Robotics and Autonomous Systems 79 (2016) 72–86

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

MVGS: A new graph signature for self-reconfiguration planning of modular robots based on Multiple Views Theory

Khalil Taheri [a,*], Hadi Moradi [a,b], Masoud Asadpour [a], Parisa Parhami [a]

[a] *Advanced Robotics and Intelligent Systems Laboratory, School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran*

[b] *Intelligent Systems Research Institute (ISRI), SKKU, South Korea*

HIGHLIGHTS

- A new graph signature for self-reconfiguration planning of modular robots is proposed.
- The Multi-View Graph Signature (MVGS) is inspired from Multiple Views Theory.
- A new similarity metric between two configuration graphs is proposed based on MVGS.
- The approach is evaluated on reconfiguration planning of M-TRAN and SuperBot modules.
- Results show the approach performs significantly much better than the previous works.

ARTICLE INFO

Article history:
Received 16 April 2015
Received in revised form
3 December 2015
Accepted 15 January 2016
Available online 22 February 2016

Keywords:
Modular robots
Self-reconfiguration planning (SRP)
Graph signature
Multiple Views Theory

ABSTRACT

Self-reconfiguration planning (SRP) of modular robots has been a major research for the past decade and still is a key issue. In this paper, we present a novel new graph signature approach for SRP based on Multiple Views Theory, called *MVGS* (Multi-View Graph Signature). In the proposed method the time complexity of the graph signature generation reduced to $O(\log n)$, in which n is the number of modules, from the best solutions with $O(n^2)$. Also, we propose a new similarity metric, between graph signatures to better guide the heuristic search toward the final configuration. The proposed approach has been implemented in a simulator and the results show significantly better performance than the other previously proposed methods.

1. Introduction

Modular robots consist of small and homogeneous parts called modules. There are different design of modules to address different issues or tasks. For instance, M-TRAN [1] is a module consists of two servo motors and six connectors (Fig. 1). Another module called SuperBot [2] that is similar to M-TRAN but has three servo motors which provides more degrees of freedom. Modules classified as lattice-type or chain-type. Lattice-type modules are connected in a lattice form and move by detaching and attaching to each other. In other words, they locomote as cluster-flow on the ground and around obstacles. Crystalline robot [3], Telecube [4], and ATRON [5] are examples of lattice-type modules. Chain-type modules use their actuated joints to locomote without changing their configuration. M-TRAN, CONRO robot [6], Polybot [7] and SuperBot are examples of the chain-type modules. In this work, we focused on chain-type modules.

Generally, modules can form different structures called configurations. The process of reconfiguration, i.e. changing from an initial configuration to a final configuration, is performed by actions such as attach or detach. The goal of Self-Reconfiguration Planning (SRP) is to find a sequence of actions to reconfigure a modular robot from an initial configuration to a final configuration. Therefore, SRP can be performed using a search process to check all possible configurations which can be generated from an initial configuration to a final configuration. For instance, Stoy [8,9] proposed a gradient-based search for SRP which guarantees that the modules does not get disconnected or get stuck in local minima during the self-reconfiguration process.

A major issue in a search process is to determine if a given configuration is the same as a desired configuration or a configuration has not been visited before. In cases in which a

* Corresponding author.
E-mail addresses: k.taheri@ut.ac.ir, k.taheri1989@gmail.com (K. Taheri), moradih@ut.ac.ir (H. Moradi), asadpour@ut.ac.ir (M. Asadpour), p.parhami@ut.ac.ir (P. Parhami).

http://dx.doi.org/10.1016/j.robot.2016.01.012
0921-8890/

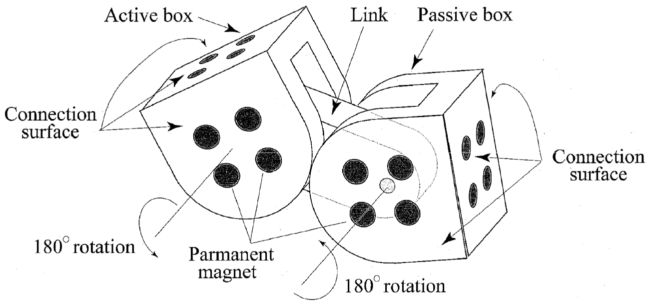

Fig. 1. Schematic view of MTRAN [1].

configuration is represented by a graph, this problem becomes a graph isomorphism problem, which is an NP problem. The graph representing a configuration is called a configuration graph, in which the vertices represent modules and the edges represent the connections between the modules [10–12].

To solve the graph isomorphism problem, the graph signature, as an isomorphism invariant representation of a configuration graph, has been introduced for fast isomorphism test [10–12]. Furthermore, to guide the search process and speed up finding a solution, several heuristics are proposed. For instance, Asadpour et al. [10,11] proposed an upper-bound of MCS (Maximum Common Sub-graph) as a similarity metric between configuration graphs.

Despite these efforts, SRP still remains a major issue in modular robots. Consequently, we propose a novel human inspired graph signature based on Multiple Views Theory. In our method the time complexity of the graph signature generation is reduced from $O\left(n^2\right)$ to $O(\log n)$, in which n is the number of modules. Also, we propose a new similarity metric, based on the new proposed graph signature, to guide our search heuristically and make it more efficient than the previous works [10–12].

The rest of the paper is organized as follows: the related works are explained in the next section. In the third section, our graph signature and similarity metric are described. Simulation results are presented in fourth section. Finally, discussion and conclusion of this work are described.

2. Related work

In the literature, different approaches have been proposed to tackle SRP. However, graph-based approaches provide global search over all possible configurations that ensures it does not get stuck in local minima. This eliminates the disadvantage of the local search approaches, such as stochastic gradient decent approaches [8], which may get stuck in local minima. Furthermore, the graph-based approaches can be applied to any design of modules while many proposed approaches are proposed for specific modules [13,14]. Therefore, we have focused on graph-based approaches.

In [15,16] Casal et al. and Yim et al. proposed divide-and-conquer strategies for closed-chain reconfiguration. At first, the initial configuration decomposed to a hierarchy of small sub-configurations and then reconfiguration between sub-configurations is done using a lookup table. The main issue with these approaches, which are based on divide-and-conquer approach, is in determining the intermediate configurations which should represent most of canonical configurations in the configuration space. Moreover, it should be less complicated to reconfigure between these canonical configurations rather than direct reconfiguration between the initial and final configurations.

In [17], Yoshida et al. proposed a hierarchical planner for SRP. In fact, blocks of modules are considered as macro-blocks and reconfiguration is done on these macro-blocks. At first, the planning is done at high level on the macro blocks which is called the global planner. Then a local planner, which is called the flow planner, tries to find possible plan to implement the designed global motion. The advantages of this approach can be handling of large configuration spaces using a rough high level planner and an exact local level planner. However, the modules cannot be moved at the same time in the process of self-reconfiguration. In other words, they should be moved one-by-one.

In [8], the author introduced a new approach to reach a desired configuration which is grown up from an initial configuration. The cellular automata is used to guide the growth toward the desired configuration. The proposed approach is evaluated in a simulator and the results show that the self-reconfiguration process always converges. Moreover, the experiments show that the self-reconfiguration time is approximately linear with the number of modules. The main disadvantage of this approach is in its huge search space.

In [18], the authors presented a hierarchical planner based on their Million Module March [19]. The high level planner designs a reconfiguration plan in the module-connector space using the distributed dynamic programming based on Markov Decision Process (MDP). The advantages of the proposed approach are in its distributed nature and the natural capability to include physical characteristics. On the other hand, their implementation has exponential time complexity based on the number of degrees of freedom in the modules. Finally, this approach is similar to [17] which its strengths and weakness described earlier. It should be noted that our proposed approach works at their high level planner category and their lower level of their planner can be used by our planner.

Aloupis et al. [13] proposed a novel algorithm for the reconfiguration of cube-style modules. The proposed approach can be performed in $O(n)$ parallel steps in which n is the number of atomic operations (expand, contract, attach, and detach) in the process of the reconfiguration. In this approach, the modules are reconfigured in the union of the bounding boxes of the initial and final configurations. Moreover, the modules do not get disconnected during the reconfiguration process. It should be noted that their algorithm is a distributed algorithm. Although the computational complexity of their algorithm is suitable for solving SRP but it only has been developed for a specific module, i.e. cube-style modules. In [14], their work is improved by introducing a parallel algorithm for reconfiguration which costs $O(\log n)$ parallel steps. However, the upper bound of the number of atomic operations increases to $\Theta(n \log n)$. Finally, in [20], they adapted their reconfiguration algorithm for M-TRAN and Molecube modules. It is shown that the other type of modules can be reconfigured within the same asymptotic time bounds as Crystalline.

In [10], Asadpour et al. proposed to represent each configuration of modules using a graph, called configuration graph. In a configuration graph, each module is represented by a vertex and the connection of two modules is represented by an edge. The label of an edge is determined based on the indices of the connectors, the direction of the connectors, and the relative orientation with respect to each other. The direction of an edge is determined based on being a male or a female connector. For genderless connectors, the direction which maximizes the label is chosen.

For each configuration graph, the authors generate an isomorphism invariant code to represent it, called graph signature. The isomorphic configuration graphs have same signatures and same signatures represent isomorphic configurations. Therefore, the graph signature can be used as an isomorphism invariant property to test isomorphic configurations fast. To find a solution for SRP, a search process is performed to check all possible configuration graphs which can be generated from an initial configuration graph by performing attach or detach actions. In the search process, the graph signature is used for two purposes: first, to avoid