Article type: Original Article

Corresponding Author:

Jamie Paik, Institute of Mechanical Engineering, EPFL STI IGM RRL, MED 11326, Station 9, CH-1015 Lausanne, Switzerland.

E-mail: jamie.paik@epfl.ch

A Reconfigurable Interactive Interface to Control Robotic Origami in Virtual Environments

Jian-Lin Huang, Zhenishbek Zhakypov, Harshal Sonar, and Jamie Paik

Reconfigurable Robotics Lab, Institute of Mechanical Engineering, EPFL, Lausanne, Switzerland

Abstract

Origami shape transformation is dictated by predefined folding patterns and their folding sequence. The working principle of robotic origami is based on the same principle: we design quasi-2D tiles and connecting hinges and define and program their folding sequences. Since the tiles are often of uniform shape and size, their final configuration is governed by the kinematic relationship. Mathematicians, computer scientists and even architects have studied a wide range of origami algorithms. However, for multiple shape transformations, the origami design parameters and consequently sequence planning become more challenging. In this work, we present a reconfigurable interactive interface, a physics-based modeling control interface to explore the design space of origami robots. We developed two interactive modes for proof of concept of a bidirectional communication interface between virtual and physical environments. The first interaction mode is origami-inspired, foldable surfaces with distributed sensors that can recreate folding sequences and shape transformations in a virtual environment via hardware-in-loop simulation. Its complementary digital transcription lays the foundation for a robotic origami design tool that provides visual representation of various design formulations as well as an intuitive controller for robotic origami. In the second interaction mode, we construct a physics-based modeling interface for intuitive user manipulation of robotic origami in a virtual environment. Algorithms for graphical representation and command transformation were developed for robotic interaction. Lastly, we tested the efficacy of the algorithms on prototypes to discover the applications and capacities of the reconfigurable interactive interface.

Keywords

Robotic origami control, rigid body virtual environment, interaction design, shape-changing interface,

1. Introduction

Origami is the art of paper folding that has also inspired dozens of engineering applications. Origami robots, robotic origami, can self-transform into desired three-dimensional (3D) shapes from nominally flat structures without manipulation by external torques or forces (Paik et al., 2012, Miyashita et al., 2016, Onal et al., 2013). This technology has appropriated the foldable and reconfigurable structures, to adapt to undefined environments or tasks. Several self-folding devices have been developed that employ shape memory alloy (SMA) or shape memory polymer (SMP) actuators (Peraza-Hernandez et al., 2014, Felton et al., 2014, Hernandez et al., 2016, Roudaut et al., 2013). Besides the hardware, planning and control algorithms have been proposed to generate automatic design processes to fold origamis into a desired shape (Byoungkwon An, 2014). However, sensing feedback has rarely been considered in the presented designs (Nisser et al., 2016) which is crucial to achieve more complex shapes with many folding steps; it is beneficial to know and control each folding step towards more complicated design**.**

The design, programming and control of robots require diverse fields of knowledge. To reduce the obstacles in development and control of robots, many approaches have been investigated. For instance, to simplify robot programming and manipulation, high-level or task-oriented programming methods (Pan et al., 2012, Billard et al., 2008) provide intuitive and comprehensive programming interfaces. Recently, the interaction interfaces between robots and humans, based on virtual reality and gesture-based control provide natural ways to program or telemanipulate robots (Kelly et al., 2011, Alonso-Mora et al., 2015). Robotic origami can serve as a platform to investigate synergy between virtual and physical reality. Due to its unique structure, kinematics and corresponding control strategies, we can gain valuable insights into different interactive methodologies via the physical interface of origami robots.

To construct a virtual model the critical step is to understand the kinematics of origami. This has been extensively studied by various research groups in mathematics, computer sciences and even architecture (Hull, 2002, Demaine and O'Rourke, 2007). To classify the different origami patterns, several works exist on kinematic analysis or representation by equivalent mechanisms and spherical mechanism is commonly used to differentiate the particular classes of origami kinematics (Bowen et al., 2014, Bowen et al., 2013). Rigid origami models regard the crease as a compliant hinge and the facet as a rigid panel that can effectively model the trajectories during folding (Greenberg et al., 2011, Tachi, 2009). In origami structure, numerical analysis and finite element simulation involving active components have been proposed by several groups (An et al., 2016, Peraza-Hernandez et al., 2013). However, finite element simulation does not allow rapid responses to dynamic input. While considerable attention has been paid to kinematic models and computational simulation, there have been also some attempts to experimentally validate models using physical devices (Hanna et al., 2014, Qiu et al., 2013, Sung and Rus, 2015, Dai and Cannella, 2008, Bowen et

al., 2016). While these studies are significant in their modeling for robotic origamis application, they lack sensing and actuation components. To incorporate a control feedback loop, we need a more extensive origami platform that includes additional parameters such as such as weight, structure thickness, torque of actuators and hinge stiffness. Progressive understanding of the realistic and physical model-based mechanics of robotic origami folding is needed to design robots from the concepts of origami. Therefore, we need a design tool that considers both the robotic hardware components as well as the origami folding mechanisms.

As well as simplifying and understanding the kinematics of foldable structures, the distributed sensing approach on deformable devices facilitates self-sensing, which could be employed as multiple DoF input devices for virtual model construction and could be compared with kinematic model or computational simulation. This type of 3D model acquisition and the input devices could have a wide range of applications, such as reverse engineering, virtual and augmented reality and human–computer interaction. Optical sensors, such as laser scanners or cameras are common methods for reconstructing 3D models. (Miller et al., 2012). Several groups have investigated 3D construction of physical structures by distributing sensors on the object itself so that the user can manipulate the object as a tangible input device for constructing the 3D model (Watanabe et al., 2004, Nakagaki et al., 2015, Huang and Eisenberg, 2012). However, due to the size of the device and the limited number of physical pieces, it inevitably has a restricted realizable DoF (below 10). Thus, distributed miniaturized sensor networks over a deformable surface for high DoF input devices were developed (Rendl et al., 2014, Chien et al., 2015, Dementyev et al., 2015). Although these devices provide the interface for high DoF, even up to 50, input at a manipulable scale, these kinds of distributed sensing systems lack a self-actuation mechanism. Recently, actuated tangible interfaces have begun to emerge and these new types of tangible interfaces provide not only one way communication but also enhance the interaction experience with digital information, making use of shape change to embody digital information in ways such as changing orientation, texture, volume, or form (Shaer and Hornecker, 2010, Rasmussen et al., 2012, Nakagaki et al., 2016a). A number of studies were conducted in the context of educational toys (Raffle et al., 2004), spatial augmented reality (Nakagaki et al., 2016b, Lindlbauer et al., 2016), tangible remote collaboration (Leithinger et al., 2014) and even conveying emotions (Strohmeier et al., 2016). Many of these devices however, have limited DoF in 3D space and quite often non-portable tabletop tangible user interfaces, especially in virtual reality and augmented reality applications. For these types of actuated tangible interfaces, Robotic origami could have advantages, namely: (1) multiple DoF actuations in 3D space, (2) light weight portable structures, and (3) are reprogrammable and reconfigurable. Hence, in this paper, we have developed an interactive interface to bridge robotic origami and human communication in both digital and physical environments.

The main contributions of this paper are:

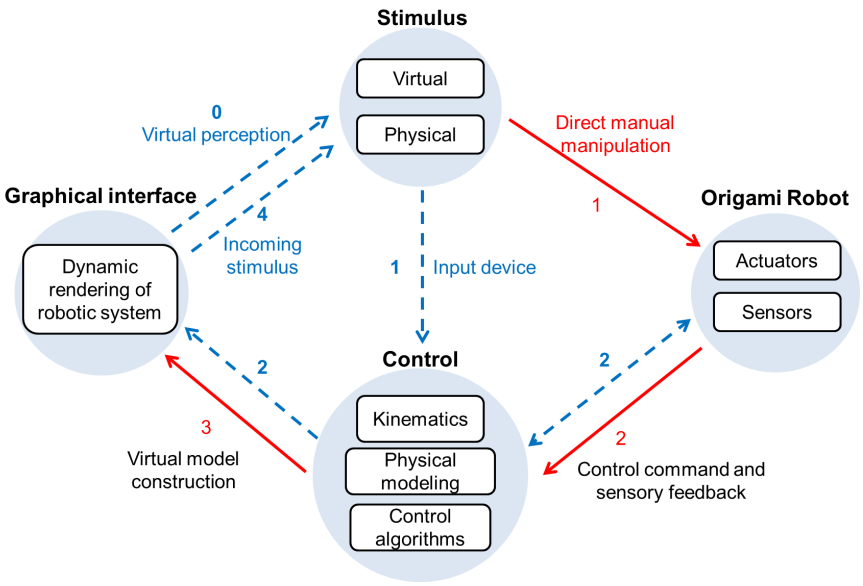


Fig. 1. The schematic of origami robot based interactive system. This behavior-based robotic interaction system can be expressed by three states. With an origami robotic platform the stimulus can be virtual and physical. The physics-based modeling forms the behavior of the robotic system that governs the responses of the robot in both the physical and graphical simulation worlds. The solid arrows show the process of interaction mode 1, and the dashed arrows show the process of interaction mode 2.

• A novel reconfigurable interactive interface to interact with robotic origami in a virtual environment. This interface provides intuitive control methodologies that can adapt to different hardware designs.
• Development of algorithms for integrating mechanical characteristics of sensors, actuators, and joints for robotic origami into an existing physics engine for the physics-based simulation.
• Development of an algorithm for rapid 3D model reconstruction by solving kinematics of origami structures embedding distributed sensors. The proposed algorithm is implementable to interactive control interfaces for visual feedback and extendable to higher DoF systems.
• Demonstration of feasibility of the interfaces and algorithms on robotic origami platforms and the integration of hardware and software to validate simulation, involving actuation and sensing components within the origami structure.

2. Concept and framework for interacting with robots in a virtual environment

For control and manipulation of high-DoF robotic systems, the kinematics and dynamic behavior of the robots should be well studied and modeled. Inspired by the concept of virtual reality-based interaction interfaces and hardware-in-loop simulation, which helped developers to reduce time, effort and cost for designing and controlling robots (Martin and Emami, 2006), we propose a

framework that enables users to interact with robots in both physical and digital domains. For an interactive system the interaction may be expressed by three stages: (1) stimulus processing, (2) behavior modeling, and (3) response generation (Chen et al., 2010). This framework consists mainly of an interactive interface involving hardware characteristics visualization and a physics-based modeling for bridging the stimulus and response, as shown in Figure 1. The physics-based modeling governs the behavior of the robotic system and can be divided into different levels of modeling: (1) the geometry of the robot body that achieves the desired functionality, (2) the mechanism employed for the geometry, which describes the relation between the robot's body, (3) the critical components of the robotic system, such as actuators and sensors, and (4) the control method implemented in the robotic system. As discussed in the Introduction, most existing origami software only focuses on the first two levels; modeling the kinematics and dynamics of origami structure.

Here, we introduce hardware-in-loop simulation into the origami modeling that involves feedback from the characteristics of physical components and the control system of the robot. The algorithms for different levels of visualization of physical objects and for the conversion of the virtual information to command the robots to generate response are also proposed. The response can be represented visually by the graphical interface or in the robotic system. To illustrate the bidirectional communication between the robot and humans, we developed two modes of interaction with the novel robotic origami system, shown by the solid (red, mode 1) and dashed (blue, mode 2) arrows in Figure 1. Below we describe the two interaction modes that show the possibility of interacting with the robotic system in physical (mode 1) and visual (mode 2) platforms.

The ultimate goal of the proposed concept and framework is to control any kind of robotic origami or even expand it to modular robots. Although various software and hardware tools exist for origami design, there are only fundamental structural and kinematic assumptions that limit their usability in dynamic robotic origamis. The principal contribution of this work is creating interactive control methodologies that consider the characteristics of physical components the origami robot such as sensors and actuators. We decipher the physics of two distinct materials of origami robots that make up the rigid tiles and compliant joints. This is important because the tiles should be flat and non-deformable during the folding motion and the joints should only allow single-axis rotation along the creases. The detailed capabilities and limitations of the proposed methodology are discussed in Section 4.6.

2.1. Interaction mode 1: the origami-based tangible interface for 3D model reconstruction

In interaction mode 1 we create physical interaction methodology that allows the user to manually operate digital information. The solid red arrows in Figure 1 show the interaction mode 1 process. The stimulus (stage 1) is given physically to the origami robot, and the

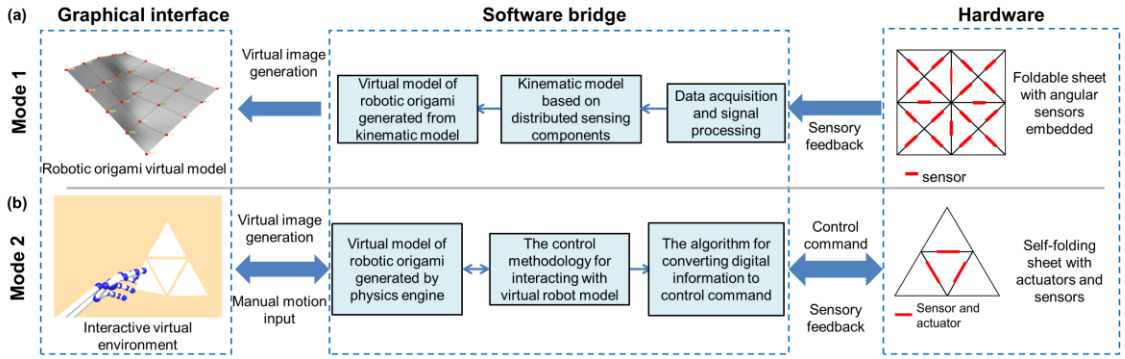


Fig. 2. System overview of the two interaction modes. Interaction mode 1 is an origami based tangible interface (a). This tangible interface collects the onboard sensory data to reconstruct a virtual 3D model of origami structure. Interaction mode 2 shows the interactive interface for controlling robotic origami in the virtual environment (b). The software bridges the graphical control interface built in a virtual environment between human and robotic systems.

response (stage 3) is generated visually by the graphical interface through the control system (stage 2). To create interaction mode 1, understanding the kinematics and construction of a virtual model for the robotic origami are required. We developed an origami-based tangible interface, which is a combination of hardware and software components as shown in Figure 2(a). The system involves not only 3D model reconstruction of the robotic origami, but also their design and control. The prototype is a low-profile, multi-DoF self-sensing device where the software runs sensor data processing, the algorithm for sensor localization, calculation of the kinematics and 3D model construction, as shown in Figure 2(a). The implementation of the software is discussed in Section 3.1. The process following the arrows in Figure 2 illustrates the workflow of this tangible input device consisting of compliant hinges and rigid tiles with embedded sensors on the hinges. The distributed sensors on the foldable sheet provide the bending angles data at each hinge and each sensor reading is collected to construct the kinematic model. Then, the software computes the relative position of each tile based on the kinematic model of the origami. Finally, the software generates a 3D image that is displayed on the graphical user interface.

Theoretically an origami has an infinite number of DoF but in practical applications we aim to have a finite number of DoF with sensors to detect all the actuated folds, rather than having arbitrary creases. Our prototype has rigid triangular tiles interconnected by compliant hinges with sensors. While this design is highly feasible and simple to fabricate, it is capable of being folded into a variety of 3D shapes. The crease pattern of the foldable sheet is shown in Figure 2(a). It has a square shape and is divided into 16 equally-sized isosceles triangles. This crease pattern design is inspired by (An and Rus, 2014, Paik et al., 2012, Demaine and O'Rourke, 2007, Hanna et al., 2014, Balkcom and Mason, 2008, Tachi, 2009). Each triangle is a basic module for constructing the 3D model. The triangular tiles are regarded

as rigid bodies that cannot be folded further. A flexible joint connects two adjacent tiles and enables bi-directional folding along the joint. To get a dihedral angle between two adjacent tiles, we distribute 20 bending sensors along each joint and by measuring the bending angle, the relative positon and orientation of adjacent tiles are calculated by the software. This tangible interface is not only a methodology for interacting with digital information, but also a kinematic modeling tool of robotic origami involving feedback hardware components.

2.2. Interaction mode 2: Intuitive interface for controlling origami robots in a virtual environment

The design and control of robotic origami can be challenging due to their multiple DoF and the complex kinematics. Moreover, the users or developers may not have adequate knowledge of kinematics of origami structure. Hence, in interaction mode 2, an interface was developed that provides a more intuitive way to command as well as simulate the kinematics of robotic origami, as shown by the dashed arrows in Figure 1. A graphical interface provides virtual perception for the user to move and manipulate the virtual robot, meaning the stimulus given virtually to the control system. The responses are generated in two ways. First, the visual responses are generated by physics-based simulation on the graphical interface, shown by the two-way arrow between graphical interface and the Software bridge in Fig. 2(b). Second, the control algorithms transfer the virtual stimulus to the control commands to control motions of the physical robot and bring onboard sensory feedback for closed-loop control and virtual model construction of robotic origami, shown by the two-way arrow between the Software bridge and hardware in Fig. 2(b). In interaction mode 2, we created an interactive virtual environment on the graphical interface to control a physical robotic origami, as shown in Figure 2(b).

The virtual environment is constructed from the 3D model of the robotic origami, the 3D model of the hands and the interaction mechanisms between the hands and robotic origami. This virtual environment enables users to interact with the robotic origami in a similar way to folding origami directly. The software is independent of the hardware. The current virtual environment setup is built on *Unity*. *Unity* is a game development platform that also provides built-in support for VR development. *Leap motion(Leap Motion, Inc.)*, a hand motion-tracking sensor works as the input device for generating the corresponding virtual 3D model of the hands on the graphical interface. To fold an origami structure, the position of the crease pattern and movement direction is required. With this virtual model of hands, users can transfer their hand motions into the virtual world including positions and gestures, that is to say, a pathway to manipulate virtual objects precisely and intuitively. For physics simulation of interaction mode 2, *Nvidia*'s *PhysX* physics engine built in the *Unity*

platform has several key features that are suitable for simulation of the interaction between the robot and the human, such as rigid body kinematics and collision detection. Furthermore, using the kinematic model in mode 1, the graphical interface is not only used to input control command to robotic origami, but also to get information from the robotic origami as visual feedback for interaction.

There are several challenges to building a virtual environment with reasonable interaction for users. The first is to create the virtual model of the robotic origami for simulating kinematics of the robotic origami. Second, the interactions between the virtual robot and virtual hands may be slightly different from those of the physical robotic origami manual folding, but still need to be intuitive to the user. That is, most of the movements and reactions in the virtual environment should represent physics as close as possible; some physics however can be modified to make the manipulation easier and more fluent in the virtual environment. These changes should not increase the learning barriers and should be understood immediately, even without extra instructions. Third, the reactions of the virtual world should be in real-time, or the user will tend to stop or keep moving until they see the desired motion finished. We addressed these three challenges using the following methodologies: (1) the virtual robotic origami model was generated based on rigid origami theory with a foldable structure of rigid flat facets connected by revolute hinges. We can calculate the dihedral angle between the tiles for the robot's control system use. (2) Rigid body modeling and collision detection between the virtual robot and the virtual hands enables users to fold and move the virtual origami tiles without penetrating the origami object. Moreover, the friction simulation of objects allows the user to manipulate or grasp the tiles of the virtual robotic origami more easily. Therefore the user can manipulate and understand the mechanics in a very intuitive way. (3) The infinite air drag force acting on the robotic origami opposes the motion direction and can stop the movement of the robotic origami immediately after the end of collision between the hands and tiles. This function enables the user to fold robotic origami to the desired shape and to operate more easily in virtual environments.

Then comes the challenge of transformation of the visual information to command the physical robot. The virtual presentation needs to consider not only the kinematics of origami structure but also the characteristics of physical components on the robots such as actuators and sensors. The detailed algorithms will be discussed in Section 3.2.

To demonstrate the capability of the interactive control interface, we built an actively, reconfigurable, self-sensing, and self-folding prototype. This prototype consists of four rigid equilateral triangular tiles with sensors and actuators attached at the edges of the triangles as shown in Figure 2(b). This design allows bending along the edges of the triangular tiles, in other words all the elements connected between two tiles are foldable. The folding is done by torsional actuators that generate motion about the central axes along the edges of the tiles.

3. Robotic origami platform: design, configuration and control

In this Section, we discuss the key elements of kinematic models of robotic origami and interactive control methodology based on a virtual environment for constructing interaction modes 1 and 2. We also discuss how to implement these interactive systems in robotic origami platforms.

3.1. Kinematic model of the robotic origami based on a distributed sensing system

In interaction mode 1, we evaluate the position of each tile on the surface and adapt the origami folding sequence using the data from angular sensors for dynamic modeling of the robotic origami on a graphical interface. Therefore, the bending angles on the creases become the input of the kinematic model. We employ the rigid origami model (Tachi, 2010, Tachi, 2009) to present equivalent mechanism of robotic origami. The rigid origami model treats planes of the origami as rigid links and their creases as revolute joints. The crease pattern and the equivalent mechanism of robotic origami are illustrated in Figure 3, where the creases are represented as dashed lines (see Figure 3 (a)) and the solid link lines in Figure 3(b) are equivalent to the triangular facets in Figure 3(a). For instance, link *A* in Figure 3(b) represents the triangular facet *A* in Figure 3(a). According to the terminology used in origami, the point where bending crease converge is called the vertex, i.e. point 1 in Figure 3(a). The vertex in origami is also the center of spherical mechanism. Spherical mechanism is a mechanical system in which any point in a moving body is constrained on concentric spherical surfaces. This mechanism is helpful for understanding the kinematics of origami. In Figure 3(b), the rectangles

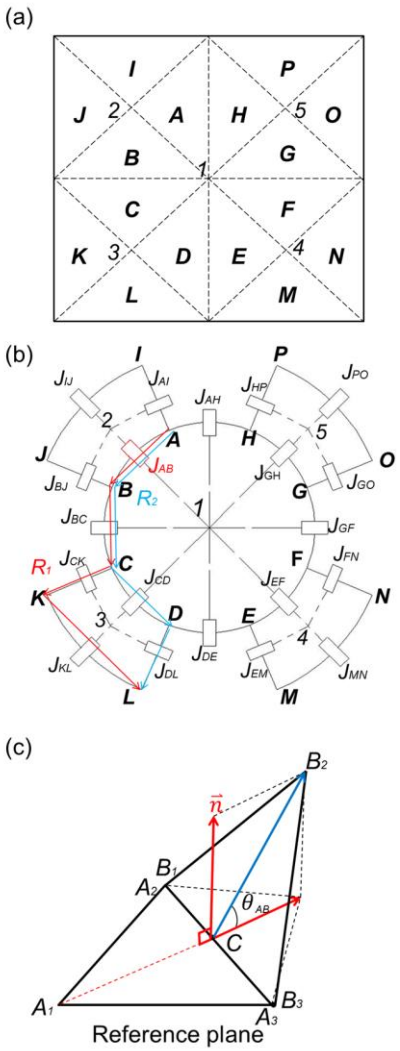


Fig. 3. Kinematics of the origami tangible interface: The crease pattern (a), the equivalent spherical mechanism (b), and the schematics for coordinate calculation of two adjacent tiles (c).

are the revolute joints rotating along the folding creases, represented as dashed lines. For example, rectangle J_{AB} is the joint between link *A* and link *B*. J_{AB} rotates along the axis passing through point 1 and point 2. The trajectories of points on links lie on concentric spherical surfaces and all the axes of revolute joints pass through the vertex. The intersections of the dashed lines are the vertices, such as point 1, 2, 3, 4, and 5. The vertex 1 is the spherical center of link *A* to link *H*, which means the trajectories of points on link *A* to link *H* will lie on concentric sphere surfaces. However, the kinematics of origami structure within multiple spherical mechanisms is not easy to solve. Since we have the rotation angles of each joint from the angular sensor data, we can simplify the analysis of multiple spherical kinematics.

Since an equivalent mechanism of the reconfigurable sheet has been found, we can coordinate each tile based on this mechanism using the revolute angle of joint. Thus, we can include sensor data into the kinematic model. The measured data taken from angular sensors represents the dihedral angle of two tiles, which is also equal to the revolute angle of joint in the spherical equivalent joint. To calculate the relative position of tiles, we set point 1 as origin of the Cartesian coordinate system. Then, given the position vectors of each of the vertices on the triangular tiles as $V = [v_A, v_B, \dots, v_P]$, and v_A ,v_B,…,v_P are 3 x 3 matrices. v_A can be shown as $v_A = [v_{A1}, v_{A2}, v_{A3}]$, where v_{A1}, v_{A2}, and v_{A3} are the position vectors of three vertices of the triangle respectively.

To find the position of all the tiles we calculate the relative position between two adjacent tiles as shown in Figure 3(c). In the beginning, link *A* is chosen for the reference plane. Then, we compute the v_B from v_A by (1)

$$v_B = f(v_A, \theta_{AB}) \tag{1}$$

θ_{AB} is the revolute angle of J_{AB}, and also the dihedral angle between link *A* and link *B*. Function *f* was derived from vector operations. Where $v_{B1} = v_{A2}$, and $v_{A3} = v_{B3}$. The coordinates of v_{B2} can be expressed in terms of cross and inner products of v_A and θ_{AB}:

In Figure 3(c), if *C* is the midpoint of A_2 and A_3, then the position vector of point *C* is written as $C = \frac{v_{A2}+v_{A3}}{2}$. Since the plane defined by point A_2, B_2, and *C* is perpendicular to $\overrightarrow{A_2A_3}$, we can divide the vector $\overrightarrow{CB_2}$ into two components. One is the projection of $\overrightarrow{CB_2}$ onto $\vec{n}$, and another is the projection of $\overrightarrow{CB_2}$ onto $\overrightarrow{A_1C}$. Where $\vec{n}$ is the normal vector of link *A*, and can be calculated as

$$\vec{n} = (v_{A3} - v_{A1}) \times (v_{A2} - v_{A1}) \tag{2}$$

We can get $\overrightarrow{CB_2}$ from the addition of two components has mentioned before by

$$\overrightarrow{CB_2} = \left|\overrightarrow{CB_2}\right| \cos\theta_{AB} \frac{\overrightarrow{A_1C}}{\left|\overrightarrow{A_1C}\right|} + \left|\overrightarrow{CB_2}\right| \sin\theta \frac{\vec{n}}{\left|\vec{n}\right|} \tag{3}$$

Finally, we can find

$$v_{B2} = \begin{bmatrix} v_{A2} \\ C + \overrightarrow{CB_2} \\ v_{A3} \end{bmatrix} \quad (4)$$

Next, we can describe the relative position of any random two tiles. For example, we calculate the relative position of link *L* based on the link *A* iteratively. That is, from Eq. (1), we find v_B from v_A. Similarly, the position vector of link *C*, v_C is expressed as $v_C = f(v_B, \theta_{BC})$. Then, the position vector of link *L*, $v_L = f(f(f(f(v_A, \theta_{AB}), \theta_{BC}), \theta_{CK}), \theta_{KL})$, takes the route R_1 as shown in Figure 4(b). The mechanism passing through R_1 is considered as a chain, and each tile can use the position of the adjacent tile as a reference. The relative position between the terminal and the initial tiles are calculated iteratively based on the two adjacent tiles' kinematics. For a given folding pattern, the chain configuration is not unique. For example, another alternative route, R_2, can calculate the relative position between link *L* and link *A* as $v_L = f(f(f(f(v_A, \theta_{AB}), \theta_{BC}), \theta_{CD}), \theta_{DL})$. We can simplify the kinematic model by hardware design parameters such as the dimension of tiles, and dihedral angles from sensory data.

The positions of links are imposed by the rotation angles on the revolute joints. The rotational angle can be measured by the angular sensor described in the next Section. In contrast to the 3D scanning technique or the gesture-based 3D modeling technique, our 3D model construction does not employ a marker or external sensing device to create a geometric model on a coordinate system. Rather, we select a tile as a fixed reference plane and calculate coordinates of adjacent tiles through the bending angle of the joint.

When considering the entire structure, several routes of kinetic joints can be applied for getting the coordinates of the standard rule of route selection to find all position vectors on the tile vertices. Our distributed sensing system has a central processor, which is usually known as the fusion center, to collect and convert the sensory data to position (Viswanathan and Varshney, 1997). The sensors are independent and there is no feedback from the fusion center to the sensors. No sensor has a pre-configured global coordinate but we assigned coordinates for one of the tiles from the reference plane. The fusion center collects relative position data which can be integrates with absolute location sensors to merge multiple distributed sensing systems. Here we investigate the tiles adjacent to the center vertex in Figure 3(b), link *A* as start nodes. By assuming the path finding problem is a single-source unweighted graphs problem, we calculate the relative position of other tiles from the nearest to the farthest tiles by the shortest path, based on breadth-first search trees (Skiena, 1998).

There are two main functions of the software bridge. The first one is to reconstruct a 3D model of the physical device, and the second is to control the actuation of the self-folding sheet. The two main functions are described below. The software uses the distributed sensor data for reconstructing the 3D model. First, the collected sensor reading is filtered by a digital signal filter. In our experiments, we employ a moving average filter to reduce the low frequency noise (< 5Hz). Second, the

software converts sensory data into bending angles based on the sensor calibration data. The calibration data of each sensor embedded in the prototype is stored in a software database before the model construction. A surface model is generated beforehand according to the kinematic model of the prototype, for simulating the physical sheet where the surface model nodes cover all vertices of the triangular tiles. The detailed model construction in the software is written in Appendix B. Last, the program updates the coordinates of each tile and generates the 3D model iteratively. For feedback control of the robotic origami, the sensory data is sent to the software controller. Then, the folding sequence of robot is converted to the angular set points.

3.2. The interactive control methodology constructed in a virtual environment for robotic origami

For interaction mode 2, we developed a methodology that employs a virtual graphical interface to control a physical robotic origami. This interface enables users to interact with robotic origami in a virtual environment in a similar way to folding physical origami directly. The physics-based modeling and the transformation of physical reality can greatly augment immersive sensation of interactivity in virtual manipulation scenarios. In Interaction Mode 2, the algorithms and the manipulation methods are designed for providing an interactive interface to control the intended actuation with folding sequences of robotic origami. In this mode, our design can teach or program the folding sequences in the virtual environment for achieving the static target geometries and final configuration rather than tracking the folding speed or motion trajectory dynamically.

3.2.1. Virtual model for robotic origami generated by a physics engine.

The first step to create a virtual environment is to construct the physical model for simulating and visualizing the kinematics of robotic origami, and this physics model should be capable of simulating kinematics of origami and interactions between human input and the robot. Thus, we created the physics-based modeling based on rigid origami and multibody system. Multibody systems utilize a set of elements such as rigid part, joints, actuators and forces. The interaction can be given by a joint, a force actuator or a sensor (Schiehlen, 1997). The kinematics of single vertex rigid origami system could be simulated by a matrix model or Gaussian curvature model (Demaine and O'Rourke, 2007, Gray et al., 2010). For the kinematics, multiple vertices origami structure is modeled by the physics engine. In this framework physics engines are not only used for visual rendering, but can also be the modeling tool of rigid origami structure.

For visualizing the physical world and building an understandable interaction model in the virtual environment, we adapted the virtual rendering algorithms (Bender et al., 2015, Deul et al., 2016, Müller et al., 2007) to approximately simulate the physics of the real world. The virtual interface is more focused on

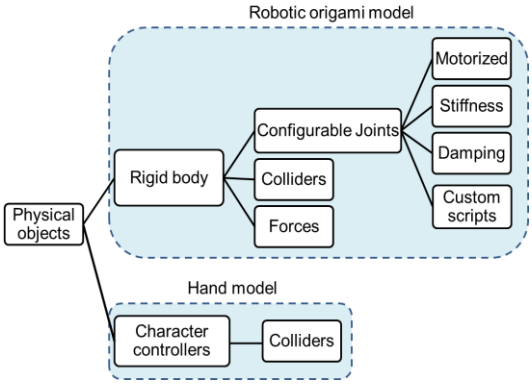


Fig.4. The architecture of physical objects for constructing a virtual environment in *Unity*.

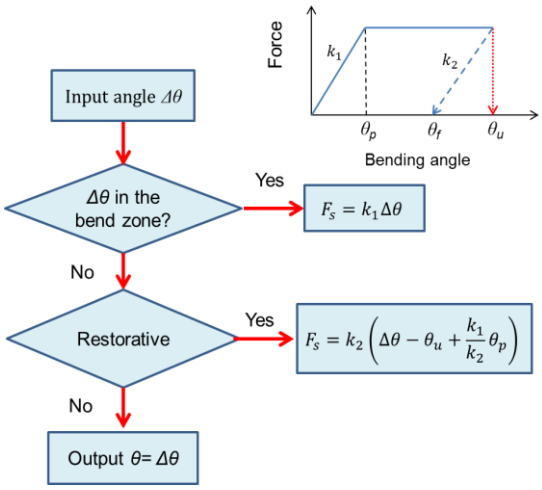


Fig. 5. The flowchart of an approximate virtual rendering algorithm for the restoration of the origami robots' Kapton hinges.

providing users with a control interface with perceptually-correct, immediate feedback and interactivity rather than simulating physics precisely in the common simulation tool. Furthermore, since we can create the virtual model of robotic origami from the on-board sensory feedback, these data can be used to fix simulation errors and refine the behavior model in the virtual environment in the future. Thus, the algorithms for building the virtual model will not be exactly the same as the physics rules, which usually simplifies the kinematics of the system. We built the model in a discrete time system with the virtual rendering rate 30 Hz, which can adequately create virtual perceptions for humans. The architecture for building a multibody system and rigid origami theory under *Unity* is shown in Figure 4 (Goldstone, 2009). We defined the rigid part of robotic origami as rigid body, which enables the objects to simulate rigid body dynamics under the control of the physics engine. This is a component that can simulate the conjunction relation between adjacent rigid parts of the models.

Due to differences in types of connecting joints of robots, they can be functional joints implemented with customized algorithms. For example, we used a Kapton layer as a passive hinge to connect tiles in our robotic origami prototypes. In this case we developed an algorithm to simulate the bending and restorative behavior of the Kapton material (Abbott, 2014). The approximate mechanical behavior and corresponding algorithm are described in Figure 5. We could implement this algorithm in the robot behavior model to give the visual perception of material properties when the user interacts with the passive hinge, even without sensory feedback from physical prototype.

The character controller is not affected by forces and will only move when

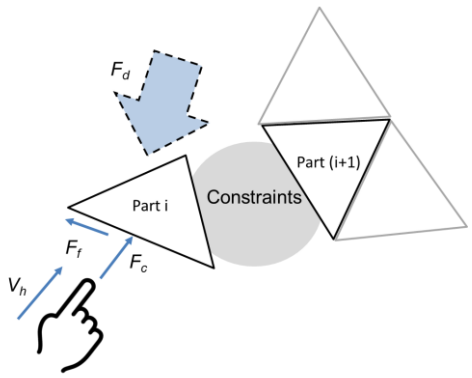


Fig. 6. Local interaction model for simulating the manipulation of robotic origami in a virtual environment. V_h is the velocity of hand motion. F_c is the contact force exerting on the part *i*.

called by the move function, which receives command from human motion input. It will then carry out the movement but be constrained by collisions. The collider allows virtual objects to collide with each other. One object can apply forces to the other via collision. It is the main mechanism to manipulate virtual objects with a user's motion.

The localized interaction model for indicating the kinematics during manipulation in the virtual environment is shown in Fig. 6. The behavior of the virtual model of the origami robot is not only dominated by the kinematic constrains but also by the forces in the virtual environment. In interaction Mode 2, the robotic origami model moves by collision forces generated by the hand model. The folding motion not only follows the user-input trajectory, but is also affected by the virtual forces that are listed in Table 1. Apart from force generated by the hand, other virtual forces like the spring force on the unactuated hinges in Fig. 5 contribute to unfolding motion. The collisions between the hand model and robot are assumed to happen on part *i*. The motion of virtual part *i* is decided by the force exerted, which F_c is the collision force generated by the hand model, F_f is the friction force between the hand and part *i*, the air drag on the rigid body. The constraints describe the relations between the parts. The virtual components generating visual rendering and converting the digital information to physical parameters for controlling robots are shown in Table 1. Once the virtual hand model collides with the robot model, it generates the collision force on the robotic origami model. The motion of the virtual hand model is only governed by the input signal from the motion tracking system. The hand model does not receive any force and the generated force can only be described by the velocity of hand motion, $F_c = M\frac{V_h}{dt}$. . The generated forces are proportional to the speed of the hands when the collision happens. However, the collision force

Table 1. The visual rendering and transformation between digital information and physical components

Virtual components	Data description	Physical components
Spring force of joints	$F_s = k\Delta\theta$	Kapton passive hinges (Elastic deformation)
Air drag	$F_d = V^2 D$	n/a
Torque of motorized joint	$T = M\frac{\omega}{dt}$	Bending sensor
Constant force	$F_c = M\frac{V}{dt}$	n/a
Hinge angle	$\Delta\theta$	SMA torsional actuator
Dynamic friction force	$\Delta V = \mu \cdot dt$	n/a

generated by the hand model will make the robot move endlessly. Thus, we created air drag force on the rigid body, which is proportional to the square of velocity, which can be expressed by $F_d = V^2 D$. Where V is the velocity of the object and D is the drag coefficient. The movement of the tile is dominated by the collision force exerted by the hands. In the actual prototype, we achieve desired folding angles thanks to the distributed sensing and consequent closed-loop control. In order to recreate the same motion of an origami in the virtual environment, we need to make sure that the each folding is completed when the hand makes no contact with the virtual model. To achieve this effect, we add an air drag effect on the origami surface after the contact is lost.

The general equation of motion used for describing the visual rendering representation of a multibody system is performed by a position-based approach (Bender et al., 2015). The vectors of generalized coordinates are denoted by q. The velocity vector can be denoted as $v = dq/dt$.From Newton-Euler equations and Lagrange's equations; we can describe the general equation motions of the rigid body by $\ddot{q} = M^{-1} f$. The mass matrix is described by *M(q)*, which depends on the deviation of constrains. f is the force exerted on central of mass. Given a known position $q_0 = q(t_0)$ and velocity $v_0 = v(t_0)$ at time $t = t_0$. Then, we define $\Delta q = q(t_0 + h) - q(t_0)$ and $\Delta v = v(t_0 + h) - v(t_0)$, where h is the updating rate of virtual rendering. The time integration is performed by implicit backward Euler method, which leads to $\Delta q = h(v_0 + \Delta v)$ and $\Delta v = hM^{-1} f(q_0 + \Delta q, v_0 + \Delta v)$. We apply Talyor series to make the first order approximation of f*:*

$$f(q_0 + \Delta q, v_0 + \Delta v) = f_0 + \frac{\partial f}{\partial q}\Delta q + \frac{\partial f}{\partial v}\Delta v \quad (5)$$

Then, we can get the virtual model of each time frame based on a fixed initial state.

To create the rigid parts of robotic origami we predefined the geometries of rigid parts by 3D CAD software such as *Blender* and imported them to *Unity*. In addition, simple geometry such as spheres and cubes can be directly edited in *Unity*. After the definition of the geometry, we added the rigid body components to the physical objects so that the objects can receive force and torque and act accordingly following physics in the real world. An important step in simulating robotic origami structure is to define the relationship between each rigid tile. We attached joints at every connection between the rigid tiles. The connected joints were defined as a revolute joint and the properties of these connected joints can be configurable as described in the previous paragraph. The integration of dynamics is performed by the symplectic Euler method for physics-based modeling, which is a popular method in computer graphics and the integration results are close to implicit backward Euler method (Bender et al., 2015).

3.2.2. Algorithms for converting virtual information to the control command of robotic origami.

In robotic control systems, the time delay for closed-loop control could result in instability of the system. Therefore we propose an algorithm to convert a continuous-time system to a discrete-time system. This algorithm includes signal processing, components behavior adaption and response generation that can divide continuous folding motions to discrete folding sequence control commands to the actuators. The algorithm enables user teaching for robotic origami by demonstrating in the virtual environment. Then, the collected data will convert to control command to control robotic origami. This is an early step towards task-oriented programming in robotic origami for high DoF mechanism control.

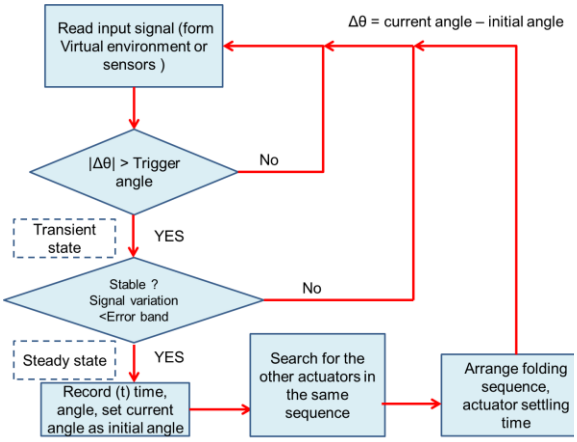


Fig. 7. The flowchart of the algorithm for converting continuous folding motions to discrete folding sequences. This algorithm converts the recorded folding angles from the virtual environment to control commands of the robotic origami, adapted to the characteristics of the hardware. In addition, this algorithm can also use the sensory data as input while folds the physical prototype to demonstrate the folding tasks.

As shown in Figure 7, the algorithm considers the mechanical characteristics of actuators and sensors in the robotic origami that can be divided into the following steps: First, the user demonstrates the folding behavior in continuous motion by the hands and the performed motions are transferred to virtual modeling by Leap motion. The graphical interface gives users immediate visual feedback of the folding process. In the first phase, the physics engine takes the simulation of interactions between user input and virtual model of robotic origami and the status of physical robotic origami does not simultaneously change while the user manipulates the virtual model. Then, the control system takes the recorded folding angles of the virtual robotic origami model and time data from the simulated results to control robotic origami. In addition to the Interaction Mode 2, this algorithm uses the sensory data from the prototype during the direct physical manipulation. Second, the control system transfers the recorded data to task structures in symbolic level representations that are the transient and stable states of robotic origami. The transient states confirm the folding motions are executed, whereas the stable state is the end of the folding sequence. Therefore, a transient state with a stable state represents a complete folding sequence. When the change of a joint angle $\Delta\theta >$ trigger angle, the task

goes to a transient state. When the signal variation is smaller than the error band for a predefined settling time, the task goes to a stable state. The trigger angle, error band and settling time can be tuned to the proficiency level of the users in the virtual environment. Third, the extraction of folding sequences represented by the task structures are relayed to actuation commands to control the robotic origami. The averaged angles of stable states will be the set points of each folding sequence for the controller. The final step of generating a discrete folding sequence is applying the converted virtual tasks on the physical robotic origami. For this robotic origami, a PID controller is employed to control the folding angles where the sensor readings dictate when to start the next folding sequence – when each sensor reading reaches a steady-state. In this step, kinematic model developed in Interaction Mode 1 can be implemented in the same graphical interface for monitoring the task sequence.

4. Experimental details

In this Section we discuss the experimental verification of the proposed interaction methodologies including hardware construction, software development and integration of the robotic systems.

4.1. Summary of required experiments

To validate the applicability of the robotic origami kinematic model based on a distributed sensing system we prototyped a multi-DoF, physical input interface and compared the actual 3D shape with the reconstructed 3D virtual model.

To achieve reconfigurability of robotic origami we implement a distributed sensing system on the robotic origami for feedback control so robotic origami can transform into different configurations from the same hardware, which is still rarely applied in robotic origami research.

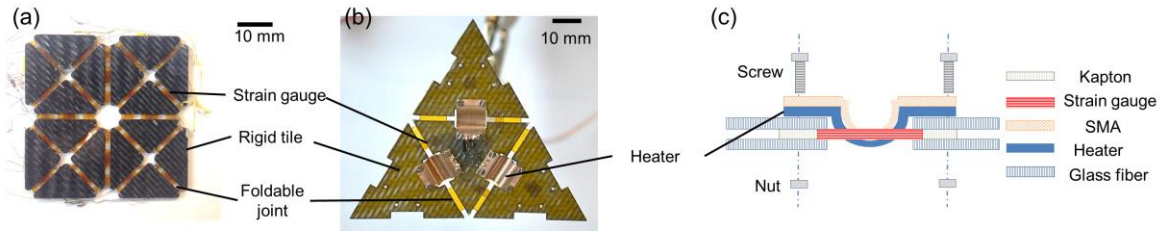


Fig. 8. Construction of two prototypes: (a) A tangible origami input device (only with sensors), and (b) robotic origami (complete with sensors and embedded actuators). The side view of the robotic origami prototype that consists of 5 functional layers (c). The robotic origami contains several functional layers including SMA actuators and sensors that enable the self-folding and self-sensing.

To demonstrate the intrinsic control methodology for robotic origami, we created a software platform with physical prototypes to teach the robot in a virtual environment and then replicate the demonstrated task on physical robots. The experimental results show the algorithm's capacity to control a multiple DoFs system in an intuitive way, as well as providing a platform for telemanipulation of the robotic origami.

4.2. Experimental platform

We prototyped two different hardware interfaces to experimentally validate the interactive interfaces: a **tangible interface** with distributed sensors and a **robotic origami** with embedded actuators and sensors. Both prototypes are assemblies of rigid tiles, foldable joints and sensing elements in a planar structure that employ a layer-by-layer fabrication method. The prototype consists of two glass fiber layers, a strain gauge, and a Kapton layer for foldable joints as depicted in Figure 8 (a). The sensory and linkage components are sandwiched between two glass fiber layers that provide a structural backing with minimal weight. The glass fiber layers, each less than 1 mm thick, form the mechanical foundation of the tiles. The shape of the glass fiber layer is an

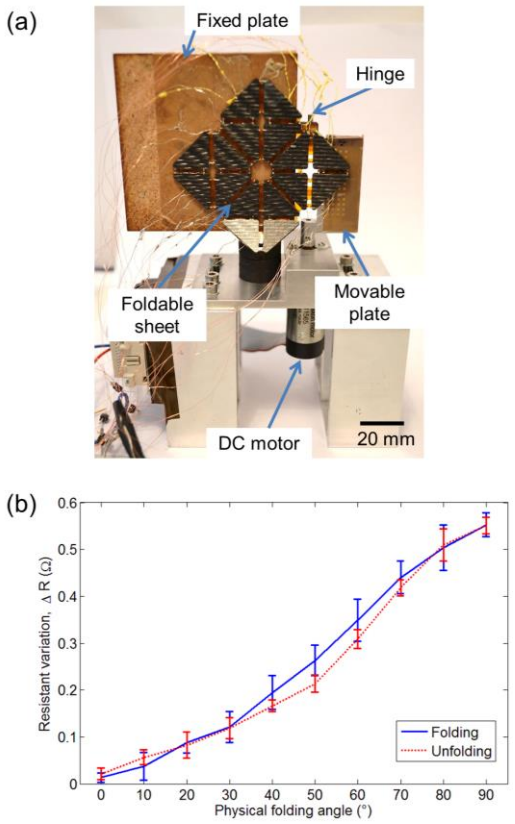


Fig. 9. The experimental setup and results of a single hinge of the tangible interface. Experimental arrangement for angular sensor testing (a) and the measured resistance variation of the strain gauge versus the bending angle (b).

isosceles triangle, of 20 mm base length and 10 mm height. The Kapton layer serves as a foldable joint between the tiles. This layer is flexible and provides

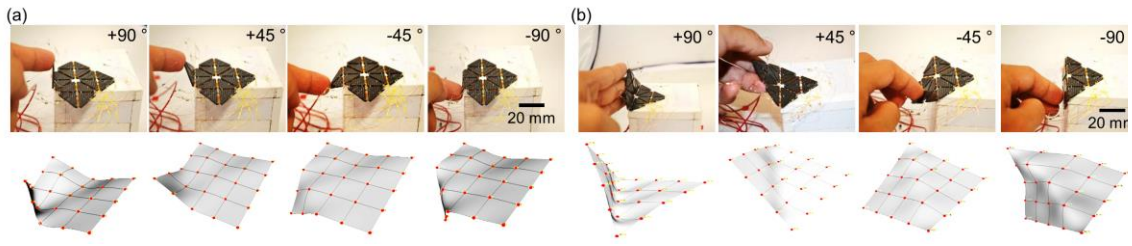


Fig. 10. Comparison between physical folding and the 3D model reconstruction. Bi-directional folding of two joints (a), bidirectional folding of four joints (b).

rotational DoF whose axis is parallel to the edges of two adjacent tiles. We adopt 120 Ω strain gauges (RS components) as bending sensors and attach them to the Kapton layer to sense the bending behavior of the layer; the sensitive direction of each strain gauge is perpendicular to the edges of two adjacent tiles.

The construction of the prototype requires several steps. We first design a 2D pattern for the glass fiber and Kapton layers and then laser micro-machine the designed pattern. The machined layers are stacked layer-by-layer and bonded by adhesive layers and after applying heat and pressure the whole structure is cured. Finally, we connect electrical wires to the strain gauges for the readout.

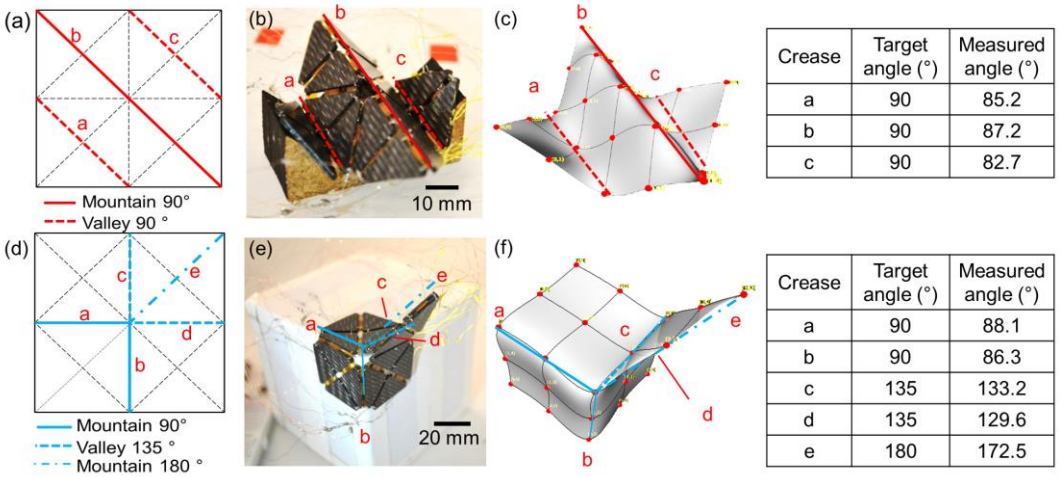
The actuated prototype uses shape memory alloy (SMA) torsional actuators (Zhakypov et al., 2016) within the layers as shown in Figure 8(b). The torsional SMA actuator generates the folding motion when it is thermally activated via a customized Copper-Kapton micro-heater. The details of the heater and SMA actuator fabrication process are discussed in (Firouzeh and Paik, 2015, Zhakypov et al., 2016). Here, strain gauges are placed along with the Kapton. All the stacks are aligned and mechanically fixed with screws and nuts that allow easy and fast assembly and dismounting. The drawing of the structure of the actuated prototype is shown in 8(c). The strain gauges were chosen for angular sensing and the resistance changes of the strain gauges are expected to be quiet small. A multiple channel resistance measurement setup with high accuracy and suitable resolution is required. For quick implementation, we selected some commercial modules to measure the sensor array rather than design and fabricate a new electrical circuit based on a Wheatstone bridge or potentiometer. Resistance measurement was accomplished with a commercial instrument (National Instruments NI USB-4065) with resolution down to 1 mΩ in the range of 1 kΩ. For multi-channel measurement we used a digital I/O device (NI 6008) to control a multiplexer (CD74HC4067). We connected the electrical measurement setup to a single sensor and measured the resistance of an unfolded sensor. The measured standard deviation was about 12 mΩ and peak-to-peak noise was about 75 mΩ.

We present an experiment to confirm and test the compatibility of the distributed sensors on the tangible interface and to evaluate the performance of the distributed sensing system, we constructed an angular sensor testing setup as shown in Figure 9(a). It consists of a DC motor with a rotary encoder and two plates; one is fixed, the other is movable, connected by a rotary hinge. The movable plate rotates along the hinge and is driven by the DC motor; the angle between the two plates is calculated from the rotary encoder data. We attached our prototype to both plates and aligned its bending axis along the rotary hinge. The prototype is folded by the DC motor and its folding angle is recorded and calculated simultaneously.

We set the DC motor to fold the prototype to 90° and unfold at a constant angular speed in a single testing cycle. At the same time the folding angle and resistance of the sensor are recorded. The resistance variation of the strain gauge is proportional to the folding angle, as shown in Figure 9(b). The resistance variation is calculated as $\Delta R = R - R_0$, where R_0 is the average resistance of the sensor in an unfolded state (with bending angle 0), R is the measured resistance and ΔR is the resistance variation corresponding to the bending angle. The sensitivity of the angular sensor as estimated from linear regression was 6 mΩ/°, and the hysteresis was 9 % of the full scale. We then use sensitivity of the angular sensor to calculate the dihedral angle between two tiles in the virtual model and after measuring the sensitivity of all angular sensors embedded in the foldable sheet, we add these calibration results into the software.

4.3. The sensor-based virtual model reconstruction of robotic origami

We validated the applicability of the multi-DoF physical input interface by comparing the actual 3D shape and the reconstructed 3D model. In the first experiment, we fold two tiles together at the corner of the sheet to 90° bi-directionally. The visualization results are shown in Figure 10(a). In the second experiment, we fold eight tiles at a time. The 3D reconstruction results are shown in Figure 10(b) and indicate that the generated virtual images are comparable to the real folding angles. However, we do observe some discrepancies on the sheet's surface that are due to signal noise. Nevertheless, the peak-to-peak noise is reduced below 5° by



Crease	Target angle (°)	Measured angle (°)
a	90	85.2
b	90	87.2
c	90	82.7

Crease	Target angle (°)	Measured angle (°)
a	90	88.1
b	90	86.3
c	135	133.2
d	135	129.6
e	180	172.5

Fig. 11. The comparison of actual folding behavior of the origami robot with its real-time 3D image generated by virtual model reconstruction software. (a) to (c) show W-shape folding and (d) to (f) show the corner shape. The folding angles of multi-DoF folding shapes in W-shape from side view (a), and a corner shape (d). The prototype is affixed on two different reference blocks to generate different folding shapes (b) and (e). The 3D model and measured bending angle of each crease are shown in (c) and (f) closely following the actual shape of the prototype.

implementing the moving average filter.

We also tested multiple shapes and the comparisons between the physical folding state and the 3D reconstruction results are shown in Figure 11. Figure 11(a) to (c) show results for the W-shape and 11(d) to (f) show results for corner-shape folding. The folding angles of creases are shown in figure 11(a) and (d). For the first folding, we fold the prototype into a W-shape by attaching the prototype to a wooden reference block to ensure that adjacent planes of the sheet are perpendicular to each other, as shown in Figure 11 (b). The image generated by the software is shown in Figure 11(c). The virtual image shows our visualization algorithm works with this multiple-DoF folding shape. Figure 11(d), (e) and (f) show another multiple-DoF folding. The prototype is laid flat against the corner of the cube. Two of the angular sensors are fully bent to 180°. These results show that the reconfigurable sheet could be a 3D shape measurement tool, with ability to measure curvature or complex topography of object surfaces while increasing the DoF of the tangible devices. Refer to the Multimedia Extension for the demonstration of interaction mode 1.

4.4. Closed-loop control of robotic origami

Using controllable actuators with feedback from distributed sensors, gains much flexibility for system reconfiguration design for any robotic origami. To control the SMA actuator, a microcontroller board (Arduino Mega) which provides pulse-width modulation (PWM) output is employed. The input power of the heater is controlled by tuning duty cycle of PWM. The electronics design was proposed in our previous paper (Zhakypov et al., 2016). However, we implemented a PID controller in *LabVIEW* for closed-loop control.

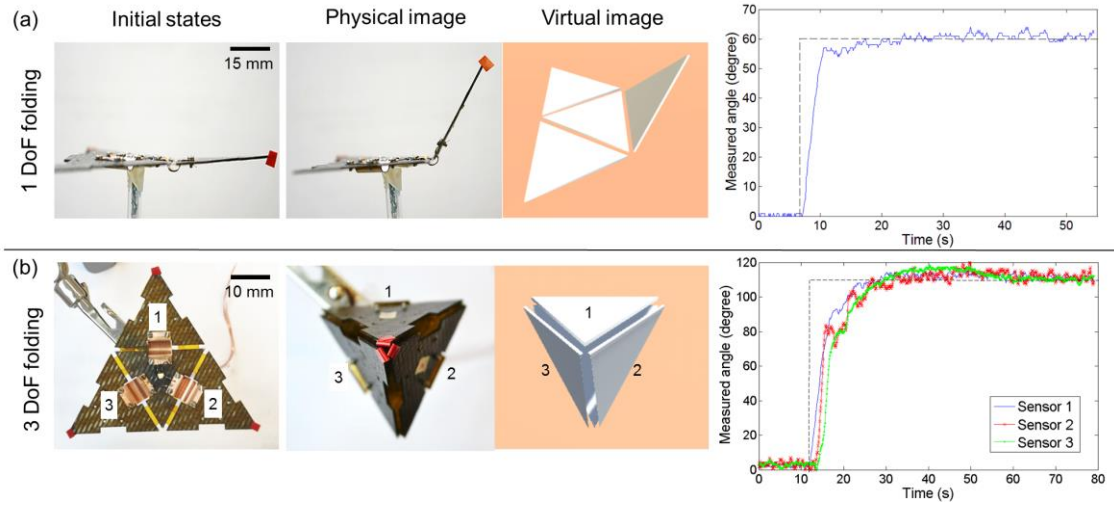


Fig. 12. Experimental results of closed-loop control of robotic origami. The pictures from left to right are i) the initial state of prototype, ii) physical folding photo, iii) the reconstructed virtual model of the folding shapes, and schemas, the sensor readings for shape 1 (a), and shape 2(b). Shape 1 folded one tile to 60°, and shape 2 folded three tiles to 110°.

The experiment includes folding the prototype into two different shapes starting from a flat configuration. The first folding attempted to valley-fold the prototype to 60° along one crease, illustrated in Figure 12 (a). The second folded all the creases to 110° to form a Tetrahedron as shown in Figure 12(b). This design-by-showing process is then transferred to the control command to control robotic origami. The set point values and response in this experiment are shown in Figure 12. The dashed line is the control input and the solid line, solid line with crosses and the solid line with points are the readings of sensors 1, 2 and 3 respectively. We can see that the module-to-module amplitudes of shape 2 have some discrepancies from the fabrication, loading effect and from the customized SMA actuators. However, this "mechanical noise" can be reduced to less than 5° by tuning the control parameters.

The images of the physical folding are shown in Figure 12 and the 3D model reconstruction is generated simultaneously with the physical folding. The simultaneous visual feedback can be employed in interactive interfaces. The folding processes of the prototype are recorded by camera and we calculate the folding angle in the physical image by tracking the red marker on the tip of the tiles. The calculated angle differences between sensor readings and results from video analysis are below 5°. Experimental results show the capabilities of the robotic origami for multiple sequence folding and possibilities for employing reinforcement learning in robotic origami with studies on the mechanism kinematics and trajectories.

4.5. Manipulation of robotic origami in a virtual environment

To evaluate the capabilities of the interface to control robotic origami, the proposed algorithm for transforming digital information into control command is implemented in the interface. We tested the algorithm with the four-tile actuated prototype, shown in Figure 8(b). We manipulated the virtual robotic origami model through a motion tracking device. We intended to fold tiles to a tetrahedron shape in 5 steps and the recorded data from this are shown in Figure 13. First, we folded hinge 3, then the bent hinge 1 and 2 together. Third, we folded hinge 1, then folded hinge 2 and 3 respectively. The scenario of folding sequences is shown at the bottom of Figure 13(a). The recorded bending angles in the virtual environment are shown in Figure 13(a). Then, the algorithm in the control system converts the recorded motion data from the virtual environment to control the commands for the robotic origami prototype, and this is shown as the dashed line in the Figure. The conversion algorithm is adapted to the human factors for manipulation in virtual environments, as well as the characteristics of the prototype, such as sensor sensitivity or settling time of the actuation system. The control results of physical prototype from sensory feedback are shown in Figure 13(b). The control command will only forward to the next sequences when all target values are achieved at a stable state. This algorithm is prominent to thoroughly fold robotic origami in multiple sequences, especially with SMA actuators whose motion is nonlinear and hardly controlled The largest oscillation

in Figure 13(b) is up to 20° all the actuators were turned on. The larger oscillation is mainly caused by the shared power sources. However, it could be decreased by changing the design of electronics and tuning PID parameters. To extend to higher DoF robot system

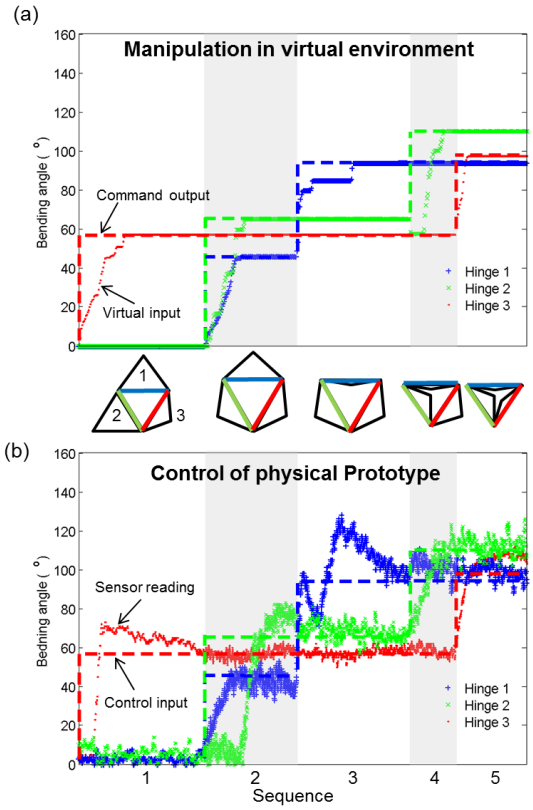


Fig. 13. Experimental results of the algorithm for conversion between digital data of manipulation and of control command. (a) The scenario of folding sequences and the schematics of folding shapes are at bottom of the figure. The bending angles acquired from the virtual environment are shown in different marks and the solid line shows the converted control commands from recorded data by the algorithm of control command conversion. (b)The control results of the physical prototype are recorded from bending sensory feedback. The solid lines show the control input for control physical prototype.

manipulation, we demonstrate a control interface with manipulation of the dual module prototype, which consists of two four-tile prototypes. The manipulation steps in the virtual environment are shown in Figure 14(a) to (d). The prototype was self-folded into the sequences from the teaching in the virtual domain, as shown in Figure 14 (e) to (h). This interactive graphical interface to manipulate robotic origami can adapt to any kind of actuator and these experimental results open up opportunities for a graphical and high-level programming interface for robotic origami. Refer to the Multimedia Extension for the demonstration of interaction mode 2.

The physics-based virtual environment can adapt to different origami structures and robotic origami, as shown in Figure 15. For interaction mode 2, we can predefine the arbitrary geometry of the rigid origami structure and then fold it in the virtual environment, as shown in Figure 15(a) to (c). This virtual environment could also serve as a simulation tool for robotic origami. Figure 15(d) to (f) show the dynamic simulation of Tribot's jumping gait (Zhakypov et al., 2015). Tribot has three SMA springs connected between each leg and we can set the spring hinges in the virtual model. Figure 15 (d) shows Tribot standing state. Then, the two SMA springs connecting the top leg and two bottom legs are virtually activated to store the energy of bottom SMA spring, shown in Figure 15(b). We release the stored energy by virtually deactivating the two SMA springs connecting the top leg and two bottom legs to activate the bottom SMA spring. The energy release makes Tribot jump, as shown in Figure 15(f). The gait control system can be

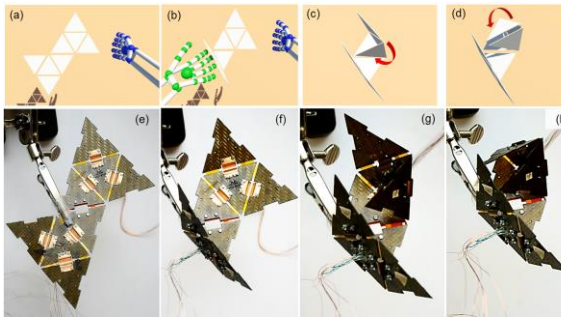


Fig. 14. Experimental results of teaching a robot by demonstration in a virtual environment. The folding motion in the virtual environment is shown in (a) to (d), and the self-folding robotic origami was achieved following the sequences that we demonstrated in the virtual environment, in (e) to (h.)

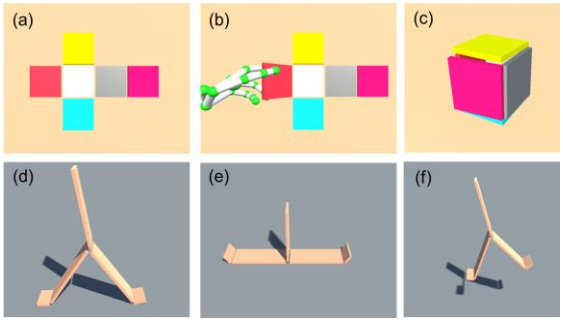


Fig. 15. The modeling with different geometries and hardware. (a) to (c) show the process of folding an origami cubic box, and (d) to (f) show simulation of Tribot jumping.

performed in virtual modeling code and shows the capability of the interface to adapt to different geometries, hardware components and fcontrol systems to manipulate and simulate.

4.6. Capabilities and limitations of the current system

The target of this paper is to design interactive control methodologies for robotic origami based on rigid origami structure. There are some capabilities and limitations arising from the design of the kinematic model and the software and hardware.

4.6.1. Capabilities and limitations of Interaction mode 1

In interaction mode 1, the effectiveness of the kinematics model is in calculating the relative coordinates of all the tiles on the robotic origami and reconstructing the geometry of robotic origami using sensory feedback. The sensory feedback provides the bending angle of each foldable joint on the robotic origami. This individual angular reading is the input for the kinematic model. Another required input is the dimensions of each tile. This model can be applied to multi-vertices and arbitrary crease pattern design. The limitations of the origami structure used in this kinematic model are assuming rigid tiles and the it should be in one piece. Kinematically this dictates the thickness to be non-zero and the creases to be straight, not curved.

The software we developed for validating the kinematic model can generate the 3D virtual model of robotic origami simultaneously with the corresponding robotic origami prototype. To generate the 3D model, all the tiles on the crease pattern of the origami structure must be described by the nodes in the software as detailed in Appendix B. Since we assume the origami structure thickness is zero, the 3D model construction has geometric bias when stacking multiple layers.

For the hardware, the fabricated prototype with 1.5 mm thickness can fold and sense the bending angle to ± 180° on single joint folding. However, it

faces foldability problems when stacking several layers, due to the thickness of the tiles and length of the joints that may be solved by structural design or kinematic synthesis (Tachi, 2011, Ku and Demaine, 2015, Chen et al., 2015).

4.6.2. Capabilities and limitations of Interaction mode 2

In interaction mode 2, the kinematics of interaction is managed by the physics engine, built in Unity system. The physics engine can handle origami structures that contain up to three vertices rather than single vertex tree structure. The created virtual environment allows the user to manipulate the virtual origami model through hand motion input. However, numerous mechanical constraints increase the calculation load for a real-time simulation. The simplification of interaction model can upgrade the robustness of the tool for handling more complex structures. Although the physics engine can handle more than tree structures only, the control methodologies and algorithms for multiple-vertices origami structure have not been thoroughly investigated in this paper.

The developed prototype is a low-profile, light-weight actuated tangible interface that has the ability to transfer to different 3D geometries from single 2D configuration. While the presented origami hardware platform only folds in a single direction, control system and algorithms allow the robotic origami to fold to static shapes within controllable bending angles on the joints, in multiple folding sequences. However, the current system lacks ability to follow a trajectory or folding speed.

5. Conclusions

In this paper we present an interactive graphical interface for controlling and simulating robotic origami. We created a virtual environment and interactive methodology to manipulate a virtual robot that enables the user to intuitively control robotic origami. For simulation, we developed an algorithm to solve the kinematics of origami using distributed sensory feedback. We designed and fabricated two prototypes to validate the capacity of the interactive interface. The first prototype is a foldable sheet with distributed angular sensors and experimental results show that it can reconstruct various 3D shapes. The second prototype uses sensors and actuators to produce a virtual model-based graphical interface to command and control the folding process of the robotic origami. These interactive interfaces and their kinematic model simplification algorithms for folding as well as sensor/actuator embedded prototypes could improve the often intuition-based origami robot design process. The experimental results reveal opportunities for further applications such as high DoF tangible interfaces and actuated tangible interfaces. In the future we will work on amalgamation of the two interaction modes to achieve a tight two-way communication and high-fidelity interaction for human-robot interface. Furthermore, the middleware such as ROS (Quigley et al., 2009) could be used to adapt to varied robotic systems and robotic simulation software. However, the current virtual environment still has some restrictions including the soft component modeling and a higher DoF interaction may be challenging to perform accurately.

Algorithms and control methodologies for more complex origami structures as well as nonlinear actuators and components are needed to push forward modeling-based control interfaces. In conclusion, we describe a novel interactive interface to advance the control methodologies of robotic origami and to gain better insight into robot-human interaction.

Funding

This research was supported by SNSF START project and partially by Swiss National Centres of Competence in Research (NCCR) Robotics.

References

ABBOTT, A. C. 2014. Characterization of Creases in Polymers for Adaptive Origami Engineering. University of Dayton.

ALONSO-MORA, J., LOHAUS, S. H., LEEMANN, P., SIEGWART, R. & BEARDSLEY, P. Gesture based human-multi-robot swarm interaction and its application to an interactive display. 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015. IEEE, 5948-5953.

AN, B. & RUS, D. 2014. Designing and programming self-folding sheets. Robotics and Autonomous Systems, 62, 976-1001.

AN, N., LI, M. & ZHOU, J. 2016. Predicting origami-inspired programmable self-folding of hydrogel trilayers. Smart Materials and Structures, 25, 11LT02.

BALKCOM, D. J. & MASON, M. T. 2008. Robotic origami folding. The International Journal of Robotics Research, 27, 613-627.

BENDER, J., MULLER, M. & MACKLIN, M. 2015. Position-based simulation methods in computer graphics. EUROGRAPHICS Tutorial Notes.

BILLARD, A., CALINON, S., DILLMANN, R. & SCHAAL, S. 2008. Robot programming by demonstration. Springer handbook of robotics. Springer.

BOWEN, L., BAXTER, W., MAGLEBY, S. & HOWELL, L. 2014. A position analysis of coupled spherical mechanisms found in action origami. Mechanism and Machine Theory, 77, 13-24.

BOWEN, L., SPRINGSTEEN, K., FRECKER, M. & SIMPSON, T. 2016. Trade space exploration of magnetically actuated origami mechanisms. Journal of Mechanisms and Robotics, 8, 031012.

BOWEN, L. A., GRAMES, C. L., MAGLEBY, S. P., HOWELL, L. L. & LANG, R. J. 2013. A classification of action origami as systems of spherical mechanisms. Journal of Mechanical Design, 135, 111008.

BYOUNGKWON AN, D. R. 2014. Designing and programming self-folding sheets. Robotics and Autonomous Systems, 62, 26.

CHEN, I. Y.-H., MACDONALD, B. A. & WÜNSCHE, B. Designing a Mixed Reality Framework for Enriching Interactions in Robot Simulation. GRAPP, 2010. 331-338.

CHIEN, C.-Y., LIANG, R.-H., LIN, L.-F., CHAN, L. & CHEN, B.-Y. FlexiBend: Enabling Interactivity of

Multi-Part, Deformable Fabrications Using Single Shape-Sensing Strip. Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, 2015. ACM, 659-663.

DAI, J. S. & CANNELLA, F. 2008. Stiffness characteristics of carton folds for packaging. Journal of mechanical design, 130, 022305.

DEMAINE, E. D. & O'ROURKE, J. 2007. Geometric folding algorithms, Cambridge university press Cambridge.

DEMENTYEV, A., KAO, H.-L. C. & PARADISO, J. A. 2015. SensorTape: Modular and Programmable 3D-Aware Dense Sensor Network on a Tape. 28th ACM User Interface Software and Technology Symposium.

DEUL, C., CHARRIER, P. & BENDER, J. 2016. Position-based rigid-body dynamics. Computer Animation and Virtual Worlds, 27, 103-112.

FELTON, S., TOLLEY, M., DEMAINE, E., RUS, D. & WOOD, R. 2014. A method for building self-folding machines. Science, 345, 644-646.

FIROUZEH, A. & PAIK, J. 2015. Robogami: A Fully Integrated Low-Profile Robotic Origami. Journal of Mechanisms and Robotics, 7, 021009.

GOLDSTONE, W. 2009. Unity game development essentials, Packt Publishing Ltd.

GRAY, S. R., SEO, J., WHITE, P. J., ZEICHNER, N. J., YIM, M. & KUMAR, V. A toolchain for the design and simulation of foldable programmable matter. ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2010. American Society of Mechanical Engineers, 1167-1176.

GREENBERG, H., GONG, M., MAGLEBY, S. & HOWELL, L. 2011. Identifying links between origami and compliant mechanisms. Mech. Sci, 2, 217-225.

HANNA, B. H., LUND, J. M., LANG, R. J., MAGLEBY, S. P. & HOWELL, L. L. 2014. Waterbomb base: a symmetric single-vertex bistable origami mechanism. Smart Materials and Structures, 23, 094009.

HERNANDEZ, E. A. P., HARTL, D. J., MALAK, R. J., AKLEMAN, E., GONEN, O. & KUNG, H.-W. 2016. Design Tools for Patterned Self-Folding Reconfigurable Structures Based on Programmable Active Laminates. Journal of Mechanisms and Robotics, 8, 031015.

HUANG, Y. & EISENBERG, M. Easigami: virtual creation by physical folding. Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, 2012. ACM, 41-48.

HULL, T. C. 2002. Modelling the folding of paper into three dimensions using affine transformations. Linear Algebra and its applications, 348, 273-282.

KELLY, A., CHAN, N., HERMAN, H., HUBER, D., MEYERS, R., RANDER, P., WARNER, R., ZIGLAR, J. & CAPSTICK, E. 2011. Real-time photorealistic virtualized reality interface for remote mobile robot

control. The International Journal of Robotics Research, 30, 384-404.

LEITHINGER, D., FOLLMER, S., OLWAL, A. & ISHII, H. Physical telepresence: shape capture and display for embodied, computer-mediated remote collaboration. Proceedings of the 27th annual ACM symposium on User interface software and technology, 2014. ACM, 461-470.

LINDLBAUER, D., GRØNBÆK, J. E., BIRK, M., HALSKOV, K., ALEXA, M. & MÜLLER, J. Combining Shape-Changing Interfaces and Spatial Augmented Reality Enables Extended Object Appearance. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2016. ACM, 791-802.

MARTIN, A. & EMAMI, M. R. An architecture for robotic hardware-in-the-loop simulation. 2006 International Conference on Mechatronics and Automation, 2006. IEEE, 2162-2167.

MILLER, A., WHITE, B., CHARBONNEAU, E., KANZLER, Z. & LAVIOLA JR, J. J. 2012. Interactive 3D model acquisition and tracking of building block structures. Visualization and Computer Graphics, IEEE Transactions on, 18, 651-659.

MIYASHITA, S., GUITRON, S., YOSHIDA, K., LI, S., DAMIAN, D. D. & RUS, D. Ingestible, Controllable, and Degradable Origami Robot for Patching Stomach Wounds. 2016 IEEE International Conference on Robotics and Automation, 2016.

MÜLLER, M., HEIDELBERGER, B., HENNIX, M. & RATCLIFF, J. 2007. Position based dynamics. Journal of Visual Communication and Image Representation, 18, 109-118.

NAKAGAKI, K., DEMENTYEV, A., FOLLMER, S., PARADISO, J. A. & ISHII, H. ChainFORM: A Linear Integrated Modular Hardware System for Shape Changing Interfaces. Proceedings of the 29th Annual Symposium on User Interface Software and Technology, 2016a. ACM, 87-96.

NAKAGAKI, K., FOLLMER, S. & ISHII, H. LineFORM: Actuated Curve Interfaces for Display, Interaction, and Constraint. Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, 2015. ACM, 333-339.

NAKAGAKI, K., VINK, L., COUNTS, J., WINDHAM, D., LEITHINGER, D., FOLLMER, S. & ISHII, H. Materiable: Rendering Dynamic Material Properties in Response to Direct Physical Touch with Shape Changing Interfaces. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2016b. ACM, 2764-2772.

NISSER, M. E., FELTON, S. M., TOLLEY, M. T., RUBENSTEIN, M. & WOOD, R. J. Feedback-controlled self-folding of autonomous robot collectives. Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, 2016. IEEE, 1254-1261.

ONAL, C. D., WOOD, R. J. & RUS, D. 2013. An origami-inspired approach to worm robots. IEEE/ASME Transactions on Mechatronics, 18, 430-438.

PAIK, J., AN, B., RUS, D. & WOOD, R. J. Robotic origamis: self-morphing modular robots. 2nd International

Conference on Morphological Computation (ICMC2011), Venice, Italy, Sept, 2012. 12-14.

PAN, Z., POLDEN, J., LARKIN, N., VAN DUIN, S. & NORRISH, J. 2012. Recent progress on programming methods for industrial robots. Robotics and Computer-Integrated Manufacturing, 28, 87-94.

PERAZA-HERNANDEZ, E. A., HARTL, D. J. & MALAK JR, R. J. 2013. Design and numerical analysis of an SMA mesh-based self-folding sheet. Smart Materials and Structures, 22, 094008.

PERAZA-HERNANDEZ, E. A., HARTL, D. J., MALAK JR, R. J. & LAGOUDAS, D. C. 2014. Origami-inspired active structures: a synthesis and review. Smart Materials and Structures, 23, 094001.

QIU, C., AMINZADEH, V. & DAI, J. S. 2013. Kinematic analysis and stiffness validation of origami cartons. Journal of Mechanical Design, 135, 111004.

QUIGLEY, M., CONLEY, K., GERKEY, B., FAUST, J., FOOTE, T., LEIBS, J., WHEELER, R. & NG, A. Y. ROS: an open-source Robot Operating System. ICRA workshop on open source software, 2009. 5.

RAFFLE, H. S., PARKES, A. J. & ISHII, H. Topobo: a constructive assembly system with kinetic memory. Proceedings of the SIGCHI conference on Human factors in computing systems, 2004. ACM, 647-654.

RASMUSSEN, M. K., PEDERSEN, E. W., PETERSEN, M. G. & HORNBÆK, K. Shape-changing interfaces: a review of the design space and open research questions. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2012. ACM, 735-744.

RENDL, C., KIM, D., FANELLO, S., PARZER, P., RHEMANN, C., TAYLOR, J., ZIRKL, M., SCHEIPL, G., ROTHLÄNDER, T. & HALLER, M. FlexSense: a transparent self-sensing deformable surface. Proceedings of the 27th annual ACM symposium on User interface software and technology, 2014. ACM, 129-138.

ROUDAUT, A., KARNIK, A., LÖCHTEFELD, M. & SUBRAMANIAN, S. Morphees: toward high shape resolution in self-actuated flexible mobile devices. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2013. ACM, 593-602.

SCHIEHLEN, W. 1997. Multibody system dynamics: Roots and perspectives. Multibody system dynamics, 1, 149-188.

SHAER, O. & HORNECKER, E. 2010. Tangible user interfaces: past, present, and future directions. Foundations and Trends in Human-Computer Interaction, 3, 1-137.

SKIENA, S. S. 1998. The algorithm design manual: Text, Springer Science & Business Media.

STROHMEIER, P., CARRASCAL, J. P., CHENG, B., MEBAN, M. & VERTEGAAL, R. An Evaluation of Shape Changes for Conveying Emotions. Proceedings of the 2016 CHI Conference on Human Factors in

Computing Systems, 2016. ACM, 3781-3792.

SUNG, C. & RUS, D. 2015. Foldable joints for foldable robots. Journal of Mechanisms and Robotics, 7, 021012.

TACHI, T. 2009. Simulation of rigid origami. Origami, 4, 175-187.

TACHI, T. Geometric considerations for the design of rigid origami structures. Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium, 2010. 458-460.

VISWANATHAN, R. & VARSHNEY, P. K. 1997. Distributed detection with multiple sensors I. Fundamentals. Proceedings of the IEEE, 85, 54-63.

WATANABE, R., ITOH, Y., ASAI, M., KITAMURA, Y., KISHINO, F. & KIKUCHI, H. 2004. The soul of ActiveCube: implementing a flexible, multimodal, three-dimensional spatial tangible interface. Computers in Entertainment (CIE), 2, 15-15.

ZHAKYPOV, Z., FALAHI, M., SHAH, M. & PAIK, J. The design and control of the multi-modal locomotion origami robot, Tribot. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, 2015. IEEE, 4349-4355.

ZHAKYPOV, Z., HUANG, J.-L. & PAIK, J. 2016. Modeling, Characterization and Control of a Novel Torsional Shape Memory Alloy (SMA) Actuator. IEEE robotics & automation magazine, vol. 23, 65-74.

Appendix A. Index to Multimedia Extensions

Table of Multimedia Extensions

Extension	Media type	Description
1	Video	Demonstration of Mode 2: the manipulation of robotic origami in a virtual environment and the description of two control modes.

Appendix B. The detailed software implementation for 3D Model construction in interaction mode 1

For visualizing the kinematic model proposed in Section 3.1, we developed 3D model visualization software in *LabVIEW*. Since the proposed kinematic model can provide the relative coordinates of all the tiles by sensory feedback, developers can use any 3D graphics software to reconstruct the 3D model of robotic origami.

We employed a 3D surface editor by Wojtek Golebiowski (https://forums.ni.com/t5/Example-Program-Drafts/3D-Picture-Interaction-3D-Surface-Editor/ta-p/3494179). This 3D surface editor uses a 2D array of nodes to describe the 3D surface. The surfaces between nodes are calculated by spline interpolation. To generate the surface model of each single triangular origami tile, we need the full coordinates

of vertices of the triangle tile. We generated a planar surface which contains 5 x 5 nodes which has enough nodes to cover all the tile vertices with a predefined distance, L, between nodes, as shown in Figure 16 (a). Thus, these nodes can be used to describe the coordinates and generate the surface of the triangular tiles, with a base length, 2L, and a height, L, as shown in Figure 16 (b). For example, the highlighted tile can be described by nodes A, B, C and D, and we can fix this tile to calculate the relative coordinates of the other tiles. By reading all bending angles between tiles from the sensors, the 3D model can be updated simultaneously. We can use different crease patterns not only in triangular shapes, but also other arbitrary patterns such as polygons, since the nodes are enough to describe the shape of each tile.

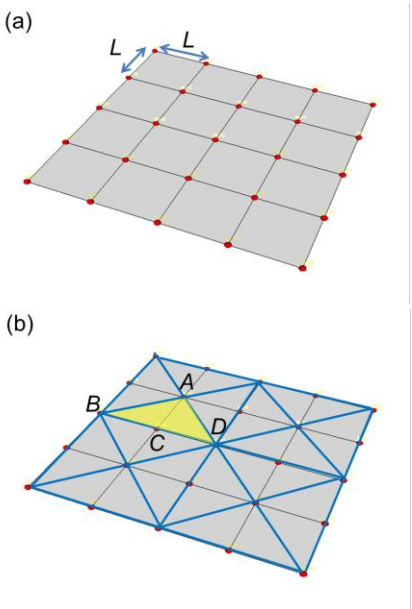


Fig. 16. The 3D model of the origami generated in *LabVIEW*. (a) A square, flat surface is predefined by a 2D array of nodes. (b) The crease pattern of the origami pattern within the triangular tiles overlays the predefined surface. The nodes used to describe the coordinates of the tiles.