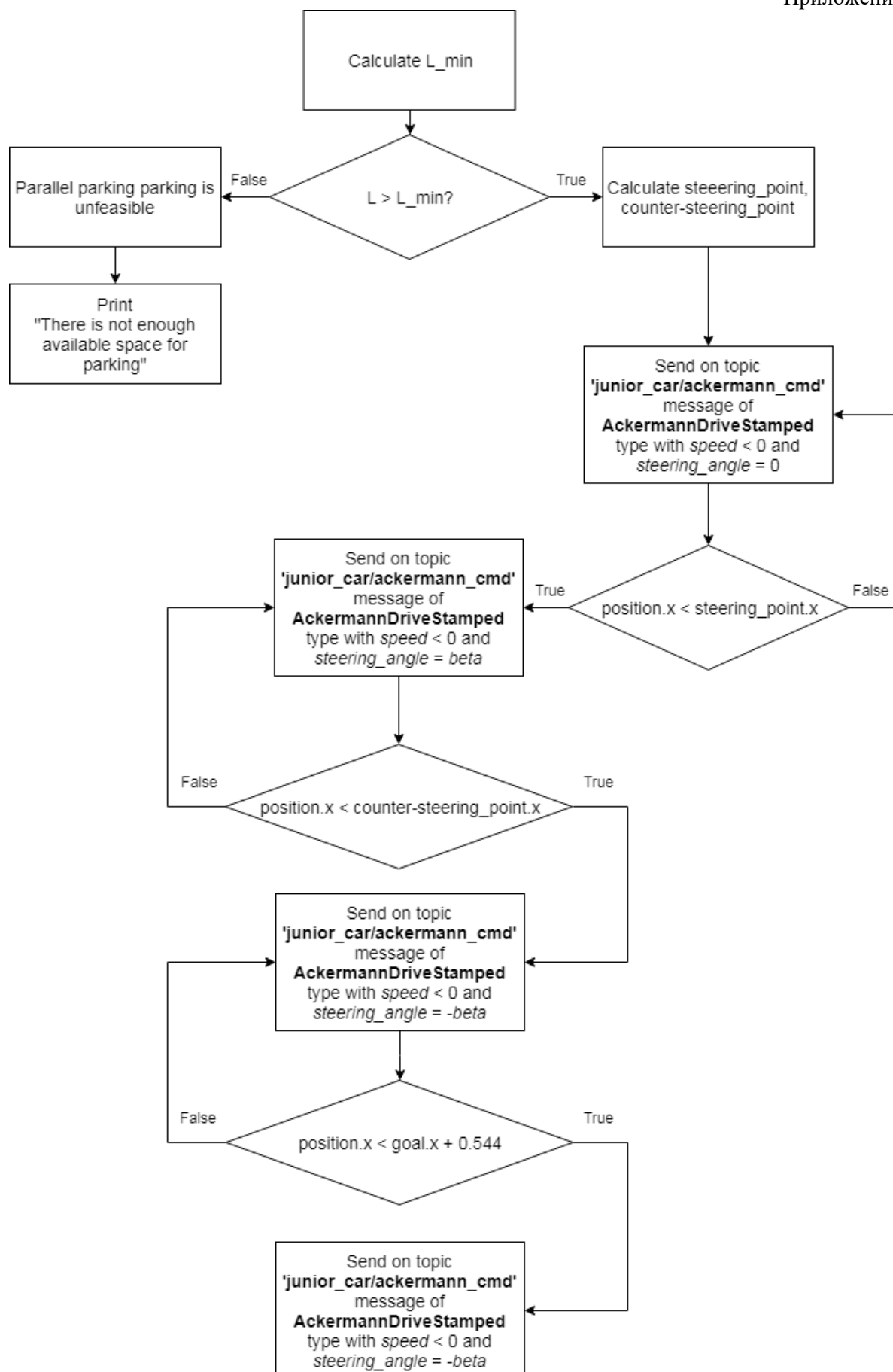
В этой работе мы применили алгоритм параллельной парковки, основанный на геометрическом подходе, на модели робомобиля Unior, но текущая реализация всё равно имеет очевидную недоработку. Процесс парковки зависит от ориентации робомобиля и точки (робомобиль должен быть расположен параллельно оси OX). Решением для проблемы является введение динамических систем координат. Также PID-контроллер обладает неидеальной конфигурацией, что влияет на повороты в ключевых точках. В дальнейшем робот будет распознавать парковочные конусы в режиме реального времени, с помощью обработки данных с его камер [6].

Исследование выполнено при финансовой поддержке Российского Фонда Фундаментальных Исследований (проект № 19-58-70002).

Список библиографических ссылок

1. J.-C. Latombe, Robot motion planning. Springer Science & Business Media, 2012, vol. 124.
2. K. Shabalina, A. Sagitov, K. Su, K.-H. Hsia, and E. Magid, "Aurora unior car-like robot in gazebo environment," International Conference on Artificial Life and Robotics (ICAROB), pp. 116–119, 2019 (in press).
3. C. Sungwoo, C. Boussard, and B. d'Andrea Novel, "Easy path planning and robust control for automatic parallel parking," IFAC Proceedings Volumes, vol. 44, no. 1, pp. 656–661, 2011.
4. "Rohith-k/autonomous-parallel-parking-car-like-robot-gazebo-ros," URL: <https://github.com/Rohith-K/Autonomous-Parallel-Parking-Car-likeRobot-Gazebo-ROS> (дата обращения: 2019-04-11).
5. Quigley M. et al. ROS: an open-source Robot Operating System //ICRA workshop on open source software. – 2009. – Т. 3. – №. 3.2. – С. 5.
6. Safin R., Lavrenov R. Implementation of ROS package for simultaneous video streaming from several different cameras //Int. Conf. on Art. Life and Robotics. – 2018. – С. 220-223.

Приложение 1



DEVELOPMENT OF THE PROCEDURE OF AUTONOMOUS PARALLEL PARKING FOR CAR-LIKE ROBOT UNIOR

Imameev D.T.¹, Lavrenov R.O.¹, Sagitov A.G.¹, Shabalina K.S.¹, Magid E.A.¹,
Martínez-García E. A.², Su Kuo-Lan³

¹ *Kazan Federal University, Kazan, Russia*

² *Autonomous University of Ciudad Juarez, Ciudad-Juarez, Mexico*

³ *National Yunlin University of Science and Technology, Yunlin, Taiwan*

Abstract

This work devotes to implement path planning and control for Unior robot car in parallel parking problem. Path planning is based on a geometrical approach. We implement the existing algorithm on the simulation Unior robot model and validate it in Gazebo simulator. In this paper we focus only in parallel parking task.

Keywords: autonomous parallel parking, Unior robot car, Gazebo.