Modeling and Control of Morphing Covers for the Adaptive Morphology of Humanoid Robots

Fabio Bergonti, Gabriele Nava, Luca Fiorio, Giuseppe L'Erario, and Daniele Pucci

*Abstract***—This article takes a step to provide humanoid robots with adaptive morphology abilities. We present a systematic approach for enabling robotic covers to morph their shape, with an overall size fitting the anthropometric dimensions of a humanoid robot. More precisely, we present a cover concept consisting of two main components: a *skeleton*, which is a repetition of a basic element called *node*, and a soft *membrane*, which encloses the cover and deforms with its motion. This article focuses on the cover skeleton and addresses the challenging problems of node design, system modeling, motor positioning, and control design of the morphing system. The cover modeling focuses on kinematics, and a systematic approach for defining the system kinematic constraints is presented. Then, we apply genetic algorithms to find the motor locations so that the morphing cover is fully actuated. Finally, we present control algorithms that allow the cover to morph into a time-varying shape. The entire approach is validated by performing kinematic simulations with four different covers of square dimensions and having 3×3, 4×8, 8×8, and 20×20 nodes, respectively. For each cover, we apply the genetic algorithms to choose the motor locations and perform simulations for tracking a desired shape. The simulation results show that the presented approach ensures the covers to track a desired shape with *good* tracking performances.**

*Index Terms***—Adaptive morphology, genetic algorithm (GA), humanoid robots, motion control, parallel robots, robot kinematics.**

I. Introduction

ADAPTIVE morphology is a fascinating natural feature that Robotics struggles with implementing into the existing platforms. Birds [1], snakes [2], and frogs [3], [4] are only a few examples of the large variety of animals that use adaptive morphology for several goals, ranging from increasing locomotion abilities to augmenting self-defense systems [5].

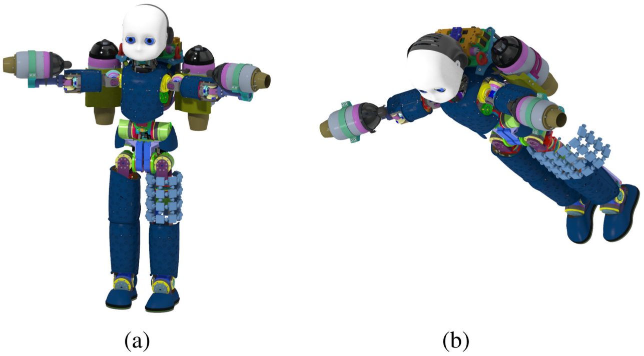


Fig. 1. Conceptual picture of the *iRonCub* robot with a morphing cover integrated in the left thigh. The cover morphs and adapts its shape to optimize aerodynamics. The real *iRonCub* is currently being tested to perform the first flight [16]. (a) Rest configuration. (b) Morphed configuration.

Although successful attempts of implementing adaptive morphology into robotic platforms exist [6]–[9], humanoid robots are still conceived with links having rigid covers, which limit their capacity to adapt to the surrounding environment and to optimize locomotion and manipulation tasks. This article proposes a concept of morphing covers for humanoid robots and addresses the associated fundamental problems of kinematic modeling, actuation positioning, and control design of the morphing system.

Most of humanoid robot design is governed by the idea of having a skeleton enclosed by rigid covers [10]–[13]. The covers are the first robot body part that makes contact with the environment, so they have a pivotal role during any robot *physical* task—e.g., locomotion, human–robot interaction, manipulation, etc. Consequently, adaptive morphology via morphing covers may lead to unprecedented features of humanoid robots. Morphing feet that optimize locomotion patterns (e.g., flat feet for balancing and point/rounded feet for walking) and adaptive chest shapes to pass through narrow cavities are only a few leaps forward that morphing covers of humanoid robots would enable. Our long-term vision is to provide future flying humanoid robots the ability to morph their shape to optimize the resulting aerodynamics acting on the robot. We are currently implementing the vision of morphing flying humanoid robots on iRonCub [14], [15], a modified version of the iCub robot powered by turbojet engines integrated in the chest and in the arms (see Fig. 1).

The idea of enhancing robot capabilities via cover and link redesign is not new to the Robotics community. Examples are the

Manuscript received 9 November 2021; accepted 21 March 2022. Date of publication 10 June 2022; date of current version 4 October 2022. This article was recommended for publication by Associate Editor R. Kramer-Bottiglio and Editor E. Yoshida upon evaluation of the reviewers' comments. *(Corresponding author: Fabio Bergonti.)*

Fabio Bergonti, Giuseppe L'Erario, and Daniele Pucci are with the Artificial and Mechanical Intelligence Laboratory, Fondazione Istituto Italiano di Tecnologia, 16163 Genoa, Italy, and also with the School of Computer Science, University of Manchester, M13 9PL Manchester, U.K. (e-mail: fabio.bergonti@iit.it; giuseppe.lerario@iit.it; daniele.pucci@iit.it).

Gabriele Nava is with Artificial and Mechanical Intelligence Laboratory, Fondazione Istituto Italiano di Tecnologia, 16163 Genoa, Italy (e-mail: gabriele.nava@iit.it).

Luca Fiorio is with iCub Tech, Fondazione Istituto Italiano di Tecnologia, 16163 Genoa, Italy (e-mail: luca.fiorio@iit.it).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2022.3170281.

Digital Object Identifier 10.1109/TRO.2022.3170281

integration of soft materials as robot skin to perceive contacts and absorb collisions [17]–[19], and the design of links with an active variable stiffness control [20]. However, how to implement effectively morphing abilities of humanoid robots is still an open issue although there is an active scientific effort in tackling several problems around the topic—see, e.g., the problems of shape morphing structures, surface control, and transformation from 2-D surfaces into 3-D bodies. We briefly review below the main approaches for tackling these problems.

The ancient art of *origami* gave impetus to researchers in proposing folding mechanisms capable of evolving from 2-D to 3-D structures [21], [22]. At the actuation level, the folding feature may be driven by pneumatic actuators [23], [24], thread-based mechanisms powered by motors [25], smart memory alloys [26], and heat sources [27]. *Origami* multi-degree-of-freedom (DoF) mechanisms can also play the role of actuation mechanisms for *reconfigurable surfaces* [28]. In this case, the resulting *reconfigurable surface* much differs from the classical configuration of having pixels that can translate only [29]–[32]. In general, *reconfigurable surfaces* are usually characterized by an actuation system that generates absolute movements of the pixels and the actuator's stators are fixed to ground. Therefore, reconfigurable surfaces little fit the application of morphing covers for humanoid robots.

Soft robots may implement morphing features since they are often made of hyperelastic materials with Young's modulus in the order of 10^4 and 10^9 [33]–[35]. Examples of soft robots that implement a degree of controllable morphing feature are active textiles with McKibben muscles [36], [37], silicon-rubber pad [38], buckling surfaces [39], groove-based mechanism [40], heat-sealing inflatable materials pneumatically activated [41], surfaces textures for camouflage [42]–[44], and underwater morphing robots to optimize hydrodynamic forces [45]. Most of these, however, are designed to achieve a unique or a limited range of configurations. A proof of concept to enable a larger spectrum of controllable motions is a morphing surface actuated with liquid crystal elastomers [46].

4-D printed variable shaping structures leverage a new frontier of manufacturing, the 4-D printing [47]. *4-D printing* consists of 3-D printed structures that evolve with time when exposed to a predetermined stimulus such as water [48], temperature [49], magnetic fields [50], etc.

Morphing wings are innovative flight supports capable of changing their shape and adapt to different aerodynamics conditions [51], [52]. The challenges when designing *morphing wings* are numerous, ranging from mechanism design to the development of deformable skins that shall withstand the aerodynamic loads [53]. *Morphing wings* can, thus, reconfigure the planform [54]–[56], perform out-of-plane transformation [57], and adjust the airfoil profile [58].

Self-reconfigurable robots change their shape by rearranging the connectivity of their constitutive parts [59]–[61]. For instance, modular self-folding robots can be composed of a repetition of triangular modules [62]. From an actuation standpoint, these modules may be either active or passive, but they are both given with the capability of coupling via a specific actuation mechanism [63]. The main characteristic of these self-reconfigurable robots is their *low* thickness that allows the mechanism to fold. Consequently, one can generate 3-D origami from an initial 2-D layout. On one hand, the concept of modular robots leads to a high structural versatility since the repetition of a basic module allows us to obtain *complex* shapes. Furthermore, the robot design pivots around the basic element to repeat and not around the entire mechanism. On the other hand, self-reconfigurable robots are designed to rearrange their connectivity, so they are not ideal in the case a single connectivity pattern should be kept by the overall platform.

The survey above highlights some of the existing technologies that may be considered when attempting to implement a degree of morphing feature onto a humanoid robot. Our specific use case of *morphing covers* for (flying) humanoid robots, however, introduces dimensional and structural constraints that call for a word of caution. From the mechanical standpoint, the cover material shall be stiff enough to resist the aerodynamic pressure without deforming or breaking apart, while the cover size shall ease its integration in a human-sized robot. From the functional standpoint, the morphing cover shall have a degree of maneuverability to induce enough variations and benefits in terms of robot shape and aerodynamic loads. Moreover, the cover shall be robust against air pressure and avoid (at most) hysteresis behaviors. The range of achievable shapes shall be various and not limited to bistable states only. Finally, the cover actuation shall be easily reproducible and of simple integration into the mechanics of the humanoid robot.

According to these use-case constraints, we believe that none of the existing methods can be applied *as such* for the design of morphing covers for humanoid robots. Rather, we believe that the morphing cover design and control shall combine ideas and principles developed by the *soft* and *self-reconfigurable* robot literature. We, thus, propose a morphing cover composed of a *skeleton*, namely a rigid mechanism that enables the cover to move, and a *soft external membrane*, namely an elastic material that encloses the cover to exploit aerodynamic forces by filling cavities and passively deforms with its motion. Although the concept of *shape morphing* is usually associated with soft structures, we refer to as *morphing cover* the proposed rigid skeleton and soft membrane because the system macroscopically morphs from one shape to another; once integrated into a humanoid robot, the covers evolution cause a morphological variation of the robot. The article, however, focuses on the cover skeleton only, and the analysis of the soft membrane will be the subject of future studies. The cover skeleton is composed of a repetition of a rigid basic element, called *node*. The node is of square shape and is connected to the four adjacent nodes via spherical joints. Joints can be actuated, and the problem of finding which joint to actuate is also investigated. Differently from the *self-reconfigurable* literature, our approach is optimized for the case where the connectivity pattern of all nodes does not change.

At the theoretical level, this article presents a systematic, model-based, and scalable approach for the modeling, actuation positioning, and control of a morphing system. The system modeling is approached using maximal coordinates, which allows us to easily tackle kinematics with closed loops. The modeling approach differs from the state of the art related to similar self-reconfigurable robots that use minimal coordinates [64]. Motor positioning is tackled using genetic algorithms (GAs), thus defining a minimal set of actuators that enable the design of global stabilizers of desired 3-D shapes. The proposed motor positioning and control design much differ from the existing methods applied to morphing systems that mostly deal with: 1) strategies to place active modules optimized for open chain mechanism [65]; 2) definitions of optimal 2-D layout to obtain a 3-D shape relying on origami theory [64]; and 3) planning algorithms to define the sequence of foldings [64].

More precisely and compactly, the contributions of this article are as follows:

1) The design of the morphing cover skeleton and node with the aim of applying it to a humanoid robot.
2) The development of the cover kinematic model.
3) a Genetic Algorithm(s) for selecting the motor locations so as to obtain a fully actuated cover.
4) The control design for the morphing cover allowing it to track a time-varying desired 3-D shape.
5) The validation with simulation results verifying the robustness of the approach against noise.

Although the contributions above can also be applied to other natures of robots, we are currently investigating the use of a specific kind of motor that induces a node dimension in the order of 25 mm. In turn, the node size implies an overall cover dimension that fits the anthropometric dimensions of existing humanoid robot covers (see Section VI).

The rest of this article is organized as follows. Section II introduces notation and recalls the basis of Genetic Algorithm(s). Section III presents the kinematic model of the proposed morphing cover. Section IV describes a Genetic Algorithm(s) for motor optimal placing. Section V presents a framework for the control design of the morphing cover. Section VI shows the simulations results. Finally, Section VII concludes this article.

II. Background

A. Notation

1) $\mathrm{SO}(3) := \{R \in \mathbb{R}^{3\times 3} \mid R^\top R = I_3,\ \det(R) = 1\}$.
2) $\mathrm{SE}(3) := \left\{ \begin{bmatrix} R & \boldsymbol{p} \\ \boldsymbol{0} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \mid R \in \mathrm{SO}(3),\ \boldsymbol{p} \in \mathbb{R}^3 \right\}$.
3) $\mathcal{S}(\boldsymbol{x}) \in \mathbb{R}^{3\times 3}$ is the skew symmetric matrix associated with the cross product in $\mathbb{R}^3$, namely, given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$, then $\boldsymbol{x} \times \boldsymbol{y} = \mathcal{S}(\boldsymbol{x})\boldsymbol{y}$.
4) The vectors $\boldsymbol{e_1}$, $\boldsymbol{e_2}$, and $\boldsymbol{e_3}$ denote the canonical basis of $\mathbb{R}^3$.
5) w denotes the inertial (or world) frame; a, b, and c are generic body-fixed frames.
6) $\boldsymbol{p}_a \in \mathbb{R}^3$ denotes the origin of the a frame expressed in the inertial frame w.
7) $R_b^a \in \mathrm{SO}(3)$ is the rotation matrix that transforms a 3-D vector expressed with the orientation of the b frame in a 3-D vector expressed in the a frame, i.e.,

$$R_b^a = \begin{bmatrix} \hat{\boldsymbol{x}}_b^a & \hat{\boldsymbol{y}}_b^a & \hat{\boldsymbol{z}}_b^a \end{bmatrix} \quad \text{with} \quad \hat{\boldsymbol{x}}_b^a = R_b^a \boldsymbol{e_1}$$

$\hat{\boldsymbol{x}}_b^a$, $\hat{\boldsymbol{y}}_b^a$, and $\hat{\boldsymbol{z}}_b^a$ are the unit vectors of the frame axes.

8) $\mathcal{Q}_b^a \in \mathbb{R}^4$ denotes the unit quaternion representation of the rotation matrix R_b^a.
9) $\boldsymbol{p}_{b,c}^a \in \mathbb{R}^3$ is the relative position of c frame w.r.t. to b frame written in a frame, i.e., $\boldsymbol{p}_{b,c}^a = R_w^a(\boldsymbol{p}_c - \boldsymbol{p}_b)$.
10) $\dot{\boldsymbol{p}}_{b,c}^a \in \mathbb{R}^3$ is the linear velocity of c frame w.r.t. to b frame written in a frame.
11) $\boldsymbol{\omega}_{b,c}^a \in \mathbb{R}^3$ is the angular velocity of c frame w.r.t. to b frame written in a frame.
12) $\boldsymbol{V_{\mathcal{B}}} \in \mathbb{R}^6$ is the 6-D velocity of base link.

Given the vectors $\boldsymbol{p}_{b,c}^a$, $\dot{\boldsymbol{p}}_{b,c}^a$, $\boldsymbol{\omega}_{b,c}^a$, and $\mathcal{Q}_b^a$ or the matrix R_b^a, if one of the terms a, b, or c is omitted, it is implicitly assumed to be equal to the inertial frame w.

B. Short Recap of Genetic Algorithm(s)

Genetic Algorithm(s) are search algorithms inspired by the process of natural selection [66]. Given an initial population composed of a set of candidates, an *evolution process* improves iteratively the characteristics of the populations according to a metric defined by a *fitness function*. The fitness function is the objective function that we want to maximize (or minimize). Often, a population member is a scalar number or an array encoded into binary strings.

The evolution of the population is performed using three operators [67]: 1) reproduction; 2) crossover; and 3) mutation. *Reproduction* is the selection process of some candidates from the current population. The selection methods are many, e.g., *fitness proportionate* defines the probability of selecting a candidate to be proportional to its fitness [68]. *Crossover* is a stochastic method to generate new population members from an existing one. *Single-point crossover* is a common implementation, where the strings of two candidates are swapped at a specific string location [69]. *Mutation* is a process that alters randomly a population member. It enables the algorithms to look for new solutions and to avoid getting stuck into local minima having suboptimal fitness values.

III. Modeling the Morphing Cover Skeleton

This section presents the skeleton of the morphing cover and the associated kinematic modeling.

A. Cover Skeleton and Mathematical Preliminaries

The proposed morphing cover is composed of two main components: a *cover skeleton*, i.e., a rigid mechanism that provides the cover with the ability to move, and a *soft membrane*, i.e., an elastic material that encloses the cover and that passively deforms with its motion. As mentioned in the introduction, this article focuses on the skeleton design, modeling, and control. In particular, the skeleton is a mechanism composed of a repetition of a rigid element, called *node*. Here, we propose a node of square section that is connected to the adjacent nodes via spherical joints. For example, Fig. 2 shows a morphing cover skeleton composed of nine nodes and 12 joints. In the remainder, we will often refer to the morphing cover skeleton as *mesh*, or simply as *morphing cover*.

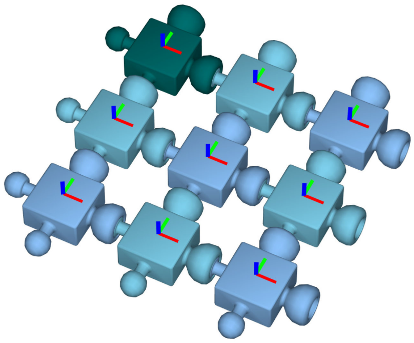


Fig. 2. 3×3 morphing cover skeleton. The coordinate frames follow RGB convention. The green-colored node is fixed to the ground.

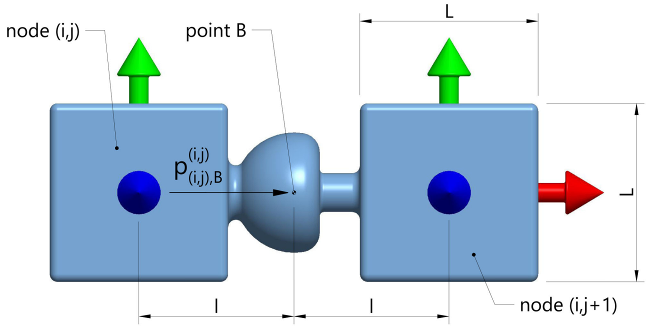


Fig. 3. Connection between node (i, j) and $(i, j + 1)$. The point B is the center of the spherical joint.

The following presents mathematical preliminaries, definitions, and assumptions that are used in the rest of this article.

Assumption 1: The morphing cover skeleton is composed of nm nodes, with n and m being two natural numbers.

Preliminary 1: Each node is uniquely identified with a pair of natural numbers (i, j), with $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, m\}$. The numbers (i, j) are called *row* and *column* indexes, respectively.

Definition 1: The node of indexes $(i, j) = (1, 1)$ is called *father node* or *base link*. Being of square section, two orthogonal directions can be associated with the father node, here referred to as *row direction* and *column direction*.

For example, the column and row directions may be represented by the red and green arrows in Fig. 3, respectively. Now, we can define how the morphing cover skeleton is built.

Preliminary 2: Arrange the nodes so that all their square sections belong to the same plane. Consider the *father node* $(1, 1)$ and its row and column directions. We, then, put the node of indexes $(2, 1)$ along the row direction at a distance $2l$ from node $(1, 1)$, and we connect the two aforesaid nodes via a spherical joint. Each joint has three DoFs. We repeat the operations by connecting in series the nodes of indexes $(3, 1)$, $(4, 1)$, …, $(n, 1)$ via spherical joints. Then, we go back to the *father node*, and we place the node of indexes $(1, 2)$ along the father node column direction. Again, we connect the nodes via a spherical joint. Then, using spherical joints, we connect in series the nodes $(2, 2)$, $(3, 2)$, …, $(n, 2)$ along the row direction. This time, however, each node is also connected by columns, e.g.,, the nodes $(2, 2)$ and $(2, 1)$ are also connected via spherical joints. We repeat the overall procedure by incrementing the row and column indexes accordingly. We stop when all nm nodes are displaced in a *matrix-like* mesh.

At the end of this procedure, we obtain a cover skeleton composed of nm nodes displaced in a matrix-like mesh: all the nodes are connected by spherical joints (see, e.g., Fig. 2). Observe that the total number N_{sj} of spherical joints in the cover is given by $N_{\text{sj}} = 2nm - n - m$. Finally, we place reference frames in each node composing the morphing cover.

Preliminary 3: Each node is associated with a coordinate frame placed at the center of the square section. More precisely, a node of indexes (i, j) is associated with the pair $(\boldsymbol{p}_{(i,j)}, R_{(i,j)}) \in \text{SE}(3)$ consisting in the vector $\boldsymbol{p}_{(i,j)} \in \mathbb{R}^3$ representing the origin of the node frame expressed w.r.t. the inertial frame and the node orientation matrix $R_{(i,j)} \in \text{SO}(3)$ composed of unit column vectors expressed w.r.t. the inertial frame. All the node frames have the same orientation when the mesh is *stretched* (namely, when all the square sections belong to the same plane)—see, e.g., Fig. 2.

B. System State

The morphing cover skeleton obtained by applying the procedure described in Preliminary 2 is, at all effects and purposes, a mechanical system. Hence, we have to define the *system state*: namely, a set of relationships from which the position and velocity of each point of the cover can be uniquely determined in the inertial frame. Classically, this problem is solved by choosing a set of minimum variables representing the system configuration and by computing their time derivative to represent the system velocity. The variables $\boldsymbol{q}, \dot{\boldsymbol{q}} \in \mathbb{R}^k$, for instance, are often used to represent the joint angles and velocities associated with fixed-base manipulators with k rotational/prismatic joints [70].

Being a *highly* parallel mechanism, the process of finding a set of minimum variables that uniquely characterize the cover configuration may not be a straightforward task. For this reason, the route we follow to characterize the system state is to consider a *complete* variable representation complemented by a set of constraints. More precisely, the system state is represented by the pair $(\boldsymbol{q}, \boldsymbol{v})$, with

$$\boldsymbol{q} = \left(\boldsymbol{p}_{(1,1)}, R_{(1,1)} \dots \boldsymbol{p}_{(i,j)}, R_{(i,j)} \dots \boldsymbol{p}_{(n,m)}, R_{(n,m)}\right) \quad \text{(1a)}$$

$$\boldsymbol{v} = \left(\dot{\boldsymbol{p}}_{(1,1)}, \boldsymbol{\omega}_{(1,1)} \dots \dot{\boldsymbol{p}}_{(i,j)}, \boldsymbol{\omega}_{(i,j)} \dots \dot{\boldsymbol{p}}_{(n,m)}, \boldsymbol{\omega}_{(n,m)}\right) \quad \text{(1b)}$$

complemented by the equations that constrain the nodes to move according to the spherical joints that connect them, i.e.,

$$g(\boldsymbol{q}) = \boldsymbol{0} \quad \text{(2a)}$$

$$J_c(\boldsymbol{q})\,\boldsymbol{v} = \boldsymbol{0}. \quad \text{(2b)}$$

In other words, the *system state* belongs to the following set:

$$\Theta := \{(\boldsymbol{q}, \boldsymbol{v}) \in \text{SE}(3)^{nm} \times \mathbb{R}^{6nm} : g(\boldsymbol{q}) = \boldsymbol{0}, J_c(\boldsymbol{q})\,\boldsymbol{v} = \boldsymbol{0}\}$$

where $\text{SE}(3)^{nm}$ is the nm-ary Cartesian power of the set $\text{SE}(3)$. What follows defines the holonomic and differential constraints $g(\boldsymbol{q}) = \boldsymbol{0}$ and $J_c(\boldsymbol{q})\boldsymbol{v} = \boldsymbol{0}$, respectively. The linear map $J_c(\boldsymbol{q})$ is here referred to as the *constraint jacobian*.

C. Holonomic and Differential Constraints

To define the holonomic constraints $g(\boldsymbol{q}) = \boldsymbol{0}$, we need to find out the geometric relationships that spherical joints impose on the motion between two nodes. Now, considering Fig. 3, observe that the midpoint B in the spherical joint must satisfy the following constraint:

$$\boldsymbol{p}_{(i,j)} + R_{(i,j)} \boldsymbol{p}^{(i,j)}_{(i,j),B} = \boldsymbol{p}_{(i,j+1)} + R_{(i,j+1)} \boldsymbol{p}^{(i,j+1)}_{(i,j+1),B} \,.$$

Note also that $\boldsymbol{p}^{(i,j)}_{(i,j),B}$ and $\boldsymbol{p}^{(i,j+1)}_{(i,j+1),B}$ are constant. The former is equal to $\boldsymbol{l_1} = \boldsymbol{e_1} l$, while the latter to $-\boldsymbol{l_1}$. Therefore, one has

$$\boldsymbol{p}_{(i,j)} + R_{(i,j)} \boldsymbol{l_1} = \boldsymbol{p}_{(i,j+1)} - R_{(i,j+1)} \boldsymbol{l_1} \,. \tag{3}$$

By repeating the above procedure for all spherical joints, and by moving the terms on the right-hand side of (3) to its left-hand side, we obtain the holonomic constraints $g(\boldsymbol{q}) = \boldsymbol{0}$.

Now, to obtain the differential constraints $J_c(\boldsymbol{q})\boldsymbol{v} = \boldsymbol{0}$, differentiate (3) w.r.t. to time, which leads to

$$\dot{\boldsymbol{p}}_{(i,j)} + \mathcal{S}\big(\boldsymbol{\omega}_{(i,j)}\big) R_{(i,j)} \boldsymbol{l_1} = \dot{\boldsymbol{p}}_{(i,j+1)} - \mathcal{S}\big(\boldsymbol{\omega}_{(i,j+1)}\big) R_{(i,j+1)} \boldsymbol{l_1}. \tag{4}$$

Rearranging the terms of (4) in a matrix form leads to

$$\left[I_3 \;\; -\mathcal{S}\Big(R_{(i,j)} \boldsymbol{l_1}\Big)\right] \begin{bmatrix} \dot{\boldsymbol{p}}_{(i,j)} \\ \boldsymbol{\omega}_{(i,j)} \end{bmatrix} = \left[I_3 \;\; \mathcal{S}\Big(R_{(i,j+1)} \boldsymbol{l_1}\Big)\right] \begin{bmatrix} \dot{\boldsymbol{p}}_{(i,j+1)} \\ \boldsymbol{\omega}_{(i,j+1)} \end{bmatrix}. \tag{5}$$

Using two selector matrices $S_{\boldsymbol{v}_{(i,j)}}$ and $S_{\boldsymbol{v}_{(i,j+1)}}$, we can express the above constraint w.r.t the velocity $\boldsymbol{v}$, namely

$$\left[I_3 \quad -\mathcal{S}\Big(R_{(i,j)} \boldsymbol{l_1}\Big)\right] S_{\boldsymbol{v}_{(i,j)}} \boldsymbol{v} = \left[I_3 \quad \mathcal{S}\Big(R_{(i,j+1)} \boldsymbol{l_1}\Big)\right] S_{\boldsymbol{v}_{(i,j+1)}} \boldsymbol{v}. \tag{6}$$

An analogous constraint can be found for the spherical joint that connects node (i, j) to $(i + 1, j)$. Then, by stacking all constraints in a matrix leads to the differential constraints:

$$J_c \boldsymbol{v} = \boldsymbol{0}, \; J_c \in \mathbb{R}^{3N_{\text{sj}} \times 6nm}. \tag{7}$$

1) Fixed Father Node Constraints: Besides the holonomic and differential constraints due to the cover spherical joints, we add another constraint that simplifies–without loss of generality–the overall modeling task: we assume that the father node does not move in the inertial frame. This translates in additional holonomic constraints, i.e., the father node position $\boldsymbol{p}_{(1,1)}$ and orientation $R_{(1,1)}$ are constant. Thus

$$\boldsymbol{v}_{(1,1)} := \begin{bmatrix} \dot{\boldsymbol{p}}_{(1,1)} \\ \boldsymbol{\omega}_{(1,1)} \end{bmatrix} := \boldsymbol{V_{\mathcal{B}}} = S_{\boldsymbol{v}_{(1,1)}} \boldsymbol{v} = 0 \tag{8}$$

where the selector matrix $S_{\boldsymbol{v}_{(1,1)}}$ is used to select only the term $\boldsymbol{v}_{(1,1)}$ from the vector $\boldsymbol{v}$. In the language of the floating base system, (8) is the so-called *fixed-base* assumption and complements the constraints (7) leading to

$$J_c \boldsymbol{v} = \boldsymbol{0}, \; J_c \in \mathbb{R}^{3N_{\text{sj}} + 6 \times 6nm}. \tag{9}$$

2) Constraint Jacobian Properties: The rows of the constraint jacobian J_c are the system's differential constraints. Some of these constraints, however, might be redundant, i.e., the jacobian J_c may be rank deficient. The Jacobian rank indicates the number of active constraints at a given cover configuration $\boldsymbol{q}$. Once the number of active constraints is identified, we can evaluate the *instantaneous* DoFs of the morphing cover at a given configuration $\boldsymbol{q}$ as follows:

$$\text{DoF} = \dim(\boldsymbol{v}) - \text{rank}\,(J_c)\,. \tag{10}$$

The rank computation of the constraint jacobian J_c is performed numerically, and its value may change along with the mesh configuration $\boldsymbol{q}$. The constraint jacobian J_c, however, will provide us with important information. For instance, its null space, the so-called $\mathcal{Z}_{\boldsymbol{v}}$, represents the vector space of all feasible cover movements in a given configuration.

D. System Modeling in Relative Coordinates

Constraints (8) and (9) are derived with the state vector $\boldsymbol{v}$, which represents the node velocities expressed in the *inertial frame*. Remind that nodes are connected via spherical joints. Since our objective is to actuate the spherical joints, then the relative angular velocities between two nodes will be affected by the joint actuation. Therefore, it is important to have a system state representation in terms of *relative node velocities*.

Definition 2: $\boldsymbol{\omega}_r^R \in \mathbb{R}^{3N_{\text{sj}}}$ is the vector containing all the relative angular velocities between consecutive links, i.e.,

$$\boldsymbol{\omega}_r^R = \left(\boldsymbol{\omega}^{(1,1)}_{(1,1),(1,2)} \cdots \boldsymbol{\omega}^{(i,j)}_{(i,j),(i,j+1)},\, \boldsymbol{\omega}^{(i,j)}_{(i,j),(i+1,j)} \cdots\right). \tag{11}$$

Definition 3: $\boldsymbol{\nu} \in \mathbb{R}^{6+3N_{\text{sj}}}$ is the vector of base velocity $\boldsymbol{V_{\mathcal{B}}}$ and relative velocities $\boldsymbol{\omega}_r^R$, i.e.,

$$\boldsymbol{\nu} = \left[\boldsymbol{V_{\mathcal{B}}}\,;\boldsymbol{\omega}_r^R\right]. \tag{12}$$

Given the system velocities $\boldsymbol{v}$ in the inertial frame, it is straightforward to define a linear transformation $\mathcal{M}_{\boldsymbol{v}}^{\boldsymbol{\nu}}$ such that $\boldsymbol{\nu} = \mathcal{M}_{\boldsymbol{v}}^{\boldsymbol{\nu}} \boldsymbol{v}$. The first term $\boldsymbol{V_{\mathcal{B}}}$ is contained in both terms $\boldsymbol{v}$ and $\boldsymbol{\nu}$. All the other terms of $\boldsymbol{\nu}$ can be computed as

$$\boldsymbol{\omega}^{(i,j)}_{(i,j),(i,j+1)} = R^{(i,j)} \boldsymbol{\omega}_{(i,j),(i,j+1)} \tag{13a}$$

$$\boldsymbol{\omega}_{(i,j),(i,j+1)} = \boldsymbol{\omega}_{(i,j+1)} - \boldsymbol{\omega}_{(i,j)}. \tag{13b}$$

Using the selector matrices $S_{\boldsymbol{\omega}_{(i,j)}}$ and $S_{\boldsymbol{\omega}_{(i,j+1)}}$, one has

$$\boldsymbol{\omega}_{(i,j),(i,j+1)} = S_{\boldsymbol{\omega}_{(i,j+1)}} \boldsymbol{v} - S_{\boldsymbol{\omega}_{(i,j)}} \boldsymbol{v} \tag{14}$$

and substituting (14) into (13a) gives

$$\boldsymbol{\omega}^{(i,j)}_{(i,j),(i,j+1)} = R^{(i,j)} \left(S_{\boldsymbol{\omega}_{(i,j+1)}} - S_{\boldsymbol{\omega}_{(i,j)}}\right) \boldsymbol{v}. \tag{15}$$

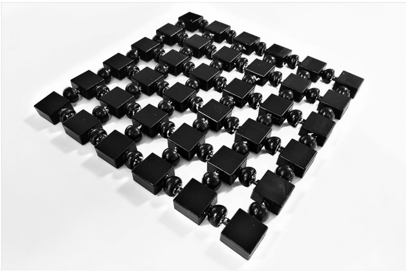


Fig. 4. 3-D printed prototype of a 6×6 morphing cover skeleton with passive spherical joints designed with the proposed approach.

By repeating this procedure for all the spherical joint, we finally get to a linear mapping between $\boldsymbol{\nu}$ and $\boldsymbol{v}$

$$\boldsymbol{\nu} = \mathcal{M}_{\boldsymbol{v}}^{\boldsymbol{\nu}} \boldsymbol{v}. \tag{16}$$

We also apply the transformation to the null space projector $\mathcal{Z}_{\boldsymbol{v}}$, so we can pass from the null space projector expressed in the inertial frame to that expressed in relative coordinates

$$\mathcal{Z}_{\boldsymbol{\nu}} = \mathcal{M}_{\boldsymbol{v}}^{\boldsymbol{\nu}} \mathcal{Z}_{\boldsymbol{v}}. \tag{17}$$

IV. Actuation Positioning

In the previous section, we designed the skeleton of the morphing cover concept, and we addressed the kinematic model that characterizes its evolution. To actuate the cover, we assume that all or some of the three DoFs of each spherical joint are actuated. This section addresses the problem of defining which DoF of the spherical joints shall be actuated. In general, actuation positioning for parallel structures is not trivial since the number of (actionable) joints is larger than the number of structure DoFs. Therefore, a possible criterion is to place a number of actuators equal to the number of the cover DoFs, i.e., render the mesh *fully actuated*. However, there may exist a large number of actuation patterns that ensure a fully actuated cover, and this number may change depending on the cover motion. What follows proposes solutions to these motor placement problems using GAs.

Let us make the following assumptions.

Assumption 2: Motors actuate the three-DoF spherical joints. Therefore, a maximum of three motors can be placed in correspondence of one spherical joint.

In practice, we make Assumption 2 true by assuming that a spherical joint is composed of three intersecting revolute joints. Therefore, we may have less than three motors in correspondence of a spherical joint, and in this case, the nonactuated revolute joints remain *passive*. Under Assumption 2, the overall objective of the GA presented below can be stated as follows: The motors shall actuate a set of revolute joints that ensure *full actuation* of the cover in a neighborhood of a cover configuration $\bar{\boldsymbol{q}}$.

An example of the several mechanisms that can actuate the revolute joints is the *Faulhaber* brushless motor 0308H003B (3-mm diameter, 8.4-mm length, and 0.026-m N$\cdot$m holding torque) with a planetary gearheads *Faulhaber* 03A 125:1. We are currently investigating its use and integration in one of the prototypes that we are currently producing (see Fig. 4).

Since the motors actuate all or some of the three joints composing the spherical joint connecting two nodes, we make the following actuation assumption.

Assumption 3: Consider the spherical joint connecting the nodes (i, j) and $(i, j+1)$. Then, a motor placed at this spherical joint imposes one and only one component of the associated relative angular velocity $\boldsymbol{\omega}_{(i,j),(i,j+1)}^{(i,j)}$.

In light of Assumption 3, the problem of positioning the cover motors corresponds to the problem of choosing a set of relative angular velocities that ensures full actuation.

To address the problem, we make use of the following property of the null space projector $\mathcal{Z}_{\boldsymbol{\nu}}$.

Corollary 1: The problem of motor positioning is equivalent to the problem of finding the indexes of the linear independent rows of the matrix $\mathcal{Z}_{\boldsymbol{\nu}}$.

The proof is given in the Appendix. We now have to find the indexes of the linear independent rows of the matrix $\mathcal{Z}_{\boldsymbol{\nu}}$.

Theoretically, we may address this problem with a *brute force approach*. The number of all possible sets to check is given by the binomial coefficient:

$$\binom{3N_{\text{sj}}}{\text{DoF}} = \binom{3 \times \text{Number spherical joints}}{\text{Number of motors}}. \tag{18}$$

Therefore, the objective is to choose one actuation pattern—among the numerous possibilities (18)—that renders the cover fully actuated in a neighborhood of the configuration $\bar{\boldsymbol{q}}$. To give the reader an insight on the size of the brute force approach, consider a 4×4 mesh and assuming that we want to place ten motors—i.e., a configuration with ten DoFs. In this case, there are about 10^{11} possible actuation patterns to check, among which one has to be chosen. Hence, the brute force approach becomes quickly unusable for large covers, especially when additional logic should be considered when choosing a pattern, e.g., to minimize the number of motors at a spherical joint.

To address this problem, we implement a simple GA based on single-point crossover and with a fitness function f_{GA} to be maximized

$$f_{\text{GA}}(h) = \left| \det \left(\mathcal{Z}_{\boldsymbol{\nu}} \ (h, :) \right) \right| \ f_R(h) \ f_P(h) \tag{19}$$

where h is a candidate string that contains the indexes of the selected rows of $\mathcal{Z}_{\boldsymbol{\nu}}$. The term $f_R(h)$ is a reward added to privilege solutions that place the lower number of motors in correspondence of a single spherical joint. The term $f_P(h)$ is a dynamic penalty cost added to maintain population diversity and create a new generation without duplicate strings.

The output of the GA is a population $\mathcal{P}$ composed by strings h. Each string contains the indexes of linearly independent rows of the matrix $\mathcal{Z}_{\boldsymbol{\nu}}$, and–thanks to Corollary 1–it corresponds to the sets of optimal motor placement. However, the optimality of the solution is only guaranteed at the specific mesh configuration $\bar{\boldsymbol{q}}$ considered by the GA.

In order to robustify the cover full actuation in a neighborhood of $\bar{\boldsymbol{q}}$ and mitigate the dependence on the cover geometry, we perform a sensitivity analysis to check the variation of the determinant w.r.t. small changes in the mesh configuration.

Define g_{SA} as

$$g_{\text{SA}}(h) = \sum_{i \in h} \left(\frac{\partial}{\partial \theta_i} \det\left(\mathcal{Z}_{\boldsymbol{\nu}}\,(h,:)\right)\right)^2 \tag{20}$$

where θ_i is a small angular variation produced by the ith actuator. We compute g_{SA} for all the strings that belong to the population $\mathcal{P}$. The candidate with lower g_{SA} is the solution whose determinant is less affected by changes in the neighborhood configuration. In the ideal case of $g_{\text{SA}} = 0$, the determinant remains constant in the neighborhood of the configuration, and the motors remain linear independent.

The above GA will be applied to locate the motors of several cover examples in Section VI. More precisely, the outputs of the algorithm that will be used in the sections next are defined by the following statements.

Definition 4: The vector $\boldsymbol{\omega}^R_{r_{\text{act}}} \in \mathbb{R}^{\text{DoF}}$ represents the relative angular velocities selected using the GA, and therefore, it represents the vector of motor velocities.

Definition 5: The submatrix $\mathcal{Z}_{\text{act}} \in \mathbb{R}^{\text{DoF}\times\text{DoF}}$ is obtained by selecting the linear independent rows of the matrix $\mathcal{Z}_{\boldsymbol{\nu}}$.

In light of the definitions above, we can state the following lemma that is useful to relate the relative angular velocities $\boldsymbol{\omega}^R_{r_{\text{act}}}$ (motor actions) to the cover velocities ν and all nodes velocities $\boldsymbol{v}$ expressed in the inertial frame.

Lemma 1: Given $\boldsymbol{\omega}^R_{r_{\text{act}}}$, the absolute and relative cover velocities $\boldsymbol{v}$ and $\boldsymbol{\nu}$ are evaluated as

$$\boldsymbol{\nu} = \mathcal{Z}_{\boldsymbol{\nu}}\,\mathcal{Z}_{\text{act}}^{-1}\boldsymbol{\omega}^R_{r_{\text{act}}} \tag{21a}$$

$$\boldsymbol{v} = \mathcal{Z}_{\boldsymbol{v}}\,\mathcal{Z}_{\text{act}}^{-1}\boldsymbol{\omega}^R_{r_{\text{act}}}. \tag{21b}$$

The GA has been developed using the node angular velocities expressed in relative coordinates because of Assumption 2. It scales, however, to the case where the cover were given with an actuation mechanism that acts directly the nodes state in the inertial frame. This additional feature, however, is not exploited in this article.

V. Control Design

In the previous section, we considered a specific cover configuration $\bar{\boldsymbol{q}}$, and we placed a number of motors equal to the number of cover DoFs at that configuration. The motor pattern ensures a minimum sensitivity of the cover DoFs, which are, thus, expected not to vary in a (relatively *large*) neighborhood of the cover configuration $\bar{\boldsymbol{q}}$. This section presents the control design for the cover motors.

The overall control objective is to allow the cover to morph (or converge) into a desired shape via the application of proper motor velocities, namely, via the relative angular velocities $\boldsymbol{\omega}^{(i,j)}_{(i,j),(i,j+1)}$—see Assumption 3. More precisely, the desired cover shape is given by the mapping $z = f(x,y,t)$, which induces the normal $\hat{\boldsymbol{n}}(t) = \nabla f(x,y,t)$. Concerning the cover desired shape, we also make the following assumption.

Assumption 4: The desired shape $z = f(x,y,t)$ passes through the constant origin of the father node (1, 1). Namely

$$z_{(1,1)} = f\left(x_{(1,1)}, y_{(1,1)}, t\right) \quad \forall t.$$

Now, consider the node (i,j). The control objective for the node (i,j) is here defined as the alignment of its normal $\hat{\boldsymbol{z}}_{(i,j)} = R_{(i,j)}\boldsymbol{e_3}$ with the desired shape normal evaluated as

$$\hat{\boldsymbol{n}}_{(i,j)}(t) = \nabla f(x,y,t)\Big|_{x_{(i,j)},y_{(i,j)}}$$

where $x_{(i,j)}$ and $y_{(i,j)}$ are the instantaneous position coordinates of the node (i,j) w.r.t. the world frame. Note that as soon as the cover moves, the desired normal associated with the node (i,j) varies even if the desired shape does not change versus time. Concerning the control law to generate an angular velocity $\boldsymbol{\omega}_{(i,j)}$ that makes $\hat{\boldsymbol{z}}_{(i,j)} = R_{(i,j)}\boldsymbol{e_3}$ converges toward $\hat{\boldsymbol{n}}_{(i,j)}(t)$, we make use of the following result.

Proposition 1 (see [71, p. 75]): Assume that $\boldsymbol{\omega}_{(i,j)} = \boldsymbol{\omega}^{\star}_{(i,j)}$ with

$$\boldsymbol{\omega}^{\star}_{(i,j)} = \boldsymbol{\omega}_{n(i,j)} + \left(k_{(i,j)}(t) + \frac{\dot{\sigma}(t)}{\sigma(t)}\right)\mathcal{S}\left(\hat{\boldsymbol{z}}_{(i,j)}\right)\hat{\boldsymbol{n}}_{(i,j)} + \lambda_{(i,j)}(t)\hat{\boldsymbol{z}}_{(i,j)}$$

and $\boldsymbol{\omega}_{n(i,j)} = \mathcal{S}(\hat{\boldsymbol{n}}_{(i,j)})\dot{\hat{\boldsymbol{n}}}_{(i,j)}$ the instantaneous angular velocity of $\hat{\boldsymbol{n}}_{(i,j)}$, $\lambda_{(i,j)}(\cdot)$ any real valued continuous function, $\sigma(\cdot)$ any smooth positive real-valued function such that $\inf_t \sigma(t) > 0$, and $k_{(i,j)}(\cdot)$ any continuous positive real-valued function such that $\inf_t k_{(i,j)}(t) > 0$ ensures the exponential stability of the equilibrium $\hat{\boldsymbol{z}}_{(i,j)} = \hat{\boldsymbol{n}}_{(i,j)}$ with the domain of attraction $\{\hat{\boldsymbol{z}}_{(i,j)} : \hat{\boldsymbol{z}}_{(i,j)}{}^{\top}\hat{\boldsymbol{n}}_{(i,j)} \neq -1\}$ at $t = 0$.

The tracking of the fictitious angular velocity $\boldsymbol{\omega}^{\star}_{(i,j)}$ for all nodes is attempted using an instantaneous optimization problem composed of a quadratic cost function and a set of linear constraints where the motors angular velocity $\boldsymbol{\omega}^R_{r_{\text{act}}}$ are the optimization variable. The problem writes

$$\min_{\boldsymbol{\omega}^R_{r_{\text{act}}}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\Lambda_{(i,j)}\|\boldsymbol{\omega}_{(i,j)} - \boldsymbol{\omega}^{\star}_{(i,j)}\|_2^2 \tag{22a}$$

$$\text{s.t.}: \boldsymbol{\omega}_{(i,j)} = S_{\boldsymbol{\omega}_{(i,j)}}\mathcal{Z}_{\boldsymbol{v}}\mathcal{Z}_{\text{act}}^{-1}\boldsymbol{\omega}^R_{r_{\text{act}}} \qquad \forall\, i,j \tag{22b}$$

$$-\boldsymbol{\omega}^R_{r_{\max}} \leq \boldsymbol{\omega}^R_{r_{\text{act}}} \leq \boldsymbol{\omega}^R_{r_{\max}} \tag{22c}$$

$$\cos(\alpha) \leq \hat{\boldsymbol{x}}_{(i,j)}{}^{\top}\hat{\boldsymbol{x}}_{(i,j+1)} \leq 1 \qquad \forall\, i,j \tag{22d}$$

$$\cos(\alpha) \leq \hat{\boldsymbol{y}}_{(i,j)}{}^{\top}\hat{\boldsymbol{y}}_{(i+1,j)} \leq 1 \qquad \forall\, i,j. \tag{22e}$$

The problem (22a) translates the task of controlling the cover normals thanks to its cost function: it aims at minimizing the difference between the absolute angular velocity of each node and the reference angular velocity $\boldsymbol{\omega}^{\star}_{(i,j)}$, which, in turn, ensures convergence in the ideal case of no inequality constraint (see Proposition 1). Note that the reference angular velocity $\boldsymbol{\omega}^{\star}_{(i,j)}$ depends on the input $\boldsymbol{\omega}^R_{r_{\text{act}}}$ due to the term $\dot{\hat{\boldsymbol{n}}}_{(i,j)}$. This dependence induces an algebraic loop. We solve the algebraic loop by discretizing the term $\dot{\hat{\boldsymbol{n}}}_{(i,j)}$ so that it no longer depends on the angular velocities, but only on the cover configuration. Although there exist other ways for evaluating the desired shape normals that do not induce algebraic loops, the choice above proves to achieve satisfactory overall tracking performances of a desired shape (see Section VI).

To regularize and avoid sharp, possibly discontinuous, variations of the control inputs, two extra terms may be added to the cost function and minimized: the norm of $\boldsymbol{\omega}^R_{r_{\text{act}}}$ and the difference between $\boldsymbol{\omega}^R_{r_{\text{act}}}$ and its value computed at the previous iteration.

The hard constraint (22b) links the control variable and the absolute angular velocity of each node. While the matrix $S_{\boldsymbol{\omega}_{(i,j)}}$ extracts $\boldsymbol{\omega}_{(i,j)}$ from $\boldsymbol{v}$. The matrix inverse of $\mathcal{Z}_{\text{act}}$ can be computed using a damping factor to mitigate instabilities caused by singularities.

The inequalities (22c), (22d), and (22e) limit the control input according to hardware limitations, such as the maximum motor speed and the range of motion of the spherical joints. More precisely, the mechanical collision between the stud and the socket of a spherical joint reduces its range of motion. The movements of the stud, in fact, must belong to a cone with semi-aperture equal to α. Since the axis of the stud is always parallel to one of the unit vectors of the frame axis, limiting the relative movement of that vector is equivalent to limit the movements of the stud. In case of a spherical joint placed in between links (i,j) and $(i,j+1)$, the scalar product between $\hat{\boldsymbol{x}}_{(i,j)}$ and $\hat{\boldsymbol{x}}_{(i,j+1)}$ must be higher than $\cos\alpha$, as stated in (22d). Substituting $\hat{\boldsymbol{x}}_{(i,j)} = R_{(i,j)}\boldsymbol{e_1}$ into (22d), we obtain

$$\cos(\alpha) \leq \boldsymbol{e_1}^\top R^{(i,j)}_{(i,j+1)}\boldsymbol{e_1} \leq 1 \tag{23}$$

with $R^{(i,j)}_{(i,j+1)}$ estimated using forward Euler integration, i.e.,

$$R^{(i,j)}_{(i,j+1)} = \left[I_3 + \mathcal{S}\left(\boldsymbol{\omega}^{(i,j)}_{(i,j),(i,j+1)}\right)dt\right] R^{(i,j)}_{(i,j+1)}(k). \tag{24}$$

$R^{(i,j)}_{(i,j+1)}(k)$ is the measured rotation matrix at iteration k, and dt is the time increment. The same analysis for the spherical joint between the joints (i,j) and $(i+1,j)$ leads to (22e).

The control laws that are obtained as a solution to (22a) ensure (quasi) *global* stability as long as $\boldsymbol{\omega}_{(i,j)} - \boldsymbol{\omega}^\star_{(i,j)} = \boldsymbol{0}\ \forall(i,j)$. Under this condition, the nodes normals converge toward those of the desired shape *whatever* (but one, i.e., $\hat{\boldsymbol{z}}_{(i,j)}{}^\top\hat{\boldsymbol{n}}_{(i,j)} = -1$) initial condition we set for the morphing cover—see the stability properties of Proposition 1. The possibility for the control laws to ensure $\boldsymbol{\omega}_{(i,j)} - \boldsymbol{\omega}^\star_{(i,j)} = \boldsymbol{0},\ \forall(i,j)$ largely depends on the cover mechanical limits and on the actuation abilities to make the skeleton move, either locally or globally. If we choose the actuators using the algorithm presented in Section IV, then the morphing cover is *locally fully actuated*. For this reason, the control laws obtained as the solution to (22a) ensures local stability under a large variety of conditions and basically independently from the desired (feasible) shape.

Since the actuation positioning ensures the cover to be locally fully actuated, critical conditions for the control laws occur when the morphing cover is initialized *far* from the desired shape. In these cases, when converging toward the desired shape, the cover may lose *full actuation* leading to ill-posed singular control laws. To mitigate these singularity issues and obtain control laws that are globally defined, regularization terms can be added in cost functions (22a), which can be implemented also by damping the inverse $\mathcal{Z}_{\text{act}}$ in (22b). Although these regularization terms alter the stability properties of the controller, they lead to globally defined laws at the cost of higher tracking errors.

Another strategy that may be attempted when the cover is initialized *far* from the desired shape $\mathcal{A}$ is that of generating a time-varying desired shape $\bar{\mathcal{A}}(t)$ that starts close to $\boldsymbol{q}_0$ and converges toward $\mathcal{A}$. The desired time-varying trajectory $\bar{\mathcal{A}}(t)$ shall ensure that the cover DoFs do not change along itself. This process of generating well-posed trajectories may be addressed by an additional control layer, namely, a *trajectory planning* layer, that feeds the control laws presented above. The trajectory planning problem, however, is beyond the scope of this article all the more so because it calls for the development of *ad hoc* numerical techniques that can be seldom used online [72].

VI. Simulation Results

In this section, we first briefly discuss the implementation of the simulation environment and the results obtained using the motor positioning algorithm presented in Section IV. Then, we show simulation results of four different meshes controlled with the control framework presented in Section V, and finally, we discuss the obtained results. The investigated meshes are of dimensions 3×3, 4×8, 8×8, and 20×20. The mesh 4×8 reproduces the *iRonCub* thigh cover integrating the morphing capability. The video of the simulations described in this section are shown in the accompanying movie https://youtu.be/kMfXb2xqGn4?t=350. The code to reproduce the results is available at https://bit.ly/morph2022.

A. Simulation Environment

The approach is tested using a custom-made MATLAB environment that integrates the cover kinematics and its constraints (9). The environment makes use of a vector representation of the system configuration space (1a). More precisely, we define the following.

Definition 6: The state space $\boldsymbol{x}$ and its time derivative is

$$\boldsymbol{x} = \left(\boldsymbol{p}_{(1,1)}, \mathcal{Q}_{(1,1)} \cdots \boldsymbol{p}_{(i,j)}, \mathcal{Q}_{(i,j)} \cdots \boldsymbol{p}_{(n,m)}, \mathcal{Q}_{(n,m)}\right) \tag{25a}$$

$$\dot{\boldsymbol{x}} = \left(\dot{\boldsymbol{p}}_{(1,1)}, \dot{\mathcal{Q}}_{(1,1)} \cdots \dot{\boldsymbol{p}}_{(i,j)}, \dot{\mathcal{Q}}_{(i,j)} \cdots \dot{\boldsymbol{p}}_{(n,m)}, \dot{\mathcal{Q}}_{(n,m)}\right). \tag{25b}$$

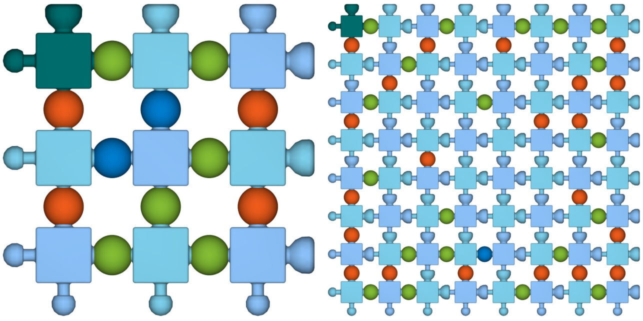
The orientation of the nodes is parameterized using quaternions to avoid singularities generated employing Euler angles.

The system evolution is obtained integrating $\dot{\boldsymbol{x}}$ using MATLAB variable step numerical integrator *ode45*. The optimization problem (22a) is solved every 0.01 s using OSQP [73] via CasADi [74]. Once motors velocity $\boldsymbol{\omega}^R_{r_{\text{act}}}$ are known, $\dot{\boldsymbol{x}}$ is computed using (21b) and the quaternion derivative definition plus an additional term to compensate numerical integration errors and ensure $\|\mathcal{Q}\|_2 = 1$ (see [75] for details). The code to model and simulate the system evolution of a multibody system with rigid links relying on a maximal coordinate approach is available at https://github.com/ami-iit/mystica.

The mesh geometries and characteristics are listed in Table I.

TABLE I
MESH DIMENSIONS AND HARDWARE LIMITS

Characteristics		Value
Node: square length	L	25[mm]
Node: distance from square center to B	l	25[mm]
Motor: Limit angular Velocity	$\boldsymbol{\omega}^R_{r_{max}}$	5 $\left[{}^{\text{deg}}/_{\text{s}}\right]$
Joints: Limit range of motion	α	50[deg]



(a)

(b)

Fig. 5. Output of the GA described in Section IV. The sphere color identifies the presence of a motor. When the color is *sky blue*, the spherical joint is passive. If the color is either *red*, *green*, or *blue*, the corresponding axis of rotation is actuated following the RGB convention, while the others axes remain passive. (a) 3 × 3 mesh. (b) 8 × 8 mesh.

B. Actuation Positioning

The algorithm presented in Section IV optimizes the motor placement at a given cover configuration $\bar{\boldsymbol{q}}$. We ran the algorithm using the cover initial configuration. The cover initial configuration is obtained sampling a *paraboloid* curve for the 3×3, 8×8, and 20×20 meshes, and a *cylinder* for the 4×8 mesh [see Fig. 1(a)]. Once the configuration $\bar{\boldsymbol{q}}$ is known, thanks to (10), it is possible to compute the number of DoFs and, consequently, the number of motors. At the chosen initial configuration, the meshes $3 \times 3, 4 \times 8, 8 \times 8,$ and 20×20 have 12, 21, 42, and 114 DoFs, respectively. To initialize the GA, we created a random population of 100 candidate strings, while the crossover and the mutation probability are set equal to 0.6 and 0.01 after a tuning process started from the values adopted in literature [76]. The evolutionary algorithm stops once the population is not evolving over the last 10^3 generations and the fitness function is higher than a desired threshold. The algorithm converged in 10^4 generations for the meshes 3×3, 4×8, and 8×8 and 10^6 generations for the mesh 20×20. To compute (20), we considered all the configurations generated applying a motor variation of $\pm 5^\circ$ from the initial state. In this way, the cover configuration stays in a neighborhood of the initial configurations. Fig. 5 shows the optimal motor pattern after the sensitivity minimization for 3×3 and 8×8. Notice the effectiveness of the term $f_R(h)$ in (19): the proposed solutions actuate at most one axis of rotation per spherical joint.

C. Control

Simulations with four meshes of different sizes are performed to test the controller and validate the ability of the considered meshes to morph into the desired shapes. To test the robustness of the proposed control approach, we also present simulations with noise in the system state used by the control action. To measure the control performances, instead, we defined two metrics. The first is the angular error in the alignment between the node normal $\hat{\boldsymbol{z}}_{(i,j)}$ and the desired normal $\hat{\boldsymbol{n}}_{(i,j)}$. For the node (i, j), the error is defined as

$$\varepsilon_{(i,j)} = \left|\arccos\left(\hat{\boldsymbol{z}}_{(i,j)}{}^\top \hat{\boldsymbol{n}}_{(i,j)}\right)\right|.$$

The minimization of $\varepsilon_{(i,j)}$ is obtained thanks to the task implemented in the cost function (22a).

The second metric is the node position error, which corresponds to the z-distance between the nodes and the curve, i.e.,

$$\zeta_{(i,j)} = \left|\boldsymbol{e_3}{}^\top \boldsymbol{p}_{(i,j)} - f\left(x_{(i,j)}, y_{(i,j)}\right)\right|.$$

Although the position error is not minimized by the optimization problem (22a), it is an indicator to understand how far the cover is from the desired shape. In the next subsections, we present some data and plots analyzing the two metrics, while meshes simulations are shown in the accompanying video.

The matrix inverse of the constraint (22b) is approximated with a damped pseudoinverse to mitigate possible effects due to singularities. In detail, $\mathcal{Z}_{\text{act}}^{-1} \simeq (\mathcal{Z}_{\text{act}}^\top \mathcal{Z}_{\text{act}} + 10^{-6} I)^{-1} \mathcal{Z}_{\text{act}}^\top$.

1) Mesh 3×3: The desired shape that the cover has to track is a function properly scaled and shifted of the form

$$f(x, y) = xy \cos y.$$

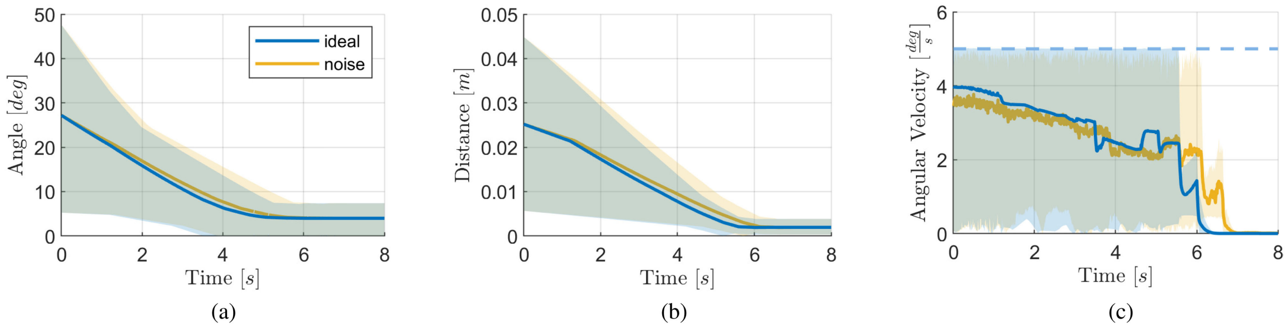
In Fig. 6, we compare two cases: the first, called *ideal*, where the actuation is considered perfect; the second, called *noise*, where we intentionally added Gaussian noise to *motor velocity*. The noise is generated such that it can only decrease motor speed to emulate friction or the disturbances generated by the elastic soft membrane. Its amplitude is proportional to motor speed, and it can reach maximum 20% of motor velocity.

After 8 s, both the *ideal* and *noise* cases decrease the average orientation error and average position error to 4° and 2 mm. The presence of actuation noise increases the response time without interfering with the task accomplishment. The high-frequency variations in the motor velocity depicted in Fig. 6(c) are due to the Gaussian noise; however, the output of the optimization problem is less oscillating.

2) Mesh 8×8: It is tested with a time-varying desired shape properly scaled and shifted in the form of

$$f(x, y, t) = \cos(x + t) + \cos(x + y + t).$$

As done for the previous mesh, we compare two simulations: the first one, called *ideal*, characterized by a state that is perfectly estimated, while the second one, called *noise*, because we intentionally added Gaussian noise in the estimated *state* used by the controller. The noise is generated guaranteeing the satisfaction of (9); in particular, it is applied directly in the actuated joints. However, owing to the closed kinematics, the variations spread in all the joints with a remarkable change in the configuration. The noise effect is added to emulate measurement errors in the encoders. The maximum amplitude applied to actuated joints was set equal to 0.05°.

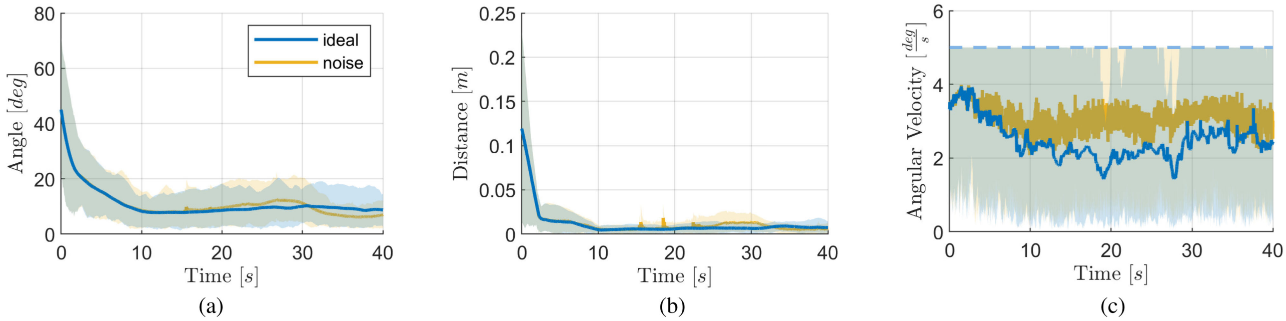


(a)

(b)

(c)

Fig. 6. Performance comparison of an ideal simulation and one with artificial noise in the actuation. The test is performed with 3 × 3 mesh. The solid lines represent the average values, while the transparent regions are the 10–90th percentile range. In (c), the dashed line is the maximum motor speed $\boldsymbol{\omega}^{R}_{r_{\max}}$. (a) Error alignment normal vectors $\varepsilon_{(i,j)}$. (b) Node position error $\zeta_{(i,j)}$. (c) Amplitude motor speed $|\boldsymbol{\omega}^{R}_{r_{\text{act}}}|$.



(a)

(b)

(c)

Fig. 7. Performance comparison of an ideal simulation and one with artificial noise in the state. The test is performed with 8 × 8 mesh. The solid lines represent the average values, while the transparent regions are the 10–90th percentile range. In (c), the dashed line is the maximum motor speed $\boldsymbol{\omega}^{R}_{r_{\max}}$. (a) Error alignment normal vectors $\varepsilon_{(i,j)}$. (b) Node position error $\zeta_{(i,j)}$. (c) Amplitude motor speed $|\boldsymbol{\omega}^{R}_{r_{\text{act}}}|$.

Fig. 7 shows 40 s of simulation; both the *ideal* and *noise* cases decrease the average orientation error and average position error to 9° and 8 mm. The presence of state noise increases response time and causes oscillations [see Fig. 7(c)]. Yet, the mesh achieves the morphing task with *acceptable* tracking performances according to the aforementioned metrics.

3) Mesh 20 × 20: It is tested with a steady desired shape properly scaled and shifted in the form of

$$f(x, y) = x^2 - y^2.$$

This simulation permitted us to check the capability of the controller to deal with meshes composed of a higher number of nodes; in particular, the number of links is 400, the number of spherical joints is 760, and the dimension of the state space $\boldsymbol{x}$ is 2800. The problem complexity increases remarkably the computational time; thus, we decided to analyze only the case with ideal actuation and ideal estimation. Fig. 8 shows 20 s of simulation, where the mesh passes from an average initial error of 41° and 180 mm to 8° and 23 mm. Even here, the mesh achieves the morphing task with *acceptable* tracking performances according to the aforementioned metrics; however, 20 s of simulation took 2 h using a PC with Intel Xeon Gold 6128 3.40 GHz and RAM 128 GB.

4) Mesh 4×8: This test presents the conceptual idea of integrating a morphing cover in a humanoid robot. We replaced the *iRonCub* thigh cover with a 4 × 8 mesh. Differently from previous experiment, we rigidly constrained four nodes. The initial configuration is a cylinder [see Fig. 1(a)]. The mesh is tested with a piecewise function properly scaled and shifted in the form of

$$f(x, y, t) = \begin{cases} x^2 - y^2, & t \leq 10 \\ x^2, & 10 < t \leq 15 \\ y^2, & 15 < t \leq 25 \\ -y^2, & t > 25 \end{cases}.$$

Fig. 9 shows 35 s of an ideal simulation without noise. The controller reacts to piecewise function discontinuities emulating four different shapes (see Fig. 10). The *paraboloids* $\pm y^2$ are reproduced with an average orientation error lower than 1° and average position error of 1 mm. Instead, the tracking of the function $x^2 - y^2$ is achieved with 10° and 6-mm error. Differently, the mesh was not able to copy the *paraboloids* x^2, i.e., 20° and 17 mm of error, mainly because the *four* fixed nodes constrained a direction of motion.

D. Discussion

Above, we have shown the performances of the proposed approach in four scenarios, each characterized by a different cover

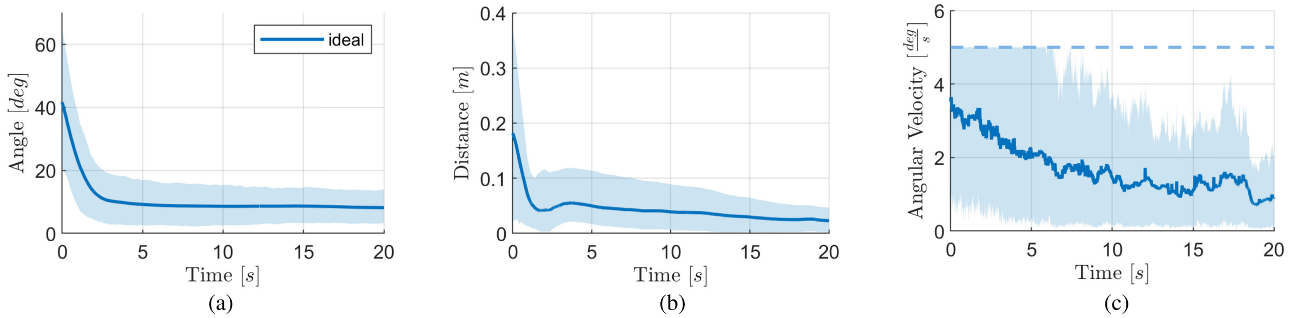


Fig. 8. Performance of an ideal simulation executed with mesh 20 × 20. The solid lines represent the average values, while the transparent regions are the 10–90th percentile range. In (c), the dashed line is the maximum motor speed $\boldsymbol{\omega}^R_{r_{\max}}$. (a) Error alignment normal vectors $\varepsilon_{(i,j)}$. (b) Node position error $\zeta_{(i,j)}$. (c) Amplitude motor speed $|\boldsymbol{\omega}^R_{r_{\text{act}}}|$.

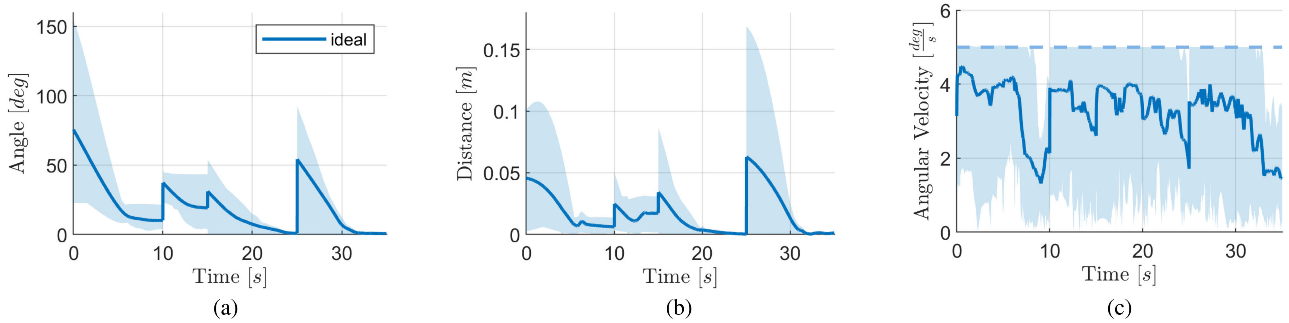


Fig. 9. Performance of an ideal simulation executed with mesh 4 × 8. The solid lines represent the average values, while the transparent regions are the 10–90th percentile range. In (c), the dashed line is the maximum motor speed $\boldsymbol{\omega}^R_{r_{\max}}$. (a) Error alignment normal vectors $\varepsilon_{(i,j)}$. (b) Node position error $\zeta_{(i,j)}$. (c) Amplitude motor speed $|\boldsymbol{\omega}^R_{r_{\text{act}}}|$.

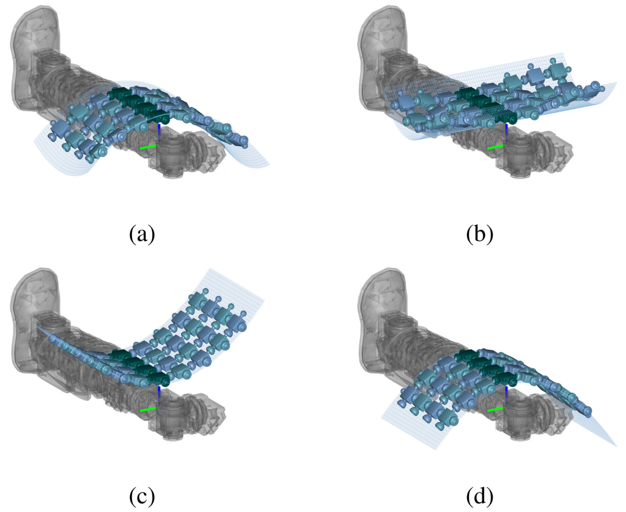


Fig. 10. Instants of the 4 × 8 cover movement discussed in Section VI-C4. The four green nodes are rigidly fixed to the leg. The light blue surface is the desired shape. The simulation video is attached in the accompanying material at https://youtu.be/kMfXb2xqGn4?t=482. (a) $t = 10$ s. (b) $t = 15$ s. (c) $t = 25$ s. (d) $t = 35$ s.

skeleton size, desired shape, and noise source. The obtained results are satisfactory since the cover tracks a desired time-varying surface with a node-normal alignment error less than 10° for most of the analyzed shapes. The presented simulations tend to show that the node-normal alignment task allows us to indirectly minimize the node position errors to a value lower than the node square length l (25 mm) in most cases. Let us remark, however, that the control laws obtained as solutions to (22a) do not, in general, imply a minimization of the node position errors.

The control problem formulated in Section V moves the skeleton to emulate a continuous function without specifying the desired node coordinates or where the mesh should lie. On the one hand, this simplifies the control process since it is not required to tessellate the desired shape and estimate the node coordinates. On the other hand, the tracking of a complex desired shape with multiple curvatures could be hard to achieve since the mesh might emulate the desired shape in a coordinate range different from the expected one. With the current control strategy, we can only select the fixed-node coordinates, but the skeleton might rotate around the fixed node for large meshes. This behavior slightly appeared with a 20 × 20 mesh (see the accompanying movie for the simulation video), where the skeleton started to “slide” on the desired shape pushed by the minimization of the node-normal alignment. A desired surface parameterization could be beneficial discouraging mesh movement along unwanted directions.

The desired shape selection is constrained by physical limitation of the mesh, like the range of motion limits or the position of fixed nodes. The desired shape must belong to a neighborhood of the mesh configuration; otherwise, the controller cannot ensure convergence, unless a trajectory planning layer is introduced. The mathematical definition of the desired shape is also another drawback since the control problem is only able to track mathematical functions, i.e., binary relations between *domain* and *codomain* that relate each element of the *domain* to exactly one element of the *codomain*. Consequently, owing to a representation reason, the mesh is not able to morph into *nonsurjective* functions.

Concerning the motor positioning algorithm, its result is an optimal solution in a configuration close to the initial one. However the full actuation is not guaranteed in the whole working space. This geometric dependence does not affect the associated performances since we can assume to work in a configuration near the starting one in most cases. In fact, in all the presented simulations, the meshes did not experience underactuation and singularity issues.

VII. Conclusion

This article proposed a solution for providing humanoid robots with adaptive morphology capabilities. We presented a novel type of humanoid robot covers whose shape can be actively controlled. The proposed morphing cover concept is composed of a skeleton and an elastic membrane. We showed how the skeleton can be obtained as a repetition of basic elements connected by spherical joints, which thus create a parallel mechanism. The design is particularly robust from the mechanical perspective. To validate its performance, we carried out extensive simulations with the objective for the cover to morph into desired shapes. The mesh evolution is governed by a kinematic controller that orients the node-normals and indirectly minimizes the node-position errors. This article also proposed a motor placement strategy aiming to minimize the number of motors without losing *full actuation*. The optimality is assured in a local configuration; however, during the presented simulations, the meshes did not experience underactuation issues.

The choice of spherical joints allows a wide range of movements. However, it increases the complexity of the mechanical design. Other limitations are the unmodeled external and self-collisions, and the absence of a dynamic model. The future work consists in testing other mechanisms to understand if we can decrease the number of DoFs without interfering on the mesh performances. For example, we can pass from spherical to universal joints adding further constraints in the kinematic model. Later, we intend to perform simulations with a dynamical model to estimate motor torques: they can be used in the presented GA to further improve the actuation position scheme. Finally, we intend to actuate the cover and perform real experiments. As for the *soft external membrane*, the design and the integration is planned to start as soon as we produced a fully working rigid skeleton.

Multimedia Material

The code of this project is available at https://bit.ly/morph2022. Additional results can be viewed in this video: https://youtu.be/kMfXb2xqGn4.

Appendix

Proof of Corollary 1: The matrix $\mathcal{Z}_{\boldsymbol{\nu}} \in \mathbb{R}^{\text{DoF}\times 6+3N_{\text{sj}}}$ contains the basis of the space of the feasible movements; it is full rank and has more rows than columns. Assume that the indexes of the linear independent rows are known and the nonsingular matrix $\mathcal{Z}_{\text{act}}$ has been constructed. Using a permutation matrix $P_{\mathcal{Z}}$ to change the order of the rows, we have

$$\mathcal{Z}_{\boldsymbol{\nu}} = P_{\mathcal{Z}} \begin{bmatrix} \mathcal{Z}_{\text{act}} \\ \mathcal{Z}_0 \end{bmatrix}. \tag{26}$$

Given the generic solution $\boldsymbol{\nu} = \mathcal{Z}_{\boldsymbol{\nu}} \boldsymbol{\xi}$, doing a change of variable $\boldsymbol{\xi} = \mathcal{Z}_{\text{act}}^{-1} \boldsymbol{\psi}$ and substituting (26), we obtain

$$\boldsymbol{\nu} = P_{\mathcal{Z}} \begin{bmatrix} I_{\text{DoF}} \\ \mathcal{Z}_0 \mathcal{Z}_{\text{act}}^{-1} \end{bmatrix} \boldsymbol{\psi} = P_{\mathcal{Z}} \begin{bmatrix} \boldsymbol{\psi} \\ \mathcal{Z}_0 \mathcal{Z}_{\text{act}}^{-1} \boldsymbol{\psi} \end{bmatrix}. \tag{27}$$

Imposing the values of $\boldsymbol{\psi}$ is equivalent to control the relative angular velocity of some joints that are also decoupled. If we imagine to actuate those joints, we can then fully control our mesh, in a range near to this configuration, with a number of motors equal to DoFs. The term $\boldsymbol{\psi}$ is called $\boldsymbol{\omega}^R_{r_{\text{act}}}$ in the remainder of this article (see Definition 4). □

Proof of Lemma 1: Given (27), substituting $\boldsymbol{\omega}^R_{r_{\text{act}}}$ instead of $\boldsymbol{\psi}$, we can prove (21a)

$$\boldsymbol{\nu} = \mathcal{Z}_{\boldsymbol{\nu}} \mathcal{Z}_{\text{act}}^{-1} \boldsymbol{\psi} = \mathcal{Z}_{\boldsymbol{\nu}} \mathcal{Z}_{\text{act}}^{-1} \boldsymbol{\omega}^R_{r_{\text{act}}}. \tag{28}$$

To proof (21b), we need to recall $J_c \boldsymbol{v} = \mathbf{0}$ and its generic solution $\boldsymbol{v} = \mathcal{Z}_{\boldsymbol{v}} \boldsymbol{\xi}$. Substituting the latter into (16), we obtain

$$\boldsymbol{\nu} = \mathcal{M}^{\boldsymbol{\nu}}_{\boldsymbol{v}} \boldsymbol{v} = \mathcal{M}^{\boldsymbol{\nu}}_{\boldsymbol{v}} \mathcal{Z}_{\boldsymbol{v}} \boldsymbol{\xi}. \tag{29}$$

Given (17), we can rewrite (29) as $\boldsymbol{\nu} = \mathcal{Z}_{\boldsymbol{\nu}} \boldsymbol{\xi}$. Both the absolute and relative cover velocities $\boldsymbol{\nu}$ and $\boldsymbol{v}$ are written as function of the same vector $\boldsymbol{\xi}$. Consequently, as done in (27) and (28), doing a change of variable $\boldsymbol{\xi} = \mathcal{Z}_{\text{act}}^{-1} \boldsymbol{\omega}^R_{r_{\text{act}}}$ in the equation $\boldsymbol{v} = \mathcal{Z}_{\boldsymbol{v}} \boldsymbol{\xi}$, we prove (21b)

$$\boldsymbol{v} = \mathcal{Z}_{\boldsymbol{v}} \mathcal{Z}_{\text{act}}^{-1} \boldsymbol{\omega}^R_{r_{\text{act}}}. \tag{30}$$

□

References

[1] C. Pennycuick, "The feathered wings of birds," in *Modelling the Flying Bird (Theoretical Ecology Series)*, vol. 5. New York, NY, USA: Academic, 2008, ch. 5, pp. 105–134.

[2] D. Holden, J. J. Socha, N. D. Cardwell, and P. P. Vlachos, "Aerodynamics of the flying snake Chrysopelea paradisi: How a bluff body cross-sectional shape contributes to gliding performance," *J. Exp. Biol.*, vol. 217, no. 3, pp. 382–394, 2014.

[3] J. M. Guayasamin, T. Krynak, K. Krynak, J. Culebras, and C. R. Hutter, "Phenotypic plasticity raises questions for taxonomically important traits: A remarkable new andean rainfrog (Pristimantis) with the ability to change skin texture," *Zool. J. Linnean Soc.*, vol. 173, no. 4, pp. 913–928, 2015.

[4] S. B. Emerson and M. A. R. Koehl, "The interaction of behavioral and morphological change in the evolution of a novel locomotor type: "flying" frogs," *Evolution*, vol. 44, no. 8, pp. 1931–1946, 1990.

[5] M. García-París and S. M. Deban, "A novel antipredator mechanism in salamanders: Rolling escape in hydromantes platycephalus," *J. Herpetol.*, vol. 29, no. 1, pp. 149–151, 1995.

[6] S. Mintchev and D. Floreano, "Adaptive morphology: A design principle for multimodal and multifunctional robots," *IEEE Robot. Autom. Mag.*, vol. 23, no. 3, pp. 42–54, Sep. 2016.

[7] D. Falanga, K. Kleber, S. Mintchev, D. Floreano, and D. Scaramuzza, "The foldable drone: A morphing quadrotor that can squeeze and fly," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 209–216, Apr. 2019.

[8] D. S. Shah, M. C. Yuen, L. G. Tilton, E. J. Yang, and R. Kramer-Bottiglio, "Morphing robots using robotic skins that sculpt clay," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2204–2211, Apr. 2019.

[9] D. Shah, B. Yang, S. Kriegman, M. Levin, J. Bongard, and R. Kramer-Bottiglio, "Shape changing robots: Bioinspiration, simulation, and physical realization," *Adv. Mater.*, vol. 33, 2020, Art. no. 2002882.

[10] L. Natale, C. Bartolozzi, D. Pucci, A. Wykowska, and G. Metta, "iCub: The not-yet-finished story of building a robot child," *Sci. Robot.*, vol. 2, no. 13, 2017, Art. no. eaaq1026.

[11] E. Guizzo, "By leaps and bounds: An exclusive look at how Boston dynamics is redefining robot agility," *IEEE Spectr.*, vol. 56, no. 12, pp. 34–39, Dec. 2019.

[12] S. Shigemi, "ASIMO and humanoid robot research at Honda," in Humanoid Robotics: A Reference. Dordrecht, The Netherlands: Springer, 2019, pp. 55–90.

[13] S. Saeedvand, M. Jafari, H. S. Aghdasi, and J. Baltes, "A comprehensive survey on humanoid robot development," *Knowl. Eng. Rev.*, vol. 34, pp. 1–8, 2019.

[14] D. Pucci, S. Traversaro, and F. Nori, "Momentum control of an underactuated flying humanoid robot," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 195–202, Jan. 2018.

[15] G. Nava, L. Fiorio, S. Traversaro, and D. Pucci, "Position and attitude control of an underactuated flying humanoid robot," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots*, 2018, pp. 1–9.

[16] H. A. O. Mohamed *et al.*, "Momentum-based extended Kalman filter for thrust estimation on flying multibody robots," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 526–533, Jan. 2022.

[17] R. Tajima, S. Kagami, M. Inaba, and H. Inoue, "Development of soft and distributed tactile sensors and the application to a humanoid robot," *Adv. Robot.*, vol. 16, no. 4, pp. 381–397, 2002.

[18] M. Hakozaki, K. Nakamura, and H. Shinoda, "Telemetric artificial skin for soft robot," in *Proc. Int. Conf. Solid State Sens. Actuators*, 1999, pp. 844–847.

[19] M. Hayashi, T. Yoshikai, and M. Inaba, "Development of a humanoid with distributed multi-axis deformation sense with full-body soft plastic foam cover as flesh of a robot," in *Sensors: Focus on Tactile Force and Stress Sensors*. Rijeka, Croatia: InTech, 2008, pp. 319–324.

[20] A. Stilli, L. Grattarola, H. Feldmann, H. A. Wurdemann, and K. Althoefer, "Variable stiffness link (VSL): Toward inherently safe robotic manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 4971–4976.

[21] R. J. Lang, "A computational algorithm for origami design," in *Proc. 12th Annu. Symp. Comput. Geometry*, 1996, pp. 98–105.

[22] D. Rus and M. T. Tolley, "Design, fabrication and control of origami robots," *Nature Rev. Mater.*, vol. 3, no. 6, pp. 101–112, 2018.

[23] J. T. B. Overvelde *et al.*, "A three-dimensional actuated origami-inspired transformable metamaterial with multiple degrees of freedom," *Nature Commun.*, vol. 7, no. 1, 2016, Art. no. 10929.

[24] W. Kim *et al.*, "Bioinspired dual-morphing stretchable origami," *Sci. Robot.*, vol. 4, no. 36, 2019, Art. no. eaay3493.

[25] E. V. Hoff, D. Jeong, and K. Lee, "OrigamiBot-I: A thread-actuated origami robot for manipulation and locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1421–1426.

[26] B. An and D. Rus, "Designing and programming self-folding sheets," *Robot. Auton. Syst.*, vol. 62, no. 7, pp. 976–1001, 2014.

[27] S. Miyashita, S. Guitron, M. Ludersdorfer, C. R. Sung, and D. Rus, "An untethered miniature origami robot that self-folds, walks, swims, and degrades," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1490–1496.

[28] M. Salerno, J. Paik, and S. Mintchev, "Ori-pixel, a multi-DoFs origami pixel for modular reconfigurable surfaces," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6988–6995, Oct. 2020.

[29] M. K. Kristensen and C. R. Jepsen, "Flexible mat for providing a dynamically reconfigurable double-curved moulding surface in a mould," U.S. Patent 9 168 678, Oct. 27, 2015.

[30] *TAS Developing a Reconfigurable Antenna*, Advanced Research in Telecommunications Systems, European Space Agency, Paris, France, 2016. [Online]. Available: https://artes.esa.int/news/tas-developing-reconfigurable-antenna

[31] *Wavehandling*, Festo Corporate, Mason, OH, USA, 2014. [Online]. Available: https://www.festo.com/group/en/cms/10225.htm

[32] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii, "inFORM: Dynamic physical affordances and constraints through shape and object actuation," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol., Assoc. Comput. Mach.*, 2013, pp. 417–426.

[33] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[34] C. Lee *et al.*, "Soft robot review," *Int. J. Control, Autom. Syst.*, vol. 15, no. 1, pp. 3–15, 2017.

[35] T. Wallin, J. Pikul, and R. Shepherd, "3D printing of soft robotic systems," *Nature Rev. Mater.*, vol. 3, no. 6, pp. 84–100, 2018.

[36] Y. Funabora, "Prototype of a fabric actuator with multiple thin artificial muscles for wearable assistive devices," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, 2017, pp. 356–361.

[37] T. Hiramitsu, K. Suzumori, H. Nabae, and G. Endo, "Experimental evaluation of textile mechanisms made of artificial muscles," in *Proc. 2nd IEEE Int. Conf. Soft Robot.*, 2019, pp. 1–6.

[38] Y. Sun, J. Guo, T. M. Miller-Jackson, X. Liang, M. H. Ang, and R. C. H. Yeow, "Design and fabrication of a shape-morphing soft pneumatic actuator: Soft robotic pad," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6214–6220.

[39] J. Kim, J. A. Hanna, M. Byun, C. D. Santangelo, and R. C. Hayward, "Designing responsive buckled surfaces by halftone gel lithography," *Science*, vol. 335, no. 6073, pp. 1201–1205, 2012.

[40] Y. Tao *et al.*, "Morphing pasta and beyond," *Sci. Adv.*, vol. 7, no. 19, 2021, Art. no. eabf4098.

[41] J. Ou *et al.*, "AeroMorph—heat-sealing inflatable shape-change materials for interaction design," in *Proc. 29th Annu. Symp. User Interface Softw. Technol.*, 2016, pp. 121–132.

[42] S. Daynes, A. Grisdale, A. Seddon, and R. Trask, "Morphing structures using soft polymers for active deployment," *Smart Mater. Struct.*, vol. 23, no. 1, Dec. 2013, Art. no. 012001.

[43] J. H. Pikul, S. Li, H. Bai, R. T. Hanlon, I. Cohen, and R. F. Shepherd, "Stretchable surfaces with programmable 3D texture morphing for synthetic camouflaging skins," *Science*, vol. 358, no. 6360, pp. 210–214, 2017.

[44] L. Yao, R. Niiyama, J. Ou, S. Follmer, C. D. Silva, and H. Ishii, "PneUI: Pneumatically actuated soft composite materials for shape changing interfaces," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, 2013, pp. 13–22.

[45] M. Ishida, D. Drotman, B. Shih, M. Hermes, M. Luhar, and M. T. Tolley, "Morphing structure for changing hydrodynamic characteristics of a soft underwater walking robot," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4163–4169, Oct. 2019.

[46] K. Liu, F. Hacker, and C. Daraio, "Robotic surfaces with reversible, spatiotemporal control for shape morphing and object manipulation," *Sci. Robot.*, vol. 6, no. 53, 2021, Art. no. eabf5116.

[47] X. Kuang *et al.*, "Advances in 4D printing: Materials and applications," *Adv. Funct. Mater.*, vol. 29, no. 2, 2019, Art. no. 1805290.

[48] D. Raviv *et al.*, "Active printed materials for complex self-evolving deformations," *Sci. Rep.*, vol. 4, no. 1, 2014, Art. no. 7422.

[49] J. W. Boley *et al.*, "Shape-shifting structured lattices via multimaterial 4D printing," *Proc. Nat. Acad. Sci.*, vol. 116, no. 42, pp. 20856–20862, 2019.

[50] Y. Kim, H. Yuk, R. Zhao, S. A. Chester, and X. Zhao, "Printing ferromagnetic domains for untethered fast-transforming soft materials," *Nature*, vol. 558, no. 7709, pp. 274–279, 2018.

[51] A. Sofla, S. Meguid, K. Tan, and W. Yeo, "Shape morphing of aircraft wing: Status and challenges," *Mater. Des.*, vol. 31, no. 3, pp. 1284–1292, 2010.

[52] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman, "A review of morphing aircraft," *J. Intell. Mater. Syst. Struct.*, vol. 22, no. 9, pp. 823–877, 2011.

[53] C. Thill, J. Etches, I. Bond, K. Potter, and P. Weaver, "Morphing skins," *Aeronaut. J.*, vol. 112, no. 1129, pp. 117–139, 2008.

[54] D. Neal, M. Good, C. Johnston, H. Robertshaw, W. Mason, and D. Inman, "Design and wind-tunnel analysis of a fully adaptive aircraft configuration," in *Proc. AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, 2004, Art. no. AIAA 2004-1727.

[55] J. Flanagan, R. Strutzenberg, R. Myers, and J. Rodrian, "Development and flight testing of a morphing aircraft, The NextGen MFX-1," in *Proc. AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, 2007, Art. no. AIAA 2007-1730.

[56] J. J. Joo, B. Sanders, T. Johnson, and M. I. Frecker, "Optimal actuator location within a morphing wing scissor mechanism configuration," *Proc. SPIE*, vol. 6166, 2006, Art. no. 616603.

[57] R. Vos, R. Barrett, R. de Breuker, and P. Tiso, "Post-buckled precompressed elements: A new class of control actuators for morphing wing UAVs," *Smart Mater. Struct.*, vol. 16, no. 3, pp. 919–926, May 2007.

[58] O. Bilgen *et al.*, "A novel unmanned aircraft with solid-state control surfaces: Analysis and flight demonstration," *J. Intell. Mater. Syst. Struct.*, vol. 24, no. 2, pp. 147–167, 2013.

[59] M. Yim *et al.*, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 43–52, Mar. 2007.

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

[61] C.-H. Yu, K. Haller, D. Ingber, and R. Nagpal, "Morpho: A self-deformable modular robot inspired by cellular structure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3571–3578.

[62] C. H. Belke and J. Paik, "Mori: A modular origami robot," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 5, pp. 2153–2164, Oct. 2017.

[63] C. H. Belke and J. Paik, "Automatic couplings with mechanical overload protection for modular robots," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 3, pp. 1420–1426, Jun. 2019.

[64] M. Yao, C. H. Belke, H. Cui, and J. Paik, "A reconfiguration strategy for modular robots using origami folding," *Int. J. Robot. Res.*, vol. 38, no. 1, pp. 73–89, 2019.

[65] M. Yao, X. Xiao, C. H. Belke, H. Cui, and J. Paik, "Optimal distribution of active modules in reconfiguration planning of modular robots," *J. Mech. Robot.*, vol. 11, no. 1, 2018, Art. no. 011017.

[66] D. E. Goldberg and J. H. Holland, *Genetic Algorithms and Machine Learning*. Norwell, MA, USA: Kluwer, 1988.

[67] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[68] J. Zhong, X. Hu, J. Zhang, and M. Gu, "Comparison of performance between different selection strategies on simple genetic algorithms," in *Proc. Int. Conf. Comput. Intell. Model., Control Autom./Int. Conf. Intell. Agents, Web Technol. Internet Commerce*, 2005, vol. 2, pp. 1115–1121.

[69] W. M. Spears and V. Anand, "A study of crossover operators in genetic programming," in *Methodologies for Intelligent Systems*, Z. W. Ras and M. Zemankova, Eds. Berlin, Germany: Springer, 1991, pp. 409–418.

[70] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016.

[71] D. Pucci, T. Hamel, P. Morin, and C. Samson, "Nonlinear feedback control of axisymmetric aerial vehicles," *Automatica*, vol. 53, pp. 72–78, 2015.

[72] S. Dafarra, G. Romualdi, G. Metta, and D. Pucci, "Whole-body walking generation using contact parametrization: A non-linear trajectory optimization approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1511–1517.

[73] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Comput.*, vol. 12, pp. 637–672, 2020.

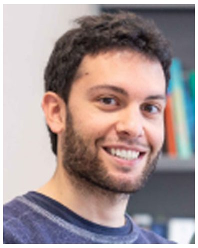
[74] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi—a software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

[75] S. Gros, M. Zanon, and M. Diehl, "Baumgarte stabilisation over the SO(3) rotation group for control," in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 620–625.

[76] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach," *Information*, vol. 10, no. 12, Dec. 2019, Art. no. 390.

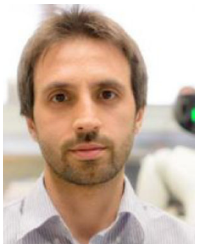
Fabio Bergonti received the bachelor's (Hons.) and master's (Hons.) degrees in mechanical engineering from the Politecnico di Milano, Milan, Italy, in 2016 and 2018, respectively. He is currently working toward the Ph.D. degree in computer science with the University of Manchester, Manchester, U.K., working at the Italian Institute of Technology, Genova, Italy, under the supervision of Daniele Pucci and Angelo Cangelosi.

He conducted his master thesis at the Dynamic Interaction and Control Laboratory, Italian Institute of Technology, focusing on the development of whole-body control algorithms to generate and control jumps of humanoid robots. From 2018 to 2019, he was a Research Fellow with the Dynamic Interaction and Control Laboratory, Italian Institute of Technology, where he then joined the iRonCub Group, a research group that aims at creating the first flying humanoid robot.



Gabriele Nava received the bachelor's and master's degrees in mechanical engineering from the Politecnico di Milano, Milan, Italy, in 2013 and 2015, respectively, and the Ph.D. degree in bioengineering and robotics from the Università degli Studi di Genova, Genoa, Italy, in 2020, in cooperation with the Italian Institute of Technology (IIT), Genoa, under the supervision of Daniele Pucci and Giorgio Metta.

He is currently a Postdoctoral Researcher with the Artificial and Mechanical Intelligence Laboratory, IIT. His main research interests include the design of control algorithms for floating base systems, with a focus on aerial humanoid robotics. He is the Scrum Master of the iRonCub Group (IIT), which pursues the objective of making the humanoid robot iCub fly.



Luca Fiorio received the bachelor's and master's degrees in mechanical engineering from the Politecnico di Torino, Turin, Italy, in 2007 and 2010, respectively, and the Ph.D. degree in bioengineering and robotics from the Italian Institute of Technology, Genoa, Italy, and the University of Genoa, Genoa, in 2015.

He conducted his master thesis with the German Aerospace Center, Munich, Germany, focusing on the simulation and control of a running hexapod robot. Subsequently, as a Postdoctoral Fellow with the Italian Institute of Technology, he worked on the design of the robot R1 and with the Dynamic Interaction Control Laboratory. He is currently a Chief Technician with the iCub Tech Facility, Italian Institute of Technology.

Giuseppe L'Erario received the bachelor's degree in aerospace engineering and master's degree in artificial intelligence and robotics from the Sapienza University of Rome, Rome, Italy, in 2015 and 2019, respectively. He is currently working toward the Ph.D. degree in computer science with the University of Manchester, Manchester, U.K., working with the iRonCub Group, Italian Institute of Technology, Genoa, Italy, under the supervision of Daniele Pucci and Angelo Cangelosi.

He conducted his master thesis with the Dynamic Interaction and Control Lab, IIT, focusing on the modeling, identification, and control of small-scale jet engines, in the context of the iRonCub project, whose aim is to create the first jet-powered flying humanoid robot.

Daniele Pucci received the bachelor's (Hons.) and master's (Hons.) degrees in control engineering from the Sapienza University of Rome, Rome, Italy, in 2007 and 2009, respectively, and the Ph.D. degree in nonlinear control applied to flight dynamics from INRIA Sophia Antipolis, Biot, France, in 2013, under the supervision of Tarek Hamel and Claude Samson.

From 2013 to 2017, he was a Postdoctoral Researcher with the Istituto Italiano di Tecnologia (IIT), Genoa, working within the EU project CoDyCo. Since August 2017, he has been the Head of the Artificial and Mechanical Intelligence Laboratory, IIT. This laboratory focuses on the humanoid robot locomotion problem, with specific attention on the control and planning of the associated nonlinear systems. Also, the laboratory is pioneering aerial humanoid robotics, whose main aim is to make flying humanoid robots. Currently, the laboratory is implementing iRonCub, the jet-powered flying version of the humanoid robot iCub. He is also the scientific principal investigator of the H2020 European Project AnDy, a task leader of the H2020 European Project SoftManBot, and a coordinator of the joint laboratory of IIT and Honda JP. Since 2020, and in the context of the split site Ph.D. supervision program, he has been a visiting Lecturer with the University of Manchester, Manchester, U.K.

Dr. Pucci received the "Academic Excellence Award" from the Sapienza University of Rome in 2009. In 2019, he was awarded as Innovator of the Year under 35 Europe from *MIT Technology Review*.