The Method of the Kinematic Structure Reconfiguration of a Multifunctional Modular Robot Based on the Greedy Algorithm

Vyacheslav Petrenko
Institute of Information Technologies and Telecommunications
North Caucasus Federal University
Stavropol, Russia
vip.petrenko@gmail.com

Andrey Pavlov
Institute of Information Technologies and Telecommunications
North Caucasus Federal University
Stavropol, Russia
losde5530@gmail.com

Mikhail Gurchinsky
Institute of Information Technologies and Telecommunications
North Caucasus Federal University
Stavropol, Russia
gurmikhail@yandex.ru

Fariza Tebueva
Institute of Information Technologies and Telecommunications
North Caucasus Federal University
Stavropol, Russia
fariza.teb@gmail.com

Abstract—The aim of the article is to provide a quick reconfiguration of a large group of robots. To solve the problem, it is proposed to use a decentralized control in the form of a multi-agent system. Each agent performs movement planning based on a greedy algorithm. The simulation showed the effectiveness of the proposed method. In the future, it is planned to increase the reliability of the proposed method and its complexity for working with more complex configurations.

Keywords—modular robotics, self-reconfiguration, group control, multi-agent system

I. INTRODUCTION

Creating a modular robot with an adaptive (reconfigurable) kinematic structure (MRAKS) in our time is one of the most promising areas in robotics. MRAKS create using the same type of modules, which are combined into a one-piece design. The connection of modules of the same type allows building mechanisms that are completely different in their structure. MRAKS have significant advantages compared to mobile robots: higher reliability and overcoming various obstacles in complexity. MRAKS combines all the latest advances in robotics, mechatronics, control theory. The newest area of application for MRAKS is extreme robotics, whose goal is the creation and implementation of complexes for working in extreme situations [1, 2].

Reconfiguration, as well as the adaptability and scalability of the kinematic structure is obtained precisely because of the modularity of the construction of the MRAKS. At the same time, it is necessary to take into account the specifics of the tasks in view of the ambiguity of the state of the environment, changes from outside, as well as the current state of all existing subsystems. This set of functionality makes it imperative to develop intelligent control systems (ICS), which have a distributed hardware structure. These funds provide both the performance of such functions and the movement of the robot in previously unfamiliar environments, as well as the automatic connection of structures and algorithms for controlling the MRAKS using the self-learning mode. Due to the construction features, as well as high functional flexibility, the MRAKS defines a wide range of possibilities and applications: domestic sphere, special tasks of military units, search and rescue operations, monitoring and mapping the terrain, studying the surface structure of various planets and space bodies and much more.

In accordance with the tasks to be solved and the design features of modular robots, many ways have been developed to reconfigure the kinematic structure of a multifunctional modular robot. According to [3], the task of reconfiguring a modular robot can be formulated as follows: the initial and target configurations of the kinematic structure of the modular robot are given, it is necessary to determine the sequence of movements of each module necessary to form the target configuration.

Existing approaches to the management of robotic systems can be divided into two categories: centralized and decentralized management. Since modular robots can consist of a huge number of modules, a centralized algorithm can exceed the available memory on a single device with a significant increase in the number of modules, while a distributed algorithm will effectively scale the system.

The recent rise in popularity of modular robotics is due to the possibility of multiple use of robot modules to perform various tasks by reconfiguring the kinematic structure. Due to the specifics of the objective functions and environmental constraints affecting the process of reconfiguring the structure of a modular robot, many different configurations can be obtained. In [4], an analysis was made of possible kinematic configurations of the ModRED II robot using graph theory. The authors aimed to identify unique variants of the robot structures for a deeper investigation.

In [5], a path planning method based on graph theory is presented, which allows speeding up path planning for a robot reconfiguration. The kinematic structure is considered as a connected graph whose vertices are the modules of the robot. The search for the path between the initial and required positions of the module is carried out by means of a randomly growing tree algorithm. In [6], the problem of forming the kinematic structure of a modular robot is considered. To solve this problem, an approach based on cooperative co-evolutionary genetic algorithms for the formation of
configurations for homogeneous modular robots supporting self-reconfiguration is proposed.

The works [7, 8] are aimed at solving the problem of forming the kinematic structure of a modular robot in the absence of information about the global state of the modules when using a decentralized control system. The authors proposed a bio-inspired way to solve this problem through the use of the Lindenmayer system and the mathematical apparatus of cellular automata for synthesizing the rules of interaction between modules in order to form the required configuration.

Another approach to the study of the reconfiguration of the kinematic structure of a modular robot is the use of optimization algorithms. In this case, the main idea is to transform the problem of reconfiguration into the task of optimizing the multiple positions of the robot modules. Thus, in [9, 10], the problem of minimizing reconfiguration time as well as collision avoidance is considered, taking into account the limitations of the dynamic characteristics of robots. To solve this problem, the authors propose a hybrid approach based on the parameterization of the control and discretization of the formation of the governing law over time based on the particle swarm method.

The article [11] addresses the problem of moving around difficult terrain with a modular robot. In order to increase the travel time while reducing energy costs, a method for planning the path of modular robots in various territories in a deterministic environment using Q-learning and optimization algorithms is proposed. The result of the method is the optimal path of movement and the corresponding configuration of the modules on each segment of the path. The work [12] explores the problem of self-reconfiguration of modular underwater robots in order to reduce energy costs. A criterion of energy efficiency and a modification of the Basic Theta* algorithm are proposed for calculating the order of movement of the modules during the self-reconfiguration of the robot from the current structure to the final one.

In [13, 14], an optimization method is presented for controlling the formation of a group of robots in an environment with static and dynamic obstacles, taking into account the geometric parameters of robots. The proposed algorithm is distributed, but the efficiency of the method is observed only when managing a group of up to 16 robots.

Each of the considered methods has its advantages and disadvantages, its specific areas of application, as well as individual features that distinguish them from other methods. Analysis of existing solutions reflects the individuality of using the developed methods for a specific situation or device. Thus, the development of a method for reconfiguring the kinematic structure of a multifunctional modular robot based on a greedy algorithm is an actual task.

II. PROBLEM FORMULATION

The specificity of hardware platform considered as a separate MRAKS unit is the absence of a vision system for individual units, which does not allow the robot to move in a non-deterministic environment on its own. The second key point is the ability to move units between positions without communication with each other. The movement is based on its own navigation system. With long movements without coupling with other robots, the accumulated error can lead to the loss of the robot. Therefore, a robot without engaging with other units can move only for short distances. A large number of robots impose restrictions on wireless communication between robots. The computational resources of an individual robot do not allow the communication of an individual robot with all the others, or of all the robots with the control center. The disadvantage of the wired interface (contact) is the large propagation delays, since each robot is a repeater. It takes time to process information, as well as complex rules for routing information, in order not to block the air and not to transmit information several times. The lack of communication of each with each, as well as the computational power of a single unit, does not allow to calculate the trajectory of movement of all units based on their current positions and velocity vectors.

MRAKS has the following important features:

- Scalability. The number of MRAKS units can vary over a wide range.
- Members simplicity. The advantage of MRAKS is the ability to create a complex configuration to perform the target function from relatively simple units.
- Local interactions. With the exception of information about the target configuration obtained from the command center, all interactions between units are local.
- Control topology is distributed. The target configuration is set by the command center, while the basic control of the movement of units is carried out at the local level.
- Mostly homogeneous. All robots are the same.
- Autonomous. All units are able to act independently, so the loss of several robots does not lead to system failure.
- Cooperation. In the context of this article, the work of units is cooperative, since they have a common goal - building a target configuration that can only be achieved collectively.
- Awareness. From the point of view of reconfiguration, all information available to units is information about neighboring units.
- Coordination. Planning the movement of robots is carried out taking into account the planned movement of the nearest neighbors.
- Anonymity. All unit interactions must be anonymous.
- General coordinate system. Despite the lack of reference to the absolute coordinate system, there is a local relative coordinate system in which the location of the unit is determined.
- The presence of memory. Each unit stores its current location in a relative coordinate system.
- Availability of communications. All robots communicate with neighbors to coordinate their plans.
- Units are able to observe only the nearest neighbors.

According to the classification proposed in [15], MRAKS is a robotic swarm.

Let the considered MRAKS include m units. The configuration is the set \( P = \{ p_i | i = 1, 2, \ldots, m \} \) of the
positions \( p_i = < x_i, y_i > \), specified in the \( xOy \), flat coordinate system in which the units are located. Without loss of generality, to simplify descriptions and calculations, we assume that the step along the \( x \) and \( y \) axes is chosen so that \( x_i \) and \( y_i \) take only integer values.

At a high level, the reconfiguration task is formulated as follows. It is necessary to develop a method that generates control signals for individual units, providing translation from one configuration to another.

Thus, it is necessary to develop a decentralized reconfiguration method, which, on the one hand, would virtually eliminate the information exchange via the wireless interface, minimize the exchange via the contact interface, and accordingly determine whether the reconfiguration is complete. The number of steps coincides with the number reported by the control center, the reconfiguration is considered complete.

The reconfiguration is chosen from the condition that all robots move in the same direction, the number of steps coincides with the number reported by the control center, and accordingly determine whether the reconfiguration is complete.

III. METHODS

To solve the problem, it is proposed to use the following strategy. Reconfiguration is carried out step by step. At each step, the MPAKS performs an intermediate reconfiguration into some intermediate configuration. Steps are performed until target configuration is achieved. One step consists of a planning stage and a moving stage. At the planning stage, each unit plans its movement on the basis of information about the target configuration, information received from its neighbors through the contact channel of communication, and information from the control center (CC) received via the wireless communication channel. At the stage of movement, each unit disconnects from its neighbors, moving and connect with new neighbors. The position of each unit during movement changes by no more than one unit and only along one axis.

The proposed strategy allows you to perform all the hardware limitations considered. The wireless interface is used only for broadcasting by the CC of a relatively small amount of information. The wired interface is used only for exchange between the nearest neighbors, so the delays are negligible. It uses relative addressing, thus avoiding the problem of excessive traffic. Autonomous movements of units are carried out only for a small distance, so the navigation error is insignificant.

Thus, the developed method should allow the reconfiguration of robots "blindly", based on the minimum information received broadcasting from the CC or from the nearest neighbors. Reconfiguration is carried out on the basis of short steps in order to minimize the use of the navigation system and, as often as possible, adjust the position of individual units relative to the bulk of the units.

The target configuration is selected based on the task that stands before the MRAKS. The start time of the reconfiguration is chosen from the condition that all robots receive a broadcast command to synchronize their movements. To synchronize their movement, the duration of the stages must be known in advance.

Each robot cannot know the current position of all robots, and accordingly determine whether the reconfiguration is complete. Therefore, he counts the number of perfect steps. If the number of steps coincides with the number reported by the control center, the reconfiguration is considered complete.

For the CC, reconfiguration also happens blindly. The CC sends data about the target configuration and counts the time necessary for reconfiguration. Moreover, the actual location of individual units is unknown to it. Location can only be modeled based on known reconfiguration rules. All units are equally anonymous for the command center.

To implement the proposed strategy, it is proposed to use the following algorithm of system actions.

1. The CC performs a reconfiguration simulation to calculate the required number of reconfiguration steps \( s \).
2. The CC sends the broadcast command a description of the target configuration \( P_{\text{target}} \), the start time of the reconfiguration \( t_{\text{start}} \), the number of reconfiguration steps \( s \), the duration of the planning stage \( t_{\text{planning}} \) and the duration of the stage of movement \( t_{\text{motion}} \). Each unit receives and stores this information.
3. Starting from the moment \( t_{\text{start}} \), each unit performs \( s \) steps consisting of the planning stage and the stage of movement. The timing diagram of the system is shown in Fig. 1. After the \( s \)-th step is completed, the target configuration is achieved.

The general strategy of the proposed algorithm is as follows (Fig. 2). The priority is the movement of outside robots, next to which there are unoccupied positions of the target configuration. Let's call them leading. All other robots are driven and their movement is determined by the movements of the leading ones.

Movement planning starts with leading robots. For this, the neighboring position is analyzed for the presence of free positions belonging to the target configuration. If such a position is found, then check for the presence of other applicants. If several extreme robots claim this position, the arbitration is performed according to the following priority: a unit for which moving to a free position corresponds to a downward movement, a unit moving to the left, a unit moving up, a unit moving to the right.
If the current position of the lead robot belongs to the target configuration, then the lead robot searches for a substitute. In general, the lead robot can have three potential substitutes. To determine his deputy, the lead robot sends to all neighbors a replacement offer.

From the consonants for the replacement of robots, one is selected according to the following priority rule. If there are extra units in the row to which the lead robot belongs, then the direction is chosen where they are the most. If there are no extra units in the row, then the alternate is taken in the direction of travel.

The algorithm of the slave robot is somewhat different. After determining that the robot is driven, the unit waits for a replacement request from the neighbors. Replacement proposals are satisfied in the following order: a proposal from a unit with the shortest chain length is accepted. If there are several such robots, then the sentences are accepted in the following order: a unit above, to the right, below, to the left. If the current position of the slave robot belongs to the target configuration, then a deputy is searched for in a manner similar to the procedure for searching for a deputy lead robot.

IV. RESULTS

To test the performance of the proposed method, it was tested on several test configurations. The simulation results are shown in Fig. 3-6.

V. DISCUSSION

The efficiency of the proposed method is proved by an experimental example with a relatively small number of robots. Although their number exceeds the number of robots than that of some analogs, the method requires testing an order of magnitude more robots. The specificity of the proposed method requires the development of a specialized simulator, work on which is already underway. Further publications on this topic and a refinement of the method are planned.

VI. CONCLUSION

To solve the problem of decentralized reconfiguration of a swarm of modular robots, a specific method was developed that minimizes the amount of information transmitted via the wireless and contact information channel. The features of the method are independence from the number of robots and low computational complexity, which makes it possible to use it on the simplest models of robots. The efficiency of the method is shown on separate examples for a set of robots of the order of several dozen. To test the reliability of the proposed method, more in-depth testing is required, which requires the development of a more complex simulation model.
Fig. 3. Square-Cross reconfiguration

Fig. 4. Arrow-H reconfiguration
Fig. 5. Square-Arrow reconfiguration

REFERENCES


Fig. 6. Square-H reconfiguration


