robotics

MDPI

Article

PARTS—A 2D Self-Reconfigurable Programmable Mechanical Structure

Michael Gerbl *, Michael Pieber, Emanuel Ulrich and Johannes Gerstmayr

Department of Mechatronics, University of Innsbruck, 6020 Innsbruck, Austria; michael.pieber@uibk.ac.at (M.P.); emanuel.ulrich@outlook.at (E.U.); johannes.gerstmayr@uibk.ac.at (J.G.)

* Correspondence: gerbl.michael@gmail.com or gerbl.michael@googlemail.com

Abstract: Modular self-reconfigurable robots hold the promise of being capable of performing a wide variety of tasks. However, many systems fall short of either delivering this promised functionality due to constraints in system architecture or validating it on functional hardware prototypes. This paper demonstrates the functional capabilities of the Planar Adaptive Robot with Triangular Structure (PARTS) and documents the versatility of this robot system using a holistic approach that combines simulations and hardware demonstrations on a prototype with nine fabricated modules. PARTS is a two-dimensional modular robot consisting of modules with a shape-shifting triangular geometry capable of forming adaptable space-covering structures. Meta-modules and mesh restructuring techniques are presented as methods for achieving topological self-reconfiguration. The feasibility of these methods is demonstrated by applying them on a simulated reconfiguration example of 62 modules. The paper showcases the versatility of PARTS on the hardware prototype using task-specific configurations, including locomotion using a meta-module and a walker configuration, module-module interaction by establishing a bridge between two separated module clusters, and interaction with the environment using a gripper and supporting structure configuration. The results validate the versatility and emphasize the potential of the system's design concept, motivating the transfer of the hardware architecture to the third dimension.

Keywords: modular robots; self-reconfiguration; triangular modules; shape-shifting geometry; locomotion; grasping; programmable mechanical structures

check for updates

Citation: Gerbl, M.; Pieber, M.; Ulrich, E.; Gerstmayr, J. PARTS—A 2D Self-Reconfigurable Programmable Mechanical Structure. *Robotics* **2024**, *13*, 77. https://doi.org/10.3390/robotics13050077

Academic Editor: Andy Tyrrell

Received: 14 April 2024
Revised: 9 May 2024
Accepted: 9 May 2024
Published: 14 May 2024

1. Introduction

In the quest for adaptable and resilient machinery to navigate the uncertainties of dynamic environments such as those encountered in space exploration and disaster response, conventional fixed-configuration systems face formidable challenges. Their inherent limitations, characterized by rigid designs and susceptibility to multiple points of failure [1], underscore the pressing need for innovative solutions [2].

Promising approaches to realizing robust and flexible mechanical systems are so-called modular robots [3]. These robotic systems consist of multiple homogeneous or heterogeneous robotic cells that can be arranged in a variable connection topology. A single module typically has limited motion capabilities; multiple modules, however, can cooperate and coordinate their movements to realize a higher level of behavior and functionality [4–6]. The ability for topological self-reconfiguration enables the autonomous spacial reorganization of modules, allowing for the generation of a variety of functional shapes such as grippers [7], support structures [8], or legs for locomotion [9–11]. The modular design further allows for the replacement of faulty modules by functioning units [12], resulting in self-repairing [13] and making the robot robust against the failure of individual system components.

A survey conducted in 2016 reported the existence of more than 90 modular robotic systems proposed by various research groups [5]. However, the majority of these systems lack the promised overarching versatility for applications in the real world [3]. Although some modular robots have been demonstrated to fulfill elaborated task sequences [5], the architecture or hardware realization of most existing systems greatly limits their ability to form complex morphological shapes and realize sophisticated assignments that require diverse functional configurations.

The *Planar Adaptive Robot with Triangular Structure* (PARTS) is a two-dimensional modular robot consisting of modules with a shape-shifting triangular geometry. In earlier work, we have constructed and improved a hardware prototype [14,15], conducted hardware modeling and calibration [14], and developed self-reconfiguration algorithms [16,17]. The unique system architecture allows the PARTS system to form space-filling, morphologically adaptable structures and realize extensive shape changes through self-reconfiguration. However, proposed self-reconfiguration strategies are due to simplified assumptions unsuited for application on the existing hardware prototype and the functional capabilities of the PARTS system have been only outlined but not validated.

In this work, we advance beyond our prior endeavors by implementing and showcasing self-reconfiguration and the envisioned versatile functionality with the system's hardware realization utilizing simulations and hardware demonstrations. The novelty lies in validating the functional capabilities of our system architecture through the deployment of algorithmic approaches on our prototype. In contrast to earlier work, we address constraints inherent to the hardware implementation, such as limited actuation lengths, and develop strategies to cope with these limitations during self-reconfiguration. Our system architecture with shape-shifting modules is capable of forming space-filling, morphologically adaptable structures and performing self-reconfiguration. To the knowledge of the authors, this combined set of functionalities has not been demonstrated in its entirety by any other modular robot.

The paper is structured as follows. Section 2 provides a review of related work in the field of modular self-reconfigurable robotic systems. We describe the PARTS modular robot and its hardware realization in Section 3. Section 4 presents methods to realize topological self-reconfiguration and demonstrates them in a simulated reconfiguration example. In Section 5, , several task-specific configurations and sample applications on the hardware prototype of PARTS are showcased, including shape formation, locomotion, and object manipulation. Finally, in Section 6, we conclude with a discussion of the results and an outlook on future research.

2. Related Work

Self-reconfigurable modular robots are ideal candidates for realizing programmable mechanical structures for versatile applications. The modular design allows the construction of various connection topologies with diverse geometries and a varying range of functionality [5]. The ability for self-reconfiguration further enables these systems to change their connection topology autonomously without external intervention. For a particular geometry or shape to be functional, the configuration needs a certain degree of morphological flexibility such that compliance and functional changes in shape are realizable without topological alteration.

The capabilities of a modular robot are determined by the design of its constituting robotic unit modules. Researchers have grouped modular robots according to their hardware architectures, broadly classifying systems into one of the following types: lattice, chain/tree, truss, and free-form [18]. We will describe each architecture type briefly and highlight designated applications and the absence of desired functionality.

Modular robots with lattice architecture consist of modules or submodules with a regular geometric shape, such as square [19–22] or circular [23–25] modules in 2D or cubic [20,26–29] or (quasi-)spherical [30–34] modules in 3D, respectively. The uniform module geometry and placement of connectors enable these systems to be organized in a regular grid where each module location corresponds to a discrete position in the lattice. The regular structure of the lattice is often exploited to realize self-reconfiguration by moving modules or meta-modules along the surface of the robot [20,35–37] or by moving modules [38] or holes [23,39,40] through the bulk of the robot.

The flexibility of lattice-based systems is limited because their module positions are constrained by the fixed lattice geometry. Achieving morphological changes is inherently tied to topological reconfiguration, which can pose challenges when continuous force or position output is required. Applications of modular robots with lattice architecture are focused on their ability to reproduce geometric shapes through self-reconfiguration, e.g., as a mechanical realization of programmable matter [18,23,31,41]. Locomotion of lattice-based systems is realized using rotational degrees of freedom from individual modules to create a wheeled-like motion [42,43] or utilizing cluster flow, which has also been successfully demonstrated in environments with obstacles [6,44–47]. Other proposed applications for these systems are supporting function [8] or grasping objects by enclosing them with growing module chains [8,48].

The unit modules of systems with chain or tree architecture generally feature one or more rotational degrees of freedom. As the name suggests, the modules can be connected to form chain-like or tree-like topologies resembling hyper-redundant kinematic chains or articulated structures. Most systems also allow the formation of loops through the interconnection of kinematic chains, which are commonly utilized to achieve topological reconfiguration. Modular robots with chain/tree architecture have been demonstrated to accomplish various types of locomotion, such as walking gaits in multi-legged configurations [49–52], rolling in loop configurations [49,52–54], or caterpillar or inchworm gaits in snake-like configurations [49–51,53–55]. Some researchers also demonstrated interaction with the environment by grasping objects with a snake-like configuration [55] or a multi-fingered gripper [7]. Some modular robots feature a hybrid architecture combining aspects of both chain and lattice systems. In principle, these are either systems with a primary chain-like architecture that allows modules to be arranged in a space-filling, typically cubic lattice, e.g., M-TRAN [9,56] or SMORES [57], or systems with a primary lattice architecture, whose modules feature one or more rotational degrees of freedom enabling the construction of articulated serial chains, such as ATRON [40] or Soldercubes [58]. Hybrid systems profit from the flexibility of chain-type systems but are also able to form closely packed space-filling structures and utilize the lattice architecture to guide self-reconfiguration. This versatility enables hybrid systems to cope with complex tasks, such as the M-TRAN system performing a search and rescue mission [5]. However, the space-filling lattice structures of hybrid systems often lack functional compliance and the ability to adapt their morphology without reconfiguration in response to a changing environment.

Modular robots with a truss architecture are built up from truss-based modules that form multi-member connections at their ends such that modules and connections represent the struts and nodes of a truss network. Individual modules can expand or contract to change their length, thus constituting active prismatic joints that realize structural deformation of the network. Some systems incorporate coupled joints, reducing the overall degrees of freedom of the system [59]. Depending on the ratio of the linear actuators forming these prismatic joints, systems with truss architecture offer a wide range of morphological flexibility. In the literature, this shape-shifting ability of truss networks has primarily been employed for locomotion purposes, realizing amoeboid motion [60] or various types of other gaits [61–64]. Qin et al. [65] proposed a modular robot composed of compliant tetrahedral truss elements and demonstrated its ability to grasp objects. However, these systems generally have limited self-reconfiguration capabilities, restricting their operational functionality to tasks that can be handled within a single configuration. The research group around Mark Yim recently proposed a Variable Topology Truss (VTT) [66], a new class of modular robots consisting of a truss network with the ability to adapt the connection topology autonomously. A considerable amount of theoretical work has been conducted regarding locomotion [67–69] and self-reconfiguration [70–72] of VTTs. Ref. [73] proposed an actuated connection mechanism for the VTT that allows for autonomous docking of nodes and tested the mechanism as a proof-of-concept in manually actuated topology modification experiments. However, to the authors' knowledge, the full self-reconfiguration capabilities of the system, including autonomous docking, have not been demonstrated. Yet, variable topology trusses hold great potential as these systems can morph into a wide range of shapes, particularly with innovative active prismatic joints that offer extension ratios beyond 10:1 [74]. As the VTT shares some common characteristics with PARTS, we will discuss similarities and differences in Section 3.3.

In modular robots of the free-form system type, modules can be arranged and clustered in semi-arbitrary positions and thus have the ability to reproduce free-form structures. This flexibility in module arrangement is achieved through a sophisticated connector design such that modules feature a multitude of connection points [75] or the ability to form connections arbitrarily around a sphere, either by magnets [23,76,77] or through temporary soldering [32]. Since these systems are not bound to a small set of limited connection points, they can, to some extent, continuously deform and dynamically adapt and interact with the environment. Demonstrated functionality includes amoeboid motion through an environment with obstacles [75,78] or self-reconfiguration and object manipulation [76]. Free-from systems hold the potential to generate forces and interact flexibly with the environment through collective actuation [79], i.e., the coordinated module movement to achieve smooth shape changes and generate greater actuation forces than individual modules can. However, much of this complex functionality has primarily been explored and demonstrated in 2D systems, with a clear need for further development to extend the concepts into three-dimensional space.

The last modular robot system we want to discuss is the origami robot Mori [80], which has recently been developed further into the Mori3 system [81]. Mori consists of triangular modules that connect and pivot along their edges, thus enabling the robot to fold from a flat geometry into a 3D structure. This innovative robot architecture defies the previously mentioned classification scheme. The update to Mori3 introduced linear actuators to alter the edge lengths of a Mori module by up to 7.5 percent. This upgrade enables the robot to form continuous three-dimensional surfaces that can change their curvature by changing the geometry of the constituent modules. In their most recent work [81], the authors demonstrated the capabilities of their robot system by showcasing diverse tasks, including user interaction, locomotion, and object manipulation. The system architecture of Mori to represent complex surfaces in 3D has great potential, but at the same time limits the viability of harnessing volume scaling effects to increase the force output of the system.

3. PARTS Modular Robot System

During the design process of a modular robotic system, research engineers must balance trade-offs between various factors like hardware architecture, complexity of mechanical design, control, or module size and geometry. The *Planar Adaptive Robot with Triangular Structure* (PARTS) was designed to create a space-filling mechanical structure with two key features: (1) the ability to autonomously change the connection topology of modules, enabling self-reconfiguration, and (2) the ability to deform without changes in the connection topology, enabling compliance and continuous interaction with the environment. In this section, we provide a brief overview of the hardware realization of the PARTS modular robot system [14,15] and describe its mode of self-reconfiguration. We also highlight distinctions in comparison to similar modular robot systems.

3.1. Hardware Design

PARTS is built up from triangular modules with variable edge lengths, which we call *Adaptive Triangular Cells* (ATCs). An ATC consists of three linear actuators connected by revolute joints to form a closed kinematic loop, as shown in Figure 1. Individual modules can change shape by adjusting their edge lengths via linear actuation, providing three degrees of freedom per ATC. ATCs can connect along edges, thus providing the ability to represent arbitrary triangular meshes within the geometric constraints of the modules, illustrated in Figure 2. As unstructured triangular meshes can represent various complex and irregular geometries, these meshes are widely used and well-established in fields such as finite element analysis or computer graphics.

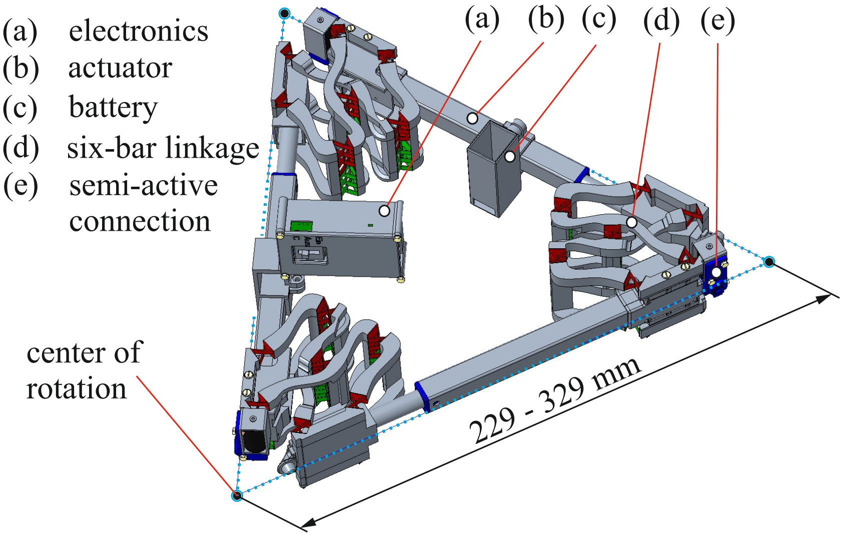

Figure 1. An Adaptive Triangular Cell (ATC) constitutes the unit cell of PARTS. The revolute joint design with 3D printable six-bar linkages realizes the center of rotation outside the extent of the mechanical structure. Refer to Figure 2 for an illustration of multiple connected ATCs with a shared vertex and coinciding center of rotation.

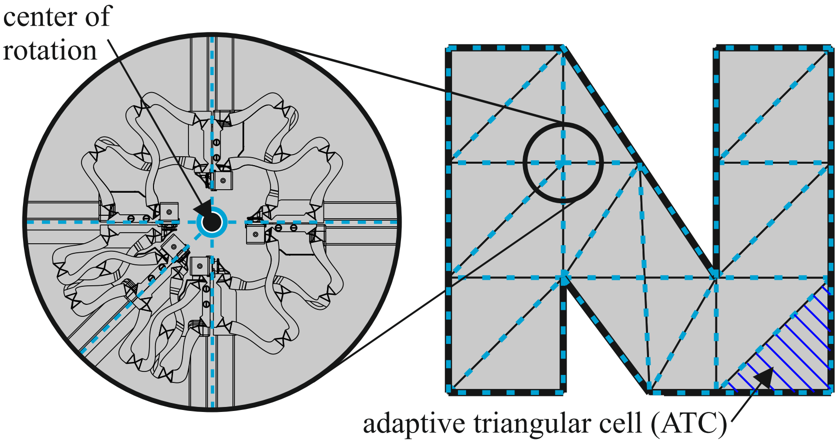

Figure 2. The mechanical design of PARTS enables multiple ATCs to form closed kinematic loops around shared vertices, allowing them to represent space-filling triangular meshes within the geometric constraints of the modules.

When modules connect along edges, vertex points are shared between multiple ATCs. Since these vertex points serve as the center of rotation for multiple revolute joints, ATCs feature a joint design where the center of rotation is positioned outside the extent of the mechanical structure. This is achieved through a modified six-bar linkage with concentrated compliance flexures instead of ball bearings to reduce the manufacturing effort.

The current ATC design employs off-the-shelf actuators *Actuonix L12* with a maximal force output of 22 N [82]. The actuators allow for an effective edge elongation of 43.7 percent, which corresponds to a minimum attainable interior angle of 40.7° and a maximum interior angle of 91.8°. Each edge features a male-female connector pair, positioned alternating to facilitate orientation-independent docking. A male-female connector pair can join passively using a catch plate, which is locked in place by a spring latch mechanism. Slanted walls of the connectors serve as guidance and compensate for misalignment during a connection attempt. ATCs detach by actively releasing the spring of the connection mechanism using a solenoid. A connection leads to a loss of one degree of freedom, as connected actuators drive the same edge of the mesh.

Each ATC features a power supply via a rechargeable lithium polymer battery with a capacity of 350 mAh and a nominal voltage of 7.4 V and is operated by an *Arduino Nano V3.0 ATMEGA328P* which controls the three actuators and solenoids. Additionally, all ATCs are equipped with a wireless module (*nRF24L01 2.4 GHz*) to communicate over the air with a central controller. This central controller acts as the master and sends motion commands to each ATC, which are executed simultaneously. For further information on the hardware design of PARTS, see [14].

3.2. Self-Reconfiguration Principle

The self-reconfiguration ability of PARTS is based on the autonomous modification of the connection topology through the formation and termination of connections. Disconnection of ATCs is feasible while ensuring overall connectivity of the configuration. However, forming new connections requires morphological changes in the robot, such that two free unconnected edges can come into contact. This alignment is typically achieved by forming a new kinematic loop around a shared vertex, as depicted in Figure 3. The modules can then establish new connections along the aligned edges, terminate existing connections, and then change the morphology again to separate previously connected edges. Due to hardware constraints and the maximum achievable angle of 91.6°, at least four ATCs are required to form such a kinematic loop, which we call the *closing condition*. Another approach to achieve reconfiguration is to connect two kinematic chains of ATCs. However, this method might necessitate additional position sensing for error correction due to accumulated errors within the chains [14] and is not considered in the present work.

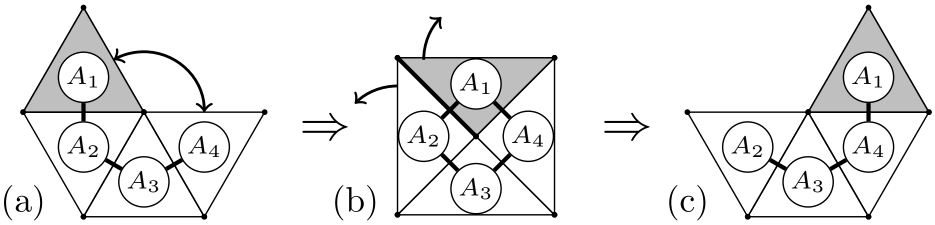

Figure 3. Movement of ATC A_1 from the initial neighboring ATC A_2 (**a**) to the new neighboring ATC A_4 (**c**) by forming the kinematic loop $A_1 - A_2 - A_3 - A_4$ (**b**) around the common vertex. Thick lines mark disconnections, single-headed arrows mark openings, double-headed arrows mark closings with subsequent connections.

3.3. Uniqueness of PARTS System Architecture

The PARTS system defies the classification scheme presented in Section 2 as it combines several properties of different architectural prototypes. PARTS consists of modules with a shape-shifting strut geometry that can be arranged in a space-filling manner. The space-filling arrangement may be of a lattice type based on structured triangular meshes but still exhibits mechanical compliance and morphological flexibility. However, the modules can also be arranged in unstructured meshes within the geometric limitations of the modules. These meshes are continuously deformable and can thus perform a diverse range of tasks, as well as change their topology via self-reconfiguration. We will demonstrate a self-reconfiguration approach on a sample configuration consisting of 62 ATCs in Section 4. In Section 5, we will showcase the morphological flexibility of PARTS in several demo applications, including the generation of a rolling and walking gait, grasping an object, connecting separate clusters of modules, and exerting forces on the environment by supporting a vertical bar.

It is worth noting that the PARTS modular robot system arose as a 2D version of the *Adaptive Robot with Tetrahedral Structure* (ARTS) [83], a modular robot system characterized by shape-shifting tetrahedral unit cells. However, the proposed hardware design could not achieve closed structures and failed to faithfully represent tetrahedral meshes since the center of rotation of the spherical joints constituting vertices does not lie outside of the mechanical assembly space. We decided to transfer the hardware architecture of ARTS to a two-dimensional framework to address and clarify major design challenges essential for a versatile and fully functional 3D robot system, such as joint and connector design and manufacturability, configuration representation, and feasible reconfiguration approaches.

The VTT is arguably the modular robot that shares the most characteristics with PARTS and its proposed 3D equivalent, ARTS. Notable commonalities include the use of linear actuators and nodal joints. The joint design, in particular, exhibits a significant resemblance, as both draw inspiration from the original work by Hamlin [84]. It could be contended that the VTT can reconstruct any PARTS mesh, thereby replicating its kinematics, albeit with substantially greater morphological flexibility due to an increased edge extension ratio. However, there are overarching differences in the design of these systems that set them apart and justify their individual existence.

The constituent elements of the VTT consist of struts which can build up variable truss networks that can change their geometry and topology. Nodes within the network serve as connection points, i.e., the end of each strut module features a chainable spherical joint connection mechanism. This system configuration gives rise to a complex nodal design, relying on two actuated degrees of freedom within each nodal connector. Additionally, owing to the large extension ratios, shape-changing algorithms must adhere to minimum and maximum angles as well as motion constraints within the nodal joints. In contrast, the unit modules of PARTS are triangular elements, with connections occurring along faces rather than nodes. This distinction results in a notably simpler design for the nodal joints, rendering them entirely passive and requiring only compliance with the minimum and maximum angles achievable through the extension ratio of a single element. PARTS builds up finite element meshes with a constant number a triangular elements. This design discrepancy also influences their respective modes of reconfiguration: while the VTT alters the topology of the truss network, PARTS achieves reconfiguration by adjusting the shape and reorganizing volume elements within its finite element mesh framework.

4. Topological Reconfiguration

Modular robots exhibit a high degree of mechanical variability concerning their system architecture, module design, and mode of actuation. Therefore, the realization of self-reconfiguration in each modular robot system requires a unique approach. Autonomous self-reconfiguration is a fundamental capability that enables modular robots to attain a diverse range of functional shapes with varying geometric extents. The ability to modify their connection topology to adapt to changing environments and tasks is a crucial aspect of their versatility.

In previous self-reconfiguration frameworks for PARTS based on single module movement along the surface of the robot [16,17], the closing condition (see Section 3.2) was relaxed to allow kinematic loops with only three participating ATCs. Although this simplifies the reconfiguration approach, such a relaxation requires an edge elongation ratio of at least $\sqrt{3}$, which is not feasible with the existing hardware implementation of PARTS.

In general, addressing hardware limitations involves either enhancing the hardware to eliminate constraints or adapting algorithms to work around hardware limitations. In our case, market constraints related to cost, size, stroke length, and availability of the actuators led us to choose the latter approach. This section outlines potential coping strategies for implementing self-reconfiguration on the current hardware platform utilizing mesh restructuring and meta-module movement.

4.1. Reconfiguration Strategy

The basic principle of topological reconfiguration involves creating new connections by forming kinematic loops, typically around a shared vertex. However, due to limitations in actuation, the hardware platform requires at least four ATCs to construct such a loop (see Figure 3 in Section 3). This constraint renders the previous reconfiguration strategy of single ATC movement along surfaces infeasible. To overcome this limitation, we propose a reconfiguration approach based on two fundamental principles:

1. Movement of meta-modules with enhanced motion capabilities along regular surfaces.
2. Mesh restructuring methods to adjust the mesh topology, create new meta-modules, and incorporate meta-modules into the existing structure to form the desired topology.

To initiate the topological reconfiguration process, we apply mesh restructuring methods on the initial configuration to generate meta-modules consisting of a closed loop of multiple ATCs. These meta-modules can move freely along the surface to regions that require additional modules. Once they reach the designated locations, mesh restructuring methods were executed again to produce the topology of the goal configuration. This process is illustrated in Figure 4. In the following, we will discuss the fundamental principles of meta-module movement and mesh restructuring in more detail.

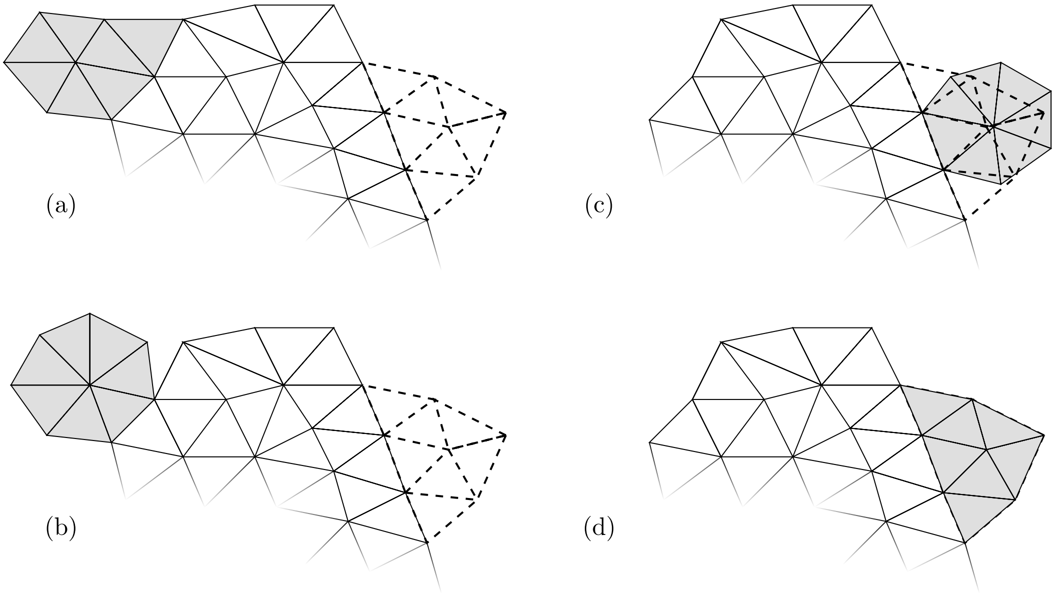

Figure 4. Topological reconfiguration can be achieved by combining mesh restructuring techniques and meta-module movement. To realize the reconfiguration from configuration (**a**–**d**), the grey modules need to be moved to new positions, marked by the dashed ATCs. Mesh restructuring (**b**) allows the generation of mobile meta-modules that can move along the surface to reach their sink region (**c**). After another mesh restructuring, ATCs can populate their designated goal positions (**d**).

4.2. Meta Modules Design and Movement

Many reconfiguration frameworks for lattice-based modular robots rely on the usage of meta-modules, as the motion capabilities of a single module are typically strongly limited [6,20,29,35–37]. A meta-module is an autonomous entity formed by grouping multiple modules, resulting in increased motion capability compared to a single module. The association of modules to meta-modules can be of fixed or variable topology. Meta-modules with a fixed size and shape typically increase the granularity of the lattice (see, e.g., [85]). In contrast, variable meta-modules, regarding the number of modules and connection topologies, emerge during the reconfiguration process from the environment created by other modules and move as an ensemble to a new position. As meta-modules dynamically emerge, no constraints on the underlying structure of the module mesh are introduced [34].

In the present work, we propose the use of emerging meta-modules consisting of a variable number of modules that form a kinematic loop around a shared vertex. Meta-modules of varying sizes are depicted in Figure 5. The meta-module design of ATCs constituting a kinematic loop around a common vertex in their middle enables surface motion along *regular surfaces*. We define a regular surface as a surface consisting of ATCs that are each connected to two other ATCs, thus having exactly one free edge. Regular surfaces contain no ATCs with only one connection to the remaining robot. Meta-modules can move along such surfaces as the closing condition of four ATCs around a common vertex is satisfied at all positions on the surface (two ATCs of the meta-module, the connected surface ATC, and one neighbor of the connected ATC). Hence, a kinematic loop can freely move along a regular surface utilizing a rolling motion.

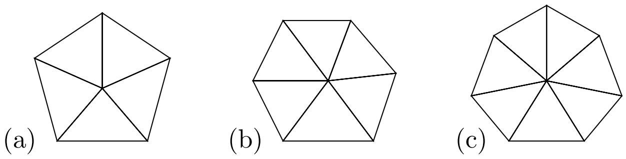

Figure 5. Meta-modules of different sizes consisting of 5 (**a**), 6 (**b**), and 7 (**c**) ATC.

To reduce kinematic constraints and facilitate loop formation, *selective disconnection* can be applied, which involves temporarily disconnecting ATCs to increase local motion capabilities. Two deterministic disconnections are employed for each meta module movement step. This method could be extended to a multi-step approach, disconnecting ATCs only if a connection is not feasible with the current topology, i.e., selective disconnection is only applied when the motion capabilities of all ATCs involved in the new loop formation are not sufficient to establish the loop. Considering Figure 6 with the meta-module $A_8 - A_9 - A_{10} - A_{11} - A_{12}$ moving in a counter-clockwise direction, we disconnect (a) the connecting ATC (A_2) and the neighboring ATC in moving direction (A_6) and (b) the loop ATC connected to the remaining robot (A_8) and the closest loop ATC opposite to the moving direction (A_9). This procedure removes sufficient kinematic constraints to enable loop formation of the meta-module (A_{12}) with the next surface ATC in moving direction (A_2). In the last step, the previous connection of the meta-module with the remaining robot is terminated (disconnection of A_8 and A_1) and the topology of the meta-module and remaining robot is restored by reconnecting ATCs A_6 and A_2 as well as A_8 and A_9. Please note that also other connections can be terminated to achieve a similar result.

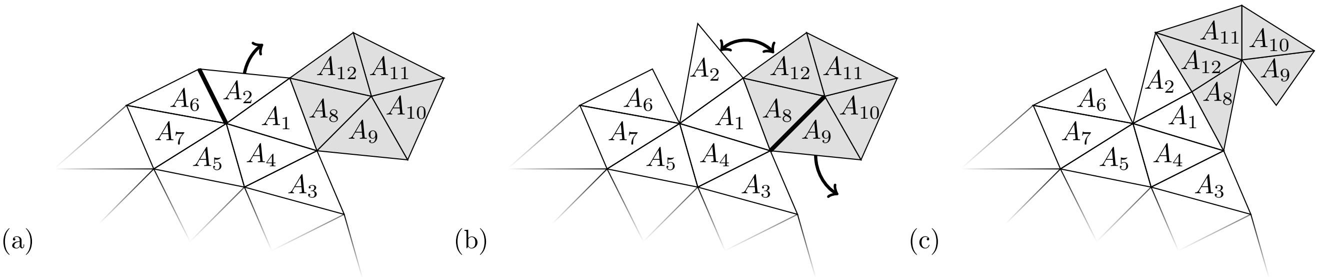

Figure 6. Selective disconnection temporarily removes kinematic constraints and enables loop formation. For the meta-module $A_8 - A_9 - A_{10} - A_{11} - A_{12}$ to move in counter-clockwise direction by forming a loop between ATCs A_2 and A_{12}, the connecting ATC A_2 and the neighboring ATC in moving direction A_6 are disonnected (**a**) and the loop ATC connected to the remaining robot A_8 and the closest loop ATC opposite to the moving direction A_9 (**b**). This removes sufficient kinematic constraints to enable loop formation and a connection of A_2 and A_{12} (**c**). In the last step, the previous connection of the meta-module with the remaining robot between A_8 and A_1 can be terminated and the topology of the meta-module and remaining robot can be restored by reconnecting ATCs A_6 and A_2 as well as A_8 and A_9 (not shown). Thick lines mark disconnections, single-headed arrows mark openings, double-headed arrows mark closings with subsequent connections.

The principle of loop rolling along the surface is depicted in Figure 7. The meta-module, composed of four ATCs, i.e., A_6, A_7, A_8, and A_9 (a), can move clockwise along the surface by forming a kinematic loop with the next surface ATC. Selective disconnection of the surface ATCs A_2 and A_4, as well as the loop ATCs A_6 and A_7, increases the motion capabilities and enables ATCs A_5 and A_8 to close the kinematic loop and form a new connection (b). Once the connection is established, the connection with the previous surface ATC can be terminated, i.e., ATCs A_4 and A_6 disconnect. The surface and meta-module morphology and topology are then restored by reconnecting ATCs A_2 and A_4, and ATCs A_6 and A_7 (c). This sequence of selective disconnection, formation of a new loop, establishment of a new connection, disconnection, and restoration of surface and meta-module geometry can be repeated to achieve rolling locomotion of the meta-module along the surface. Supplementary Material Video S1 showcases a simulation and hardware demonstration of meta-module movement as presented in Figure 7. The surface motion of PARTS by module deformation is similar to the metamorphic robot system [86]. However, due to geometric constraints, the motion in PARTS cannot be achieved by single module deformation as in the metamorphic robot.

Meta-modules emerge from source regions in the initial configuration with an excess of ATCs. These meta-modules can be modified by adding or removing ATCs from the kinematic loop, thereby changing the size of the meta-module. In the following sections, we will examine the mesh restructuring techniques required to modify the size of the meta-modules and adapt the internal topology of the mesh.

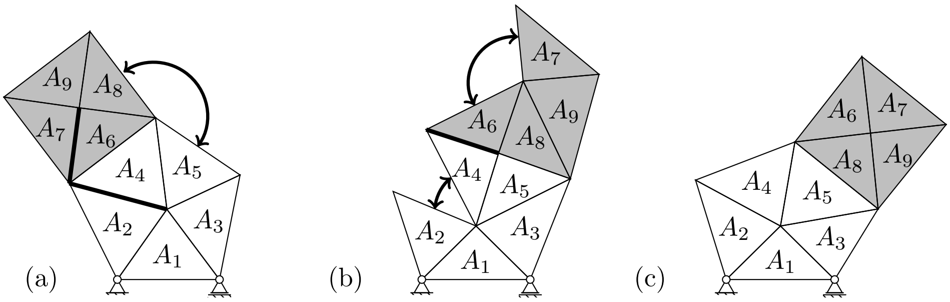

Figure 7. Meta-module movement along a regular surface: The meta-module, consisting of four ATCs (A_6, A_7, A_8, and A_9), can move clockwise along the surface by forming a kinematic loop with the next surface ATC. Selective disconnection of ATCs (A_2 and A_4, A_6 and A_7) is applied to increase the motion capabilities, allowing ATCs A_5 and A_8 to close the kinematic loop and form a new connection (**b**). The disconnection of ATCs A_2 and A_4, and restoration of the surface and meta-module morphology and topology results in the clockwise motion of the meta-module (**c**). Thick lines mark disconnections, double-headed arrows mark closings with subsequent connections.

4.3. Mesh Restructuring

Fully partitioning an arbitrary mesh into meta-modules is generally not possible, as it would require every internal vertex to belong to an even number of modules. Removing a meta-module can result in the creation of *simply connected ATCs*, which have two free edges and only one connection to the rest of the robot, or kinematic chains on the surface. Furthermore, the size of a given meta-module may not be appropriate for placing the next modules in the goal configuration, highlighting the need for mesh restructuring methods. These methods involve motion sequences for altering the internal topology of the mesh, such as excluding, integrating, or transferring individual ATCs from or into existing kinematic loops.

In this section, we present several mesh restructuring methods and describe situations where they can be applied during reconfiguration. We do not aim to present a complete self-reconfiguration algorithm and acknowledge that this set of restructuring primitives may not be sufficient to guarantee self-reconfiguration. Rather, our goal is to demonstrate that topological self-reconfiguration is feasible with the current system architecture and a sophisticated planning framework. However, the detailed implementation of such a framework is outside the scope and intent of the present work.

All mesh restructuring techniques are based on distinct sequences of opening and closing operations for kinematic loops close to the surface. The motion primitives utilize the fact that every internal vertex of the mesh belongs to at least four ATCs, which guarantees that the closing condition of four ATCs around every shared vertex for new kinematic loops is always fulfilled. In the following, we will analyze the *loop transfer primitive* as a basic motion sequence for topological mesh restructuring and describe how this primitive can be altered or queued to achieve various restructuring results.

The *loop transfer primitive* allows for the transfer of an ATC between two interconnected kinematic loops that share two ATCs. We will examine the necessary steps in the example configuration shown in Figure 8a, with the two kinematic loops $A_1 - A_2 - A_4 - A_6 - A_5 - A_3$ and $A_6 - A_8 - A_9 - A_7 - A_5$ sharing the ATCs A_5 and A_6. To transfer an ATC, one loop next to the surface between a shared ATC (A_6) and a surface ATC (A_4) is opened, as depicted in (b). Next, the other loop is opened between the ATCs previously shared by both loops, i.e., ATC A_5 and A_6 in (c). The two loops reform in the same order as they opened by first closing and connecting ATCs A_4 and A_5 in (d), followed by A_4 and A_6 in (e). The net result of this mesh restructuring primitive is the transfer of an ATC from one loop to the other. Please note that the loop transfer motion primitive can be applied regardless of the distinct number of ATCs in both loops.

The loop transfer primitive can be adapted to serve various purposes. For example, the motion sequence can be repeated on the other side of the connected loops to restore the initial number of ATCs per kinematic loop, as shown in Figure 8f–i. This sequence also results in displacing the loops with respect to each other when comparing the initial configuration in (a) and the final configuration in (i). When sequenced multiple times, a meta-module can be moved within the robot structure close to the surface to a more favorable position, such as adjacent to a regular surface, to initiate surface motion. Supplementary Material Video S2 showcases a simulation of the loop transfer primitive as presented in Figure 8 conducted twice.

Furthermore, the loop transfer primitive can be partially applied to integrate simply connected surface ATCs into a moving meta-module, defining a *meta-module integration primitive*. For example, in Figure 8b, suppose the ATCs A_3, A_1, and A_2 belong to a surface section in a larger PARTS system (not shown). The meta-module $A_6 - A_8 - A_9 - A_7 - A_5$ can incorporate the simply connected ATC A_4 by following the loop transfer steps from Figure 8c–e and additionally disconnecting ATCs A_2 and A_4 instead of ATCs A_5 and A_7 in the last step. Similarly, excluding ATCs from a meta-module can also be accomplished by executing the respective steps in reverse order.

Another modification of the loop transfer primitive is the *loop integration primitive*, depicted in Figure 9. In this motion primitive, a simply connected ATC on the surface can be integrated into an existing loop in the robot structure. Please note that the motion sequence is essentially the same as for the loop transfer and also entails the transfer of an ATC from one loop to the other, in addition to integrating the ATC. To achieve the sole integration of the simply connected ATC into one of the participating kinematic loops, an additional loop transfer has to be performed. Loop integration can also be performed in reverse order to exclude an ATC from an existing kinematic loop. A video demonstrating the loop integration primitive conducted on the hardware prototype can be found in Supplementary Material Video S3.

The mesh restructuring methods around the loop transfer primitive can hence be applied in various ways to adapt the internal topology of the mesh close to the surface, to generate suitable meta-modules for surface motion, or to deconstruct meta-modules and fill empty positions in the goal configuration. We will demonstrate these methods in the following section using a simulation example.

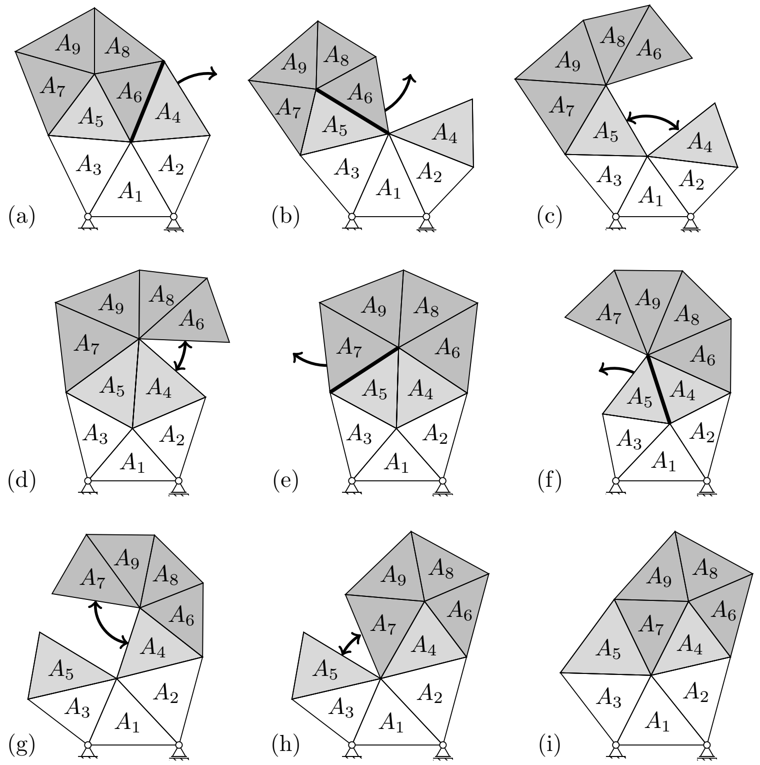

Figure 8. Loop transfer mesh restructuring primitive: Two adjacent kinematic loops with shared ATCs A_5 and A_6 (**a**) can transfer an ATC shared between them from one loop to the other by applying coordinated opening and closing operations. The first kinematic loop $A_1 - A_2 - A_4 - A_6 - A_5 - A_3$ opens by disconnecting ATCs A_4 and A_6 (**b**), followed by an opening of the second loop $A_5 - A_6 - A_8 - A_9 - A_7$ between ATCs A_5 and A_6 (**c**). The kinematic loops are re-established in the same order as they were opened, with the first loop closing (now with one ATC less than before the sequence) by connecting ATCs A_4 and A_5 (**d**), followed by the second loop closing (now with one ATC more than before the sequence) by connecting ATCs A_6 and A_7 (**e**). To restore the original number of ATCs for each loop, the same steps can be repeated on the other side of the loops by opening the second loop (**f**), opening the first loop (**g**), closing the second loop (**h**), and closing the first loop (**i**). This results in a net displacement of the loops with respect to each other (Observe the position of ATCs A_6, A_7, A_8, and A_9). Thick lines mark disconnections, single-headed arrows mark openings, double-headed arrows mark closings with subsequent connections.

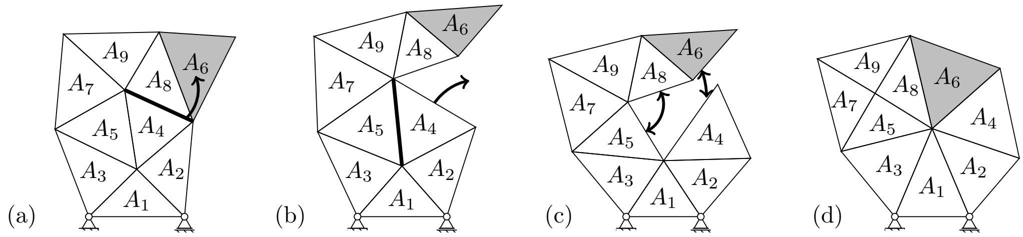

Figure 9. Loop integration primitive: The simply connected ATC A_6 can be integrated into the existing loop $A_4 - A_2 - A_1 - A_3 - A_5$ by applying an adapted loop transfer operation. Two adjacent kinematic loops with shared ATCs and a simply connected ATC on the surface (**a**) are opened subsequently by terminating the connections between ATCs A_4 and A_8 (**b**), and A_4 and A_5 (**c**). The kinematic loops are closed in the same order as they were opened by connecting first ATCs A_5 and A_8, followed by connecting ATCs A_4 and A_6 (**d**). Please note that this process also involves a loop transfer of the shared ATC A_4. Loop integration can also be executed in reverse to exclude an ATC from an existing loop. Thick lines mark disconnections, single-headed arrows mark openings, double-headed arrows mark closings with subsequent connections.

4.4. Reconfiguration Example

In this section, we demonstrate the proposed methods for topological self-reconfiguration,, i.e., meta-module movement and mesh restructuring, on a simulated reconfiguration example. The reconfiguration workflow presented in Section 4.1 is employed on a sample configuration consisting of 62 ATCs. Similar to our previous work [17], the largest common topology is identified and source regions and sink regions for modules are determined. Source regions define topological locations with a surplus of ATCs, while sink regions are surface positions that need to be occupied to form the final configuration. These regions prescribe the flow of ATCs along the surface to realize the reconfiguration and serve as a basis for deciding the direction along the surface in which modules in source regions are supposed to move. Morphological shape changes which are required for module transport along the surface are realized utilizing the constraint optimization framework briefly summarized below.

A finite element representation of the configuration is used to calculate the vertex coordinates required for closing a kinematic loop. The algorithm relies on minimizing the objective function $f(\boldsymbol{n})$ representing the distance between two connecting vertices v_a and v_b with corresponding coordinates $\boldsymbol{n}_a$ and $\boldsymbol{n}_b$, respectively, i.e.,

$$f(\boldsymbol{n}) = |\boldsymbol{n}_a - \boldsymbol{n}_b|^2. \tag{1}$$

Constraints are given by the minimal edge length l_{min} and maximal edge lengths l_{max} between two connected vertices v_i and v_j, i.e.,

$$\begin{aligned} c_{min} &= |\boldsymbol{n}_i - \boldsymbol{n}_j|^2 - l_{min}^2 \geq 0 & (2) \\ c_{max} &= l_{max}^2 - |\boldsymbol{n}_i - \boldsymbol{n}_j|^2 \geq 0. & (3) \end{aligned}$$

Additionally, constraints are included to guarantee the relative node orientation of ATCs. For an ATC defined by its three vertex points v_i for $i = 1, 2, 3$ in a counter-clockwise manner with corresponding coordinates $\boldsymbol{n}_i = (n_{ix}, n_{iy})$, the orientation is given by the half-space constraint

$$\begin{aligned} c_{hs} = {} & n_{1x}n_{2y} + n_{2x}n_{3y} + n_{3x}n_{1y} \\ & - n_{1x}n_{3y} - n_{2x}n_{1y} - n_{3x}n_{2y} > 0. \end{aligned} \tag{4}$$

Half-space constraints are also utilized to resolve collisions between ATCs, ensuring that vertex positions reside on distinct half-spaces and preventing vertex penetration into other ATCs. It is worth noting, that Equation (4) is derived from the signed volume of a triangle, also referred to as a 2-simplex. The volume of a k-dimensional simplex is given by

$$V = \frac{1}{k!}\begin{vmatrix} \boldsymbol{n}_1 & \boldsymbol{n}_2 & \cdots & \boldsymbol{n}_{k+1} \\ 1 & 1 & \cdots & 1 \end{vmatrix}, \tag{5}$$

where k is the dimension, and $\boldsymbol{n}$ represents the coordinate vectors of the $k+1$ vertices [87]. Setting $k = 2$ and omitting the constant prefactor, Equation (4) is obtained. Notably, if one vertex changes sides with respect to the plane defined by the other vertices, the sign of the volume changes.

The complete constraint optimization for changing the morphology and establishing new connections is thus given by

$$\min_{\boldsymbol{n}} f(\boldsymbol{n}) \text{ such that} \begin{cases} c_{min} & \geq 0 \\ c_{max} & \geq 0 \\ c_{hs} & > 0 \end{cases}, \tag{6}$$

with the vector $\boldsymbol{n}$ containing the stacked vertex coordinates. The number of variable vertices included in the optimization is extended in a multi-step process until the objective function converges to a desired minimum, i.e., the coordinates of the connecting vertices are equal. The objective function $f(\boldsymbol{n})$ can be adapted to achieve different shape formation goals besides connecting vertices, e.g., opening a kinematic loop and separating vertices or minimizing or maximizing angles between edges. Figure 10 depicts an overview of the algorithm for morphological adaptation.

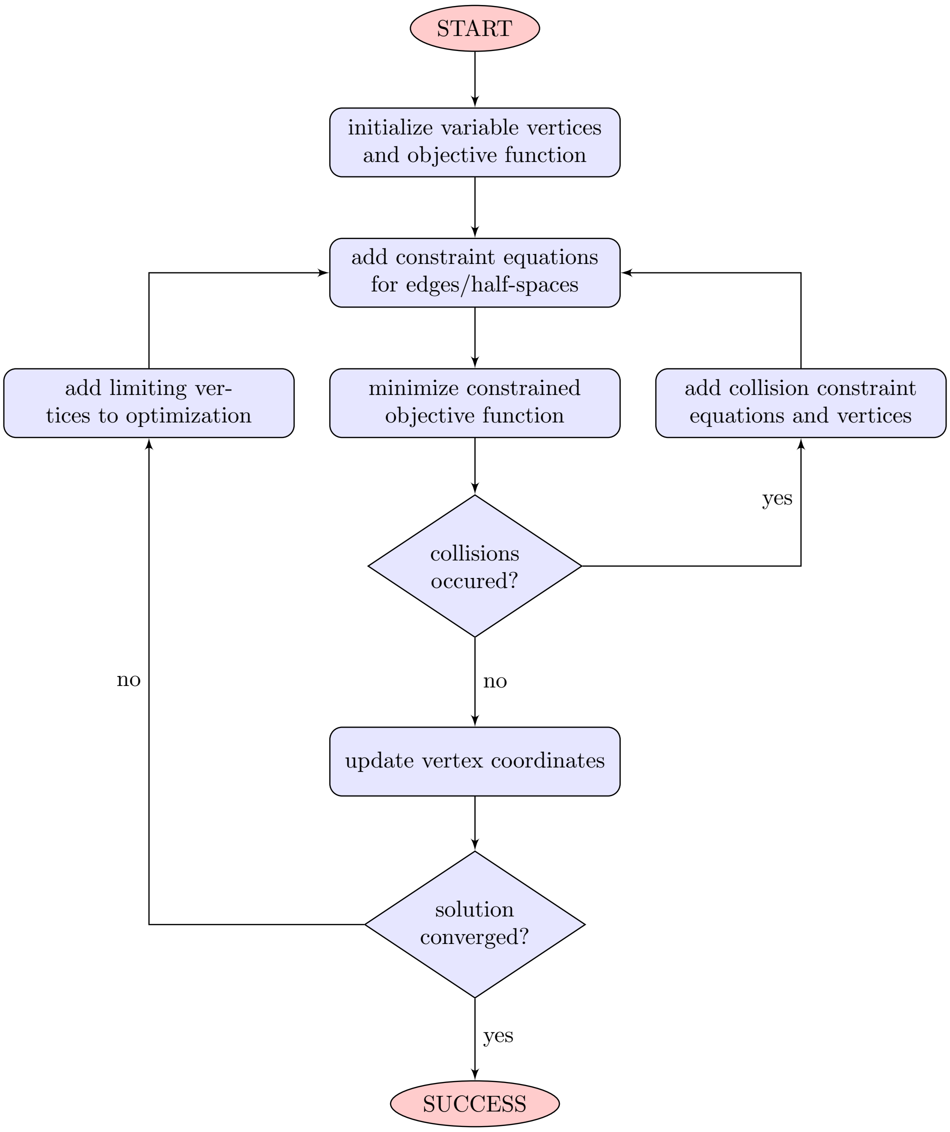
START
initialize variable vertices and objective function
add constraint equations for edges/half-spaces
add limiting vertices to optimization
minimize constrained objective function
add collision constraint equations and vertices
yes
collisions occured?
no
no
update vertex coordinates
solution converged?
yes
SUCCESS

Figure 10. Overview of the iterative optimization framework: Intermediate solutions are checked after each minimization step for collisions. In case of collision, the minimization is repeated with additional constraints. The optimization domain is extended by including additional vertices until the solution converges.

The major difference to our previous work lies in the mesh restructuring and the moving of meta-modules instead of single ATCs. Meta modules are selected from the source region such that the number of surface ATCs to traverse is minimized for succeeding modules. Source ATCs not associated with a specific meta-module can either (1) move by themselves when the surface topology permits it, or (2) be integrated into an existing meta-module. Mesh restructuring methods are applied in the source region to generate meta-modules of sufficient size to populate the next unoccupied positions. In this step, additional ATCs are included into the meta-module that may be excluded once reaching the sink region to fill places of ATCs with no meta-module affiliation in the goal configuration. Kinematic chains can connect the base of the chain with a neighboring surface ATC to reduce the chain length and form new meta-modules.

Consider the initial and final configuration depicted in Figure 11a,d, respectively. The reconfiguration involves a change in shape from an initial bar-like geometry to a T-like shape. Intermediate configurations are depicted in Figure 11b,c. A simulation of the reconfiguration is available online and can be viewed in Supplementary Material Video S4.

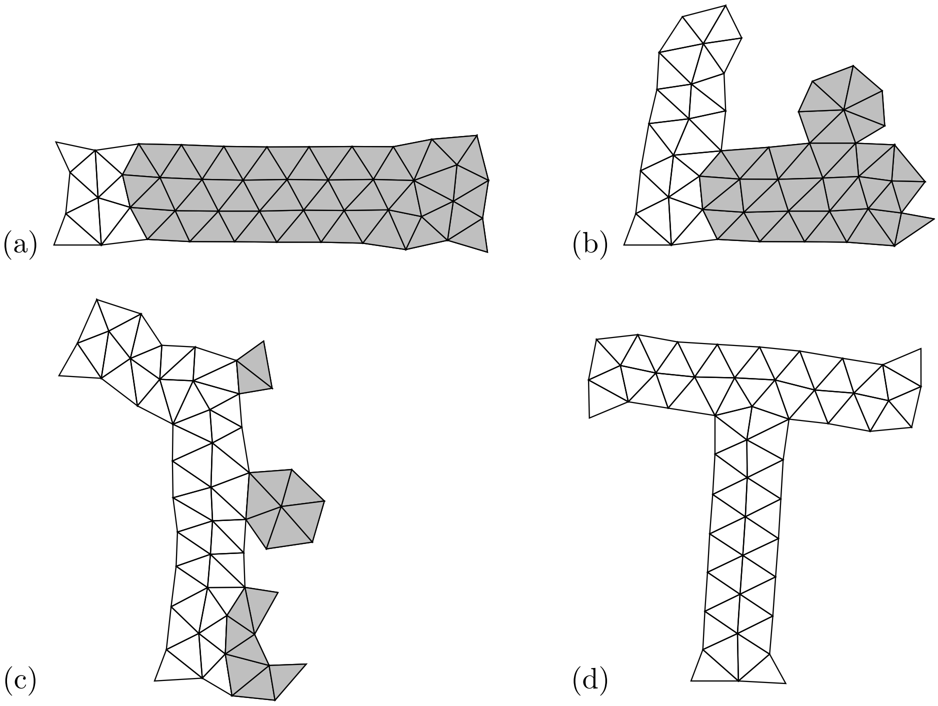

Figure 11. Reconfiguration Example with 62 ATCs. Modules that need to be moved to establish the goal configuration are highlighted in gray: The initial bar-like configuration (**a**) undergoes a reconfiguration into a T-like shape (**d**). The intermediate configurations after 30 (**b**) and 98 (**c**) reconfiguration steps are also shown in the figure. The complete reconfiguration process required 117 steps. A video of the entire simulation is available in Supplementary Material Video S4.

Please note that the proposed reconfiguration methods require a sufficient number of modules to fully utilize the mesh restructuring capabilities. In the presented example, ATCs need to be moved from a single source region to a single sink region. More complex reconfigurations may rely on populating goal regions with fewer ATCs than those contained in a meta-module. The surface flow reconfiguration planning of our previous work [17] may still be employed, however, further extensions will be necessary to guarantee completion in such cases. A comprehensive reconfiguration algorithm must incorporate thorough planning to ensure that module demands are met and no single ATCs are left behind, incapable of movement without meta-module assistance. A possible solution may involve utilizing meta-modules solely dedicated to incorporating and transporting individual modules to their goal region, returning after delivery to their initial position in the topology.

In addition, collision detection and avoidance must be addressed during the self-reconfiguration process. While a detailed analysis of the feasibility of self-reconfiguration for various configurations and the implementation and evaluation of the algorithm's efficiency and complexity are well beyond the scope and intent of this paper, these aspects are essential considerations for future research in this field.

5. Functional Change in Morphology

Topological reconfiguration is a crucial capability to realize large *structural changes* in the geometry of a modular robot. However, in most real-world applications, the robot needs to undergo *functional morphological changes*, i.e., adapt its geometric shape to task-specific intermediate configurations, often without modifying the underlying connection topology. In this section, we will examine the morphological flexibility of the PARTS system and showcase several application examples. These example tasks can be regarded as constituents to realize more complex behavior, thus demonstrating, as a whole, the system's versatile functionality. We will outline how the morphological changes can be realized for each application type since the robot's motion is typically driven by task-specific requirements that differ between applications.

Figure 12 depicts the measurement setup for the experiments investigating morphological changes of PARTS. The measuring range of camera 1 (*DFK AFU050-L34* with 2592 × 1944 MJPG) was calibrated to 1115 mm × 710 mm and has a relative error in marker detection of 0.22 mm. Additionally, the measurement setup has a second camera (2) (*Intel RealSense D415*) for gesture recognition, which is not used in the present work. In most experiments, the PARTS system (3) is fixed to the ground (4). Measured positions of PARTS are primarily used to compare the computed positions of the simulation with the experiment. In all experiments and for all modules, the same firmware was used. Pre-defined high-level configuration commands, consisting of three edge lengths and connector states for each ATC, were sent wirelessly from a central master controller to all modules simultaneously. As the actuators are not able to follow synchronized motion profiles but only perform point-to-point trajectories between configurations, intermediate subconfigurations are introduced in the motion plan to avoid the loss of topological integrity due to internal stresses. Empirical data showed that the compliance introduced by the flexible joints is sufficient to account for forces and misalignment during the transition between these subconfigurations. The pose of all modules is estimated via a camera using colored markers. This position information is used in some of the experiments as an initial geometric configuration for motion planning. All photographs of the experiments include the measured (blue) and estimated (orange) positions of these markers. Videos of all demo applications performed on the hardware prototype can be found in the Extensions.

5.1. Locomotion

Locomotion is an important ability for modular robots as mobility enables them to change their position in space and move autonomously to locations where their presence is required for other expedient tasks. The system architecture of PARTS allows the realization of a wide range of different locomotion patterns. We will focus on gait-based locomotion and present two different gaits—a *rolling gait* for meta-module movement and a *walking gait* for larger robot structures. It is worth noting that flow locomotion utilizing cluster flow (as discussed, e.g., in [6,46,88]) can also be achieved by applying the reconfiguration methods presented in Section 4. However, as flow methods are considered inefficient in terms of time and energy requirements [4], we will not further elaborate on this locomotion strategy.

In gait-based locomotion, the robot's topology remains fixed, and movement is achieved solely through morphological changes. These changes are typically coupled to cyclic actions and require synchronization of the modules. The periodic motion is commonly generated by *central pattern generators* (CPGs) [89–91]. Incorporating sensory feedback can enhance the motion output of the CPG and enable the robot to traverse uneven terrain and obstacle-filled environments [92].

A common locomotion strategy for modular robots with truss architecture is known as rolling or tumbling gait [63,67,93]. In this gait, the modular robot shifts its center of gravity (COG) outside the supporting area of its base, causing the robot to tip over. The weight-shifting and toppling process can then be repeated from the new base that is in contact with the ground, resulting in a rolling type of motion. The direction of motion is determined by the predominant direction of the COG shift and the relative position of the new supporting surface to the old.

Rolling motion has already been discussed in Section 4.2 for meta-modules of PARTS during surface motion to achieve reconfiguration. However, the rolling gait can also be applied for individual isolated meta-modules to traverse even surfaces, as depicted in Figure 13. The initial meta-module consisting of 6 ATCs has contact with one ATC to the ground (a). Through deformation, the next ATC in the movement direction also establishes contact with the ground (b). The meta-module then shifts its center of mass (c), allowing the ATC with initial ground contact to be lifted without toppling backward (d). In the last step, the meta-module geometry is restored (e). By performing these steps periodically, the meta-module can move forward in a rolling motion. A video of the rolling gait performed with the hardware prototype can be viewed in Supplementary Material Video S5.

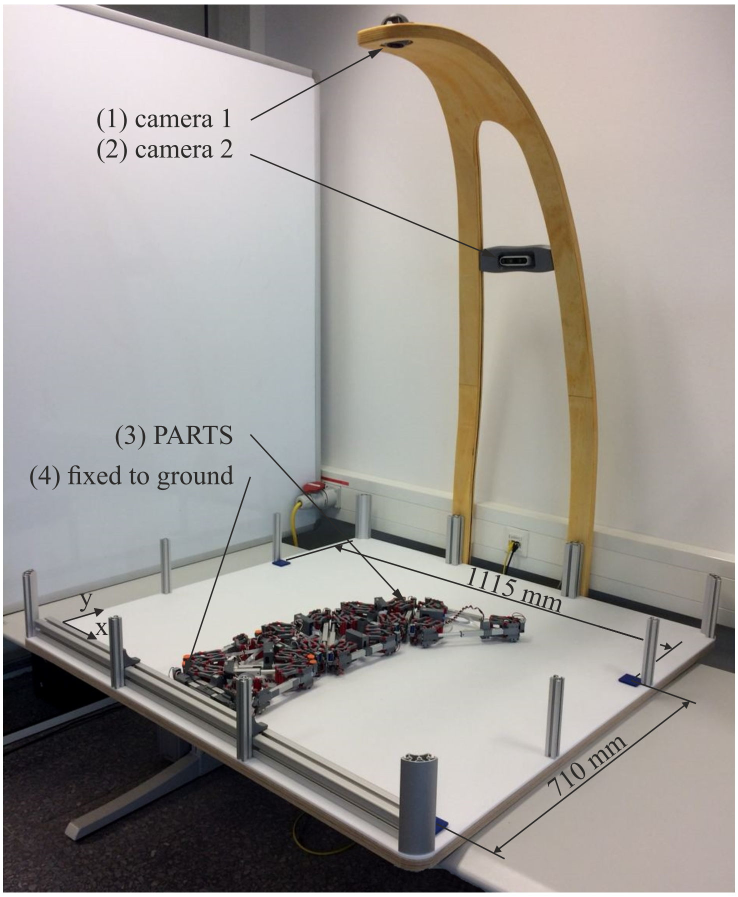

Figure 12. Experimental setup for testing funcional changes in morphology.

Configurations for toppling were estimated assuming a uniform weight distribution with the geometric centroid as the COG. However, this approach did not account for the asymmetric weight distribution caused by the control unit and battery pack, each of which is mounted to one of the actuators (see Figure 1). Consequently, calculating the required configuration to achieve a weight shift for toppling depends on the orientation of each ATC, making the use of CPGs for gait realization challenging. The experiment was conducted on an inclined plane of 15° to reduce the necessary weight shift for toppling. A solution to this problem involves achieving a symmetric, orientation-independent weight distribution by balancing the masses on edge actuators using dummy weights.

With coordinated control and selective disconnection, it would be possible to prevent meta-modules from toppling over. Instead, two consecutive supporting edges can maintain simultaneous contact with the ground during the COG transition, resulting in a smoother motion without any impacts due to falling. This has particular relevance for movement on unstable surfaces since excess forces on impact due to toppling can cause the surface support to collapse, such as when crossing heavily damaged bridges during a disaster response mission. It is worth noting that a rolling gait can be achieved using any PARTS topology that can sufficiently shift its center of gravity so that the projection of the COG onto the ground lies outside the convex hull of the contacting area.

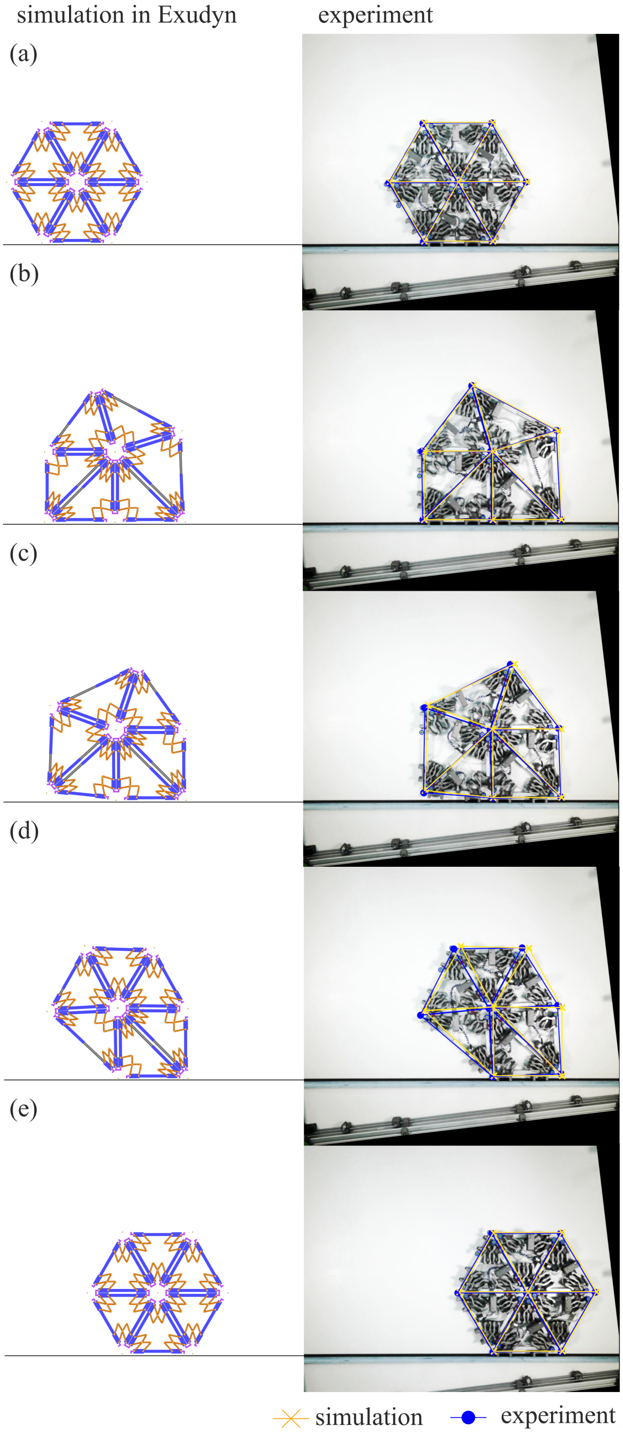

Figure 13. Rolling gait of a PARTS meta-module on an inclined plane with an angle of 15°: the initial meta-module consisting of 6 ATCs has contact with one ATC to the ground (**a**). Through deformation, the next ATC in the movement direction also establishes contact with the ground (**b**). In the following step, the meta-module shifts its center of mass (**c**), so that the ATC with initial ground contact can be lifted without toppling backward (**d**). Finally, the initial meta-module geometry is restored (**e**). By performing these steps periodically, the meta-module can move forward in a rolling motion. Please note that the images of the experiment were rotated to agree with the simulation.

The second gait we want to showcase for PARTS is the *walking gait*. In this type of locomotion, coordinated control of the legs in a multi-legged robot configuration enables the robot to move forward while maintaining support. Walking gaits have been predominantly realized by modular robots with a chain or hybrid architecture [11,40,56], but have also been successfully demonstrated on truss-type robots [63].

The biped walker configuration shown in Figure 14 was chosen to demonstrate the walking gait of PARTS. The walking gait is achieved through a cyclic control pattern that involves sequential steps of weight-shifting (b), extension of the front leg (c), weight-shifting (d), and contraction of the rear leg (e). A video of this walking gait demonstrated on the biped walker configuration is provided in Supplementary Material Video S6. Please note that repositioning can also be accomplished solely by sliding, taking advantage of the differing friction resulting from the weight shift [62]. However, for larger systems, multi-legged walker configurations are generally more desirable since they can achieve smoother motion without relying on additional weight-shifting steps to maintain stability. Additionally, the unique system architecture of PARTS allows for generating a peristaltic-like gait in such multi-legged configurations [16].

5.2. Module-Module Interaction

In the following section, we will demonstrate the interaction of independent clusters of modules by connecting and combining them into a larger structure with broader functionality. This process is commonly referred to as *self-assembly* [18,94]. As the initial positions of the modules or clusters of modules with respect to each other are generally not precisely known, statistical methods [4] or external guidance through additional sensory inputs, such as cameras [95], infrared sensors [96], or hall sensors [97], are employed to achieve alignment and docking.

Consider the experimental setup depicted in Figure 15, consisting of two independent clusters of modules, separated by a certain distance from each other and fixed to the ground to prevent tipping (a). In this example, the two clusters will connect to form a bridge that spans the gap between them (b). The position of the connecting edges was determined using a constrained optimization procedure similar to the one presented in Section 4.4. The position information of the modules determined by a visual localization system was used as the initial input for the optimization process. A video of the experiment can be viewed in Supplementary Material Video S7.

Additional sensor systems may be necessary to ensure a successful connection between separate clusters of ATCs during field deployment. A camera-based localization system, such as the one used in the experimental setup, may be impractical for this purpose due to environmental conditions or limited visibility. Additionally, errors due to gravity or joint compliance can affect the internal state of the robot system, leading to deviations from the ideal internal kinematic state. To overcome these challenges, possible solutions could include combinations of local or global positioning systems with additional sensory inputs, such as infrared sensors, to aid in the docking process. Laser tracking of the docking modules could also be utilized to accurately recognize the current pose of the modules and ensure precise alignment during connection.

Besides using the presented methods for self-assembly to form structural systems like bridges or halls, they can also be used in reverse as a form of “self-disassembly" to transport clusters of modules to hard-to-access places, such as across rivers or onto the tops of buildings.

5.3. Interaction with the Environment

Real-world applications typically involve interaction between the robot and the environment. Thus, one of the key functionalities of a modular robot system is its ability to interact flexibly with its surroundings to accomplish a wide range of tasks and mission objectives. This interaction can be achieved through the robot’s shape and form or by the controlled generation of forces applied to the environment. To illustrate both possibilities, we will provide an application example for each. Interaction through form is demonstrated by showcasing the robot’s ability to grasp an object, while interaction through force is exemplified by the robot’s ability to support a vertical structure.

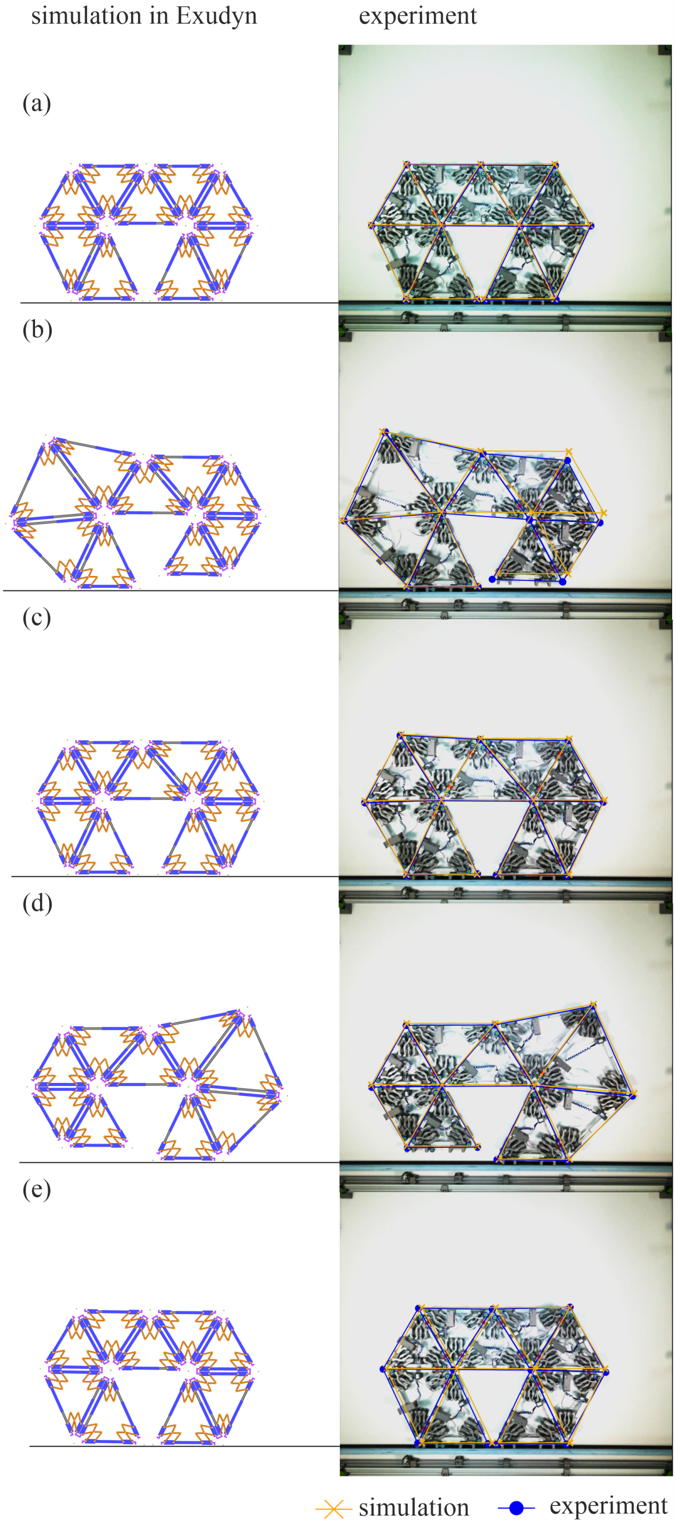

Figure 14. Walking gait of a PARTS biped walker on an inclined plane with an angle of 5.2°: the initial configuration consists of a two-legged structure constructed from 9 ATCs (**a**). The front leg is lifted off the ground and moved forward through the extension of the middle ATC by simultaneous deformation of the ATCs constituting the rear leg for counterbalance (**b**). Next, the front leg makes contact with the ground (**c**) and the rear leg is lifted and moved forward by contraction of the middle ATC (**d**). Deformation of the ATCs at the front is here also required for counterbalance. After the rear leg is placed on the ground (**e**), the walker reaches its initial morphology, and the steps (**b**–**e**) can be repeated to achieve a continuous forward motion.

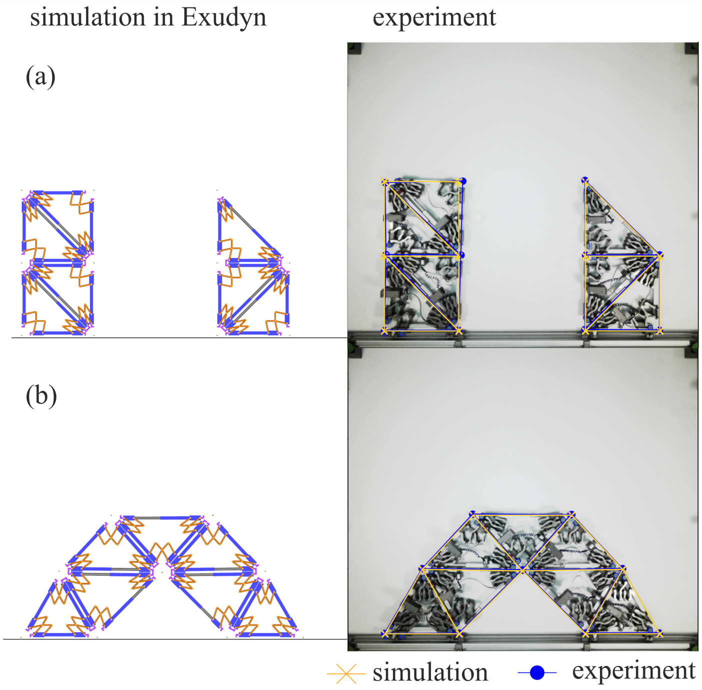

Figure 15. Two separate module clusters, one consisting of 4 ATCs and the other consisting of 3 ATCs, are initially fixed to the ground with a certain distance between them (**a**). The clusters can change their morphology and connect to form a bridge structure (**b**).

Objects can be grasped by utilizing a gripper configuration featuring multiple kinematic chains acting as fingers for grasping. The exact gripper configuration can be adapted to the shape and size of the object that has to be picked up. We considered a two-fingered gripper grasping a circular object, as shown in Figure 16. The object is grasped by creating an enclosure around it, enabling the robot to displace and manipulate the object. A video of the grasping process can be found in Supplementary Material Video S8. More elaborate grasping approaches for multi-fingered grippers utilizing form closure can be realized by defining gap functions and first-order closure conditions (see, e.g., [98]).

In the second example, we aim to showcase the ability of PARTS to actively generate directed forces on the environment. For this purpose, the setup depicted in Figure 17 is used, consisting of a PARTS system with 9 ATCs and a horizontally suspended vertical bar. The bar represents a supporting structure, such as a wall or a building, that requires stabilization. To quantify the forces exerted by PARTS, a measurement setup, depicted in Figure 17a, is employed. An 80 cm wide bar (2) is affixed by two linear guides (1) and two parallel springs (4) each featuring a stiffness of $k = 0.165\,\mathrm{N/mm}$. The bar's displacement is measured by an inductive displacement transducer (3). The system was simulated using the multibody system dynamics code Exudyn [99] to generate forces through strains. Initially, the morphology of the PARTS system is adapted to establish a continuous contact area with the bar (b). The robot system can generate a directional deformation, which allows it through strains to exert forces on the environment. The generated forces are indicated by the horizontal displacement of the suspended bar (c). The measured displacement, $x = 46.8\,\mathrm{mm}$, corresponds to a force of

$$F = 2kx = 15.44\,\mathrm{N}. \tag{7}$$

A video of the experiment is provided in Supplementary Material Video S9. Note the deformation of the robot's structure (orange markers) from the ideal configuration (blue markers). The pose information of the camera system could be used for error correction to compensate for the effects of gravitational and external forces.

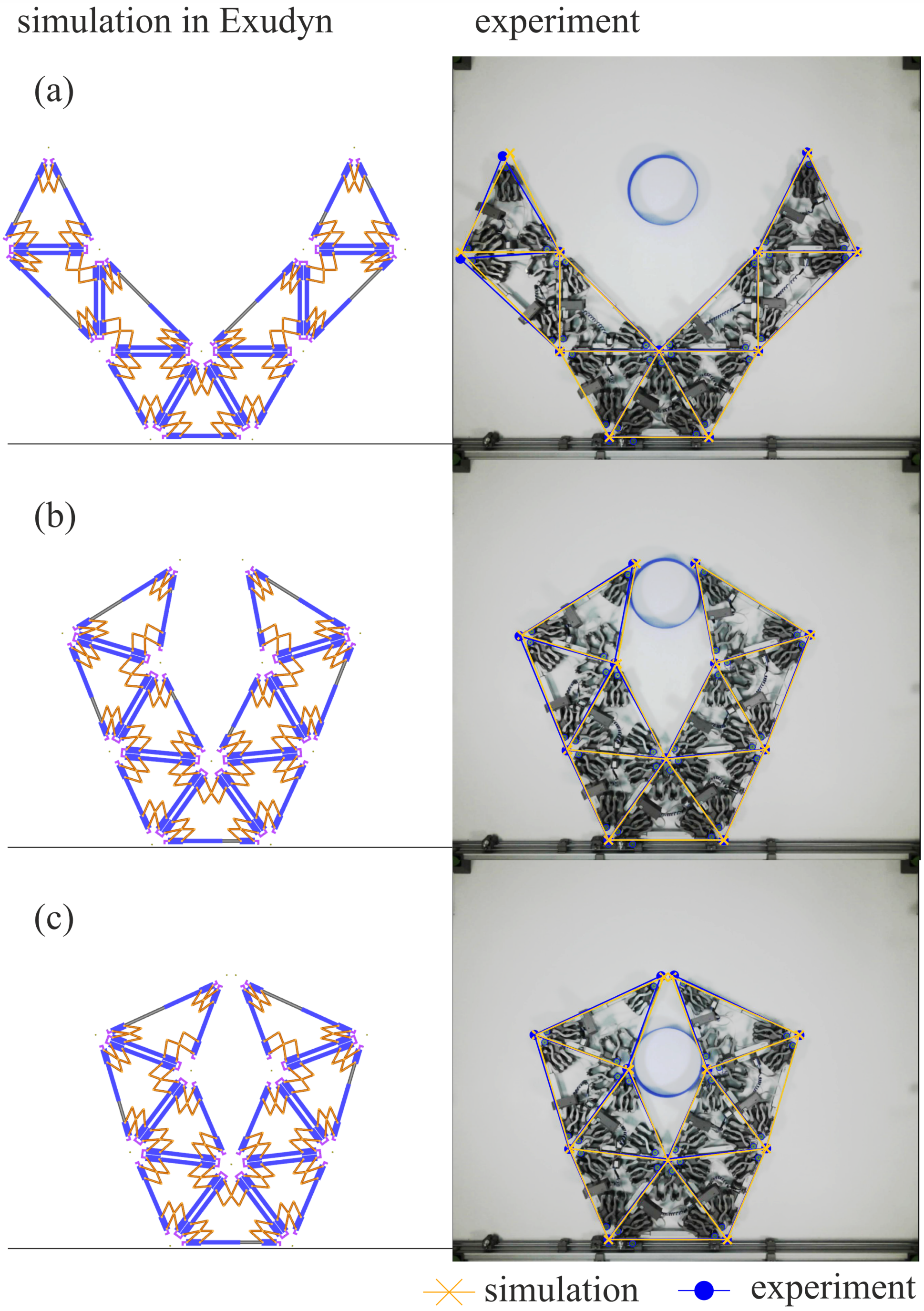

Figure 16. The gripper configuration is composed of a base and two kinematic chains, each consisting of 3 ATCs (**a**). The kinematic chains act as fingers that can close around an object to grasp it (**b**). By enclosing the object, it can be held securely in place and be manipulated (**c**).

The module architecture of PARTS enables the utilization of scaling effects in the system's force output through collective actuation. By coordinately changing their shapes, modules can generate directional strains and accumulate actuator forces to achieve a higher total force output. This functionality can also be used in reverse, such that the PARTS system reacts to external forces from the environment by deforming, resulting in functional compliance of the system. This feature can be applied, for example, to manually adjust the robot's configuration via hand guidance or acting as support with active force damping.

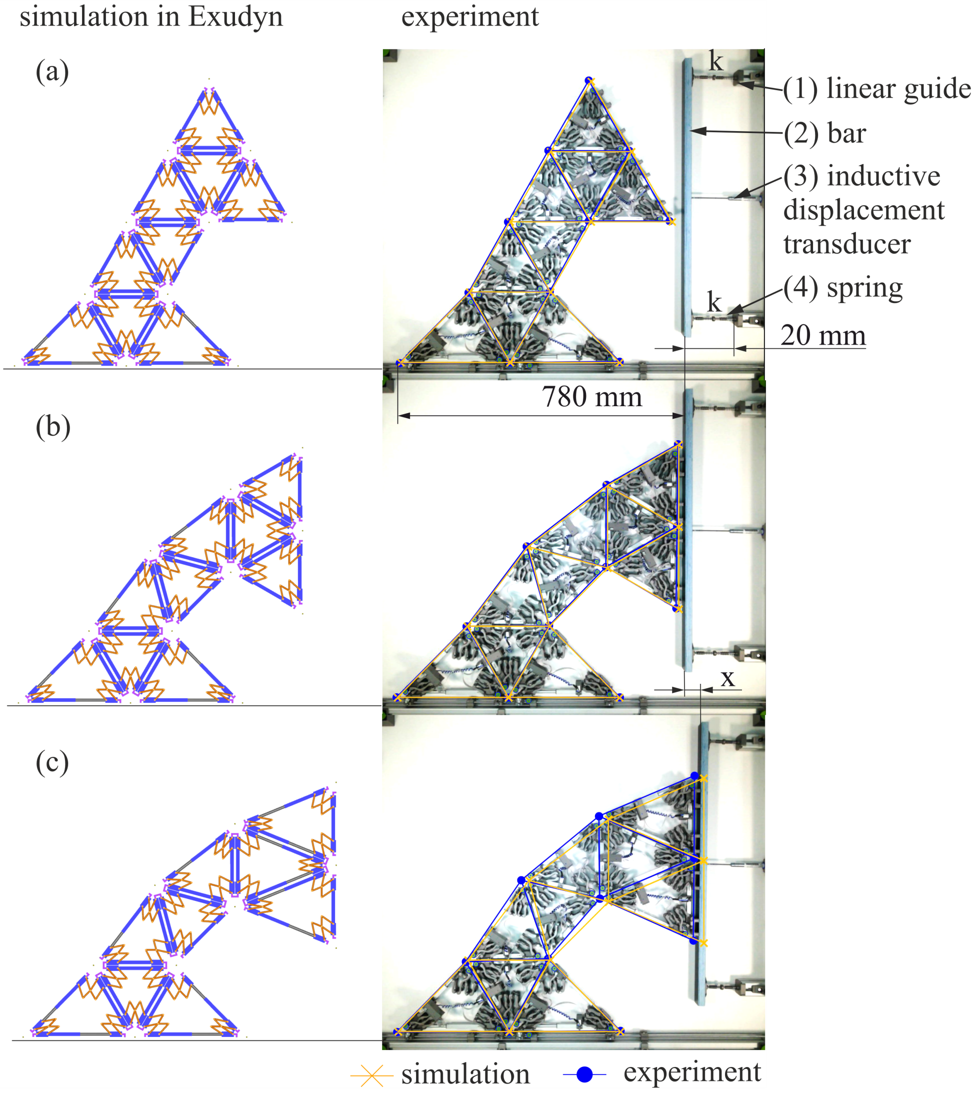

Figure 17. PARTS generating forces on the environment: A PARTS configuration composed of 9 ATCs is capable of supporting a horizontally suspended vertical bar (**a**). The robot adapts its morphology to establish a continuous contact area with the bar (**b**). Through collective actuation, the PARTS system induces directional deformation, exerting strains that generate forces on the environment, ultimately causing a horizontal displacement of the bar (**c**).

6. Conclusions

In this study, we validated the functional capabilities of the PARTS modular robot system and highlighted the versatility of its system architecture. Meta-modules and mesh-restructuring strategies were introduced to manage hardware constraints and realize self-reconfiguration of PARTS. These strategies were demonstrated through a reconfiguration example of a simulated system with 62 ATCs. We showcased the diverse functionality and capabilities of PARTS through various use cases and exemplary configurations using the hardware prototype. Specifically, its abilities in locomotion, object manipulation, and interaction with the environment and other modules was demonstrated.

Coping with these tasks and possessing the ability for self-reconfiguration, the unique system architecture of PARTS combines a set of functionalities that has, to the authors' knowledge, not been realized fully and to this extent by any other modular robot system. PARTS can form space-filling structures with a wide range of shapes and configurations, and the system can change between these shapes through topological self-reconfiguration. Moreover, all configurations of PARTS inherently feature morphological flexibility and compliance without altering their connection topology. These abilities also make it possible for PARTS to apply forces to the environment in a controlled manner, permitting a flexible interaction.

The demonstrated capabilities make the system architecture of PARTS a highly versatile and adaptable modular robot system with great potential. PARTS serves as a stepping stone system in the developmental journey aimed at realizing the system architecture in 3D, ultimately creating a fully functional version of ARTS, an Adaptive Robot with a Tetrahedral Structure. Many authors have employed two-dimensional designs to assess the viability of hardware and algorithms for the successful spatial implementation of their system architecture. Examples include the *FireAnt* [25] and *FireAnt3D* [32] systems, as well as systems like *Crystalline* [100], *Telecube* [26,101], *2D Catoms* [77], and *3D Catoms* [31]. The constraint optimization framework for morphological adaptation as presented in Equation (6) can be extended to the 3D space, as none of the constituting Equations (1)–(3) and (5) depend inherently on the dimension. Supplementary Material Video S10 and Supplementary Material Video S11 show animations of ATC surface motion and the connection of two kinematic chains with ATC transport, generated by a spatial version of the optimization algorithm. The mesh restructuring primitives presented in Section 4.3, utilizing a sequence of opening and closing kinematic loops, may, in a spatially adapted form, also prove expedient in modifying the topology of a 3D system with tetrahedral modules.

We are currently working on a functional hardware realization of an adaptive robot with a tetrahedral structure, the spatial version of PARTS, with the capability to represent tetrahedral meshes. The PARTS system provides valuable insights and the foundation for transferring the module architecture to the third dimension, including topological representation methods, mechanical design approaches, and reconfiguration strategies. Future research will focus on completing the functional prototype of the adaptive robot with a tetrahedral structure and on transferring the methods of PARTS for self-reconfiguration and morphological adaptation to the spatial realm and real-world applications.

Supplementary Materials: The following supplementary material can be downloaded at: https://www.mdpi.com/article/10.3390/robotics13050077/s1, Video S1: Meta-module movement; Video S2: Loop Transfer; Video S3: Loop Integration; Video S4: Reconfiguration Example; Video S5: Rolling Gait; VideoS6: Walking Gait; Video S7: Forming a Bridge; Video S8: Gripper; Video S9: Support Structure; Video S10: ARTS surface motion; Video S11: ARTS module transfer.

Author Contributions: Conceptualization, M.G., M.P. and J.G.; methodology, M.G. and M.P.; software, M.G., M.P. and E.U.; investigation, M.G., M.P. and E.U.; resources, J.G.; writing—original draft preparation, M.G.; writing—review and editing, M.G., M.P. and J.G.; supervision, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: The open access fee was funded by the publication fund of the University of Innsbruck.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jonsson, A.K.; Morris, R.A.; Pedersen, L. Autonomy in Space Exploration: Current Capabilities and Future Challenges. In Proceedings of the 2007 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2007; pp. 1–12. [CrossRef]
2. Post, M.A.; Yan, X.T.; Letier, P. Modularity for the future in space robotics: A review. *Acta Astronaut.* **2021**, *189*, 530–547. [CrossRef]
3. Yim, M.; Shen, W.M.; Salemi, B.; Rus, D.; Moll, M.; Lipson, H.; Klavins, E.; Chirikjian, G.S. Modular self-reconfigurable robot systems [Grand challenges of robotics]. *IEEE Robot. Autom. Mag.* **2007**, *14*, 43–52. [CrossRef]
4. Ahmadzadeh, H.; Masehian, E. Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization. *Artif. Intell.* **2015**, *223*, 27–64. [CrossRef]
5. Ahmadzadeh, H.; Masehian, E.; Asadpour, M. Modular Robotic Systems: Characteristics and Applications. *J. Intell. Robot. Syst. Theory Appl.* **2016**, *81*, 317–357. [CrossRef]

6. Shokri, A.; Masehian, E. A meta-module approach for cluster flow locomotion of modular robots. In Proceedings of the 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), Tehran, Iran, 7–9 October 2015; pp. 425–431. [CrossRef]
7. Yu, C.H.; Nagpal, R. Self-adapting modular robotics: A generalized distributed consensus framework. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 1881–1888. [CrossRef]
8. Bojinov, H.; Casal, A.; Hogg, T. Multiagent control of self-reconfigurable robots. *Artif. Intell.* **2002**, *142*, 99–120. [CrossRef]
9. Kurokawa, H.; Tomita, K.; Kamimura, A.; Kokaji, S.; Hasuo, T.; Murata, S. Distributed self-reconfiguration of M-TRAN III modular robotic system. *Int. J. Robot. Res.* **2008**, *27*, 373–386. [CrossRef]
10. Fang, Y.; Zhang, H.; Li, X.; Chen, S. The Mathematical Model and Control Scheme of a Four-Legged Robot Based on GZ-I and Note Module. In *Intelligent Robotics and Applications*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6424, pp. 300–309. [CrossRef]
11. Shen, W.M.; Salemi, B.; Will, P. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Trans. Robot. Autom.* **2002**, *18*, 700–712. [CrossRef]
12. Tucci, T.; Piranda, B.; Bourgeois, J. A distributed self-assembly planning algorithm for modular robots: Robotics track. In Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, Stockholm, Sweden, 10–15 July 2018; Volume 1, pp. 550–558.
13. Zykov, V.; Mytilinaios, E.; Desnoyer, M.; Lipson, H. Evolved and designed self-reproducing modular robotics. *IEEE Trans. Robot.* **2007**, *23*, 308–319. [CrossRef]
14. Pieber, M.; Neurauter, R.; Gerstmayr, J. An Adaptive Robot for Building In-Plane Programmable Structures. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–9. [CrossRef]
15. Pieber, M.; Gerstmayr, J. Six-Bar Linkages With Compliant Mechanisms for an Adaptive Robot. In Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Infomation in Engineering Conference, Virtual, Online, 1–19 August 2020; pp. 1–9.
16. Gerbl, M.; Gerstmayr, J. Self-reconfiguration of shape-shifting modular robots with triangular structure. *Robot. Auton. Syst.* **2022**, *147*, 103930. [CrossRef]
17. Gerbl, M.; Gerstmayr, J. Self-reconfiguration of PARTS: A parallel reconfiguration algorithm based on surface flow. *Robot. Auton. Syst.* **2023**, *164*, 104417. [CrossRef]
18. Gilpin, K.; Knaian, A.; Rus, D. Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2485–2492. [CrossRef]
19. Rus, D.; Vona, M. Crystalline Robots: Self-Reconfiguration with Compressible Unit Modules. *Auton. Robot.* **2001**, *10*, 107–124. [CrossRef]
20. Sung, C.; Bern, J.; Romanishin, J.; Rus, D. Reconfiguration planning for pivoting cube modular robots. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1933–1940. [CrossRef]
21. Piranda, B.; Bourgeois, J. A Distributed Algorithm for Reconfiguration of Lattice-Based Modular Self-Reconfigurable Robots. In Proceedings of the 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Heraklion, Greece, 17–19 February 2016; pp. 1–9. [CrossRef]
22. Moussa, M.; Piranda, B.; Makhoul, A.; Bourgeois, J. Cluster-Based Distributed Self-Reconfiguration Algorithm for Modular Robots. In *Advanced Information Networking and Applications*; Springer: Cham, Switzerland, 2021; pp. 1–12.
23. Goldstein, S.; Campbell, J.; Mowry, T. Programmable Matter. *IEEE Comput.* **2005**, *38*, 99–101. [CrossRef] [PubMed]
24. Naz, A.; Piranda, B.; Bourgeois, J.; Goldstein, S.C. A distributed self-reconfiguration algorithm for cylindrical lattice-based modular robots. In Proceedings of the 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 31 October–2 November 2016; pp. 254–263. [CrossRef]
25. Swissler, P.; Rubenstein, M. FireAnt: A Modular Robot with Full-Body Continuous Docks. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 6812–6817. [CrossRef]
26. Vassilvitskii, S.; Yim, M.; Suh, J. A complete, local and parallel reconfiguration algorithm for cube style modular robots. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, USA, 11–15 May 2002; Volume 1, pp. 117–122. [CrossRef]
27. An, B.K. EM-Cube: Cube-Shaped, Self-reconfigurable Robots Sliding on Structure Surfaces. In Proceedings of the - IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008, pp. 3149–3155. [CrossRef]
28. Romanishin, J.W.; Gilpin, K.; Rus, D. M-blocks: Momentum-driven, magnetic modular robots. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 4288–4295. [CrossRef]
29. Kawano, H. Distributed tunneling reconfiguration of cubic modular robots without meta-module's disassembling in severe space requirement. *Robot. Auton. Syst.* **2020**, *124*, 103369. [CrossRef]

30. Meng, Y.; Zhang, Y.; Sampath, A.; Jin, Y.; Sendhoff, B. Cross-Ball: A new morphogenetic self-reconfigurable modular robot. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 267–272. [CrossRef]
31. Piranda, B.; Bourgeois, J. *Geometrical Study of a Quasi-Spherical Module for Building Programmable Matter*; Springer International Publishing: Cham, Switzerland, 2018; pp. 387–400. [CrossRef]
32. Swissler, P.; Rubenstein, M. FireAnt3D: A 3D self-climbing robot towards non-latticed robotic self-assembly. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 3340–3347.
33. Liang, G.; Luo, H.; Li, M.; Qian, H.; Lam, T.L. FreeBOT: A Freeform Modular Self-reconfigurable Robot with Arbitrary Connection Point-Design and Implementation. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 6506–6513.[CrossRef]
34. Christensen, D.J.; Stoy, K. Selecting a meta-module to shape-change the ATRON self-reconfigurable robot. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2532–2538. [CrossRef]
35. Christensen, D.; Østergaard, E.; Lund, H. Metamodule control for the ATRON self-reconfigurable robotic system. In Proceedings of the 8th Conference on Intelligent Autonomous Systems, Amsterdam, The Netherlands, 1 January 2004; pp. 685–692.
36. Stoy, K. Using cellular automata and gradients to control self-reconfiguration. *Robot. Auton. Syst.* **2006**, *54*, 135–141. [CrossRef]
37. Yim, M.; Zhang, Y.; Lamping, J.; Mao, E. Distributed control for 3D metamorphosis. *Auton. Robot.* **2001**, *10*, 41–56. [CrossRef]
38. Kawano, H. Tunneling-based self-reconfiguration of heterogeneous sliding cube-shaped modular robots in environments with obstacles. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 825–832. [CrossRef]
39. De Rosa, M.; Goldstein, S.; Peter, L.; Campbell, J.; Pillai, P. Scalable shape sculpting via hole motion: Motion planning in lattice-constrained modular robots. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 1462–1468. [CrossRef]
40. Brandt, D.; Christensen, D.J.; Lund, H.H. ATRON robots: Versatility from self-reconfigurable modules. In Proceedings of the 2007 International Conference on Mechatronics and Automation, Harbin, China, 5–8 August 2007; pp. 26–32. [CrossRef]
41. Bourgeois, J.; Piranda, B.; Naz, A.; Boillot, N.; Mabed, H.; Dhoutaut, D.; Tucci, T.; Lakhlef, H. Programmable matter as a cyber-physical conjugation. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 2942–2947. [CrossRef]
42. Spröwitz, A.; Moeckel, R.; Vespignani, M.; Bonardi, S.; Ijspeert, A.J. Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. *Robot. Auton. Syst.* **2014**, *62*, 1016–1033. [CrossRef]
43. Ryland, G.G.; Cheng, H.H. Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 60–65. [CrossRef]
44. Bie, D.; Wang, Y.; Zhang, Y.; Liu, C.; Zhao, J.; Zhu, Y. Parametric L-systems-based modeling self-reconfiguration of modular robots in obstacle environments. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 172988141875447. [CrossRef]
45. Butler, Z.; Kotay, K.; Rus, D.; Tomita, K. Cellular Automata for Decentralized Control of Self-Reconfigurable Robots. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, USA, 11–15 May 2002; pp. 21–26.
46. Fitch, R.; Butler, Z. Million module march: Scalable locomotion for large self-reconfiguring robots. *Int. J. Robot. Res.* **2008**, *27*, 331–343. [CrossRef]
47. Kawano, H. Complete reconfiguration algorithm for sliding cube-shaped modular robots with only sliding motion primitive. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3276–3283. [CrossRef]
48. Chirikjian, G.; Pamecha, A.; Ebert-Uphoff, I. Evaluating efficiency of self-reconfiguration in a class of modular robots. *J. Field Robot.* **1996**, *13*, 317–338. [CrossRef]
49. Yim, M.; Duff, D.G.; Roufas, K.D. PolyBot: A Modular Reconfigurable Robot. In Proceedings of the IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 24–28 April 2000. [CrossRef]
50. Liu, C.; Whitzer, M.; Yim, M. A Distributed Reconfiguration Planning Algorithm for Modular Robots. *IEEE Robot. Autom. Lett.* **2019**, *4*, 4231–4238. [CrossRef]
51. Wei, H.; Li, H.Y.; Tan, J.; Wang, T. Self-assembly control and experiments in swarm modular robots. *Sci. China Technol. Sci.* **2012**, *55*, 1118–1131. [CrossRef]
52. Shen, W.M.; Krivokon, M.; Chiu, H.; Everist, J.; Rubenstein, M.; Venkatesh, J. Multimode locomotion via SuperBot robots. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2552–2557. [CrossRef]
53. Støy, K.; Shen, W.; Will, P.M. A simple approach to the control of locomotion in self-reconfigurable robots. *Robot. Auton. Syst.* **2003**, *44*, 191–199. [CrossRef]
54. Baca, J.; Hossain, S.; Dasgupta, P.; Nelson, C.A.; Dutta, A. ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration. *Robot. Auton. Syst.* **2014**, *62*, 1002–1015. [CrossRef]

55. Gonzalez-Gomez, J.; Gonzalez-Quijano, J.; Zhang, H.; Abderrahim, M. Toward the sense of touch in snake modular robots for search and rescue operations. In Proceedings of the ICRA 2010 Workshop on Modular Robots: State of the Art, Anchorage, AK, USA, 3–7 May 2010; pp. 63–68.
56. Murata, S.; Yoshida, E.; Kamimura, A.; Kurokawa, H.; Tomita, K.; Kokaji, S. M-TRAN: Self-Reconfigurable Modular. *IEEE/ASME Trans. Mechatronics* **2002**, *7*, 431–441. [CrossRef]
57. Davey, J.; Kwok, N.; Yim, M. Emulating self-reconfigurable robots-design of the SMORES system. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 4464–4469. [CrossRef]
58. Neubert, J.; Lipson, H. Soldercubes: A self-soldering self-reconfiguring modular robot system. *Auton. Robot.* **2016**, *40*, 139–158. [CrossRef]
59. Usevitch, N.S.; Hammond, Z.M.; Schwager, M.; Okamura, A.M.; Hawkes, E.W.; Follmer, S. An untethered isoperimetric soft robot. *Sci. Robot.* **2020**, *5*, eaaz0492. [CrossRef] [PubMed]
60. Curtis, S.; Brandt, M.; Bowers, G.; Brown, G.; Cheung, C.; Cooperider, C.; Desch, M.; Desch, N.; Dorband, J.; Gregory, K.; et al. Tetrahedral robotics for space exploration. *IEEE Aerosp. Electron. Syst. Mag.* **2007**, *22*, 22–30. [CrossRef]
61. Hamlin, G.J.; Sanderson, A.C. TETROBOT modular robotics: Prototype and experiments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96, Osaka, Japan, 8 November 1996; Volume 2, pp. 390–395. [CrossRef]
62. Izadi, M.; Mahjoob, M.J.; Soheilypour, M. Walking Gait of a Single-Tetrahedral Robot: Design, Modeling and Implementation. In Proceedings of the ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, Istanbul, Turkey, 12–14 July 2010; Volume 3, pp. 627–631. [CrossRef]
63. Motahari-Bidgoli, S.M.; Mahjoob, M.J.; Davaria, S. Simulation and analysis of a TET-walker robot motion. In Proceedings of the 2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014, Tehran, Iran, 15–17 October 2014; pp. 914–919. [CrossRef]
64. Belisle, R.; Yu, C.h.; Nagpal, R. Mechanical Design and Locomotion of Modular Expanding Robots. In Proceedings of the ICRA-2010 Workshop on Modular Robots: State of The Art, Anchorage, AK, USA, 3 May 2010.
65. Qin, Y.; Ting, L.; Saven, C.; Amemiya, Y.; Tanis, M.; Kamien, R.D.; Sung, C. TrussBot: Modeling, Design, and Control of a Compliant, Helical Truss of Tetrahedral Modules. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4218–4224.
66. Spinos, A.; Carroll, D.; Kientz, T.; Yim, M. Variable topology truss: Design and analysis. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2717–2722. [CrossRef]
67. Jeong, S.; Kim, B.; Park, S.; Park, E.; Spinos, A.; Carroll, D.; Tsabedze, T.; Weng, Y.; Seo, T.W.; Yim, M.; et al. Variable Topology Truss: Hardware Overview, Reconfiguration Planning and Locomotion. In Proceedings of the 2018 15th International Conference on Ubiquitous Robots, Honolulu, HI, USA, 26–30 June 2018; pp. 610–615. [CrossRef]
68. Bae, J.; Park, S.; Yim, M.; Seo, T.W. Polygon-Based Random Tree Search Algorithm for a Size-Changing Robot. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8100–8105. [CrossRef]
69. Bae, J.; Park, I.; Yim, M.; Seo, T. Locomotion Planning of a Truss Robot on Irregular Terrain. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 824–829. [CrossRef]
70. Liu, C.; Yim, M. Reconfiguration Motion Planning for Variable Topology Truss. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Macau, China, 3–8 November 2019; pp. 1941–1948. [CrossRef]
71. Spinos, A.; Yim, M. Collision-Free Reconfiguration Planning for Variable Topology Trusses Using a Linking Invariant. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 2210–2215. [CrossRef]
72. Liu, C.; Yu, S.; Yim, M. Motion Planning for Variable Topology Trusses: Reconfiguration and Locomotion. *IEEE Trans. Robot.* **2023**, *39*, 2020–2039. [CrossRef]
73. Spinos, A.; Carroll, D.; Kientz, T.; Yim, M. Topological reconfiguration planning for a variable topology truss. *J. Mech. Robot.* **2021**, *13*, 040902. [CrossRef]
74. Collins, F.; Yim, M. Design of a spherical robot arm with the Spiral Zipper prismatic joint. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2137–2143. [CrossRef]
75. Li, S.; Batra, R.; Brown, D.; Chang, H.D.; Ranganathan, N.; Hoberman, C.; Rus, D.; Lipson, H. Particle robotics based on statistical mechanics of loosely coupled components. *Nature* **2019**, *567*, 361–365. [CrossRef] [PubMed]
76. Liang, G.; Lam, T.L.; Tu, Y. FreeSN: A Freeform Strut-node Structured Modular Self-reconfigurable Robot-Design and Implementation. In Proceedings of the IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, 23–27 May 2022; pp. 4239–4245.
77. Kirby, B.T.; Aksak, B.; Campbell, J.D.; Hoburg, J.F.; Mowry, T.C.; Pillai, P.; Goldstein, S.C. A modular robotic system using magnetic force effectors. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 2787–2793. [CrossRef]
78. Shimizu, M.; Ishiguro, A.; Kawakatsu, T. A modular robot that exploits a spontaneous connectivity control mechanism. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 1899–1904. [CrossRef]

79. Campbell, J.; Pillai, P. Collective actuation. *Int. J. Robot. Res.* **2008**, *27*, 299–314. [CrossRef]
80. Belke, C.H.; Paik, J. Mori: A Modular Origami Robot. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 2153–2164. [CrossRef]
81. Belke, C.H.; Holdcroft, K.; Sigrist, A.; Paik, J. Morphological flexibility in robotic systems through physical polygon meshing. *Nat. Mach. Intell.* **2023**, *5*, 669–675. [CrossRef]
82. Actuonix Motion Devices Inc. Datasheet Miniature Linear Motion Series L12. Available online: https://www.actuonix.com/assets/images/datasheets/ActuonixL12Datasheet.pdf (accessed on 8 May 2024).
83. Gerstmayr, J.; Pieber, M. An Adaptive Robot with Tetrahedral Cells. In Proceedings of the The 4th Joint International Conference on Multibody System Dynamics, Montreal, QC, Canada, 29 May–2 June 2016; Volume 5, pp. 1–15.
84. Hamlin, G.J.; Sanderson, A.C. A Novel Concentric Multilink Spherical Joint with Parallel Robotics Applications. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; pp. 1267–1272.
85. Yim, M.; Zhang, Y.; Duff, D. Modular robots. *IEEE Spectr.* **2002**, *39*, 30–34. [CrossRef]
86. Pamecha, A.; Chiang, C.J.; Stein, D.; Chirikjian, G. Design And Implementation Of Metamorphic Robots. In Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Infomation in Engineering Conference, Irvine, CA, USA, 18–22 August 1996; Volume 10, pp. 1–10.
87. Stein, P. A Note on the Volume of a Simplex. *Am. Math. Mon.* **1966**, *73*, 299–301. [CrossRef]
88. Kubica, J.; Casal, A.; Hogg, T. Complex Behaviors from Local Rules in Modular Self-reconfigurable Robots. In Proceedings of the 2001 IEEE International Conference on Robotics and Automation, Seoul, Republic of Korea, 21–26 May 2001; pp. 360–367.
89. Kamimura, A.; Kurokawa, H.; Yoshida, E.; Tomita, K.; Kokaji, S.; Murata, S. Distributed adaptive locomotion by a modular robotic system, M-TRAN II. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2370–2377. [CrossRef]
90. Sproewitz, A.; Moeckel, R.; Maye, J.; Asadpour, M.; Ijspeert, A. Adaptive locomotion control in modular robotics. In Proceedings of the Workshop on Self-Reconfigurable Robots/Systems and Applications IROS07, San Diego, USA, 29 October–2 November 2007; pp. 81–84.
91. Moreno, R.; Gomez, J. Central pattern generators and hormone inspired messages: A hybrid control strategy to implement motor primitives on chain type modular reconfigurable robots. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1014–1019. [CrossRef]
92. Pouya, S.; Aydin, E.; Möckel, R.; Ijspeert, A.J. Locomotion Gait Optimization For Modular Robots; Coevolving Morphology and Control. *Procedia Comput. Sci.* **2011**, *7*, 320–322. [CrossRef]
93. Abrahantes, M.; Silver, A.; Wendt, L. Gait design and modeling of a 12-tetrahedron walker robot. In Proceedings of the 2007 Thirty-Ninth Southeastern Symposium on System Theory, Macon, GA, USA, 4–6 March 2007; pp. 21–25. [CrossRef]
94. Wei, H.; Cai, Y.; Li, H.; Li, D.; Wang, T. Sambot: A self-assembly modular robot for swarm robot. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 66–71. [CrossRef]
95. Murata, S.; Kakomura, K.; Kurokawa, H. Toward a scalable modular robotic system. *IEEE Robot. Autom. Mag.* **2007**, *14*, 56–63. [CrossRef]
96. Rubenstein, M.; Payne, K.; Will, P.; Shen, W.M. Docking among independent and autonomous CONRO self-reconfigurable robots. In Proceedings of the International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; pp. 2877–2882. [CrossRef]
97. Zhu, Y.; Jin, H.; Zhang, X.; Yin, J.; Liu, P.; Zhao, J. A multi-sensory autonomous docking approach for a self-reconfigurable robot without mechanical guidance. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 1–10. [CrossRef]
98. Prattichizzo, D.; Trinkle, J.C., Grasping. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 671–700. [CrossRef]
99. Gerstmayr, J. Exudyn—A C++ based Python package for flexible multibody systems. *Preprint* **2023**. [CrossRef]
100. Rus, D.; Vona, M. Self-reconfiguration planning with compressible unit modules. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 10–15 May 1999; Volume 4, pp. 2513–2520.[CrossRef]
101. Suh, J.W.; Homans, S.B.; Yim, M. Telecubes: Mechanical design of a module for self-reconfigurable robotics. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, USA, 11–15 May 2002; Volume 4, pp. 4095–4101. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.