УДК 004.9

ПОСТРОЕНИЕ НАВИГАЦИОННЫХ РЕКОНФИГУРИРУЕМЫХ МОДЕЛЕЙ МОБИЛЬНЫХ РОБОТОВ

^{1,2}Нгуен Чанг Хоанг Тхуй

¹Национальный исследовательский Томский государственный университет, Томск;

²Вьетнамский морской университет, Хайфон, Вьетнам, e-mail: trangnht@vimaru.edu.vn

Ключевой проблемой разработки мобильных роботов является поиск и перемещение робота в неизвестной среде. В методах навигации и определения местоположения часто приходится использовать систему мейнфрейма для обработки полученной от робота информации об окружающей среде, в которой он путешествовал. Фактически мобильные роботы обычно очень маленькие, поэтому для их использования необходимо соответствующее оптимальное, компактное и техническое решение. Существует много предлагаемых методов, но эти методы решаются только алгоритмически, и способ обработки информации, полученной от роботов на аппаратных устройствах, не упоминается. Использование правильных аппаратных технологий для мобильных роботов также является проблемой для научных исследователей. В этой статье описан способ построения высокопроизводительной реконфигурируемой модели на основе принципов параллельной архитектуры в сочетании с алгоритмом SLAM для отображения мобильных роботов в помещениях. Мобильные роботы будут построены на платформе ROS в среде Gazebo, где будут установлены препятствия, еще не известные роботам, в сочетании с алгоритмом SLAM и методом Octomap. Реконфигурируемая модель разработана в среде Matlab, состоящей из множества идентичных вычислительных компонентов, которые настраиваются автоматически в зависимости от местоположения робота и параметра расстояния. Результаты показывают, что построенная модель будет быстро получать высокоточную карту одновременно с движением робота. Кроме того, эта модель может быть установлена на ИС, ПК или ПЛИС, поэтому она имеет высокую практическую ценность.

Ключевые слова: реконфигурируемая вычислительная среда, высокопроизводительная реконфигурируемая модель, высокопроизводительная вычислительная система, SLAM, модель бинарной логики

BUILDING NAVIGATION RECONFIGURABLE MODELS FOR MOBILE ROBOTS

^{1,2}Nguyen Trang Hoang Thuy

¹National Research Tomsk State University, Tomsk;

²Vietnam Maritime University, Hai Phong, Viet Nam, e-mail: trangnht@vimaru.edu.vn

The key issue of mobile robot development is locating and navigating the robot in an unknown environment. Navigation and location techniques often have to use a mainframe system to process information received from the robot about the environment in which it traveled. In fact, mobile robots are usually very small, so robot also needs optimal, compact and appropriate technical problem-solving. There are many proposed methods, but these methods only solve algorithmically and the way to process information received from robots on hardware devices is not mentioned. Using the right hardware techniques for mobile robots is also a challenge for scientific researchers. This article describes the way to build a high-performance reconfigurable model based on the principles of parallel architecture combined with the SLAM algorithm for mapping mobile robots in indoor environments. Mobile robots will be built on the ROS platform in the Gazebo environment, putting obstacles that robots are not known, combining in parallel with the SLAM algorithm and the Octomap method. The reconfigurable model is designed in a Matlab environment consisting of many identical computational components, being controlled automatically based on the robot's location and distance parameter. The results show that the built model will receive high-precision map, quickly and simultaneously with robot movement. In addition, this model can be installed on an IC, PC or FPGA, so this model has high practical applicability

Keywords: Reconfigurable computing environment, high-performance reconfigurable model, high-performance computing system, SLAM, binary logic model

Появление роботов в значительной степени поддержало человеческую производственную деятельность. Робототехника стала неотъемлемой частью научных исследований. Сегодняшние автономные роботы способны заменить людей, работающих в опасных средах. Одним из методов, который ознаменовал большой скачок в применимости роботов, является метод SLAM – он позволяет роботам перемещаться в закрытых или недоступных пространствах, анализировать свое местоположение в реальном времени, позиционировать препятствия и при передвижении картировать

окружающее их пространство [1]. Целью исследования является повышение эффективности компьютерной обработки и синтеза данных путем разработки и изучения реконфигурируемых алгоритмов и моделей компьютерных сред для обработки информации, картографирования для помощи в навигации робота в среде, в которой он движется.

Slam и Octomap

Slam (одновременная локализация и отображение) – это вычислительная задача, обычно состоит из двух частей (рис. 1).

Одна заключается в построении карты неизвестного окружения или неизвестного пространства. В то же время можно отслеживать местоположение или движение камеры или датчиков робота в этом пространстве. Следовательно, конкретный объект, движущийся в этом пространстве, можно описать достаточно ясно и с большой точностью [2–5].

В настоящее время Slam используется на многих рынках и во многих приложениях, существующих в мире. Можно начать с мобильных телефонов и множества приложений дополненной реальности для большого количества беспилотных автомобилей и дронов, перемещающихся и летающих полностью автономно. Эта технология в настоящее время достаточно широко используется и продолжает исследоваться [6–8].

Раньше Slam развертывался на серверах, требующих больших вычислительных ресурсов (рис. 2). По этой причине изготовление таких систем было очень громоздким.

С годами алгоритм совершенствовался, аппаратная часть также улучшалась. Внедрение этой технологии уже происходит от CPU к GPU, а теперь даже на мобиль-

ных телефонах. Таким образом, повышение энергоэффективности и снижение массогабаритов – это вопрос исследований, представляющий сегодня большой интерес для молодых ученых. Существует множество алгоритмов SLAM, опубликованных для повышения производительности вычислений картирования и полной траектории. В этой статье предлагается метод аппаратной обработки, а именно создание микроципа с использованием параллельных методов для оптимизации получаемой информации, а также для достижения высокой производительности [1; 3; 9].

Для оптимизации в работе с данными используется метод Octomap – это метод представления эффективно размещающихся сеточных карт в 3D. Он основан на октодереве с использованием вероятностных оценок занятости. Метод Octomap обеспечивает максимальную гибкость для области карты и ее разрешения. Кроме того, этот метод помогает эффективно представить моделирование последовательных данных при минимальных требованиях к памяти [10; 11].

Рис. 1. SLAM

Рис. 2. Сервер SLAM для платформенной установки

Рис. 3. Связь элементарных вычислений построена в 4 слоя на основе метода Оостомар

На основе метода Оостомар строится модель автоматического картирования с 4 слоями, каждый слой будет состоять из 4 групп, каждая группа будет иметь 4 основных вычислительных компонента (рис. 3). Соединение этих элементарных компонентов в группах, а также между группами и слоями вместе осуществляется по логическому правилу. Когда элементарные вычисления в группе получают значение, равное единице, три других вычислительных элемента в группе также получают значение, равное единице. Когда элементарные вычисления в группе получают нулевое значение, три оставшихся элементарных вычисления в группе также получают значение, равное нулю.

Каждый элементарный вычислитель (ЭВ) имеет 10 входов, 5 выходов и 5 разрядов управляющего сигнала z . Для каждого из возможных наборов значений сигнала z мы будем иметь свой (индивидуальный) выходной сигнал. ЭВ, рассматриваемый в работе, описывается следующим образом [12–14]:

$$f_1 = x_1$$

$$f_2 = x_1$$

$$f_3 = x_1$$

$$\begin{aligned} f_{4_T} = & (x_1 \cdot \bar{z}_1) \vee (x_2 \cdot \bar{z}_1) \vee (x_3 \cdot \bar{z}_1) \vee (x_4 \cdot \bar{z}_1) \vee (q_5 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_5) \vee (q_5 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot z_5) \vee \\ & (q_5 \cdot \bar{z}_1 \cdot z_3 \cdot \bar{z}_5) \vee (q_5 \cdot \bar{z}_1 \cdot \bar{z}_4 \cdot z_5) \vee (q_6 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_4) \vee (q_6 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot \bar{z}_3 \cdot z_4) \vee (q_6 \cdot \bar{z}_1 \cdot z_3 \cdot \bar{z}_4) \vee \\ & (q_6 \cdot \bar{z}_1 \cdot z_4 \cdot z_5) \vee (q_7 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_4) \vee (q_7 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot z_4) \vee (q_7 \cdot \bar{z}_1 \cdot z_3 \cdot z_4 \cdot z_5) \vee (q_7 \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot \bar{z}_4) \vee \\ & (q_7 \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot \bar{z}_5) \vee (q_8 \cdot \bar{z}_1 \cdot z_2 \cdot z_3 \cdot z_5) \vee (q_8 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_3 \cdot \bar{z}_5) \vee (q_8 \cdot \bar{z}_1 \cdot z_2 \cdot z_4) \vee \\ & (q_8 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot z_3 \cdot \bar{z}_5) \vee (q_8 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot \bar{z}_3 \cdot \bar{z}_4) \vee (q_8 \cdot \bar{z}_1 \cdot z_4 \cdot z_5) \vee (q_9 \cdot \bar{z}_1 \cdot \bar{z}_2) \vee (q_9 \cdot \bar{z}_1 \cdot z_3 \cdot z_4) \vee \\ & \vee (q_9 \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot z_5) \vee (q_{10} \cdot \bar{z}_1 \cdot z_2 \cdot z_3) \vee (q_{10} \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot \bar{z}_4) \vee (q_{10} \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot z_4) \cdot \\ f_{5_T} = & (x_1 \cdot \bar{z}_1) \vee (x_2 \cdot \bar{z}_1) \vee (x_3 \cdot \bar{z}_1) \vee (x_4 \cdot \bar{z}_1) \vee (q_5 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_5) \vee (q_5 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot z_5) \vee \\ & (q_5 \cdot \bar{z}_1 \cdot z_3 \cdot \bar{z}_5) \vee (q_5 \cdot \bar{z}_1 \cdot \bar{z}_4 \cdot z_5) \vee (q_6 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_4) \vee (q_6 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot \bar{z}_3 \cdot z_4) \vee (q_6 \cdot \bar{z}_1 \cdot z_3 \cdot \bar{z}_4) \vee \\ & (q_6 \cdot \bar{z}_1 \cdot z_4 \cdot z_5) \vee (q_7 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_4) \vee (q_7 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot z_4) \vee (q_7 \cdot \bar{z}_1 \cdot z_3 \cdot z_4 \cdot z_5) \vee (q_7 \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot \bar{z}_4) \vee \\ & (q_7 \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot \bar{z}_5) \vee (q_8 \cdot \bar{z}_1 \cdot z_2 \cdot z_3 \cdot z_5) \vee (q_8 \cdot \bar{z}_1 \cdot z_2 \cdot \bar{z}_3 \cdot \bar{z}_5) \vee (q_8 \cdot \bar{z}_1 \cdot z_2 \cdot z_4) \vee \\ & (q_8 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot z_3 \cdot \bar{z}_5) \vee (q_8 \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot \bar{z}_3 \cdot \bar{z}_4) \vee (q_8 \cdot \bar{z}_1 \cdot z_4 \cdot z_5) \vee (q_9 \cdot \bar{z}_1 \cdot \bar{z}_2) \vee (q_9 \cdot \bar{z}_1 \cdot z_3 \cdot z_4) \vee \\ & \vee (q_9 \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot z_5) \vee (q_{10} \cdot \bar{z}_1 \cdot z_2 \cdot z_3) \vee (q_{10} \cdot \bar{z}_1 \cdot \bar{z}_2 \cdot \bar{z}_4) \vee (q_{10} \cdot \bar{z}_1 \cdot \bar{z}_3 \cdot z_4), \end{aligned} \quad (*)$$

где f_1, f_2, f_3, f_4, f_5 – выходы ЭВ, $x_1, x_2, x_3, x_4, q_5, q_6, q_7, q_8, q_9, q_{10}$ – входы, z_1, z_2, z_3, z_4, z_5 – настроенные входы.

*Построение высокопроизводительной
реконфигурируемой среды*

Для повышения эффективности, экономии энергии, компактности, простоты в использовании – идеальным будет создание перестраиваемой вычислительной среды. Эта среда на самом деле представляет собой модель, разработанную с множеством небольших перестраиваемых автоматов. В частности, автомат внутри среды реализует новый подходящий для нее алгоритм для ускорения вычислительной обработки. Автоматами в среде будут элементарные вычислительные компоненты с идентичной структурой. Такой автомат будет настроен на основе регулируемого сигнала разряда z [15–18].

Построение перестраиваемых автоматов (ПА) также называется элементарными вычислениями, основанными на построении автоматических отображений. Автоматическое отображение построено по принципу зависимости изменения выходного сигнала от входного сигнала [15; 16].

Каждый ЭВ имеет 10 входов, 5 выходов и 5 разрядов управляющего сигнала z . С каждым набором из 5 контактов сигнала z будет иметь разный выходной сигнал.

Имитационная модель, основанная на формуле (*), показана на рис. 4.

Моделирование в среде Gazebo

На базовой диаграмме процесса (выше) показана стандартная работа SLAM в большинстве приложений, используемых сегодня. Многие лазерные датчики, камеры, звук и т.д. используются совместно с роботами в приложениях SLAM. Из них лазерный датчик является одним из лучших, потому что он может получать данные при слабом освещении и на него не оказывает влияние датчик шума.

В среде Gazebo, когда робот движется, положение робота может быть определено на основе каждого координатного кадра, переданного роботом. Через лазерный датчик собираются данные о расстоянии от робота до препятствия. Эти данные о расстоянии сравниваются с размерами робота, далее определяется положение ячейки, содержащей препятствие, и устанавливается для этой ячейки значение 1. То есть во сколько раз расстояние от робота до объекта больше размера робота, столько раз будет вычислено количество квадратов от робота до препятствия, при том что размещение указанной ячейки с препятствием равно единице [15; 16].

На следующем шаге для каждого координатного кадра значение сгенерированной карты будет сохранено в памяти. При хранении эта карта постоянно обновляется, пока робот не покинет эту среду.

Рис. 4. Элементарный вычислитель создан в перестраиваемой вычислительной среде

Рис. 5. Робот Turtlebot 3 Waffle в Gazebo

С помощью этого метода можно получить информацию об облаке точек от датчика или сканера, затем эта информация передается автоматически не только на обрабатывающее устройство, но также с помощью вышеуказанного расчета можно рассчитать и среду, создаваемую для восстановления автоматов.

Когда робот перемещается в симуляционной обстановке Gazebo (рис. 5), вокруг него возникают препятствия. Робот получает данные о расстоянии от него до объекта с лазерного датчика. Затем для каждого координатного кадра также фиксируются данные о координатах положения робота, и обрабатывается соответствующая информация.

В этом методе предполагается, что размер робота соответствует ячейке. И текущая построенная область обладает размерами 6×6. Когда робот движется прямо вперед, собранные данные о расстоянии обрабатываются и сравниваются с размером робота, исходя из чего можно вычислить, какие ячейки содержат препятствия, а какие нет.

Ячейкам, содержащим препятствия, будет присвоено значение 1, а ячейкам, их

не содержащим, будет присвоено значение ноль.

При применении метода Octomap связь между элементами в этой реконфигурируемой среде показана на рис. 6.

При создании этой среды в программном обеспечении Matlab результаты, полученные, как показано на рис. 7, будут реконфигурированы ЭВ в этом чипе путем настройки разрядов управляющих сигналов. Эти ЭВ конструктивно похожи, они изменятся только при изменении управляющих ими сигнальных разрядов. Для среды размером 6×6 необходимо построить ЭВ, соответствующие данной среде (рис. 8), т.е. потребуется 36 ЭВ.

Параллельная комбинация вышеперечисленных методов помогает роботу получать точную карту при его движении, ячейки с препятствиями получают значение единицы, ячейки без препятствий получают значение нуля. Отсюда легко увидеть, что карта движущегося робота в окружающей среде Gazebo, созданная на рис. 5, будет автоматически генерироваться роботом, когда он перемещается из начального положения в конечное положение окружения.

Рис. 6. Подключение среды реконфигурации по методу Остотар

Рис. 7. Среда реконфигурации, встроенная в Matlab

Рис. 8. Карта робота получается при прямом движении из начальной позиции в конечную позицию окружающей среды

Заключение

Создание модели реконфигурируемой вычислительной среды, возможность ее размещения на мобильных роботах, программирование и автоматическое отображение каждого движения робота в данной среде – это новый подход к SLAM. Кроме того, становится совершенно выгодно создавать подобные модели, поскольку возможность реконфигурировать электромобили во время программирования, а не проектировать компоненты оборудования, не может быть установлена на электронных устройствах. Использование этой модели на ПЛИС или на компьютере, на микрочипе или на других устройствах имеет широкое практическое применение и может широко использоваться в мире.

Список литературы

1. Аносов В.Л., Черномаз В. Учебник по курсу «Симуляторы» для студентов специальностей 7.050102 «Экономическая кибернетика» и 7.080404 «Интеллектуальные системы при принятии решений». Краматорск: ДГМА, 2007. С. 156.
2. Nguyen Hoang Thuy Trang, Shydouski S. Situations in Construction of 3D Mapping for Slam. 2018. DOI: 10.1051/mateconf/201815501055.
3. Nguyen Hoang Thuy Trang, Shydouski S. Tunable computing Slam navigation environments. 2019. DOI: 10.1088/1757-899X/516/1/012053.
4. An Z., Lina Hao, Liu Y., Dai L. Development of Mobile Robot SLAM Based on ROS. 2016. DOI: 10.18178/ijmerr.5.1.47-51.
5. Fernando A. Auat Cheein, Natalia Lopez, Carlos M. Soria, Fernando A. di Sciascio, Fernando Lobo Pereira, Ricardo Carelli. SLAM algorithm applied to robotics assistance for navigation in unknown environments. Journal of Neuro Engineering and Rehabilitation. 2010. Vol. 7. № 10. DOI: 10.1186/1743-0003-7-10.

6. Aria M. Real-Time 2D Mapping and Localization Algorithms for Mobile Robot Applications. 2019. DOI: 10.1088/1757-899X/662/2/022131.
7. Ebrahim A. Mattar. Mobile Robot Feature-Based SLAM Behavior Learning and Navigation in Complex Spaces. 2018. DOI: 10.5772/intechopen.81195.
8. César Debeunne, Damien Vivet. A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping. 2020. DOI: 10.3390/s20072068.
9. Riaz Z., Pervez A., Ahmer M., Iqbal J. A Fully Autonomous Indoor Mobile Robot using SLAM, Department of Mechatronics Engineering College of E.M.E National University of Sciences and Technology Rawalpindi, Pakistan. 2020. DOI: 10.1109/ICIET.2010.5625691.
10. Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. 2013. V. 34. P. 189–206. DOI: 10.1007/s10514-012-9321-0.
11. Джеркер Бергстрем. Планирование пути с регионами с взвешенной стеной с использованием OctoMap. 2018. С. 7–8.
12. Шашев Д.В., Шидловский С.В. Морфологическая обработка двоичных изображений с использованием настраиваемых вычислительных сред // Автометрия. 2015. Т. 51. № 3. С. 19–26.
13. Иванов В.А., Киричук В.С. Особенности алгоритмов обнаружения упавшего человека по последовательности изображений сцены // Автометрия. 2011. Т. 47. № 2. С. 15–25.
14. Шапиро Л., Стокман Дж. Компьютерное зрение / Пер. с англ. М.: БИНОМ. Лаборатория знаний, 2006. 752 с.
15. Шидловский С.В. Автоматическое управление. Реконфигурируемые системы: учебное пособие. Томск: Изд-во Том. ун-та, 2010. 168 с.
16. Шидловский С.В. Автоматическое управление. Перестраиваемые конструкции. Томск: Изд-во Том. ун-та, 2006. 288 с.
17. Калаяев И.А., Левин И.И., Семерников Е.А., Шмоилов В.И. Реконфигурируемые многоконвейерные вычислительные структуры. Ростов н/Д.: Изд-во УНЦ РАН, 2008. 393 с.
18. Хорошевский В.Г., Курносоев М.Г., Мамоиленко С.Н. Пространственно-распределенная многокластерная вычислительная система: архитектура и программное обеспечение // Вестн. ТГУ. Менеджмент, компьютерная инженерия и информатика. 2011. № 1 (14). С. 79–84.