УДК 62-526 + 004.932.4

DOI 10.35211/1990-5297-2020-9-244-52-56

*А. Е. Марков, С. Е. Терехов, А. Е. Андреев, Р. Е. Горелик***СИСТЕМА ДВИЖЕНИЯ ПО ЗАДАННОЙ ТРАЕКТОРИИ
ДЛЯ БЕСПИЛОТНОГО АВТОМОБИЛЯ****Волгоградский государственный технический университет**

vm@vstu.ru, markovalex95@gmail.com

Представлен алгоритм управления моделью беспилотного автомобиля, осуществляющий движение по заданной траектории с обратной связью с помощью стереокамеры. Описаны основные подсистемы системы управления беспилотным автомобилем.

Ключевые слова: беспилотный автомобиль, SLAM, движение по траектории, техническое зрение.

*A. E. Markov, S. E. Terekhov, A. E. Andreev, R. E. Gorelik***TRAJECTORY FOLLOWING SYSTEM FOR AUTONOMOUS CAR****Volgograd State Technical University**

Trajectory following algorithm for autonomous car model with feedback by stereo-camera is presented. Main subsystems of autonomous car control system are described.

Keywords: autonomous car, SLAM, trajectory following, machine vision.

Управление беспилотными автомобилями, автоматизированное управление транспортом – это актуальные задачи, которые на сегодняшний день имеют важное значение. Существенные результаты в данной области были достигнуты такими компаниями, как Tesla, Google, Uber и др. В России в качестве основных результатов можно представить разработки компании Яндекс, однако среди российских компаний так же имеются проекты по созданию беспилотных автомобилей, которые находятся в стадиях запуска или начального развития. В данной работе описываются теоретические и практические подходы к разработке и реализации системы автономного управления, которые могут быть использованы в проекте ВолгГТУ по созданию беспилотного автомобиля на базе LADA KALINA, и рассматривается разработанная подсистема движения по заданной траектории.

Задача автоматизированного управления без участия человека, может быть решена различными способами, но можно выделить ключевые стадии и части (подзадачи) решение которых необходимо для реализации общей системы.

Можно выделить следующие подзадачи:

1) позиционирование автомобиля в пространстве;

2) выделение различных препятствий и объектов (дорога, пешеходы, другие автомобили, знаки и т. п.) с помощью систем технического зрения и определение их характеристик (положение, габариты, скорость);

3) построение локальной траектории движения автомобиля с учетом объектов, полученных на предыдущем шаге;

4) следование по заданной траектории.

Для решения задач, связанных с построением виртуальной карты окружающего пространства и выделения на ней объектов и их кластеризации, применяются алгоритмы одновременной картографии и навигации (SLAM) [14]. Существуют SLAM-алгоритмы, которые могут использовать данные с таких датчиков, как:

- обычные камеры;
- стереокамеры;
- лидары (LIDAR).

Примерами таких алгоритмов являются Lidar Odometry and Mapping [1], работающий с LIDAR, ORB-SLAM 2 [2], работающий с моно- и стереокамерами, а также с камерами глубины, LSD SLAM [3], работающий со монокамерами, RTAB-Map [4]. Тестирование большого количества SLAM-алгоритмов опубликовано на сайте [5] бенчмарка KITTI [6].

Также существуют наборы разработчика, включающие в себя комплекты аппаратного и программного обеспечения, необходимого для решения поставленных подзадач. Примером такого решения можно назвать NVIDIA DRIVE AGX Platform [7], масштабируемая платформа ИИ для автономной езды. Решение позволяет работать с данными камер, радаров, лидаров и совмещает в себе технологии глубокого обучения, sensor fusion и машинного зрения для высокопроизводительного управления.

В данной работе рассматривается задача, связанная со следованием автомобиля вдоль построенной траектории (полученной на предыдущих этапах от какой-либо из систем).

В нашей работе в качестве датчиков системы компьютерного зрения мы используем ZED-камеру от компании Stereolabs [8]. ZED-камера – это стереокамера с сенсорами инерциальной сис-

темы навигации. Технические характеристики камеры представлены в таблице, приведенной ниже. Поставляемое с камерой программное обеспечение позволяет получать карту глубины, облако точек, осуществлять построение трехмерной модели окружающего пространства и сохранение его в файл, а также определение ориентации и положения камеры в пространстве.

Технические характеристики ZED-камеры

Разрешение, пикселей	1344x376, 2560x720, 3840x1080, 4416x1242
Частота кадров (в зависимости от разрешения), кадров/с	100, 60, 30, 15
Расстояние между объективами, мм	120
Диапазон расстояний, м	0.5–20

В представленной работе ZED-камера применялась для локализации мобильной платформы, на которой она была закреплена.

Для проведения экспериментов, датчики были закреплены на колесной платформе, имитирующей настоящий автомобиль, показанной на рис. 1. Использование для экспериментов колесной платформы на первых этапах оправдано тем, что для реализации управления реальным автомобилем требуется решение задач, связанных с размещением оборудования, датчиков, реверс-инжинирингом и разработкой аппаратного и программного обеспечения для электронного управления узлами автомобиля (газ, руль, тормоз, сцепление и т. д.).

В качестве основы была использована колесная платформа Dagu Wild Thumper 6WD All-Terrain Chassis [9], оснащенная шестью колесами с электродвигателями постоянного тока. Для управления платформой применяется встраиваемый компьютер NVidia Jetson TX2 [10], оснащенный центральным процессором ARM A57 + HMP Dual Denver (суперскалярная микроархитектура с широким командным словом от NVidia на базе ARM), графическим ускорителем с архитектурой NVidia Pascal, содержащим 256 ядер CUDA, 8 Гб оперативной памяти и 32 Гб постоянной памяти eMMC. Для управления двигателями использован драйвер Pololu MD03A с двумя H-мостами, позволяю-

щий независимо управлять приводами левого и правого бортов.

Рис. 1. Колесная платформа для экспериментов

Для связи NVidia Jetson и драйвера служит микроконтроллер STM32F103C8T6. Он принимает команды управления двигателями от бортового компьютера с помощью интерфейса UART и управляет драйвером двигателей.

Питание платформы осуществляется от двух литий-полимерных аккумуляторов, отдельно для питания силовой части и электроники. Электрическая схема платформы приведена на рис. 2.

Рис. 2. Электрическая схема колесной платформы

Для решения задачи следования по траектории применяется SLAM-алгоритм, определяющий локальное положение и ориентацию камеры и, соответственно, мобильной платформы, в пространстве относительно точки старта. Локальные координаты мобильной платформы преобразуются в систему координат траектории,

таким образом получается положение и ориентация мобильной платформы относительно траектории. Вычисляется отклонение реального положения платформы от заданной траектории и формируются сигналы управления для двигателей, чтобы осуществить поворот колесной платформы в сторону возврата на траекторию.

Рис. 3. Схема определение угла поворота

Для реализации движения по траектории применяется следующий подход. В текущем положении мобильной платформы строится окружность радиуса R (см. рис. 3). Затем определяется точка I пересечения окружности и требуемой траектории движения автомобиля. Определяется вектор $\vec{OI}$ от точки положения платформы в направлении точки пересечения окружности и траектории. Затем определяется угол α между вектором текущего направления платформы $\vec{v}$ и вектором $\vec{OI}$:

$$\alpha = \operatorname{atan} 2 \left(\frac{|\vec{v} \times \vec{OI}|}{\vec{v} \cdot \vec{OI}} \right). \quad (1)$$

На основе полученного угла α формируются сигналы управления двигателем с помощью ПИД-регулятора:

$$= \text{speed} + \text{PID}(\alpha) = \text{speed} - \text{PID}(\alpha). \quad (2)$$

Окружность служит для того, чтобы определять точку траектории, лежащую немного впереди по движению автомобиля. Таким образом, регулируя радиус окружности, можно предусмотреть, насколько заранее регулятор будет поворачивать автомобиль.

Точка пересечения окружности и траектории определяется следующим образом. Траектория задается в виде набора точек, таким образом достаточно найти точку пересечения отрезка с окружностью.

Рассмотрим окружность с центром в точке O и отрезок P_1P_2 (рис. 4). Рассмотрим прямую, заданную каноническим уравнением прямой:

$$\frac{x - p_{1x}}{p_x} = \frac{y - p_{1y}}{p_y}. \quad (3)$$

В данном случае прямая задана двумя точками P_1 и P_2 , поэтому

$$p_x = p_{2x} - p_{1x} \quad (4)$$

$$p_y = p_{2y} - p_{1y}$$

$$\frac{x - p_{1x}}{p_{2x} - p_{1x}} = \frac{y - p_{1y}}{p_{2y} - p_{1y}}. \quad (5)$$

Определим расстояние от точки O до прямой и координаты ближайшей к O точки C на прямой [12]:

$$d = |\overline{OC}| = \frac{|aO_x + bO_y + c|}{\sqrt{a^2 + b^2}} \quad (6)$$

$$C_x = \frac{b(bO_x - aO_y) - ac}{a^2 + b^2} \quad (7)$$

$$C_y = \frac{a(-bO_x - aO_y) - bc}{a^2 + b^2}. \quad (8)$$

Рис. 4. Определение точки пересечения отрезка и окружности

В зависимости от соотношения радиуса окружности d и расстояния от точки O до прямой, возможно различное количество точек пересечения: ни одной; одна (в том случае, если окружность касается прямой) и две (если окружность не пересекается с прямой). В данном случае нас не интересуют обе точки пересечения, и необходимо выбрать одну. Положим, что точка P_1 имеет меньший порядковый номер в траектории, чем точка P_2 . В таком случае, вектор $\overline{P_1P_2}$, являющийся также направляющим вектором прямой, будет указывать на направление траектории в этой окрестности. Поэтому мы рассматриваем только ту точку пересечения, которая лежит по направлению вектора $\overline{P_1P_2}$ от точки C :

$$I = C + \sqrt{R^2 - d^2} \cdot \frac{\overline{CI}}{|\overline{CI}|}. \quad (9)$$

На основе рассмотренных принципов было реализовано программное обеспечение для движения мобильной платформы по заданной траектории. Исходные коды программного обеспечения доступны на GitHub [13].

Были произведены эксперименты с движением мобильной платформы по заданной траектории, в частности, по кругу. На рис. 5 представлено программное и реальное движение мобильной платформы по кругу.

Рис. 5. Движение мобильной платформы по кругу

Таким образом, была решена одна из подзадач, связанная с движением автономного автомобиля по заданной траектории. Как известно из начала статьи, это одна из подзадач, решение которой позволяет перейти к следующим этапам разработки, таким как, например, планирование траектории, навигация и сегментация, выделение целей и обработка различных ситуаций (встречи с пешеходами и другими машинами, алгоритмы действий при возможных авариях и наезде на пешеходов и т. п.).

Реализация решения для всех вышеперечисленных подзадач позволит создать полноценную систему автономного управления автомобилем и применить ее на разрабатываемом беспилотном автомобиле в ВолГТУ на основе LADA KALINA.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Zhang, J. LOAM: Lidar Odometry and Mapping in Real-time / Ji Zhang, Sanjiv Singh // Robotics: Science and Systems Conference, July 12–16, 2014. – 2014.
2. Mur-Artal R. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras / Raul Mur-Artal, Juan D. Tard // CoRR. – 2016. – Vol. 33, issue. 5. – P. 1255–1262.
3. Engel, J. LSD-SLAM: Large-Scale Direct Monocular SLAM / J. Engel, S. Schöps, D. Cremers // Lecture Notes in Computer Science. – 2014. – Vol. 8690.
4. Labbe, M. Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM / M. Labbe, F. Michaud // Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. – 2014. – P. 2661–2666.
5. Visual Odometry / SLAM Evaluation 2012 [Электронный ресурс]. – Режим доступа : http://www.cvlibs.net/datasets/kitti/eval_odometry.php (дата обращения 14.10.2018).
6. Geiger, A. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite / A. Geiger, P. Lenz, R. Urtasun // Conference on Computer Vision and Pattern Recognition (CVPR). – 2013.
7. Autonomous Car Driving Platform | NVidia Drive APX [Электронный ресурс]. – Режим доступа : <https://www.nvidia.com/en-us/self-driving-cars/drive-platform/> (дата обращения: 14.10.2018).
8. Meet the ZED Stereo Camera [электронный ресурс]. – Режим доступа : <https://www.stereolabs.com/zed/> (дата обращения: 14.10.2018).
9. Dagu Wild Thumper 6WD All-Terrain Chassis [Электронный ресурс]. – Режим доступа : <https://www.pololu.com/product/1561> (дата обращения: 14.10.2018).
10. Решения NVidia Jetson для разработки встраиваемых систем [Электронный ресурс]. – Режим доступа : <https://www.nvidia.com/ru-ru/autonomous-machines/embedded-systems-dev-kits-modules/> (дата обращения 14.10.2018).
11. Dual VNH3SP30 Motor Driver Carrier MD03A [Электронный ресурс]. – Режим доступа : <https://www.pololu.com/product/707> (дата обращения: 14.10.2018).
12. Larson, R. Precalculus / Ron Larson. – Brooks/Cole/Cengage Learning, 2007. – 452 p.
13. https://github.com/Garrus007/jetson_car
14. Системы машинного зрения мобильных роботов / В. А. Беликов, В. В. Жога, В. Н. Скакунов, С. Е. Терехов, И. В. Шаманов // Известия ВолГТУ : межвуз. сб. науч. ст. № 25 (152) / ВолГТУ. – Волгоград, 2014. – (Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах» ; вып. 22). – С. 184–190.