JOURNAL OF MARITIME RESEARCH

Vol XXI. No. II (2024) pp 146–151

ISSN: 1697-4840, **www.jmr.unican.es**

Reconfigurable Computing Environment in Mobile Robot Navigation

Trang Nguyen Hoang Thuy[1,*], Cuong Nguyen The[2]

ARTICLE INFO

Article history:
Received 5 Dec 2023;
in revised from 18 Jan 2024;
accepted 27 Mar 2024.

Keywords:
reconfigurable computing environment (RCE), tunable computing environment, robotic, information technology, tunable automat, reconfigurable automat.

ABSTRACT

Navigational robots use technology to navigate movement and orient the work plan. The navigation technology used in the robot allows to decide how to move, identify and plan the robot's work to adapt to the external environment. The operation of the navigational robot is based on the control algorithm built into the controller of the robot. Depending on the environmental conditions received by the robot, the controller will decide the direction of movement for the robot. However, environmental conditions are often customizable, so to have the right direction of movement requires the algorithm and control system installed on each robot to be flexible. With the above approach, this article focuses on giving algorithms for reconfigurable computing environment models and building maps that robot moves in 2D space.

1. Introduction.

Robot is considered an integrated system of hardware and software, capable of independent operation. In general, when building robots, focus on the basic field that is to propose algorithms or operating scenarios for robots and suitable hardware architecture models for robots. Navigational robot is no exception, the navigation algorithm for robots is considered as the basis for building the map that the robot moves in 2D space. The environment in which the robot moves is usually represented in 2D space. In this space, the robot can move in different directions depending on the position of the obstacles around it. The robot's behavior is based on its current location and its surroundings. The process of building a map that the robot moves in 2D space is established by:

- Structuring robot operations.
- Mapping of robot movement in state space.
- Building a map of the environment in which the robot moves.

In order to structure the robot's operations as well as map the robot's movement in the state space, it is necessary to determine the appropriate navigation technology for the robot. Popular navigation technologies include: sensor navigation, optical camera, or laser navigation. The navigation technique using sensors is quite common due to its simple structure, but the effectiveness of this navigation method is limited when the received signal is noisy. Meanwhile, the limitations of robots using optical cameras are the light environment and the complexity of image processing. In contrast to the above two methods, the robots use laser technology that allows the detection of obstacles by the laser quickly based on the frequency characteristic of the light wave. As a result, it is possible to determine the position and distance from the robot to the obstacles in real time. With a 360-degree rotation mechanism that allows the robot to continuously scan its surroundings, the floor covering performance of robots using this technology is highly effective. In addition, with the development of SLAM technology, the robot information obtained from the laser sensor allows to build detailed and customized maps of the robot's movements

[1]Vietnam Maritime University (484 Lach Tray - Kenh Duong, Le Chan - Hai Phong, Vietnam), E-mail: trangnht@vimaru.edu.vn.

[2]Vietnam Maritime University (484 Lach Tray - Kenh Duong, Le Chan - Hai Phong, Vietnam), E-mail: cuongntit@vimaru.edu.vn.

[*]Corresponding author: Trang Nguyen Hoang Thuy. E-mail Address: trangnht@vimaru.edu.vn.

according to any environment. Through structuring the robot's operations to form basic elementary computings, the state space of the robot will be established. Thereby, a reconfigurable automat model of the robot's movement will also be built (Nguyen Trang Hoang & Shidlovskiy, 2018).

2. Materials and Methods.

2.1. Method of building reconfigurable automats and the reconfigurable computing environment in mobile robot navigation.

The process of building an automat for a robot begins by structuring the robot's operation using logic functions based on the current state of the robot and environmental parameters (obstacles). The process of building an automat for a robot begins by structuring the robot's operation using logic functions based on the current state of the robot and environmental parameters (obstacles). The mapping of the robot's movement on the state space matrix is performed based on the current state and possible movements of the robot, then the automat represents each state and reconfigurable automats for the robot will be built.

With the hard control method: the quantity and structure of logic elements are set when forming the automat based on the structure of the input signal. With the soft control method, the change of the output value f can be achieved by changing the value of input signals (x_1, x_2,...), setting signals (z_1, z_2, ...) or the connection between logic elements in the circuit. This is completely achievable based on the control algorithm installed for each automat, then the automat can be reconfigured based on its input and output automatic mapping.

A reconfigurable computing environment (RCE) is a homogeneous environment, consisting of structurally identical cells connected together in the same way as neighboring cells, which can be reconfigured by setting signals. The reconfigurable automats receive input signals and go in a certain direction, performing internal functions to give the outputs of each cell in the environment.

By changing the connections between the basic logic elements (AND, OR, NOT) that configure the RCE, the hardware system can be restructured, through which the operation of the control algorithm becomes flexible and adapts to the robot's operation, allowing the robot to adapt to the environment, ensuring accuracy and reliability. Proposing the RCE model to be used in navigation robots in particular and autonomous robots in general can lead to a new trend in intelligent control and handling techniques.

2.2. Method of building the process for handling navigation information for mobile robots and algorithm to build reconfigurable computing environment model for mobile robot.

The robot's movement depends on its current state and its actions and next states. From there, the robot's operating state space is formed in 2D and 3D environments. This space is considered as the basis for building control algorithms as well as reconfigurable automats for the Robot, forming the process of handling navigation information:

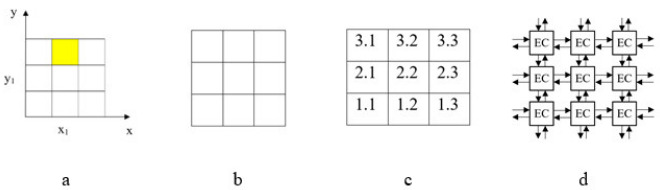
1. Structuring the robot's operation by logic functions: 2D, 3D space partitioning by matrices. Determining the distance between the position of the robot and the obstacle. Formalizing robot motion by input/output signals and their connections.
2. Building a function representing the input/output connection (position of robot and obstacle / next position of robot and direction of movement).
3. Structuring functions by basic logic elements AND, OR, NOT, each element represents a state of the robot.
4. Connecting elements to form an elementary computing (EC) (reconfigurable automat).
5. Building the active state space for the robot (matrix m × n elements) by connecting the EC. The output value of each EC element corresponds to the presence or absence of obstacles in the environment.
6. Building a reconfigurable computing environment model – RCE for robots. RCE operation will be established through the state space.
7. Building map.

From the navigation information handling process, the basic steps in the robot building process can be seen: proposing control algorithms, reconfigurable automat models for robots, building a reconfigurable computing environment. From there, building 2D, 3D state space maps for robots.

The process of structuring the robot's operation by logical functions:

Assume that the robot is moved in two-dimensional space with the Oxy axis. Performing a two-dimensional space partition Oxy according to a two-dimensional matrix n×m (n rows, m columns) (Figure 1a). Without losing the generality of the problem, consider the robot's movement environment in a 3×3 matrix (Figure 1b), and compare its movement with the information processing in RCE. Each cell will correspond to the size of the robot in the environment. The positions of the cells will be numbered as shown in Figure 1c.

Figure 1: Representation of the position of the obstacle in the space of the robot's movement, the environment of the robot's movement, the RCE representation for the 3x3 environment. a - the coordinate position of the obstacle in the environment; b - 3x3 environment; c - 3x3 environment; d - RCE representation for a 3x3 environment.

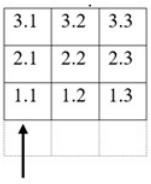


Source: Authors.

Testing the cases for each cell at that time to see if there is an obstacle in that cell, and the connection between that cell and its neighbors. Comparing the space around the robot (Figure 1a) with the two-dimensional representation of RCE (Figure 1d).

The robot moves from the outside in the direction from position 1.1 to position 3.1 (Figure 2).

Figure 2: The robot position is in the lower left corner, the robot goes straight from the outside of the cell 1.1 to the end of the cell 3.1.



Source: Authors.

Assuming position 1.2 has an obstacle, then position 1.3 will also receive the value of one as having an obstacle because it is hidden behind position 1.2. Likewise with positions 2.2, 2.3, 3.2, 3.3. So the logic function built for the positions will be represented as follows:

Position 1.2: $f_1 = x_1, f_2 = x_1, f_3 = x_1, f_4 = x_1 \vee x_2 \vee x_5$
Position 1.3: $f_1 = x_1, f_2 = x_1, f_3 = x_1, f_4 = x_1 \vee x_2 \vee x_4 \vee x_5$
Position 2.2: $f_1 = x_1, f_2 = x_1, f_3 = x_1,\ f_4 = x_1 \vee x_2 \vee x_5$
Position 2.3: $f_1 = x_1, f_2 = x_1, f_3 = x_1, f_4 = x_1 \vee x_2 \vee x_4 \vee x_5$
Position 3.2: $f_1 = x_1, f_2 = x_1, f_3 = x_1, f_4 = x_1 \vee x_2 \vee x_4 \vee x_5$
Position 3.3: $f_1 = x_1, f_2 = x_1, f_3 = x_1, f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$

From the logic functions obtained above, it can be seen that there are 6 cases and there are 3 overlapping cases, so 3 different cases will be received, so the installer (z_1, z_2, z_3) is used to control. Match each case with the resulting set z cases:

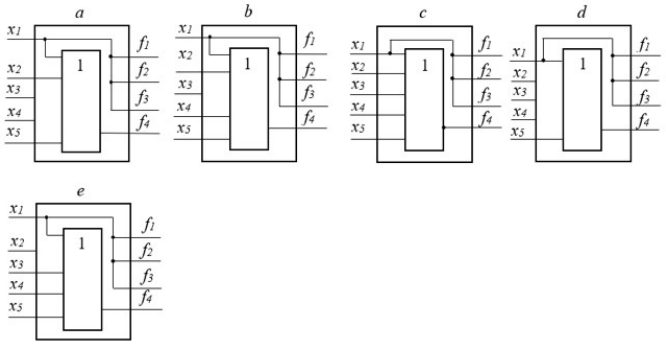
1) If $z_1 = 0, z_2 = 0, z_3 = 0$
$f_1 = x_1, f_2 = x_1, f_3 = x_1$
$f_4 = x_1 \vee x_2 \vee x_5$
2) If $z_1 = 0, z_2 = 0, z_3 = 1$
$f_1 = x_1, f_2 = x_1, f_3 = x_1$
$f_4 = x_1 \vee x_2 \vee x_4 \vee x_5$
3) If $z_1 = 0, z_2 = 1, z_3 = 0$
$f_1 = x_1, f_2 = x_1, f_3 = x_1$
$f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$

Similarly consider the remaining cases, obtain the general function after minimization:

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_4 = (x_1.\bar{z}_1) \vee (x_1.\bar{z}_2.\bar{z}_3) \vee (x_2.\bar{z}_1.\bar{z}_2) \vee (x_2.\bar{z}_1.\bar{z}_3) \vee (x_3.z_1.\bar{z}_2.\bar{z}_3) \vee \\ (x_3.\bar{z}_1.z_2.\bar{z}_3) \vee (x_4.z_1.\bar{z}_2.\bar{z}_3) \vee (x_4.\bar{z}_1.z_2.\bar{z}_3) \vee (x_4.\bar{z}_1.\bar{z}_2.z_3) \vee (x_5.\bar{z}_1) \vee (x_5.\bar{z}_2.\bar{z}_3) \end{cases}$$

On the basis of the synthesized automat, the RCE model will be built in the case of 2D space. In this space, as mentioned, the RCE environment of the robot's movement is given in a 3×3 matrix. In order to save storage space, an Octomap algorithm was used, grouping 4 cells in a 2×2 matrix into 1 group (Hornung et al., 2013). When the robot detects 1 cell

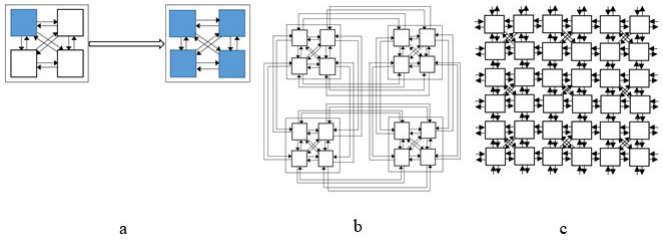
Figure 3: Basic elementary computings (EC) representing robot states in 2D space.



Source: Authors.

with obstacles, the remaining 3 cells in the group will be considered as containing obstacles. Therefore, cells with obstacles will all receive a value of one, cells without obstacles will receive a value of zero. By convention, the cell with an obstacle will have a logical value of one, otherwise will have a logical value of zero, if an element takes on a value of one, the other three elements in this group will also have a value of one, which means the area of these four elements will be the area areas with obstacles (Figure 4-a). Elements in one group are connected to the corresponding elements of another group and receive the values returned by the elements of the other group at the output of the corresponding elements of this group (Figure 4-b). The reconfigurable computing environment (RCE) built in 2D when the robot moves is shown in Figure 4-c.

Figure 4: The connection between elements in the same group and with neighboring groups (a, b), the space of RCE (c) in the 2D environment.

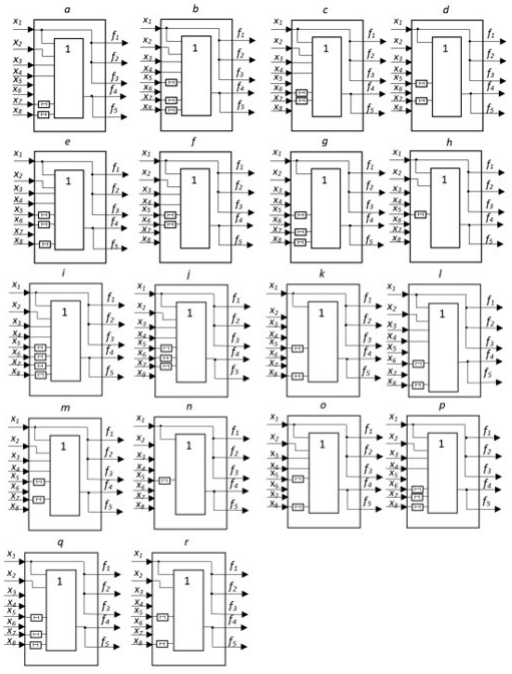


Source: Authors.

Input x_1 of elements in the same group will receive the processed information from the robot and the distance from the robot to the environment to determine if the location has obstacles or not. Inputs x_2, x_3, x_4 receive connections from the corresponding outputs of surrounding elements in the same group, inputs x_5, x_6, x_7, x_8 receive connections from the output corresponding to the elements of neighboring groups. Each element in the group will have 5 outputs, where the output f_1, f_2, f_3 will be the output of the elements in the group, output f_4 and output f_5 give the final result of the element there in the group, where f_5 gives the result displayed on the map. By connecting the input and output signals between the elements in the group,

and at the same time considering the connection between the elements and neighboring groups, the state space of the robot's movement is characterized by 18 representative element is built (Figure 5).

Figure 5: Automatic mapping for 2D environment: x_i – input information; f – output of automat; f_4, f_5 - output of connected automats.



Source: Authors.

For each automatic mapping case illustrated in Figure 5, each of these cases will be assigned a corresponding of z-setting codes. After that, a synthetic automat (EC) will be performed, the map formation algorithm on the RCE environment will be built based on the identical EC. To handle automatic mapping in 2D environment, the developed system of synthetic logic formulas for EC is given in equation (1).

$$
\begin{cases}
f_1 = x_1 \\
f_2 = x_1 \\
f_3 = x_1 \\
f_4 = (x_1.\bar{z}_1) \vee (x_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_2.\bar{z}_1.z_2.z_3.z_4) \vee (x_2.\bar{z}_1.z_2.\bar{z}_3.z_5) \vee (x_2.\bar{z}_1.\bar{z}_2.z_3.z_5) \\
\vee (x_2.\bar{z}_1.\bar{z}_3.\bar{z}_4) \vee (x_2.\bar{z}_1.\bar{z}_4.\bar{z}_5) \vee (x_2.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_3.\bar{z}_1.z_2.z_3.z_4) \vee (x_3.\bar{z}_1.z_2.\bar{z}_3.z_5) \\
\vee (x_3.\bar{z}_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_3.\bar{z}_1.\bar{z}_2.\bar{z}_4) \vee (x_3.\bar{z}_1.\bar{z}_4.\bar{z}_5) \vee (x_4.\bar{z}_1.z_2.z_3.z_4) \vee (x_4.\bar{z}_1.z_2.\bar{z}_3.z_5) \vee \\
(x_4.\bar{z}_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_4.\bar{z}_1.\bar{z}_2.\bar{z}_4) \vee (x_4.\bar{z}_1.\bar{z}_4.\bar{z}_5) \vee (q_5.z_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (q_5.\bar{z}_1.z_2.\bar{z}_5) \vee \\
(q_5.\bar{z}_1.\bar{z}_2.z_5) \vee (q_5.\bar{z}_1.z_3.\bar{z}_5) \vee (q_5.\bar{z}_1.\bar{z}_4.z_5) \vee (q_6.\bar{z}_1.z_2.\bar{z}_3.\bar{z}_4) \vee (q_6.\bar{z}_1.z_2.z_4.z_5) \vee \\
(q_6.\bar{z}_1.\bar{z}_2.z_3.\bar{z}_4) \vee (q_6.\bar{z}_1.\bar{z}_2.\bar{z}_3.z_4.\bar{z}_5) \vee (q_7.\bar{z}_1.z_2.z_3.z_4.z_5) \vee (q_7.\bar{z}_1.z_2.\bar{z}_4.\bar{z}_5) \vee \\
(q_7.\bar{z}_1.\bar{z}_2.\bar{z}_3) \vee (q_7.\bar{z}_1.\bar{z}_2.z_4.\bar{z}_5) \vee (q_7.\bar{z}_1.\bar{z}_3.\bar{z}_4) \vee (q_7.\bar{z}_2.\bar{z}_3.\bar{z}_4.\bar{z}_5) \vee (q_8.z_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \\
\vee (q_8.\bar{z}_1.z_2.z_4) \vee (q_8.\bar{z}_1.\bar{z}_2.z_3.\bar{z}_5) \vee (q_8.\bar{z}_1.\bar{z}_2.\bar{z}_3.\bar{z}_4) \vee (q_8.\bar{z}_1.\bar{z}_3.\bar{z}_4.\bar{z}_5) \\
f_5 = (x_1.\bar{z}_1) \vee (x_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_2.\bar{z}_1.z_2.z_3.z_4) \vee (x_2.\bar{z}_1.z_2.\bar{z}_3.z_5) \vee (x_2.\bar{z}_1.\bar{z}_2.z_3.z_5) \vee \\
(x_2.\bar{z}_1.\bar{z}_3.\bar{z}_4) \vee (x_2.\bar{z}_1.\bar{z}_4.\bar{z}_5) \vee (x_2.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_3.\bar{z}_1.z_2.z_3.z_4) \vee (x_3.\bar{z}_1.z_2.\bar{z}_3.z_5) \vee \\
(x_3.\bar{z}_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_3.\bar{z}_1.\bar{z}_2.\bar{z}_4) \vee (x_3.\bar{z}_1.\bar{z}_4.\bar{z}_5) \vee (x_4.\bar{z}_1.z_2.z_3.z_4) \vee (x_4.\bar{z}_1.z_2.\bar{z}_3.z_5) \vee \\
(x_4.\bar{z}_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (x_4.\bar{z}_1.\bar{z}_2.\bar{z}_4) \vee (x_4.\bar{z}_1.\bar{z}_4.\bar{z}_5) \vee (q_5.z_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee (q_5.\bar{z}_1.z_2.\bar{z}_5) \vee \\
(q_5.\bar{z}_1.\bar{z}_2.z_5) \vee (q_5.\bar{z}_1.z_3.\bar{z}_5) \vee (q_5.\bar{z}_1.\bar{z}_4.z_5) \vee (q_6.\bar{z}_1.z_2.\bar{z}_3.\bar{z}_4) \vee (q_6.\bar{z}_1.z_2.z_4.z_5) \vee \\
(q_6.\bar{z}_1.\bar{z}_2.z_3.\bar{z}_4) \vee (q_6.\bar{z}_1.\bar{z}_2.\bar{z}_3.z_4.\bar{z}_5) \vee (q_7.\bar{z}_1.z_2.z_3.z_4.z_5) \vee (q_7.\bar{z}_1.z_2.\bar{z}_4.\bar{z}_5) \vee \\
(q_7.\bar{z}_1.\bar{z}_2.\bar{z}_3) \vee (q_7.\bar{z}_1.\bar{z}_2.z_4.\bar{z}_5) \vee (q_7.\bar{z}_1.\bar{z}_3.\bar{z}_4) \vee (q_7.\bar{z}_2.\bar{z}_3.\bar{z}_4.\bar{z}_5) \vee (q_8.z_1.\bar{z}_2.\bar{z}_3.\bar{z}_5) \vee \\
(q_8.\bar{z}_1.z_2.z_4) \vee (q_8.\bar{z}_1.\bar{z}_2.z_3.\bar{z}_5) \vee (q_8.\bar{z}_1.\bar{z}_2.\bar{z}_3.\bar{z}_4) \vee (q_8.\bar{z}_1.\bar{z}_3.\bar{z}_4.\bar{z}_5)
\end{cases}
\tag{1}
$$

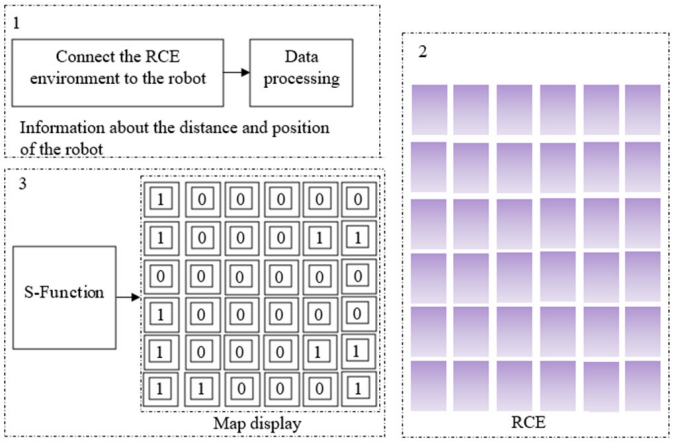
When processing the navigation information of a mobile robot in a 2D environment represented by the formula system (1), information about each cell is fed into the x input of the corresponding EC, which is adjusted by the tuning code z_i to perform automatic mapping on demand. With the help of the setup code, the EC will be set up to perform transformations according to the customized environment the robot moves in.

The construction algorithms in the research process of building maps for robots in 2D space were implemented in the reconfigurable computing environment (RCE). The navigation information processing based on the reconfigurable computing environment has been developed, RCE algorithms for processing 2D maps are performed line by line in the EC cycle corresponding to RCE.

3. Modeling simulation of robot operation based on reconfigurable computing environment.

From the study of modeling methods to simulate the robot's operation based on the reconfigurable computing environment structure. From there, it is proposed to model the robot's operation, a method to build a reconfigurable computing environment model in 2D space when the robot moves. The behavior of the robot in its state space is considered on the basis of a deterministic finite automat. However, these operations must be modeled and simulated in order to implement algorithms, from which a physical structure can be built for the robot. The selection of the reconfigurable computing environment model to simulate the robot's operation, from which the reconfigurable computing environment model is generated will include identical reconfigurable automats that can be signal change by setting signals. The output signals from the automats will be changed due to the change of the input signals based on the distance measurements from the sensor mounted on the robot and the position of the robot. Creating a RCE model will help evaluate the system more accurately, confirming that the given algorithms have higher reliability. Moreover, it is confirmed that this model can also be applied in practice.

Figure 6: RCE simulation model implements map processing algorithm in 2D environment.



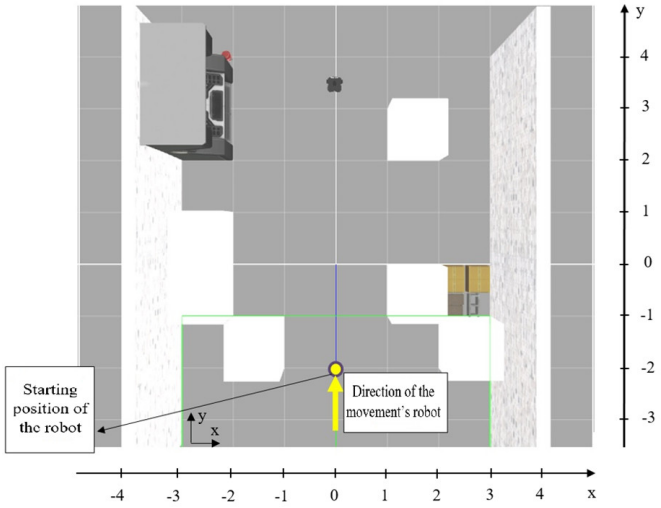
Source: Authors.

For this purpose, a library of corresponding electronic models has been developed in the Simulink environment, which are implemented in combinational circuits on the models of Boolean logic elements provided by the Simulink environment.

Using the connections of the structures shown in Figure 4-b, simulation models of RCE are designed in such a way that the cells in the RCE coincide with the blocks in the 2D space that the robot moves (Figure 6). The cells in the RCE environment will then be tested for their accuracy and operability to match those in the test environment in which the robot moves through space.

Testing RCE's integrated simulation model when connected to a moving robot model (Figure 7).

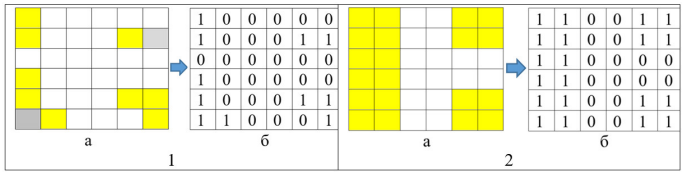
Figure 7: Gazebo environment simulation for moving Turtlebot Waffle 3 robot.



Source: Authors.

The simulation model developed by RCE for the 2D environment consists of 6 × 6 cells (Figure 7) that will be connected to a moving robot (simulating the environment of a moving robot in Figure 8 (1). In the case of using the 1×1-cell Octomap algorithm, the step-by-step motion map of the robot is shown in Figure.8 (1), the z-set signals for the corresponding EC in the RCE are shown in Table 1.

Figure 8: Map obtained in Matlab when the robot moves in Gazebo.



Source: Authors.

1 - In the case of the Octomap algorithm 1×1 cells; 2- In the case of the Octomap algorithm 2×2 cells.

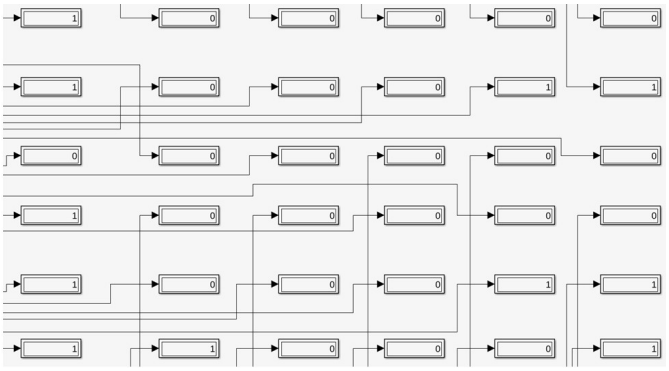
Table 1: Values of setting codes z corresponding to EC positions in RCE.

Competency in	School A		School B	
	Mean	**Interpretation**	**Mean**	**Interpretation**
Function 1: Navigation	2.02	Partly Competent	2.28	Partly Competent
Function 2: Cargo handling and stowage	1.65	Partly Competent	2.34	Partly Competent
Function 3: Controlling the operation of the ship and care for persons on board	1.79	Partly Competent	2.30	Partly Competent
As a whole	**1.87**	**Partly Competent**	**2.29**	**Partly Competent**

Source: Authors.

The resulting map in RCE is received exactly as the map in the environment in which the Robot moves in the case of the Octomap algorithm 1×1 cells.

Figure 9: Map obtained in Matlab when the robot moves in Gazebo in the case of the Octomap algorithm 1x1 cells.



Source: Authors.

Similarly, considering the option of using the 2×2-cell Octomap algorithm, the robot movement map is shown in Figure 8 (2), the z-setting code for the corresponding EC in the RCE is shown in Table 2.

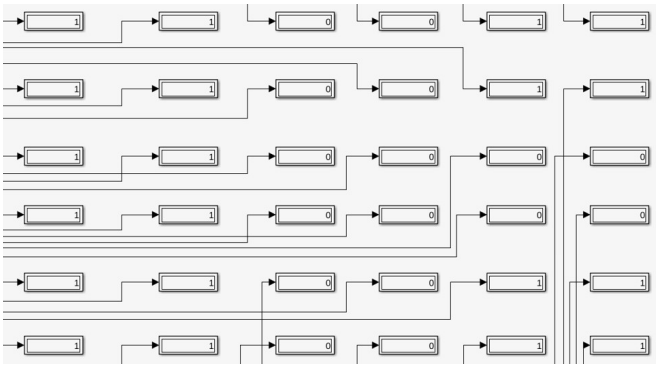
Table 2: Setting value of z-codes corresponding to EC positions in RCE.

y / x	1	2	3	4	5	6
1	00010	00010	00000	00000	00010	00010
2	00010	00010	00000	00000	00010	00010
3	00010	00010	00000	00000	01000	01000
4	00010	00010	00000	00000	01000	01000
5	00100	00100	00000	00000	00100	00100
6	00100	00100	00000	00000	00100	00100

Source: Authors.

The resulting map in RCE is received exactly as the map in the environment in which the Robot moves in the case of the Octomap algorithm 2×2 cells.

Figure 10: Map obtained in Matlab when the robot moves in Gazebo in the case of the Octomap algorithm 2x2 cells.



Source: Authors.

4. Discussion.

Previously, I gave a part of the results of my method in (Nguyen Trang Hoang, Nguyen Cuong The & Shidlovskiy, 2019), the authors (Kalyaev et al., 2008), (Kalyaev, Levin & Semernikov, 2012), (Shidlovskiy, 2006, 2010) presented a high-performance reconfigurable computing system application for digital signal processing as well as structural processing of reconfigurable systems. However, these methods are only in general form. In this paper, I present methods, algorithms as well as a new reconfigurable computing environment model built by myself as well as the results of the test program in the RCE environment and the real environment where the robot moves completely match each other.

Conclusions.

Because of the ability to reconstruct the logic functions in the algorithm for RCE, and flexibly adjust the setting signals of the automats inside the reconfigurable computing environment, the construction of the environment map for the mobile robot brings high efficiency, accuracy, opens up a new way to build maps for mobile robots.

Acknowledgements.

The research presented in this paper was supported by the Vietnam Maritime University. The authors also want to express our deep gratitude to the supervisor Professor Shidlovskiy Stanislav Victorovich

References.

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C. & Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34, 189–206. https://doi.org/10.1007/s10514-012-9321-0.

Kalyaev, I.A, Levin, I.I, & Semernikov, E.A. (2012). *Application of high-performance reconfigurable computing systems for digital signal processing.* Proceedings of the Russian Scientific and Technical Society of Radio Engineering, Electronics and Communications named after A.S. Popov. Collection of reports of the 14th International Conference ”Digital Signal Processing and its Application DSPA-2012”. Moscow, 377-381.

Kalyaev, I.A., Levin, I.I., Semernikov, E.A., & Shmoilov, V.I. (2008). *Reconfigurable multiconveyor computing structures.* Rostov, NY: Publishing House of the YUNTS RAS.

Nguyen Trang Hoang, Thuy & Shidlovskiy, S.V. (2018). Situations in Construction of 3D Mapping for Slam. *8th International Scientific and Practical Conference on Information and Measuring Equipment and Technologies, IME and T* 2017. Tomsk, Russian Federation, 155. DOI: 10.1051/matecconf/20-1815501055.

Nguyen Trang Hoang, Thuy, Nguyen Cuong, The & Shidlovskiy S.V. (2019). Tunable computing Slam navigation environments. *III International Conference “Cognitive Robotics”*. Tomsk, Russian Federation, 516.

Shidlovskiy, S. V. (2006). *Automatic control. Rebuilt structures*. Tomsk, 288-289.

Shidlovskiy, S. V. (2010). *Automatic control. Reconfigurable systems: a textbook*. Tomsk, 168-169.