УДК 004.9

МОДЕЛИРОВАНИЕ РЕКОНФИГУРИРУЕМОЙ СРЕДЫ ДЛЯ НАВИГАЦИИ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

[1,2]Нгуен Чанг Хоанг Тхуи

[1]Национальный исследовательский Томский государственный университет, Томск;
[2]Вьетнамский морской университет, Хайфон, e-mail: trangnht@vimaru.edu.vn

Использование БПЛА с годами неуклонно растет, и сегодня БПЛА стали стандартным исследовательским инструментом для выполнения локализации в небольшом узком пространстве, где устройства не могут использовать GPS-позиционирование для определения положения конкретного объекта. Для решения проблемы локализации внутренней среды был создан метод SLAM. В нем используются независимые мобильные инструменты для картирования исследованного пространства с отслеживанием текущего местоположения и пройденного расстояния. Разработка SLAM облегчила ориентацию мобильных роботов как с наземным, так и с беспилотным оборудованием. Однако сложность современных методов позиционирования и точное параллельное сопоставление очень важны, поскольку робот очень мал, а для обработки этой информации требуется мэйнфрейм, решающий технические проблемы. Следовательно, для решения данных задач наиболее эффективным является использование вычислительной архитектуры параллельного конвейерного типа. В этой статье описаны алгоритмы построения и реконфигурируемые модели окружающей среды для навигации БПЛА. На основе соответствующего набора логических формул И, ИЛИ, НЕ будут построены вычислительные компоненты для создания автоматических сопоставлений для реконфигурируемой среды. Эти вычислительные компоненты, по сути, являются регулируемыми автоматами, производительность автомата зависит от входных сигналов автомата. Эта реконфигурируемая модель среды представляет собой комбинацию нескольких машин автоматизации (вычислительных компонентов) вместе. Эта модель позволяет пользователю настраиваться по внешним сигналам, не влияя на внутреннее оборудование устройства. Эта реконфигурируемая среда может быть установлена на компьютере, на микрочипе или на ПЛИС. Фактически использование систем для управления БПЛА требует хорошей основы для обработки данных в среде мобильных БПЛА. Построение реконструированной модели позволит создавать высокоточные карты, пригодные для использования в исследованиях БПЛА.

Ключевые слова: реконфигурируемая вычислительная среда, высокопроизводительная реконфигурируемая модель, высокопроизводительная вычислительная система, SLAM, логическая модель, беспилотный летательный аппарат, БПЛА

MODELLING OF A RECONFIGURABLE ENVIRONMENT FOR NAVIGATION OF UNMANNED AERIAL VEHICLE

[1,2]Nguyen Trang Hoang Thuy.

[1]National Research Tomsk State University, Tomsk;
[2]Vietnam Maritime University, Hai Phong, e-mail: trangnht@vimaru.edu.vn

The use of UAVs has steadily increased over the years, and today UAVs have become the standard research tool for performing localization in small, narrow spaces where devices cannot use GPS positioning to determine the position of a specific object. To solve the problem of localizing the internal environment, the SLAM method was created. It uses independent mobile tools to map the explored area, keeping track of the current location and distance traveled. The development of SLAM has made condition for navigating mobile robots with both on the ground and unmanned equipment. However, the sophistication of modern positioning techniques and accurate parallel matching are very important because the robot is very small, and to process this information requires a mainframe to solve technical problems. Therefore, to solve these problems, the most effective is the use of a parallel pipeline type computing architecture. This article describes the construction of the algorithms and reconfigurable environment models for UAV navigation. Based on an appropriate set of AND, OR, and NOT logical formulas, computational components will be built to generate automatic mappings for a reconfigurable environment. In fact, These computational components are control automation, the performance of the automatic machine depends on the input signals of the automatic machine. This reconfigurable environment model is a combination of several automatic machines (computing components) together. This model allows the user to adjust with external signals without affecting the device's internal hardware. This reconfigurable environment can be installed on a computer, on a microchip, or on an FPGA. In fact, the use of systems for UAV control requires a good basis for data processing in a mobile UAV environment. The construction of the reconfigurable model will make it possible to create high-precision maps suitable for use in UAV research.

Keywords: reconfigurable computing environment, high performance reconfigurable model, high performance computing system, SLAM, logic model, unmanned aerial vehicle, UAV

Беспилотный летательный аппарат – одно из важных достижений в развитии науки и техники. Он может быть запрограммирован на автоматический полет по заранее определенным маршрутам, и им можно управлять удаленно в режиме реального времени. Использование беспилотных летательных аппаратов растет во всех сферах [1, 2]. Существует множество способов управления мобильными роботами, а также беспилотным летательным аппаратом, но SLAM – один из самых интересных.

SLAM занимается отображением неизвестной среды с помощью мобильного робота во время навигации по среде с использованием картирования [3, 4]. В последнее время было проведено множество исследований для определения вычислительной эффективности SLAM и объединения данных, а также и для определения всего необходимого для мощного метода SLAM. Целью исследования является разработка и исследование реконфигурируемых алгоритмов и моделей компьютерной среды для обработки информации, создание карт для БПЛА во время его передвижения, что помогает повысить эффективность компьютерной обработки и синтеза данных.

1. Создание алгоритма автоматического отображения для перестраиваемой вычислительной среды (ПВС)

Картирование окружающей среды включает создание математической модели пространственной информации реальной окружающей среды. SLAM расширяет требования этой математической модели, показывая то же состояние робота и положение извлеченных ориентиров относительно его положения [3, 4].

В статье будет показано построение вычислительной среды, настраивающейся в соответствии с предопределенными алгоритмами. Автоматическое реконфигурационное устройство называется автоматом, мощность которого зависит от значения входного сигнала. В настройке этого аппарата используются только базовые входные сигналы 0 и 1. Следовательно, на основе булевых формул можно использовать простые логические вычисления И, ИЛИ, НЕ для настройки реконфигурируемых логических схем [5, 6]. Это автоматическое устройство работает в строго определенное время таким образом, что имеет только одно состояние за раз [7–9].

Для корректировки состава расчет будет основываться на управляющем сигнале разряда z. Это означает, что вычислительный компонент будет состоять из входов x, управляющего сигнала разрядки z и выводов выходного сигнала f. Значения внутреннего и выходного сигналов каждый раз могут быть установлены в соответствии с функциями входного и внутреннего сигналов. Чтобы гарантировать, что логические формулы определены только в классе, для каждой из этих логических формул создаются автоматические сопоставления. Эти булевы формулы построены следующим образом (рис. 1) [10–12]:

1) При $z_1 = z_2 = z_3 = z_4 = z_5 = 0$ (рис. 1, а)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_1} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_7 \vee q_8 \vee q_9 \vee q_{10} \\ f_{5_1} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_7 \vee q_8 \vee q_9 \vee q_{10} \end{cases}$$

2) При $z_1 = z_2 = z_3 = z_4 = 0$, $z_5 = 1$ (рис. 1, б)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_2} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_7 \vee q_8 \vee q_9 \vee q_{10} \\ f_{5_2} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_7 \vee q_8 \vee q_9 \vee q_{10} \end{cases}$$

3) При $z_1 = z_2 = z_3 = 0$, $z_4 = 1$, $z_5 = 0$ (рис. 1, в)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_3} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_7 \vee q_9 \vee q_{10} \\ f_{5_3} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_7 \vee q_9 \vee q_{10} \end{cases}$$

4) При $z_1 = z_2 = z_3 = 0$, $z_4 = 1$, $z_5 = 1$ (рис. 1, г)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_4} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee x_8 \vee q_9 \vee q_{10} \\ f_{5_4} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee x_8 \vee q_9 \vee q_{10} \end{cases}$$

5) При $z_1 = z_2 = 0$, $z_3 = 1$, $z_4 = z_5 = 0$ (рис. 1, д)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_5} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_8 \vee q_9 \vee q_{10} \\ f_{5_5} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_8 \vee q_9 \vee q_{10} \end{cases}$$

6) При $z_1 = z_2 = 0$, $z_3 = 1$, $z_4 = 0$, $z_5 = 1$ (рис. 1, е)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_6} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_9 \vee q_{10} \\ f_{5_6} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_9 \vee q_{10} \end{cases}$$

7) При $z_1 = z_2 = 0$, $z_3 = z_4 = 1$, $z_5 = 0$ (рис. 1, ё)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_7} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_7 \vee q_8 \vee x_9 \\ f_{5_7} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_7 \vee q_8 \vee x_9 \end{cases}$$

8) При $z_1 = z_2 = 0$, $z_3 = z_4 = z_5 = 1$ (рис. 1, ж)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_8} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_8 \vee q_9 \\ f_{5_8} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_8 \vee q_9 \end{cases}$$

9) При $z_1 = 0$, $z_2 = 1$, $z_3 = z_4 = z_5 = 0$ (рис. 1, з)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_9} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_8 \\ f_{5_9} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_8 \end{cases}$$

10) При $z_1 = 0$, $z_2 = 1$, $z_3 = z_4 = 0$, $z_5 = 1$ (рис. 1, и)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_10} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_9 \\ f_{5_10} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_9 \end{cases}$$

11) При $z_1 = 0$, $z_2 = 1$, $z_3 = 0$, $z_4 = 1$, $z_5 = 0$ (рис. 1, й)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_11} = x_1 \vee q_2 \vee x_3 \vee x_4 \vee q_5 \vee q_7 \vee q_8 \vee q_{10} \\ f_{5_11} = x_1 \vee q_2 \vee x_3 \vee x_4 \vee q_5 \vee q_7 \vee q_8 \vee q_{10} \end{cases}$$

12) При $z_1 = 0$, $z_2 = 1$, $z_3 = 0$, $z_4 = 1$, $z_5 = 1$ (рис. 1, к)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_12} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_8 \vee q_9 \vee q_{10} \\ f_{5_12} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_8 \vee q_9 \vee q_{10} \end{cases}$$

13) При $z_1 = 0$, $z_2 = 1$, $z_3 = 1$, $z_4 = z_5 = 0$ (рис. 1, л)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_13} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_{10} \\ f_{5_13} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_{10} \end{cases}$$

14) При $z_1 = 0$, $z_2 = z_3 = 1$, $z_4 = 0$, $z_5 = 1$ (рис. 1, м)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_14} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_8 \vee q_{10} \\ f_{5_14} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \vee q_8 \vee q_{10} \end{cases}$$

15) При $z_1 = 0$, $z_2 = z_3 = z_4 = 1$, $z_5 = 0$ (рис. 1, н)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_15} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_8 \vee q_9 \vee q_{10} \\ f_{5_15} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_8 \vee q_9 \vee q_{10} \end{cases}$$

16) При $z_1 = 0$, $z_2 = z_3 = z_4 = z_5 = 1$ (рис. 1, п)

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_{4_16} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_7 \vee q_8 \vee q_9 \vee q_{10} \\ f_{5_16} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_7 \vee q_8 \vee q_9 \vee q_{10} \end{cases}$$

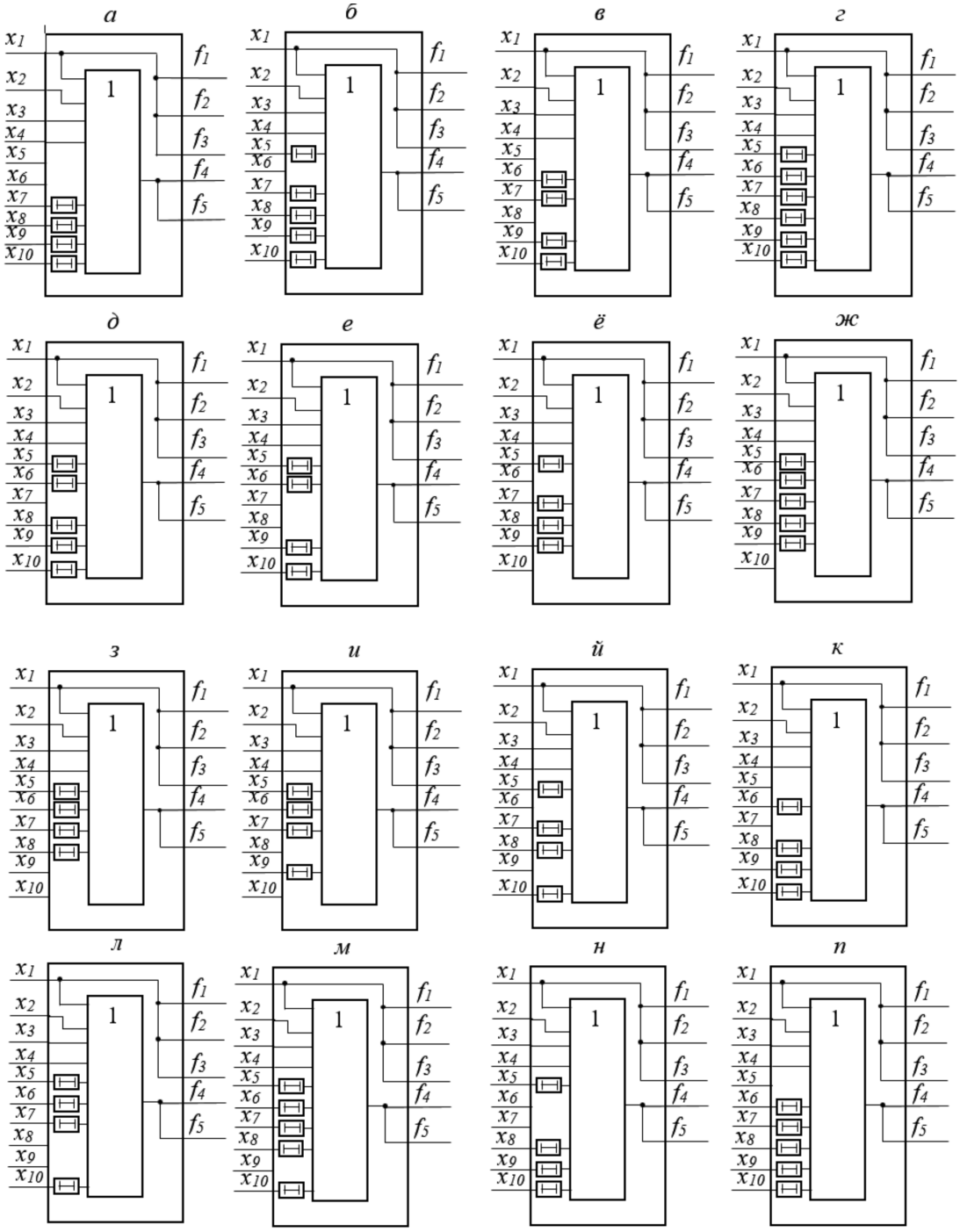

Рис. 1. Модель автоматического картирования

Здесь нам представлен построенный алгоритм элементарных вычислений для 16 блоков ЭВ (элементарных вычислений), каждое из которых соответствует случаю корректировки значений управляющего сигнала при разряде z. Оттуда соединение этих блоков вместе также строится по общей формуле

$$f_1 = x_1,$$

$$f_2 = x_1,$$

$$f_3 = x_1,$$

$$f_{4_T} = (x_1.\overline{z}_1) \vee (x_2.\overline{z}_1) \vee (x_3.\overline{z}_1) \vee (x_4.\overline{z}_1) \vee (q_5.\overline{z}_1.z_2.\overline{z}_5) \vee (q_5.\overline{z}_1.\overline{z}_2.z_5) \vee$$
$$(q_5.\overline{z}_1.z_3.\overline{z}_5) \vee (q_5.\overline{z}_1.\overline{z}_4.z_5) \vee (q_6.\overline{z}_1.z_2.\overline{z}_4) \vee (q_6.\overline{z}_1.\overline{z}_2.\overline{z}_3.z_4) \vee (q_6.\overline{z}_1.z_3.\overline{z}_4) \vee$$
$$(q_6.\overline{z}_1.z_4.z_5) \vee (q_7.\overline{z}_1.z_2.\overline{z}_4) \vee (q_7.\overline{z}_1.\overline{z}_2.z_4) \vee (q_7.\overline{z}_1.z_3.z_4.z_5) \vee (q_7.\overline{z}_1.\overline{z}_3.\overline{z}_4) \vee$$
$$(q_7.\overline{z}_1.\overline{z}_3.\overline{z}_5) \vee (q_8.\overline{z}_1.z_2.z_3.z_5) \vee (q_8.\overline{z}_1.z_2.\overline{z}_3.\overline{z}_5) \vee (q_8.\overline{z}_1.z_2.z_4) \vee$$
$$(q_8.\overline{z}_1.\overline{z}_2.z_3.\overline{z}_5) \vee (q_8.\overline{z}_1.\overline{z}_2.\overline{z}_3.\overline{z}_4) \vee (q_8.\overline{z}_1.z_4.z_5) \vee (q_9.\overline{z}_1.\overline{z}_2) \vee (q_9.\overline{z}_1.z_3.z_4) \vee \quad (*)$$
$$\vee (q_9.\overline{z}_1.\overline{z}_3.z_5) \vee (q_{10}.\overline{z}_1.z_2.z_3) \vee (q_{10}.\overline{z}_1.\overline{z}_2.\overline{z}_4) \vee (q_{10}.\overline{z}_1.\overline{z}_3.z_4),$$

$$f_{5_T} = (x_1.\overline{z}_1) \vee (x_2.\overline{z}_1) \vee (x_3.\overline{z}_1) \vee (x_4.\overline{z}_1) \vee (q_5.\overline{z}_1.z_2.\overline{z}_5) \vee (q_5.\overline{z}_1.\overline{z}_2.z_5) \vee$$
$$(q_5.\overline{z}_1.z_3.\overline{z}_5) \vee (q_5.\overline{z}_1.\overline{z}_4.z_5) \vee (q_6.\overline{z}_1.z_2.\overline{z}_4) \vee (q_6.\overline{z}_1.\overline{z}_2.\overline{z}_3.z_4) \vee (q_6.\overline{z}_1.z_3.\overline{z}_4) \vee$$
$$(q_6.\overline{z}_1.z_4.z_5) \vee (q_7.\overline{z}_1.z_2.\overline{z}_4) \vee (q_7.\overline{z}_1.\overline{z}_2.z_4) \vee (q_7.\overline{z}_1.z_3.z_4.z_5) \vee (q_7.\overline{z}_1.\overline{z}_3.\overline{z}_4) \vee$$
$$(q_7.\overline{z}_1.\overline{z}_3.\overline{z}_5) \vee (q_8.\overline{z}_1.z_2.z_3.z_5) \vee (q_8.\overline{z}_1.z_2.\overline{z}_3.\overline{z}_5) \vee (q_8.\overline{z}_1.z_2.z_4) \vee$$
$$(q_8.\overline{z}_1.\overline{z}_2.z_3.\overline{z}_5) \vee (q_8.\overline{z}_1.\overline{z}_2.\overline{z}_3.\overline{z}_4) \vee (q_8.\overline{z}_1.z_4.z_5) \vee (q_9.\overline{z}_1.\overline{z}_2) \vee (q_9.\overline{z}_1.z_3.z_4) \vee$$
$$\vee (q_9.\overline{z}_1.\overline{z}_3.z_5) \vee (q_{10}.\overline{z}_1.z_2.z_3) \vee (q_{10}.\overline{z}_1.\overline{z}_2.\overline{z}_4) \vee (q_{10}.\overline{z}_1.\overline{z}_3.z_4).$$

Общая формула выше – это формула для настраиваемой вычислительной среды. Для каждого набора переменной *z* и входного сигнала эта модель будет получать выходной сигнал, соответствующий 1 из 16 ЭВ, указанных выше. Для каждого набора формул построение модели автоматического отображения будет соответствовать приведенным выше 16 моделям вычислительных компонентов. Например, набор формул 1 будет соответствовать модели 1 (а), набор формул 2 соответствует модели 1 (б)… набор формул номер 16 соответствует модели 1 (п). Наборы этих формул с 1 по 16 определяют набор формул, генерируемых автоматами, и алгоритм настройки для каждого из них. Регулируя значение сигнала на его входной части, можем использовать некоторые преобразователи, чтобы автоматически преобразовывать значения выходного сигнала, поступающего с другого входа, в выходной сигнал [1, 13].

Такая настраиваемая модель вычислительной среды создается в программном обеспечении Matlab, как показано на рис. 2 ниже, включая 10 входов x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, 5 управляющих сигналов разряда z_1, z_2, z_3, z_4, z_5 и 5 выходов f_1, f_2, f_3, f_4, f_5.

В зависимости от смоделированной спроектированной среды, элементарные вычисления генерируются для соответствия этой среде.

Эти модели структур Tr-struct действуют как ячейки во всем облаке точек (пространстве, в котором движется робот). В этом исследовании метод Octomap использовался в комбинации. Один из самых эффективных методов создания гибких, постоянно обновляющихся и компактных карт. Начните группировать 4 модели клеток в группу. Когда элемент конструкции изменяется, группа меняет сигнал одновременно с ним. Предполагая, что первый

элемент в группе получает значение входного сигнала 1, тогда выходной сигнал первого элемента также будет равен 1, и выход остальных трех элементов также получит сигнал 1. Если элемент в группе принимает значение входного сигнала 0, выходной сигнал участников группы будет нулевым. Таким образом, отображение для робота будет более точным, быстрым и эффективным. Кроме того, этот подход обеспечивает производительность за счет моделирования последовательных данных при минимизации требований к памяти. Кроме того, метод Octomap обеспечивает максимальную гибкость с точки зрения площади и разрешения карты. Потому что этот метод выполняет оценку вероятности для обеспечения возможности обновления и работы с датчиками шума. Octomap обеспечивает эффективный экран сетки в 3D с октодеревьями. Эта модель представлена на рис. 3 [14–16].

2. Новый метод SLAM и имитационного моделирования

Во-первых, создается совершенно произвольная среда для перемещения робота в имитируемой модели Gazebo. Установленный робот Hector-Quadrotor перемещается в среде с препятствиями, спроектированными, как показано на рис. 4 [14].

Когда робот перемещается через систему координат, он будет отслеживать свое положение (получать параметры положения робота) и получать параметры расстояния робота от препятствия через лазерный датчик. Здесь метод SLAM построен следующим образом: предполагается, что размер робота равен размеру ячейки. Вышеупомянутая проектируемая среда имеет размер 12x12 ячеек. Обозначим размер робота как «sr»; ячейку, содержащую препятствие – «CC». По мере движения робота в каждой системе координат отображается его положение, а лазерный датчик указывает расстояние от робота до препятствия. Если расстояние больше, чем k * sr, тогда расстояние между роботом и ячейкой «CC» также будет k * sr (k – действительное число больше 0). Оттуда определяем ячейку, содержащую препятствие. Размерность (или Значение) этих ячеек установлена в 1, у ячеек без препятствий она устанавливается в ноль. Таким образом, при установке на Matlab результат, показанный на рис. 5, является правильной картой для робота, когда он движется в неизвестной среде.

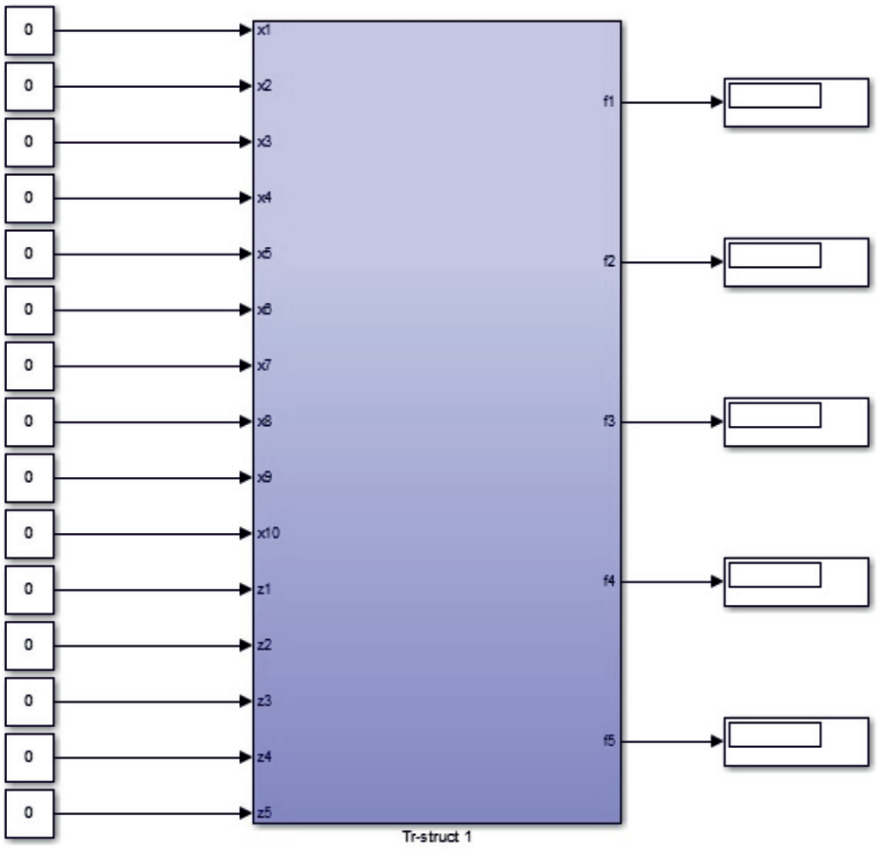

Рис. 2. Модель элементарных вычислений после проектирования

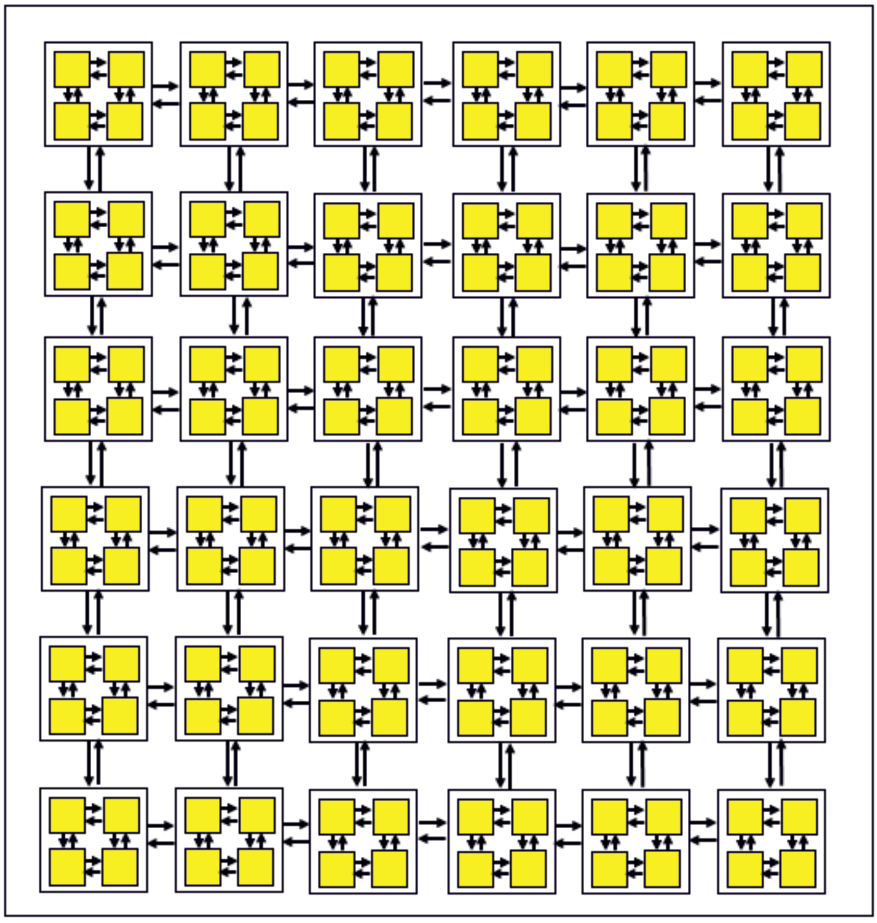

Рис. 3. Модель перестраиваемой вычислительной среды

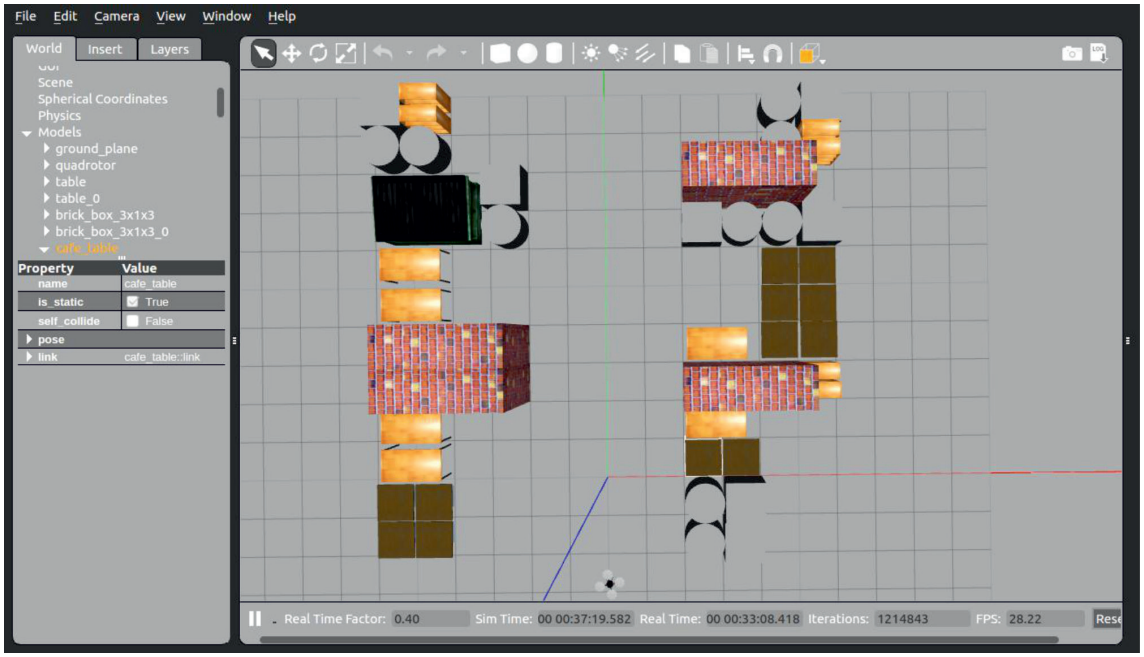

Рис. 4. Искусственная среда для квадрокоптера Hector-Quadrotor

Рис. 5. Карта робота получается перемещением на последнюю позицию

Метод группировки ячеек, описанный выше, реализован в соответствии с методом Octomap для минимизации памяти для элементарных вычислений в реконфигурируемой вычислительной среде, размещаемой на роботе по мере его движения. Благодаря универсальности и компактности робототехнических систем включение Octomap для развертывания карт позволит роботу быстро определять области с препятствиями.

Полученную карту можно визуализировать, как показано на рис. 6.

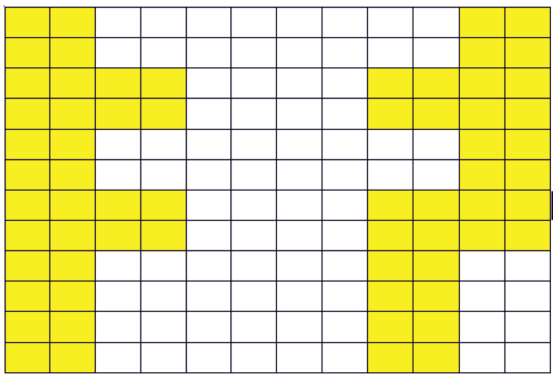

Рис. 6. Карта движения дрона Hector-Quadrotor в среде Gazebo

Создание нового алгоритма для проектирования реконфигурируемой вычислительной среды – это новый подход к SLAM. С помощью этого метода робот может легко получать карту и постоянно обновлять ее, перемещаясь в неизвестном пространстве или в окружающей среде. Таким образом была создана новая вычислительная парадигма, способствующая эволюции сегодняшнего SLAM.

Заключение

Создание нового алгоритма и модели ПВС для картирования UAV является результатом, который данная статья хочет продемонстрировать. Использование адаптируемой модели ПВС на самоходных роботах для картирования ими окружения внесло значительный вклад в разработку новой методологии SLAM. Эта методология улучшает скорость обработки, а также производительность вычислений.

Список литературы

1. Daniele Giordan, Marc S. Adams, Irene Aicardi, Maria Alicandro, Paolo Allasia, Marco Baldo, Pierluigi De Berardinis, Donatella Dominici, Danilo Godone, Peter Hobbs, Veronika Lechner, Tomasz Niedzielski, Marco Piras, Marianna Rotilio, Riccardo Salvini, Valerio Segor, Bernadette Sotier, Fabrizio Troilo. The use of unmanned aerial vehicles (UAVs) for engineering geology applications. Bulletin of Engineering Geology and the Environment. 2020. V. 79. P. 3437–3481. DOI: 10.1007/s10064-020-01766-2.

2. Pedro Castillo-García, Laura Elena Muñoz Hernandez, Pedro García Gil. Indoor Navigation Strategies for Aerial Autonomous Systems. 2017. P. 3–30.

3. Nguyen Hoang Thuy Trang, Shydlouski S. Situations in Construction of 3D Mapping for Slam. VIII International Scientific and Practical Conference «Information and Measuring Equipment and Technologies». MATEC Web Conf. 2018. V. 155. DOI: 10.1051/matecconf/201815501055.

4. Nguyen Hoang Thuy Trang, Shydlouski S. Tunable computing Slam navigation environments. IOP Conference Series: Materials Science and Engineering, III International Conference «Cognitive Robotics» (22–24 November 2018). Tomsk, 2018. V. 516. DOI: 10.1088/1757-899X/516/1/012053.

5. Шашев Д.В., Шидловский С.В. Морфологическая обработка бинарных изображений с использованием перестраиваемых вычислительных сред. 2015. С. 19–26.

6. Гонсалес Р., Вудс Р. Цифровая обработка изображений / Пер. с англ. М.: Техносфера, 2005. С. 1072.

7. Иванов В.А., Киричук В.С. Особенности работы алгоритмов обнаружения упавшего человека по последовательности изображений сцены // Автометрия. 2011. Т. 47. № 2. С. 15–25.

8. Шапиро Л., Стокман Дж. Компьютерное зрение. 2006. 752 с.

9. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультиконвейерные вычислительные структуры. Ростов н/Д.: Издательство ЮНЦ РАН, 2008. 393 с.

10. Шидловский С.В. Автоматическое управление. Реконфигурируемые системы: учеб. пособие. Томск: Изд-во Том. ун-та, 2010. 168 с.

11. Шидловский С.В. Автоматическое управление. Перестраиваемые структуры. Томск: Томский государственный университет, 2006. 288 с.

12. Boolean algebra. [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Boolean_algebra (дата обращения: 08.10.2020).

13. Хорошевский В.Г., Курносов М.Г., Мамойленко С.Н. Пространственно-распределенная многокластерная вычислительная система: архитектура и программное обеспечение // Вестн. ТГУ. Менеджмент, компьютерная инженерия и информатика. 2011. № 1 (14). С. 79–84.

14. Roman Lavrenov, Aufar Zakiev, Evgeni Magid. Automatic mapping and filtering tool: From a sensor-based occupancy grid to a 3D Gazebo octomap. 2017. P. 1. DOI: 10.1109/ICMSC.2017.7959469.

15. Армин Хорнунг, Кай М. Вурм, Марен Бенневиц, Кирилл Стахнисс, Вольфрам Бургард. OctoMap: эффективная вероятностная структура трехмерного отображения на основе октодеревьев, 2013. С. 189–206. DOI: 10.1007/s10514-012-9321-0.

16. Джеркер Бергстрем. Планирование пути с регионами с взвешенной стеной с использованием OctoMap. 2018. С. 7–8.