2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

An Actuation Fault Tolerance Approach to Reconfiguration Planning of Modular Self-folding Robots

Meibao Yao*
School of Artificial Intelligence
Jilin University
Changchun, China
meibaoyao@gmail.com

Xueming Xiao
School of Opto-Electronic Engineering
Changchun University of Science and Technology
Changchun, China
alexcapshow@gmail.com

Yang Tian
Deep Space Exploration Research Center
Harbin Institute of Technology
Harbin, China
tian0yang@163.com

Hutao Cui
Deep Space Exploration Research Center
Harbin Institute of Technology
Harbin, China
cuiht@hit.edu.cn

Jamie Paik
Reconfigurable Robotics Lab
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
jamie.paik@epfl.ch

***Abstract*—This paper presents a novel approach to fault tolerant reconfiguration of modular self-folding robots. Among various types of faults that probably occur in the modular system, we focus on the tolerance of complete actuation failure of active modules that might cause imprecise robotic motion and even reconfiguration failure. Our approach is to utilize the reconfigurability of modular self-folding robots and investigate intra-module connection to determine initial patterns that are inherently fault tolerant. We exploit the redundancy of actuation and distribute active modules in both layout-based and target-based scenarios, such that reconfiguration schemes with user-specified fault tolerant capability can be generated for an arbitrary input initial pattern or 3D configuration. Our methods are demonstrated in computer-aided simulation on the robotic platform of Mori, a modular origami robot. The simulation results validate that the proposed algorithms yield fault tolerant initial patterns and distribution schemes of active modules for several 2D and 3D configurations with Mori, while retaining generalizability for a large number of modular self-folding robots.**

I. Introduction

Modular self-folding robots are autonomous machines that can reconfigure themselves from a collective pattern on a two-dimensional (2D) surface into user-specified shapes. While this versatility is popular in quite a few robotic scenarios that require emergent command response, such as search and rescue [1], [2], underwater exploration [3], [4], and space applications [5], [6], robotic faults of any kind might become catastrophic in such highly autonomous environments. Faults occurred in the modular robotic system can lead to imprecise motion control, collisions among modules, and even reconfiguration failures. Therefore, the capability for fault tolerance is of great significance and should be taken into consideration while designing and planning for modular self-folding robots. However, due to increase of complexity in fault tolerant design procedures as well as larger degrees-of-freedom (DoFs) with the number of modules going up, this topic is highly challenging. In this paper, we put forward a novel fault tolerant approach to reconfiguration planning of modular self-folding robots, while relieving both hardware and algorithmic complexities simultaneously.

Despite their intrinsic characteristics of fault tolerance due to modularity and repeatability in the robotic systems, such as swarm [7] and large-scale meta-module system [8], the problem of fault tolerant design of modular robots has been investigated from several perspectives, including mechanical design of modular systems [9]–[11], reconfigurable software in fault responsive control and re-planning [12], [13], and task-based reconfiguration methods [14], [15].

From a perspective of mechanical design, modular systems such as Sambot [9] and M-Blocks [11] can tolerate misalignment between modular surfaces by embedding active docking mechanism or disc magnets in the surfaces. Yim [10] presents a connecting mechanism for intra-module communication and power transfer among modules, allowing switch of electrical components for fault tolerance. While the mechanically fault tolerant design of modular systems generally demand implementation of hardware redundancy or complication, reconfigurable software design [12], [16], [17] has similar requirement. Commuri [13] presents a hierarchical architecture based on reconfigurable FPGAs that allows dynamical modification of the behavior of a team of mobile robots when faults occur. Shen [18] proposes a distributed control and adaptive communication approach inspired by the concept of hormone, enabling the modules to accommodate

*Meibao Yao is corresponding author.

unexpected changes of the configuration. Another approach to fault tolerant reconfiguration is task-based methods [14], [15] that designed for target configurations of modular robots, such as manipulators [14], [19] and walking robots [15], [20], as well as a universal approach [21] for configuration design based on fault tolerant indices.

Although the above-mentioned methods provide comprehensive perspectives for fault tolerant design in modular robots, they either cast high demand on the hardware implementation in mechanical design, or complicate the algorithmic procedures to tolerate faults. In this work, we put forward an alternative approach that fully utilizes the reconfigurability of modular self-folding robots for fault tolerance in their reconfiguration.

Note that even for an identical three-dimensional (3D) configuration of a modular self-folding robot, there are quite a number of initial patterns that can finally transform into the target shape. Since the mechanical connection among modules, as well as overall impact suffered by a fault occurred differ from one initial pattern to another, each candidate of initial patterns has different level of fault tolerant capability in its intrinsicality. Therefore, our approach is to determine initial configurations of modular self-folding robots that are inherently more fault-tolerant than their counterparts.

In general there are two types of modules in a large number of modular robotic systems, namely active modules and passive ones. The active modules are a collective of actuators, controllers and related electronic components, while the passive ones act as simple connectors. Although there are different types of faults that can lead to final reconfiguration failures, such as motion fault resulting in module collisions, here we focus on actuation fault of the active modules that turns them into passive ones. Note that actuation fault can be tolerated by exploiting the redundancy of active modules, our approach is to determine a distribution scheme of active modules in the robotic configuration, so that a predefined metric of fault tolerance can be regulated on demands, trading off between the number of distributed active modules and overall fault tolerant capability.

The main contributions of this paper are:

1. Methodology to evaluate fault tolerance capability of reconfiguration schemes with various 2D or 3D configurations of modular self-folding robots;

2. Reconfiguration planners in both layout-based and target-based scenarios to generate initial patterns and distribution schemes of active modules with desired fault tolerant capability;

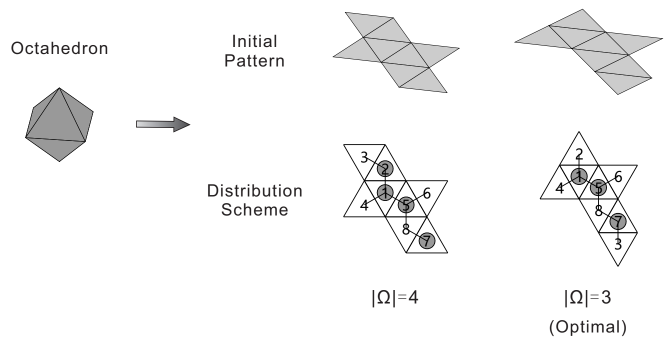
3. Demonstration on the modular robotic platform of Mori in simulation, validating the proposed methods and algorithms.

II. Methodology

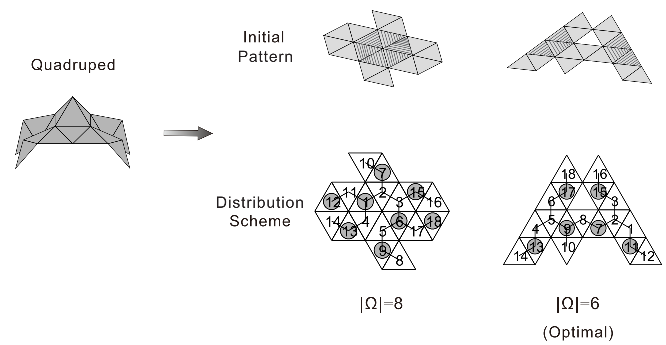
A. System overview

In this section we investigate the reconfiguration schemes of modular self-folding robots, and discuss their capability for fault tolerance with different distribution schemes of active modules.

In our previous work [22], we inferred that for a predefined 3D shape of modular self-folding robot, there exist a large number of reconfiguration schemes that can finally achieve the target shape. The reconfiguration scheme consists of initial pattern, distribution scheme of active modules, denoted as Ω, and other elements. We show some of the reconfiguration schemes with two different 3D shapes, an octahedron and a quadruped, as illustrated in Fig. 1. The rightmost column of each subfigure is the optimal reconfiguration scheme with minimally distributed active modules.



(a) An octahedron



(b) A quadruped

Fig. 1: The reconfiguration schemes of a modular self-folding robot with two different 3D shapes.

For each initial pattern in Fig. 1, minimum number of active modules are distributed while the modular self-folding robot has sufficient actuation capability to reconfigure and perform tasks after its transformation. With the lowest number of active modules distributed, the robot cannot tolerate any actuation fault, since it cannot reconfigure into the target shape with the failure of any active modules.

A straightforward approach to enhance the capability for fault tolerance is to exploit the redundancy of active modules. With more active modules distributed in the robotic configuration, the fault tolerant capability of the robotic system improves accordingly. However, active modules are generally much more costly than the passive ones, since most mechanical and electronic components are assembled on active modules. Therefore, from a cost-saving perspective, fewer active modules should be distributed in the robotic configuration, in contrast to the fault tolerant demands. To this end, we propose algorithms to determine the distribution scheme of active modules, trading off between cost and fault tolerant capability.

Note that from Fig. 1a, even though the number of active modules distributed in the first initial pattern is one more than that in the second one, both reconfiguration schemes lack the capability of fault tolerance. We denote the probability of actuation fault occurred in each active module as δ. The overall fault occurrence probability in the robotic configuration rises with working-hour increasing due to more serious mechanical abrasion or circuits aging. When this probability rises from near 0 to $\frac{1}{3}\delta$, the first reconfiguration scheme in Fig. 1a might suffer the fault while the second scheme would not. This is because the probability of any one of the active modules that suffers actuation fault in the first scheme is $\mathbf{C}_4^1\delta = \frac{1}{4}\delta$, while this probability in the second scheme is $\mathbf{C}_3^1\delta = \frac{1}{3}\delta$. For Fig. 1b we have similar inference. Therefore, we put forward the following corollary:

Corollary 1. *The initial pattern generated by minimum distribution algorithm*[1] *is inherently more fault tolerant and less costly than other candidates in the initial configuration space of a target 3D shape of modular self-folding robot.*

Corollary 1 will be further verified in Sec. II-B from a more analytical point of view. Our method for fault tolerant design is thus based on the reconfiguration schemes with minimally distributed active modules, and to utilize the redundancy of actuation to tolerate faults.

B. Metric of fault tolerance capability

Given a user-specified configuration of modular self-folding robot, the essential in designing a fault tolerant reconfiguration scheme is to evaluate the capability of fault tolerance of each candidate. We investigate the metric of fault tolerance in reconfiguration of modular self-folding robots as follows.

Although there are numerous factors that can cause actuation fault in modular self-folding robots, such as faults occurred in electric circuits and mechanical failure of components, here we assume that the probability of actuation fault occurred in each active module is identical, for computational simplification. We further suppose that the actuation fault in active modules results in a complete failure and turns them into passive ones.

Based on Sec. II-A, we inferred that the initial pattern with minimally distributed active modules is the base for fault tolerant reconfiguration planning in modular self-folding robots. For an input configuration of modular self-folding robot, the reconfiguration scheme with minimally distributed active modules can be generated according to [22]. We denote the number of active modules in this reconfiguration scheme as n' and the total number of modules as n. We assume that there are additional m active modules to be distributed, so as to enhance the fault tolerant capability of the robot. The probability of actuation fault occurred that can be tolerated by the redundancy of active modules is denoted as Δ. The minimum of Δ occurs when all the minimally distributed active modules work normally and at least one of the additionally distributed active modules suffers actuation fault, and can be calculated as:

$$\Delta_{\min} = (1-\delta)^{n'} \times \left[1-(1-\delta)^m\right],\ m \geq 1 \tag{1}$$

Since the analytical expression of Δ cannot be determined without knowing the exact reconfiguration scheme a priori, we utilize the minimum of Δ, $\Delta_{\min}$, as a metric to evaluate the overall fault tolerant capability of an initial pattern in the design procedures. The maximum of $\Delta_{\min}$ occurs when m reaches its maximum, as $m_{\max} = n - n'$. Under this circumstance, all modules of the robotic configuration are assigned as active modules, and the fault tolerant capability of the robot as well as the cost caused by active modules reach their peak simultaneously.

On the other hand, it is deduced from Eq. 1 that the value of $\Delta_{\min}$ increases with n' going down in number, thus the minimal distribution scheme of active modules can facilitate maximum fault tolerant capability for an initial pattern and a 3D shape of modular self-folding robot. This verifies Corollary 1 in Sec. II-A from an analytical point of view.

Based on the above discussion, the user can specify the fault tolerant capability of the modular self-folding robot while trading off the cost according to actual demands. We define $\bar{\Delta}$ as another metric of fault tolerant capability with m as the only variable, since n' is constant with a given initial pattern. $\bar{\Delta}$ is defined as:

$$\bar{\Delta}(m) = 1-(1-\delta)^m,\ m \geq 1 \tag{2}$$

$\bar{\Delta}$ is a monotone increasing function of m, and its range of value can be calculated as: $\delta \leq \bar{\Delta} \leq \bar{\Delta}(m_{\max}), m_{\max} = n - n'$. We infer from [22] that the lower bound of n' is roughly $\frac{1}{2}n - 1$, thus $m_{\max} \leq \left\lceil \frac{1}{2}n + 1 \right\rceil$. Hence, the range of $\bar{\Delta}$ can be deduced as:

$$\delta \leq \bar{\Delta} \leq \bar{\Delta}\left(\left\lceil \frac{1}{2}n + 1 \right\rceil\right) \tag{3}$$

Therefore, the user can specify the value of $\bar{\Delta}$ according to Eq. 3. For a user-defined value of the function of metric $\bar{\Delta}$, denoted as Δ', the number of additionally distributed active modules excluding the minimally distributed ones, denoted as $\bar{m}$, can be calculated as:

$$\bar{m} = \left\lceil \bar{\Delta}^{-1}(\Delta') \right\rceil = \left\lceil \log_{(1-\delta)}(1-\Delta') \right\rceil \tag{4}$$

The range of Δ' is consistent with that of $\bar{\Delta}$, as:

$$\delta \leq \Delta' \leq \bar{\Delta}\left(\left\lceil \frac{1}{2}n + 1 \right\rceil\right) \tag{5}$$

C. Algorithmic approach to fault tolerant reconfiguration

Based on the metric of fault tolerance introduced in Eq. 2, we propose an algorithmic approach to reconfiguration planning of modular self-folding robots with user-defined fault tolerant capability that compromises the cost of active modules.

[1]Minimum distribution algorithm [22] yields the initial pattern that can reconfigure into the target 3D shape with minimally distributed active modules.

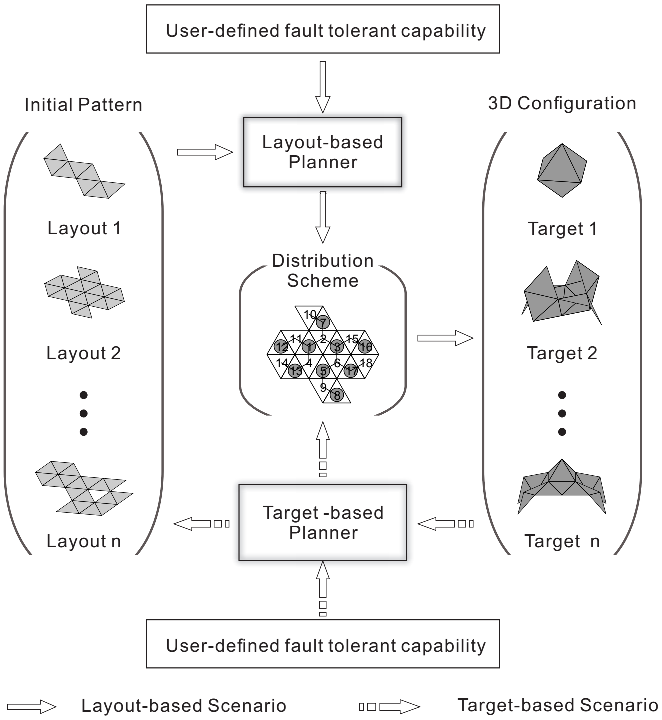


Fig. 2: Method overview of fault tolerant reconfiguration planning of modular self-folding robots.

Inheriting the idea from our previous work [22], we formulate the fault tolerant reconfiguration planning in two scenarios on different demands, namely layout-based and target-based scenario, as illustrated in Fig. 2. In the layout-based scenario, the input of the layout-based planner is an arbitrary initial pattern of the modular self-folding robot, and a user-defined parameter that represents desired fault tolerant capability of the robot, while the output is a distribution scheme of active modules with sufficient actuation redundancy to tolerate faults. In the target-based scenario, the input of the target-based planner is a target 3D configuration and a user-defined parameter of fault tolerance, while the output is the optimal initial pattern with minimally distributed active modules.

The layout-based planner allows an arbitrary initial pattern to reconfigure into as many 3D shapes as possible, enhancing its adaptability under complex environment, especially when the task-oriented configuration is undefined. On the other hand, the target-based planner generates the optimal reconfiguration scheme of a user-specified 3D configuration, where connected coplanar surface might exist, further reducing the number of active modules with unnecessary actuation taken into account. Here we present the layout-based planner and target-based planner corresponding to the two scenarios as follows.

1) Layout-based planner: The algorithmic procedures of the fault tolerant layout-based planner is shown in Alg. 1. The user should specify the value of Δ' according to the range in Eq. 5. The probability for the robot to tolerate fault become larger with the value of Δ' going up, while the cost of active modules increases. We utilize graph representation [23] to mathematically describe the modular architecture of a modular self-folding robot. In our previous work [22], we model the input initial layout of modular self-folding robot as a mesh $\boldsymbol{M}$, and its dual graph $\boldsymbol{\Psi}(\boldsymbol{M})$ and graph matrix $\boldsymbol{H}(\boldsymbol{\Psi})$ are calculated accordingly. We make use of Alg. 1 in [22] to generate the minimal distribution scheme of active modules of $\boldsymbol{M}$, denoted as $\boldsymbol{\Omega}^*$. In this planner we distribute the additional $\bar{m}$ active modules in the modular architecture.

Algorithm 1 Fault tolerant layout-based algorithm.

Input: An initial pattern $\boldsymbol{M}$ and desired fault tolerance Δ';
Output: Fault tolerant distribution scheme $\boldsymbol{\Omega}$;
1: Dual graph $\boldsymbol{\Psi}(\boldsymbol{M})$, graph matrix $\boldsymbol{H}(\boldsymbol{\Psi})$;
2: Initialization: $\boldsymbol{\Omega} \leftarrow \emptyset$, $\bar{\boldsymbol{\Omega}} \leftarrow \emptyset$;
3: Generate $\boldsymbol{\Omega}^*$ by Alg. 1 in [22];
4: Connection set $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_i \mid i \in \boldsymbol{V}_d\} \leftarrow$ Eq. 6a;
5: $\bar{m} \leftarrow$ Eq. 4, $\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega}^*$;
6: **while** $\bar{m} > 0$ **do**
7: $\quad \boldsymbol{\Gamma} \leftarrow \arg\max_{i \in \bar{\boldsymbol{V}}_d} |\boldsymbol{\Lambda}_i|$, $\bar{\boldsymbol{V}}_d = \mathrm{C}_{\boldsymbol{V}_d}\boldsymbol{\Omega}$;
8: $\quad$ **if** $|\boldsymbol{\Gamma}| \geq \bar{m}$ **then**
9: $\quad\quad \bar{\boldsymbol{\Omega}} = \mathrm{Rand}\,(\boldsymbol{\Gamma}, \bar{m})$;
10: $\quad$ **else** $\bar{\boldsymbol{\Omega}} = \boldsymbol{\Gamma}$;
11: $\quad$ **end if**
12: $\quad \bar{m} = \bar{m} - |\bar{\boldsymbol{\Omega}}|$, $\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega} \cup \bar{\boldsymbol{\Omega}}$;
13: **end while**
14: **return** $\boldsymbol{\Omega}$

It is straightforward that if the added active modules connect with more units, they can play a more important role when others suffer actuation faults. Therefore, modules with more connection to other individuals have priority to be chosen as active ones. We define $\boldsymbol{\Lambda}_i$ as the connection set of module i, representing the set of connected units of the module.

$$\boldsymbol{\Lambda}_i = \{j \mid h_{ij} = 1, i, j \in \boldsymbol{V}_d\} \tag{6a}$$

$$|\boldsymbol{\Lambda}_i| = \sum_{i,j \in \boldsymbol{V}_d} h_{ij} \tag{6b}$$

where h_{ij} is the corresponding element of graph matrix $\boldsymbol{H}$; $\boldsymbol{V}_d$ is the set of all modules in the robotic configuration.

We utilize the connection set defined in Eq. 6 to select the active modules among the modules with maximum connection to other units, denoted as $\boldsymbol{\Gamma}$ in Alg. 1 Line 7. We denote $\bar{\boldsymbol{V}}_d$ as the complement of $\boldsymbol{\Omega}$ in $\boldsymbol{V}_d$, and randomly choose modules with maximum connection among $\bar{\boldsymbol{V}}_d$, until $\bar{m}$ additional active modules are assigned. Based on the above algorithmic procedures, the distribution scheme of active modules with fault tolerant capability is generated.

2) Target-based planner: The algorithmic procedures of the fault tolerant target-based planner is shown in Alg. 2. The algorithm is similar with Alg. 1 except that the modules in connected coplanar areas of the target shape are excluded from candidates of active modules.

The input of the planner is a 3D configuration modeled as a mesh $\boldsymbol{M}$ and the user-defined fault tolerance Δ'. The normal vector of each module in the 3D shape is calculated and

Algorithm 2 Fault tolerant target-based algorithm.

Input: A 3D configuration $\boldsymbol{M}$ and desired fault tolerance Δ';

Output: A fault tolerant initial pattern $\boldsymbol{\Upsilon}^*$ and its distribution scheme $\boldsymbol{\Omega}$;

1: Normal vectors $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \cdots, \boldsymbol{\sigma}_i, \cdots, \boldsymbol{\sigma}_n$;
2: Generate $\boldsymbol{\Upsilon}^*$ and $\boldsymbol{\Omega}^*$ by Alg. 2 in [22];
3: Graph matrix $\boldsymbol{H}(\boldsymbol{\Psi}^*)$, dual graph $\boldsymbol{\Psi}^*$;
4: $\bar{\boldsymbol{E}}_d^* \leftarrow \emptyset$, $\boldsymbol{H}(\bar{\boldsymbol{\Psi}}^*) \leftarrow \boldsymbol{H}(\boldsymbol{\Psi}^*)$;
5: **for** $(i,j) \in \boldsymbol{E}_d^*$ **do**
6: $\quad \boldsymbol{\alpha}_{i,j} = \boldsymbol{\sigma}_i \times \boldsymbol{\sigma}_j$;
7: $\quad$ **if** $\boldsymbol{\alpha}_{i,j} = \boldsymbol{0}$ **then**
8: $\qquad \bar{\boldsymbol{E}}_d^* \leftarrow \boldsymbol{E}_d^* - (i,j)$;
9: $\qquad \boldsymbol{H}(\bar{\boldsymbol{\Psi}}^*)\,(i,j) \leftarrow 0$;
10: $\quad$ **end if**
11: **end for**
12: Connection set $\bar{\boldsymbol{\Lambda}}_i^*, i \in \boldsymbol{V}_d \leftarrow$ Eq. 6a;
13: $\bar{m} \leftarrow$ Eq. 4, $\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega}^*$;
14: **while** $\bar{m} > 0$ **do**
15: $\quad \boldsymbol{\Gamma} \leftarrow \arg\max_{i \in \bar{\boldsymbol{V}}_d} \left|\bar{\boldsymbol{\Lambda}}_i^*\right|, \bar{\boldsymbol{V}}_d = \mathrm{C}_{\boldsymbol{V}_d}\boldsymbol{\Omega}$;
16: $\quad$ **if** $|\boldsymbol{\Gamma}| \geq \bar{m}$ **then**
17: $\qquad \bar{\boldsymbol{\Omega}} = \mathrm{Rand}\,(\boldsymbol{\Gamma}, \bar{m})$;
18: $\quad$ **else** $\bar{\boldsymbol{\Omega}} = \boldsymbol{\Gamma}$;
19: $\quad$ **end if**
20: $\quad \bar{m} = \bar{m} - \left|\bar{\boldsymbol{\Omega}}\right|$, $\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega} \cup \bar{\boldsymbol{\Omega}}$;
21: **end while**
22: **return** $\boldsymbol{\Upsilon}^*$ and $\boldsymbol{\Omega}$

denoted as σ. We utilize Alg. 2 in [22] to generate the optimal initial pattern $\boldsymbol{\Upsilon}^*$ with minimal distribution scheme of active modules $\boldsymbol{\Omega}^*$. The corresponding graph matrix $\boldsymbol{H}(\boldsymbol{\Psi}^*)$ and dual graph $\boldsymbol{\Psi}^*$ are calculated. $\boldsymbol{E}_d^*$ denotes the edge set of the dual graph $\boldsymbol{\Psi}^*$. Lines 5-12 in Alg. 2 calculate the graph matrix and connection set of $\boldsymbol{\Upsilon}^*$ with the connection among coplanar modules excluded, denoted as $\boldsymbol{H}(\bar{\boldsymbol{\Psi}}^*)$ and $\bar{\boldsymbol{\Lambda}}_i^*$. The procedures in lines 13-21 are similar with that in Alg. 1, generating $\bar{m}$ active modules with maximum connection according to $\bar{\boldsymbol{\Lambda}}_i^*$.

III. Simulation

To evaluate the effectiveness of the proposed methods and algorithms in this paper, we carry out simulation on a modular origami robot named Mori [1] as the hardware platform. There are both active and passive modules in the Mori platform, and each individual is triangular-shaped and identical in geometry. Three stepper motors are embedded in each lateral of the active module. We present simulation results of the proposed layout-based and target-based planners as follows.

A. Layout-based reconfiguration scheme

We utilize the layout-based planner shown in Alg. 1 to generate a fault tolerant distribution scheme of active modules for an input initial pattern. We first investigate the results of distribution schemes in various initial patterns with desired fault tolerant capability.

The probability of actuation fault occurred in each active module is set as $\delta = 10\%$. For an initial pattern with 18 modules, as $n = 18$, the range of Δ' is $[10\%, 65.13\%]$, according to Eq. 5. We specify the value of Δ' as $\Delta' = 40\%$, and the number of additionally distributed active modules is $\bar{m} = 5$, according to Eq. 4. Three initial patterns of the quadruped in Fig. 1b, their minimal distribution schemes, and fault tolerant distribution schemes of active modules are shown in Fig. 3. Distribution scheme I represents the minimal distribution schemes generated by Alg. 1 in [22], and distribution scheme II is generated by the layout-based planner.

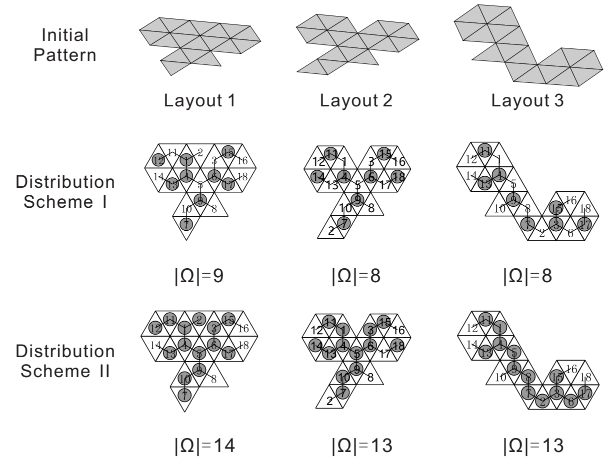


Fig. 3: Three initial patterns of a quadruped, their minimal distribution schemes, and fault tolerant distribution schemes generated by the layout-based planner.

It can be inferred from Fig. 3 that the first initial pattern with 9 active modules in distribution scheme I is potentially less fault tolerant than the other two, according to Eq. 1. Although the distribution scheme of active modules varies in Layout 2 and Layout 3, the minimum probability of actuation fault occurred and tolerated is identical. However, the exact fault tolerant capability of the three initial patterns might be different, since the connection among modules varies and the additionally distributed active modules might be able to tolerate the fault occurred in the minimally distributed ones. For example, in distribution scheme I of Layout 1, if module 1 suffers actuation fault, the additionally distributed modules 2 and 11 can make it up, enhancing the fault tolerant capability of the robot.

On the other hand, for a specific initial pattern, we investigate its fault tolerant capability with different distribution schemes of active modules. We illustrate an initial pattern of modular self-folding robot with different fault tolerant capability due to redundancy of active modules in Fig. 4.

The initial pattern in Fig. 4 is assigned with 9 active modules in the minimal distribution scheme, and its fault tolerant capability increases gradually with the number of additionally distributed active modules going up. When each module is assigned as active module, the fault tolerant capability of the initial pattern reaches maximum.

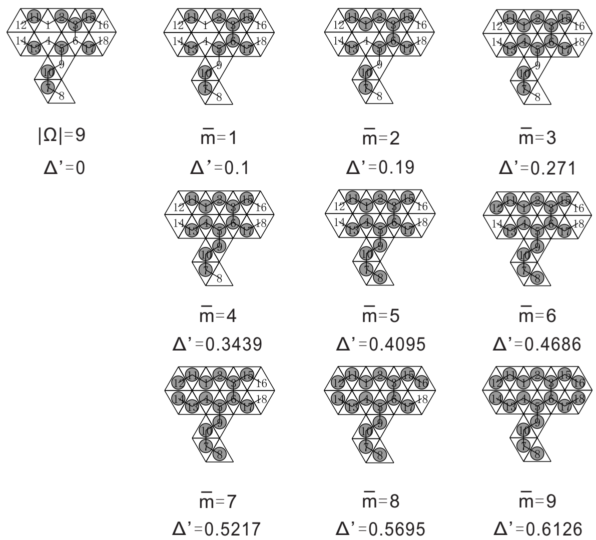


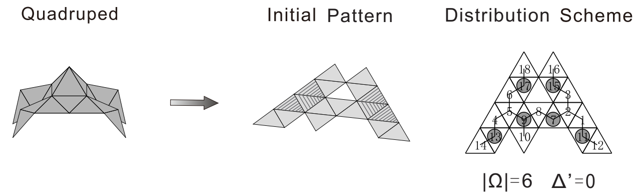
Fig. 4: An initial pattern with different fault tolerant capability due to redundancy of active modules.

The simulation results demonstrate that the layout-based planner in Alg. 1 can yield reconfiguration scheme with desired fault tolerant capability for an arbitrary input initial pattern of modular self-folding robot.

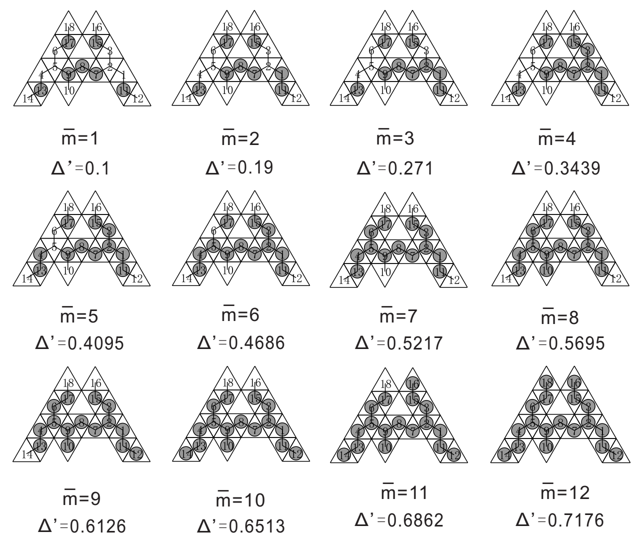
B. Target-based reconfiguration scheme

In the target-based scenario, we utilize the planner in Alg. 2 to generate the reconfiguration scheme with desired fault tolerant capability. The input of target-based planner is a user-defined 3D configuration of the modular self-folding robot.

For the quadruped shown in Fig. 1b, the planner yields its optimal initial pattern with minimum distribution scheme of active modules, as well as the distribution schemes with various fault tolerant capability, as illustrated in Fig. 5.



(a) A quadruped and its optimal reconfiguration scheme



(b) The optimal initial pattern with various fault tolerant capability

Fig. 5: The optimal reconfiguration scheme of a quadruped and its distribution schemes with various fault tolerant capability, generated by the target-based planner.

In Fig. 5a, there are 6 active modules distributed minimally in the initial pattern, 2 modules less than that of Layout 2 and 3 in Fig. 3. This is because there exist two coplanar areas of connected modules in the quadruped, shown as shaded areas in the figure, and no actuation is needed among the coplanar modules for reconfiguration. Hence, the target-based planner can potentially generate reconfiguration schemes with less active modules than the layout-based planner, while the layouts can transform into an identical 3D shape.

Fig. 5b shows that fault tolerance capability of the initial pattern enhances with the number of additional active modules rising. The range of Δ' is $[10\%, 71.76\%]$ if additional active modules are assigned, according to Eq. 2. The maximum fault tolerance metric reaches $\Delta' = 71.76\%$, higher than that in Fig. 4. This is because the initial pattern in Fig. 5b can distribute 12 additional active modules, while the one in Fig. 4 can only distribute 9 active modules at most. However, the initial pattern distributed with 6 active modules is not as versatile as that with 9 active modules, since it is not fully actuated. This demonstrates that the initial pattern with minimal distribution scheme of active modules is potentially more fault tolerant than others in the initial configuration space of a target 3D shape, verifying Corollary 1 in Sec. II-A.

The simulation results validate that the target-based planner in Alg. 2 can yield reconfiguration scheme with desired fault tolerant capability for an arbitrary input 3D shape of modular self-folding robot.

IV. Conclusion

This paper investigates the problem of fault tolerant reconfiguration of modular self-folding robots. We discuss this problem from the perspectives of layout-based and target-based planning, based on the optimal initial pattern with minimal distribution scheme of active modules. We propose methodology to evaluate fault tolerant capability of the candidates of reconfiguration schemes of an arbitrary 2D or 3D configuration as input of the planners. The modular robotic platform of Mori is utilized to validate the effectiveness of the proposed planning algorithms, while the presented work retains generalizability for the modular robotics community.

Acknowledgment

This work is sponsored by the National Natural Science Foundation of China (NSFC) through grants No. 61503102 and No. 61673057, and partially by the CUST Youth Fund Project (grant XJJLG-2018-04). In addition, we appreciate the support by Civil Aviation Advanced Research Project of China.

References

[1] C. H. Belke and J. Paik, "Mori: a modular origami robot," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 5, pp. 2153–2164, 2017.

[2] M. Yao, C. H. Belke, H. Cui, and J. Paik, "A reconfiguration strategy for modular robots using origami folding," *The International Journal of Robotics Research*, vol. 38, no. 1, pp. 73–89, 2019.

[3] I. Vasilescu, P. Varshavskaya, K. Kotay, and D. Rus, "Autonomous modular optical underwater robot (amour) design, prototype and feasibility study," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1603–1609.

[4] D. J. Christensen, J. C. Andersen, M. Blanke, L. Furno, R. Galeazzi, P. N. Hansen, and M. C. Nielsen, "Collective modular underwater robotic system for long-term autonomous operation," in *ICRA 2015*, 2015.

[5] J. Baca, S. Hossain, P. Dasgupta, C. A. Nelson, and A. Dutta, "Modred: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 1002–1015, 2014.

[6] M. Yao, X. Xiao, Y. Tian, and H. Cui, "A two-time scale control scheme for on-orbit manipulation of large flexible module," *Acta Astronautica*, vol. 154, pp. 92–102, 2019.

[7] B. Jakimovski, B. Meyer, and E. Maehle, "Swarm intelligence for self-reconfiguring walking robot," in *2008 IEEE Swarm Intelligence Symposium*. IEEE, 2008, pp. 1–8.

[8] D. J. Christensen, "Experiments on fault-tolerant self-reconfiguration and emergent self-repair," in *2007 IEEE Symposium on Artificial Life*. IEEE, 2007, pp. 355–361.

[9] H. Wei, Y. Chen, J. Tan, and T. Wang, "Sambot: A self-assembly modular robot system," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 745–757, 2010.

[10] M. H. Yim, "Fault tolerant connection system for transiently connectable modular elements," May 15 2001, uS Patent 6,233,502.

[11] J. W. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4288–4295.

[12] M. Park and M. Yim, "Distributed control and communication fault tolerance for the ckbot," in *2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*. IEEE, 2009, pp. 682–688.

[13] S. Commuri, V. Tadigotla, and L. Sliger, "Task-based hardware reconfiguration in mobile robots using fpgas," *Journal of Intelligent and Robotic Systems*, vol. 49, no. 2, pp. 111–134, 2007.

[14] B. Zhao and Y. Li, "Local joint information based active fault tolerant control for reconfigurable manipulator," *Nonlinear dynamics*, vol. 77, no. 3, pp. 859–876, 2014.

[15] D. J. Christensen, J. C. Larsen, and K. Stoy, "Fault-tolerant gait learning and morphology optimization of a polymorphic walking robot," *Evolving Systems*, vol. 5, no. 1, pp. 21–32, 2014.

[16] H. Ahmadzadeh and E. Masehian, "Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization," *Artificial Intelligence*, vol. 223, pp. 27–64, 2015.

[17] D. J. Christensen, J. Campbell, and K. Stoy, "Anatomy-based organization of morphology and control in self-reconfigurable modular robots," *Neural Computing and Applications*, vol. 19, no. 6, pp. 787–805, 2010.

[18] W.-M. Shen, B. Salemi, and P. Will, "Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots," *IEEE transactions on Robotics and Automation*, vol. 18, no. 5, pp. 700–712, 2002.

[19] C. J. Paredis and P. K. Khosla, "Designing fault-tolerant manipulators: How many degrees of freedom?" *The international journal of robotics research*, vol. 15, no. 6, pp. 611–628, 1996.

[20] J.-M. Yang, "Fault-tolerant gaits of quadruped robots for locked joint failures," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 4, pp. 507–516, 2002.

[21] Q. Li and J. Zhao, "A universal approach for configuration synthesis of reconfigurable robots based on fault tolerant indices," *Industrial Robot: An International Journal*, vol. 39, no. 1, pp. 69–78, 2012.

[22] M. Yao, X. Xiao, C. H. Belke, H. Cui, and J. Paik, "Optimal distribution of active modules in reconfiguration planning of modular robots," *Journal of Mechanisms and Robotics*, vol. 11, no. 1, p. 011017, 2019.

[23] R. Straub and H. Prautzsch, *Creating optimized cut-out sheets for paper models from meshes*. Citeseer, 2011.