Robotica (2022), **40**, pp. 213–233
doi:10.1017/S0263574721000473

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Design, modeling, and constraint-compliant control of an autonomous morphing surface for omnidirectional object conveyance

Ioannis A. Raptis[1*] Christopher Hansen[2] and Martin A. Sinclair[2]

[1]Autonomous Robotic Systems Laboratory, Department of Electrical and Computer Engineering, North Carolina A&T State University, Greensboro, North Carolina 27411, USA and [2]Department of Mechanical Engineering, University of Massachusetts Lowell, Lowell, Massachusetts 01854, USA
*Corresponding author. Email: iraptis@ncat.edu

Received: 18 August 2020; **Revised:** 31 March 2021; **Accepted:** 1 April 2021; **First published online:** 5 May 2021

Keywords: Distributed manipulation, actuator networks, large-scale spatial actuation, control of robotic systems, force control, novel applications of robotics.

Abstract

In this paper, we conceptualize, analyze, and assemble a prototype adaptive surface system capable of morphing its geometric configuration using an array of linear actuators to impose omnidirectional movement of objects that lie on the surface. The principal focus and contribution of this paper is the derivation of feedback control protocols–for regulating the actuators' length in order to accomplish the object conveyance task–that scale with the number of actuators and the nonlinear kinematic constraints of the morphing surface. Simulations and experimental results demonstrate the advantages of distributed manipulation over static-shaped feeders.

1. Introduction

The term Large-Scale Actuator Networks (LSANs) is used to designate robotic systems comprised of interconnected and spatially distributed, simplistic actuators. Their actuation is autonomously coordinated to achieve global objectives that exceed the capabilities of the individual actuating elements. The underlying promise of LSAN systems is to obviate the traditional use of complex and costly monolithic robotic manipulators devoted to a single task or function. The actuating network operates in a cooperative fashion that is capable of maintaining performance even in the event of the failure of individual elements, thereby increasing the robustness and scalability of the network.

LSANs have direct application to the field of distributed manipulation, which applies to object handling using a large number of individual manipulators. Exemplar scalable distributed manipulators have used multiple types of actuators; these individual components are typically simplistic actuating components, that is, air jets, vibrating plates, or rotary rollers. Progress in Micro-Electro-Mechanical Systems (MEMS) fabrication has led to the inexpensive manufacture of microscale manipulators that possess high areal density of actuating elements within small regions. This high manipulator density results in improved actuation resolution [1, 2, 3, 4].

The field of distributed manipulation has many impressive examples showing the rapid progress leading to the current state of the art. A typical example is vibrating plate-based part feeders reported

This work is the complete version of the synoptic conference proceedings paper [5]. This full paper includes additional sections, clarifying remarks, and results compared to its conference counterpart.

in refs. [6, 7, 8, 9, 10]. Vibratory feeders utilize inclined vibrations to transport parts along a track. In refs. [6, 7] parallel manipulation of multiple parts (translation and orientation) is achieved with a single vibrating plate. Programmable vector techniques for vibratory part feeders and MEMS actuator arrays have been expounded upon in refs. [8, 11, 12]. These works present an analytic derivation of an artificial force field method in order to transport and rotate planar objects to pre-defined locations; the artificial force field is implemented using a massive number of microscopic actuators ("motion pixels") to apply frictional forces to the planar objects. Microscale manipulations using MEMS microactuators are further investigated in refs. [13, 14]. These systems implement a very large number of actuators to generate even small motions. The modular "Polybot" system is an example that uses out-of-plane, cilia-like manipulators to convey nonplanar items [15]. A survey on coupling mechanism for reconfigurable robotic systems is given in ref. [16].

Air jets are another widely employed method to direct object transport, as shown in refs. [29, 30, 31, 32, 15, 33, 34]. Air-jet conveyance is a preferred means to interact with objects that are subject to damage by refs. [15, 34, 35] or contamination from ref. [33] traditional part feeders. The air-jet actuators utilize both static and dynamic potential flow fields [31, 32]. The implementation of sinks to control a single object in a fixed plane is described in ref. [29].

Conveyor belts are a commonplace industrial method to transport objects to a target destination. The system detailed in ref. [36] uses a single joint robot to move and orient parts in a speed conveyor belt. While conveyor systems are cost-effective and widely implemented, they are typically restricted to a limited set of predetermined exit locations. Distributed manipulation using an array of wheels was used to exert forces on an object in ref. [37]. Investigation of the implementation and development of an array of soft and compliant actuators to move fragile objects is shown in ref. [38]. A surface containing protrusions to actuate via frictional contacts is used in ref. [39] to direct part feeding. Additional distributed manipulable surfaces are considered in ref. [40] with a focus on maintaining a level platform despite changing ground conditions. The surface in ref. [41] changes its shape to drive objects on a single axis. The use of a morphing surface as a means to display 3D objects is shown in ref. [42]. Self-organized systems of actuator networks, with a focus on controlling networks of up to 10,000 nodes, are investigated in ref. [10].

Another transformative application of LSANs is the emerging field of soft robotics [43, 44], which refers to continuum robotic manipulators made of soft materials that are subjected to elastic deformation to produce backbone curved bending motion [45]. The actuation and sensorization of "soft matter" for autonomous grasping, or locomotion, significantly enlarged the design parameter space, yielding new challenges in modeling and control of robotic manipulators. The derivation of analytically attractive models and the design of feedback control laws, which scale gracefully with the complexity of the soft manipulator and are compliant with its kinematic limitations, is still an open challenge in the community of soft robotics. An up-and-coming class of "robotic materials" are reconfigurable intelligent surfaces [46]; namely, materials that reconfigure under the influence of electromagnetic waves.

An additional example of innovative–and distinctive–large-scale actuation is multi-UAV aerial manipulation [17, 18], which is an elaborate extension of the earlier object handling problem with collectives of mobile ground robots [19, 21, 20, 22]. A survey paper for multi-robot coordination is written by Yan *et al*. [25] and for multi-robot object transport by Tuci *et al*. [26]. Grasping with robotic hands also qualifies as multi-actuated manipulation [23, 24]. Interestingly, the Stewart-Gough platform [27], conceived to simulate flight conditions for the training of helicopter pilots, classifies as an example of a constrained multi-actuated mechanism, where the control law accounts for both the multi-actuation and nontrivial kinematics. In ref. [28], a compelling connection is made between the dexterity and the workspace of serial manipulators.

Hitherto, we have reasoned that large-scale actuation appears in various forms and an assortment of applications. The common characteristic of the foregoing body of work involves robotic systems with a large number of actuators and, typically, kinematics with nontrivial constraints. Such systems' mechanics and control requirements are distinguishably different from conventional industrial robotic

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

manipulators forming an open kinematic chain. The common need of such multi-actuated robotic systems is low-complexity control laws that scale (preferably linearly) with the order of actuation and, at the same time, are sympathetic to the mechanics of the configuration to realize–for example–desired motion, locomotion, or object handling. Due to the high variability of configurations of contactual robotic systems that exhibit large-scale actuation, in this paper, we narrow down our focal point to a well-defined conceptualization of an LSAN from which basic theory can be established. The motivation behind this work is the manifestation of a comprehensive analytical framework that can display–in a concise, methodological fashion–the benefits and challenges of large-scale actuation for distributed manipulation.

Herein, we introduce a morphing surface that autonomously adjusts its topography to convey an arbitrary number of objects to a variable reference location. The surface shape reconfiguration is controlled by a grid of vertical linear actuators that adjust its vertical position at specific coordinates in two-dimensional (2D) Cartesian space. The multi-actuator grid results in a mesh of interconnected rectangular flat cells that can adjust their local inclination. This inclination transports the objects that lie on top of the cell by regulating their weight components.

The surface kinematics and object dynamics are detailed and analyzed. The surface constraint equations are explicitly derived and the number of available independent control inputs associated with the system is determined. In addition, the objects' equations of motion reveal how the multi-actuator input controls a multitask process governed by continuous dynamics. This analysis is used to design control schemes that are sympathetic to the physical constraints of the system. The primary novelty of this approach is the creation of control algorithms that are computationally efficient, of low complexity, conform to the physical limitations of the actuators, and scale to systems comprising a substantial quantity of actuating elements. To validate the applicability of this approach, a prototype system is developed. The prototype demonstrates how simplistic and computationally attractive control algorithms combined with minimalistic hardware can be used to transport objects to multiple directions of the workspace. Such control algorithms could be applied to automated warehouses or assembly lines.

This paper is organized into eight sections: Section 2 details the mathematical representation of the morphing surface. Section 3 contains the kinematic analysis of a single cell, and outlines a preliminary control law for transporting objects over a single cell. The kinematic description of a multicell surface is provided in Section 4. The control algorithms governing the transport of multiple objects over an arbitrary surface are described in Section 5. A description of a prototype surface that was developed to validate the applicability of the proposed mechanism is given in Section 6. Both experimental and simulation results of the single- and multicell surfaces are presented in Section 7. Finally, Section 8 presents concluding remarks.

Mathematical notation

The set of real numbers is denoted by $\mathbb{R}$. The trigonometric functions $\cos(t)$, $\sin(t)$, and $\tan(t)$ are, respectively, abbreviated as C_t, S_t, and T_t – with $t \in \mathbb{R}$. The transpose of a vector or matrix is represented with the superscript T. The operand $||\cdot||$ denotes the Euclidean norm and $|\cdot|$ denotes the $|\cdot|_1$ norm of a vector. The element in the ith row and jth column of a matrix A is denoted by $[A]_{ij}$.

2. System description: Overview

This work considers a system of a 2D surface comprising a cell array and deformable in the third dimension. Every cell in the array is a 2D tetrahedron (parallelogram) whose shape is defined at every time interval by the tip positions of four linear actuators. The actuators are positioned orthogonal to the global inertial plane within an ordered array (grid). To reduce the system model complexity, each cell is constrained to constitute a planar surface. Due to each cell's treatment as a two-dimensional (2D) surface, it is assumed that the seams between adjacent cells always form straight lines and do not arch.

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

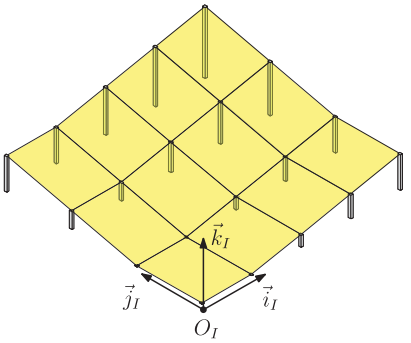

Figure 1. *The definition of the inertial reference frame and its origin for an arbitrary surface S* (n, m).

The position of each grid element (i.e., cell) is uniquely defined by its column and row numbers. The surface $\mathcal{S}$ (n, m) contains a total number of $n \times m$ cells, where n and m represent the number of columns and rows, respectively. For row and column sets $\mathcal{R} = \{1, \ldots, m\}$ and $\mathcal{C} = \{1, \ldots, n\}$, respectively, a cell entry is denoted by $C_{I,J}$ and is positioned at the $I \in \mathcal{R}$ row and $J \in \mathcal{C}$ column of the array.

The origin of the inertial reference frame is based in the cell $C_{1,1}$ and located at the lower left actuator when retracted to its minimum position. The actuator extension to its maximum length is designated as l. The inertial frame remains stationary with respect to the origin even as the actuators retract or expand their length. The inertial frame, as shown in Fig. 1, is defined by its origin O_I and three orthonormal vectors, such that $\mathcal{F}_I = \{O_I, \vec{i}_I, \vec{j}_I, \vec{k}_I\}$. The vectors $\vec{i}_I$ and $\vec{j}_I$ are co-planar with the surface, while $\vec{k}_I$ points upward such that $\{\vec{i}_I, \vec{j}_I, \vec{k}_I\}$ constitutes a right-handed Cartesian coordinate frame ($\vec{k}_I = \vec{i}_I \times \vec{j}_I$).

Let W and L be, respectively, the fixed distances between two adjacent actuators in the $\vec{i}_I$ (horizontal) and $\vec{j}_I$ (vertical) direction of the inertial plane $[\vec{i}_I, \vec{j}_I]$. The coordinates of the four corners where the actuators contact the cell $C_{I,J}$ are

$$P_1^{I,J} = \begin{bmatrix} (I-1)\cdot W \\ (J-1)\cdot L \\ Z_1^{I,J} \end{bmatrix}, \qquad P_2^{I,J} = \begin{bmatrix} I\cdot W \\ (J-1)\cdot L \\ Z_2^{I,J} \end{bmatrix},$$
$$P_3^{I,J} = \begin{bmatrix} I\cdot W \\ J\cdot L \\ Z_3^{I,J} \end{bmatrix}, \qquad P_4^{I,J} = \begin{bmatrix} (I-1)\cdot W \\ J\cdot L \\ Z_4^{I,J} \end{bmatrix}. \tag{1}$$

In the above equation, $Z_i^{I,J}$, for $i = 1, \ldots, 4$, represents the extended distance in the $\vec{k}_I$ direction of the inertial frame of the actuator $P_i^{I,J}$. At each time instance during transport of the object, the surface of every cell remains planar, thereby constraining the actuator positions. Given the planar cell constraint, the four actuator contact positions with the cell will satisfy the following determinant:

$$\begin{vmatrix} \left[P_4^{I,J}\right]^T - \left[P_1^{I,J}\right]^T \\ \left[P_4^{I,J}\right]^T - \left[P_2^{I,J}\right]^T \\ \left[P_4^{I,J}\right]^T - \left[P_3^{I,J}\right]^T \end{vmatrix} = 0. \tag{2}$$

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

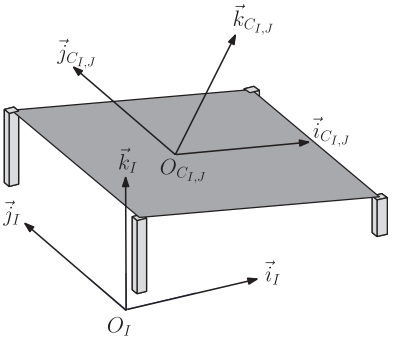

Figure 2. *A schematic illustration of the inertial $\mathcal{F}_I$ and cell-fixed $\mathcal{F}_{C_{I,J}}$ reference frames for a cell configuration.*

The above relation holds for any four points that belong to the same flat plane. The enumeration of each cell's actuator starts from the southwest actuator and continues in a counterclockwise manner. Substitution of the coordinates from (1) into (2) results in the equality

$$Z_3^{I,J} = -Z_1^{I,J} + Z_2^{I,J} + Z_4^{I,J}. \tag{3}$$

This relation explicitly demonstrates that only three actuator positions can be independently chosen, thereby constraining the position of the fourth actuator. In reality, each actuator is activated with a motor that determines its motion response. Each cell surface is defined by four actuator positions; the desired actuator lengths for the cell are commanded by the control signals $Z_{i,com}^{I,J}$. The actuator responses to these commands can be satisfactorily described by the first-order differential equation

$$\tau \dot{Z}_i^{I,J} + Z_i^{I,J} = Z_{i,com}^{I,J}, \text{ with } i = \{1, ..., 4\}, I \in \mathcal{C}, \text{ and } J \in \mathcal{R}; \tag{4}$$

where the time constant for the actuator motor, τ, is a positive value. A large time constant, τ, corresponds to a slow actuator. In order to simplify the subsequent analysis, $Z_i^{I,J}$ will be assumed to be equal to $Z_{i,com}^{I,J}$, such that the actuator dynamics can be neglected.

3. Single cell kinematics and control

In this section, we derive the kinematic equations of a single cell that changes its orientation in space, forming a flat plane, and the dynamic equations of an object that transports on top of a cell, as a result of the inclination of the cell. We also show how the actuator extensions, which determine the inclination of the cell, should be dynamically configured, based on a feedback control law, to move an object at a predefined reference location of the cell's plane.

3.1. Cell kinematics

The objective of this subsection is to detail the mapping between the orientation of each cell and the coordinates of its four actuators' tip points. At a later point, this map will determine the kinematic constraints of a surface configuration and thereby its available degrees of control freedom. The kinematic equations of each cell are derived using an auxiliary, cell-fixed reference frame. This frame of reference, for the arbitrary cell $C_{I,J}$, is denoted $\mathcal{F}_{C_{I,J}} = \{O_{C_{I,J}}, \vec{i}_{C_{I,J}}, \vec{j}_{C_{I,J}}, \vec{k}_{C_{I,J}}\}$ with its origin $O_{C_{I,J}}$ located at the geometric center of cell $C_{I,J}$. The vectors $\vec{i}_{C_{I,J}}, \vec{j}_{C_{I,J}}$ are orthonormal and lie in the plane of the cell, and their direction changes with time to maintain their orthogonality, while the basis vector $\vec{k}_{C_{I,J}}$ is constantly normal to the plane of the cell. The frames $\mathcal{F}_I$ and $\mathcal{F}_{C_{I,J}}$ coincide when all the actuators lie in a level plane at their minimum positions. Figure 2 shows the inertial and cell-fixed frames.

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

The orientation of cell $C_{I,J}$ can be determined by applying three consecutive rotations of the inertial reference frame $\mathcal{F}_I$ until the frame aligns with $\mathcal{F}_{\mathrm{C_{I,J}}}$. The oriented frame $\mathcal{F}_{\mathrm{C_{I,J}}}$ is found rotating $\mathcal{F}_I$ by an angle $\phi_{I,J}$ about the axis $\vec{i}_I$, next rotating by an angle $\theta_{I,J}$ about the axis $\vec{j}_I$, and finally rotating an angle $\psi_{I,J}$ about the axis $\vec{k}_I$. These angles $\phi_{I,J}$, $\theta_{I,J}$, and $\psi_{I,J}$ are conventionally referred to as roll, pitch, and yaw, respectively. The right-hand rule is used to define the positive direction for each angle. The formulation of the problem prevents rotation about $\vec{k}_I$, therefore $\psi_{I,J} = 0$.

The rotation matrix maps vectors from frame $\mathcal{F}_{\mathrm{C_{I,J}}}$, rigidly attached to the cell, to the $\mathcal{F}_I$ inertial frame. The relative orientation of these frames are systematically expressed with the parametrization of the rotation matrix $R_{I,J}$ with the roll ($\phi_{I,J}$), pitch ($\theta_{I,J}$), and yaw ($\psi_{I,J}$) angles. Considering that $\psi_{I,J} = 0$, the rotation matrix is

$$R_{I,J} = \begin{bmatrix} C_{\theta_{I,J}} & S_{\theta_{I,J}} S_{\phi_{I,J}} & C_{\phi_{I,J}} S_{\theta_{I,J}} \\ 0 & C_{\phi_{I,J}} & -S_{\phi_{I,J}} \\ -S_{\theta_{I,J}} & C_{\theta_{I,J}} S_{\phi_{I,J}} & C_{\theta_{I,J}} C_{\phi_{I,J}} \end{bmatrix}. \tag{5}$$

The roll–pitch–yaw angle parametrization results in singularities at $\theta_{I,J} = \pm\pi/2$. In reality, the finite actuator lengths prevent their extension to these extreme orientations for these singularity conditions.

Each cell's orientation is controlled by changing the extension of the actuators located on the cell corners. The coordinates of the basis vector $\{\vec{i}_{C_{I,J}}, \vec{j}_{C_{I,J}}, \vec{k}_{C_{I,J}}\}$ with respect to the inertial frame are the columns of the orientation matrix $R_{I,J}$; thus, $R_{I,J}$ can be expressed as

$$R_{I,J} = \begin{bmatrix} i^I_{C_{I,J}} & j^I_{C_{I,J}} & k^I_{C_{I,J}} \end{bmatrix}. \tag{6}$$

The superscript indicates the reference frame corresponding to the expressed basis vector. In order to associate $Z_1^{I,J}$, $Z_2^{I,J}$, and $Z_3^{I,J}$ with the roll and pitch angles $\phi_{I,J}$, $\theta_{I,J}$, the right-hand side of (6) must be expressed with respect to the actuator tip coordinates, then equated to the entries of (5).

By definition, the basis vectors $\vec{i}_{C_{I,J}}$ and $\vec{j}_{C_{I,J}}$ of the cell frame $\mathcal{F}_{\mathrm{C_{I,J}}}$ are rigidly placed on the cell plane. This plane can be uniquely defined by the two vectors $\iota^I = \begin{bmatrix} W & 0 & Z_2^{I,J} - Z_1^{I,J} \end{bmatrix}^T$ and $\eta^I = \begin{bmatrix} 0 & L & Z_4^{I,J} - Z_1^{I,J} \end{bmatrix}^T$, which describe the two sides of the seam connecting the cells at actuators 1 and 2 and actuators 1 and 4.

Orthogonality cannot be assumed for the two vectors ι^I and η^I for each $Z_i^{I,J} \in \begin{bmatrix} 0 & l \end{bmatrix}$, with $i = 1, \ldots, 4$–it can be shown that $\iota^I \cdot \left(\eta^I\right)^T = 0$ only when all actuators of the cell are aligned. For our purpose, we only need to explicitly define the coordinates, with respect to the inertial frame $\mathcal{F}_I$, of the vectors $\vec{i}_{C_{I,J}}$ and $\vec{k}_{C_{I,J}}$; then, the third basis vector is calculated by the cross product $\vec{j}_{C_{I,J}} = \vec{k}_{C_{I,J}} \times \vec{i}_{C_{I,J}}$ (but its calculation is not required by the subsequent analysis). The straightforward choice for the unitary vector $\vec{i}_{C_{I,J}}$ is along the direction of ι^I, so we set $i^I_{C_{I,J}} = \iota^I / ||\iota^I||$. The vector $\vec{k}_{C_{I,J}}$ is normal to both $\vec{\iota}$ and $\vec{\eta}$; hence, by selecting $k^I_{C_{I,J}} = \kappa^I / ||\kappa^I||$ with

$$\kappa^I = \begin{bmatrix} \dfrac{Z_1^{I,J} - Z_2^{I,J}}{W} & \dfrac{Z_1^{I,J} - Z_4^{I,J}}{L} & 1 \end{bmatrix}^T \tag{7}$$

the plane vectors satisfy the inner product orthogonality condition $\iota^I \cdot \left(\kappa^I\right)^T = \eta^I \cdot \left(\kappa^I\right)^T = 0$, and $\vec{k}_{C_{I,J}}$ is always normal to the cell $\mathcal{F}_{\mathrm{C_{I,J}}}$. The $R_{I,J}$ elements $\left[R_{I,J}\right]_{1,1}$, $\left[R_{I,J}\right]_{3,1}$, $\left[R_{I,J}\right]_{2,3}$, and $\left[R_{I,J}\right]_{3,3}$ are equated respectively from the two equivalent representations of the rotation matrix, given in (5) and (6), to derive (after a few elementary algebraic steps) the following kinematic relations:

$$S_{\theta_{I,J}} = \frac{\Delta Z_1^{I,J}}{W} C_{\theta_{I,J}}, \tag{8}$$

$$-S_{\phi_{I,J}} = \frac{\Delta Z_2^{I,J}}{L} C_{\theta_{I,J}} C_{\phi_{I,J}}, \tag{9}$$

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

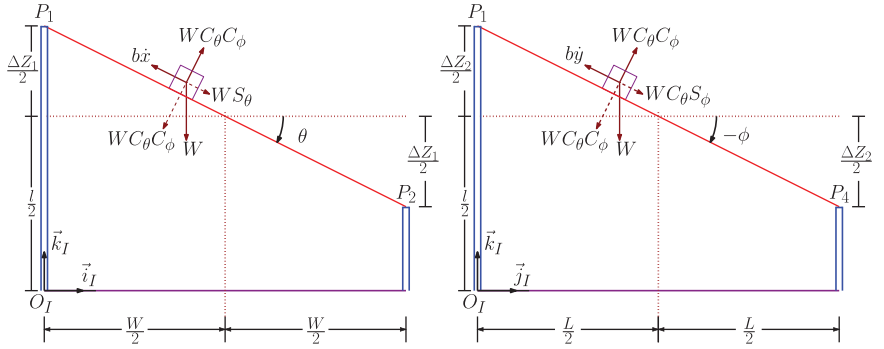

Figure 3. *Schematic of the force components acting on the object as viewed from the cell edges.*

where $\Delta Z_1^{I,J} = Z_1^{I,J} - Z_2^{I,J}$ and $\Delta Z_2^{I,J} = Z_1^{I,J} - Z_4^{I,J}$. With these nonlinear expressions, the actuator tip coordinates are related to the roll and pitch angles of the cell-fixed frame to establish later the total number of independent control variables.

3.2. Object dynamics

In this subsection, we derive the equations of motion of the conveying objects on top of inclined cells of the surface. For simplicity, the objects are treated as point masses. The position of an object, which resides on the surface, can be described within the inertial frame by the position coordinate vector p^I=$[x\ y\ z]^T$. The motion analysis is restricted over a single-cell surface configuration, that is, $\mathcal{S}(1,1)$; as the relations derived in this Subsection refer to all cells, the indices that correspond to the cell number (i.e., I, J) are dropped for brevity. The net forces experienced by the object are captured by the force vector ΣF^I in the inertial frame; and the object motion is governed by Newton's second law, represented by

$$m\ddot{p}^I = \Sigma F^I \tag{10}$$

for the object mass m. Three forces act on the object: (i) the object's weight in the $-\vec{k}_I$ direction of the inertial frame, (ii) the frictional resistance to the object motion, and (iii) the surface reaction force, which is normal to the cell and is oriented upward. The object dynamics are controlled by adapting the surface inclination using the actuators. The object weight is expressed in the inertial frame as $W^I = -\begin{bmatrix}0\,0\,\mathrm{mg}\end{bmatrix}^T$. Using the rotation matrix in (5), the weight components in the cell-fixed frame are given by

$$W^C = R^T W^I = \mathrm{mg}\begin{bmatrix} S_\theta & -C_\theta S_\phi & -C_\theta C_\phi \end{bmatrix}^T. \tag{11}$$

The surface reaction forces on the object are opposite and equal to the weight component in the $\vec{k}_C$ direction; thus,

$$N^C = \mathrm{mg}\begin{bmatrix} 0 & 0 & C_\theta C_\phi \end{bmatrix}^T. \tag{12}$$

The force opposing the object's motion due to the friction of the surface–object contact is modeled as

$$F^C = -bv^C, \tag{13}$$

where the coefficient of friction b is greater than zero and $v^C \in \mathbb{R}^3$ is the object velocity in the object-fixed frame. Because the object is constrained to lie on the surface, the component of v^C in the $\vec{k}_C$ direction is zero. Using Newton's second law, $F^I = -bv^I$ where $v^I = [\dot{x}\ \dot{y}\ \dot{z}]^T$. Figure 3 illustrates these

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

force components acting on the object. Substituting the force Eqs. (11)–(13) into Newton's second law, the object experiences the accelerations

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = g \begin{bmatrix} C_\theta C_\phi^2 S_\theta \\ -C_\theta C_\phi S_\phi \\ C_\theta^2 C_\phi^2 - 1 \end{bmatrix} - b \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \tag{14}$$

3.3. Elementary reference point control

The autonomous transport of an object to a single cell position with inertial coordinates $\left(x_r^I, y_r^I\right)$ requires the development of a feedback control law. The reference coordinate of the object in the $\vec{k}_I$ directions is not considered because the surface orientation constrains the object's z^I coordinate. The feedback law is translated into height differences of the four actuators' lengths.

The control law analysis is pursued by a simple change of coordinates to the error variables $e_x = x^I - x_r^I$ and $e_y = y^I - y_r^I$. From (14), the error dynamics become

$$\ddot{e}_x + b\dot{e}_x + C_\theta^2 C_\phi^2 \frac{g}{W} \Delta Z_1 = 0, \tag{15}$$

$$\ddot{e}_y + b\dot{e}_y + C_\theta^2 C_\phi^2 \frac{g}{L} \Delta Z_2 = 0, \tag{16}$$

where $\Delta Z_1 = Z^1 - Z^2$ and $\Delta Z_2 = Z^1 - Z^4$. The $C_\theta^2 C_\phi^2$ term is contained within the (0 1] interval because $\theta, \phi \in (-\pi/2\, \pi/2)$. The error dynamics, therefore, can be represented by two identical second-order nonlinear differential equations. The control objective can be achieved with a feedback law based on the position error. The following saturated feedback functions relevant to finite actuator lengths are chosen:

$$\Delta Z_1 = K_x sat_W\, e_x, \qquad \Delta Z_2 = K_y sat_L\, e_y, \tag{17}$$

where K_x, K_y, M_x, and M_y are positive constants. The saturation function $sat_M(\cdot)$ is defined as

$$sat_M(x) = \begin{cases} x, & |x| \le M; \\ sign(x) \cdot M, & \text{else.} \end{cases}$$

Elementary Lyapunov stability arguments can show the convergence of the above control law. The parameters of the second-order differential Eqs. (15) and (16), for the control laws given in (17), are positive; thus, the error dynamics (e_x, e_y) are rendered exponentially stable.

The friction coefficient b damps the object's motion. To inject additional damping to the object's motion, the control inputs in (17) may be augmented with a velocity feedback term. Finally, the actuator lengths are configured based on the values of ΔZ_1 and ΔZ_2. In order to orient the surface about the axes located in the midpoints of the cell width and length, the following extensions for the actuator tips are chosen:

$$Z_1 = \frac{l}{2} + \frac{\Delta Z_1}{2} + \frac{\Delta Z_2}{2}, \qquad Z_2 = \frac{l}{2} - \frac{\Delta Z_1}{2} + \frac{\Delta Z_2}{2},$$

$$Z_3 = \frac{l}{2} - \frac{\Delta Z_1}{2} - \frac{\Delta Z_2}{2}, \qquad Z_4 = \frac{l}{2} - \frac{\Delta Z_1}{2} + \frac{\Delta Z_4}{2}.$$

To re-ensure that the tip positions of the actuators remain within the [0 l] interval, the control inputs are bounded by $|\Delta Z_1|, |\Delta Z_2| \le l/2$; therefore, the feedback gains are restricted to $K_x \le l/2W$ and $K_y \le l/2L$, respectively.

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

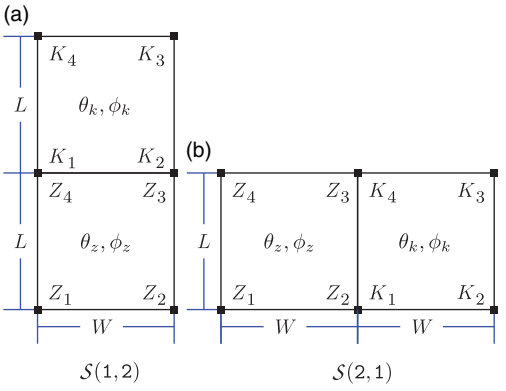

Figure 4. *Two-cell configurations (vertical and horizontal).*

4. Multicell kinematics

In this section, we progressively derive–using simple surface layouts–the available DOF of a generic surface configuration by identifying (i) the total independent variables needed to define its shape and (ii) the number of the surface kinematic constraints, which emerge from the preliminary requirement that all cells form flat planes. We later show how the DOF of the surface are interpreted to the available number of control inputs for the morphing system.

4.1. Multicell constraints

Each cell of the surface $\mathcal{S}(\mathtt{n}, \mathtt{m})$ possesses two Degrees of Freedom (DOF) and can change the surface pitch and roll angles to convey the object. Neighboring cells share two actuators along their common edge. Therefore, the extended lengths of the actuators for each cell at any given point in time have to also satisfy (3) to prevent discontinuities on the surface. This constraint reduces the total system DOF.

The total constraints for the surface $\mathcal{S}(\mathtt{n}, \mathtt{m})$ are systematically derived from a foundation of the two simplest cases involving cell pairs arranged either vertically or horizontally.

4.1.1. Vertical configuration

The surface configuration for $\mathcal{S}(1, 2)$ is provided in Fig. 4(a). For this cell pair, two actuators are shared on their common edge. The height profile, which is directly determined by the actuator extensions, is thus equal for this shared edge. The two cells are labeled as C_Z and C_K for the following analysis.

The schematic in Fig. 4(a) shows that the shared points in space are $Z_4 = K_1$ and $Z_3 = K_2$. Relation (3) applies to each cell; thus,

$$K_2 - K_1 = Z_3 - Z_4 = Z_2 - Z_1. \tag{18}$$

The pitch angles θ for the two cells can, via rearrangement of the terms of (8), be related to the actuator extensions of the shared edge:

$$T_{\theta_z} = \frac{Z_1 - Z_2}{W}, \qquad T_{\theta_k} = \frac{K_1 - K_2}{W}. \tag{19}$$

Substitution of (18) into (19) establishes that the pitch orientation is equal for the shared edge of the cells; that is,

$$T_{\theta Z} = T_{\theta K} \rightarrow \theta_Z = \theta_K. \tag{20}$$

Thus, two vertically aligned cells possess the same pitch angle θ. The roll angle of each cell is not limited, so the $\mathcal{S}(1, 2)$ surface has three DOF.

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

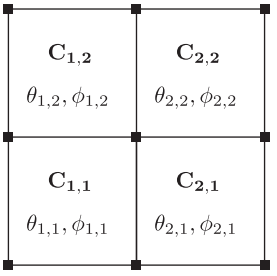

Figure 5. *A $\mathcal{S}(2,2)$ surface in a square configuration.*

4.1.2. Horizontal configuration

The horizontal configuration of $\mathcal{S}(2,1)$ is provided in Fig. 4(b). The actuators on the shared edge physically result in the equality of $Z_2 = K_1$ and $Z_3 = K_4$. The constraint that both cells form a continuous surface corresponds to the case of (3), yielding:

$$Z_2 - Z_3 = Z_1 - Z_4 = K_1 - K_4. \tag{21}$$

The pitch and roll angles of the actuators on the shared edge are found by substitution of (9) into (21):

$$\frac{T_{\phi_Z}}{C_{\theta_Z}} = \frac{Z_1 - Z_4}{L}, \qquad \frac{T_{\phi_K}}{C_{\theta_K}} = \frac{K_1 - K_4}{L}. \tag{22}$$

The following nonholonomic constraint relating the orientation between adjacent cells is derived from (21) and (22):

$$\frac{T_{\phi_K}}{C_{\theta_K}} = \frac{T_{\phi_Z}}{C_{\theta_Z}}. \tag{23}$$

The $\mathcal{S}(2,1)$ surface has four independent generalized coordinates describing the system orientation (θ_Z,ϕ_Z, θ_K, and ϕ_Z) and one constraint is given in (23). As for the vertically arranged cells, the system has three DOF for the horizontal case.

4.1.3. Square configuration

The vertical and horizontal constraints in both grid directions will be combined in an illustrative example for the case of a square $\mathcal{S}(2,2)$ surface, as shown in Fig. 5. From the vertical alignment:

$$\begin{aligned} T_{\theta_{1,1}} = T_{\theta_{1,2}} \quad &\longrightarrow \quad \theta_{1,2} = \theta_{1,1}, \\ T_{\theta_{2,1}} = T_{\theta_{2,2}} \quad &\longrightarrow \quad \theta_{2,2} = \theta_{2,1}. \end{aligned} \tag{24}$$

The constraints equations due to the horizontal placement are

$$\begin{aligned} \frac{T_{\phi_{1,1}}}{C_{\theta_{1,1}}} = \frac{T_{\phi_{2,1}}}{C_{\theta_{2,1}}} \quad &\longrightarrow \quad T_{\phi_{2,1}} = \frac{C_{\theta_{1,1}}}{C_{\theta_{2,1}}} T_{\phi_{1,1}}, \\ \frac{T_{\phi_{1,2}}}{C_{\theta_{1,2}}} = \frac{T_{\phi_{2,2}}}{C_{\theta_{2,2}}} \quad &\longrightarrow \quad T_{\phi_{2,2}} = \frac{C_{\theta_{1,1}}}{C_{\theta_{2,1}}} T_{\phi_{1,2}}. \end{aligned} \tag{25}$$

In total, there are eight generalized coordinates determining the orientation of the $\mathcal{S}(2,2)$ surface. Due to the constraints of the neighboring cells, the total number of DOF drops to four. It becomes apparent that the control resources available to manipulate the surface are a common pitch angle for every column, and a common roll angle for every row. Every independent pitch and roll inclination translates to a difference $\Delta Z_1^{I,J} = Z_1^{I,J} - Z_2^{I,J}$ and $\Delta Z_2^{I,J} = Z_1^{I,J} - Z_4^{I,J}$ of the actuators' heights, respectively.

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

4.1.4. Arbitrary configuration

The available DOF for an arbitrary surface $\mathcal{S}(\mathtt{n},\mathtt{m})$ is determined by the total number of control inputs available to the system. Following the case of the vertical configuration, it is found that

$$T_{\theta_{I,J}} = T_{\theta_{I,1}} \forall\, I \in \mathcal{C}, J \in \mathcal{R} \rightarrow \theta_{I,J} = \theta_{I,1} = \Theta_I. \tag{26}$$

In the horizontal configuration, the roll angles have a dependency on the pitch angles. This nonholonomic constraint results in the relationship:

$$\frac{T_{\phi_{I,J}}}{C_{\theta_{I,J}}} = \frac{T_{\phi_{I+1,J}}}{C_{\theta_{I+1,J}}} \quad \longrightarrow \quad T_{\phi_{I+1,J}} = \frac{C_{\Theta_{I+1}}}{C_{\Theta_I}} T_{\phi_{I,J}}\,. \tag{27}$$

Each row of the array can be iteratively calculated as follows:

$$T_{\phi_{I,J}} = \frac{C_{\Theta_I}}{C_{\Theta_1}} T_{\Phi_J} \text{where } \Phi_J = \phi_{1,J} \text{ and } I \in \{2,\ldots,n\}\,, J \in \mathcal{R}. \tag{28}$$

From (26) and (27), for any cell $C_{I,J}$ of surface $\mathcal{S}(\mathtt{n},\mathtt{m})$, the pitch and roll angles can be described by a function of a single row of n pitch angles Θ_I (with $I \in \mathcal{C}$) , and a single column of m roll angles Φ_J (with $J \in \mathcal{R}$) . Specifically, for each cell $C_{I,J}$:

$$\theta_{I,J} = \Theta_I, \tag{29}$$

$$\phi_{I,J} = \arctan\left(\frac{C_{\Theta_I}}{C_{\Theta_1}} T_{\Phi_J}\right). \tag{30}$$

These relationships show that the surface DOF total to $n+m$. More specifically, the rotational motion of the surface has $2{\cdot}n{\cdot}m$ generalized coordinates and is subject to $2{\cdot}n{\cdot}m - n - m$ constraints. The DOF dictate the number of independent coordinates that are needed to completely define the surface orientation. The pitch and roll angles of a generic cell $C_{I,J}$ are calculated by subtracting the actuator extensions along each edge: $\Delta Z_1^{I,J} = Z_1^{I,J} - Z_2^{I,J}$ and $\Delta Z_2^{I,J} = Z_1^{I,J} - Z_4^{I,J}$. The constraints of (26) and (27) can be expressed with respect to these height differences $\Delta Z_1^{I,J}$ and $\Delta Z_2^{I,J}$ for all cells. Thus, for every $I \in \mathcal{C}$ and $J \in \mathcal{R}$:

$$\Delta Z_1^{I,J} = \Delta Z_1^{I,1} = \Delta Z_1^I, \tag{31}$$

$$\Delta Z_2^{I,J} = \Delta Z_2^{1,J} = \Delta Z_2^J, \tag{32}$$

where $\Delta Z_1^I = Z_1^{I,1} - Z_2^{I,1}$ and $\Delta Z_2^J = Z_1^{1,J} - Z_4^{1,J}$.

The control algorithm is designed to transport each object on the surface to the reference cell C_{I_r,J_r} by coordination of the values of the $n+m$ control inputs ΔZ_1^I $(I \in \mathcal{C})$ and ΔZ_2^J $(J \in \mathcal{R})$. Figure 6 summarizes the independent control variables and the constraints in the orientation angles for each cell.

5. Multicell control

In Section 4, the unique definition of the overall surface orientation for the generic configuration $\mathcal{S}(\mathtt{n},\mathtt{m})$ was found by the specification of each of its $n+m$ DOF. The objective of the control algorithm is to allocate these DOF so that the system most efficiently achieves the global goal of transporting each object to the reference cell C_{I_r,J_r}.

The control law accomplishes this objective by adjusting $\Delta Z_1^I = Z_1^{I,1} - Z_2^{I,1}$ $(I \in \mathcal{C})$ and $\Delta Z_2^J = Z_1^{1,J} - Z_4^{1,J}$ $(J \in \mathcal{R})$ defined in Section 3.1. Initially, the height of the target cell is leveled to attain the lowest potential energy. The heights of all other cells are chosen to be proportional to the distribution of objects between the target cell and the external boundaries of the surface. Two implementations of this general logic–distributed allocation and wave control–have been explored and are described in detail. The final

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

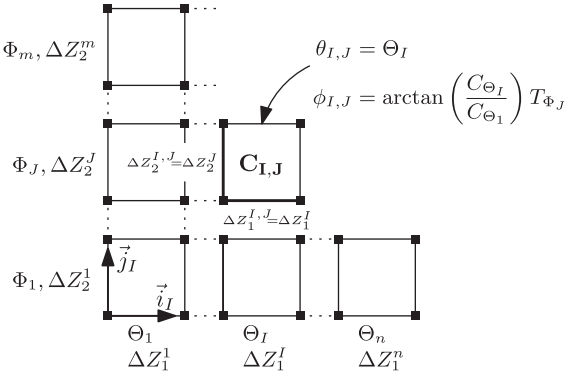

Figure 6. *Schematic illustration of the independent control variables and the constraints of the orientation angles for each cell.*

step of the collective control law is to translate the inclinations of the cells into height adjustments of the individual actuators of the grid.

5.1. Distributed allocation control logic

The rationale underpinning the distributed allocation control algorithm is to maximize the utilization of the actuators' finite heights by restricting the surface to incline only for the rows and columns that hold objects. This control logic is built on the hypothesis that a faster convergence time will be achieved with a dynamic rather than a static inclination of the cells.

The algorithm determines the column and row sets $\bar{\mathcal{C}} \subset \mathcal{C}$ and $\bar{\mathcal{R}} \subset \mathcal{R}$ that have cells containing objects. These two sets are divided into two subsets:

$$\begin{aligned} \bar{\mathcal{C}}_l &= \left\{ I \in \bar{\mathcal{C}} | I < I_r \right\} \qquad \bar{\mathcal{R}}_d = \left\{ J \in \bar{\mathcal{R}} | J < J_r \right\} \text{ and} \\ \bar{\mathcal{C}}_r &= \left\{ I \in \bar{\mathcal{C}} | I > I_r \right\} \qquad \bar{\mathcal{R}}_u = \left\{ J \in \bar{\mathcal{R}} | J > J_r \right\}. \end{aligned} \tag{33}$$

If $\bar{C}_l, \bar{C}_r$ and $\bar{R}_d, \bar{R}_u$ denote the cardinality of $\bar{\mathcal{C}}_l, \bar{\mathcal{C}}_r$ and $\bar{\mathcal{R}}_d, \bar{\mathcal{R}}_u$, respectively, then the desired actuator height changes ΔZ_1^I $(I \in \mathcal{C})$ and ΔZ_2^J $(J \in \mathcal{R})$ are given by

$$\Delta Z_1^I = \begin{cases} a\dfrac{l}{\bar{C}_l}, & \text{if } I \in \bar{\mathcal{C}}_l, \\ -a\dfrac{l}{\bar{C}_r}, & \text{if } I \in \bar{\mathcal{C}}_r \\ 0, & \text{else}, \end{cases} \quad \text{and} \quad \Delta Z_2^J = \begin{cases} b\dfrac{l}{\bar{R}_l}, & \text{if } I \in \bar{\mathcal{R}}_d, \\ -b\dfrac{l}{\bar{R}_r}, & \text{if } I \in \bar{\mathcal{R}}_r, \\ 0, & \text{else}, \end{cases}$$

where a and b are positive percentiles satisfying $a + b = 1$.

As the maximum extensions of the actuators have a physical limit, the constants a and b represent the percentile distributions of the actuator lengths assigned in the $\vec{i}_i$ and $\vec{j}_i$ directions of the surface. The actuator lengths are updated based on the calculated height differences, starting with $Z_i^{I_r,J_r} = 0$ for $i = 1, \ldots, 4$. The sets $\bar{\mathcal{C}}$ and $\bar{\mathcal{R}}$ are updated as objects translate on the surface. This approach is illustrated in Figs. 7, 8, and 9 for the calculation of ΔZ_1^I and ΔZ_2^J for an $\mathcal{S}(5,4)$ surface. Using this control scheme, the surface inclination is allotted uniformly to all objects, while not allocating any height different to cells without objects.

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

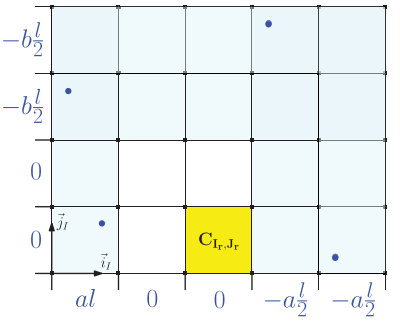

Figure 7. *A schematic illustration of sets $\bar{\mathcal{C}}$ and $\bar{\mathcal{R}}$ for an $\mathcal{S}(5,4)$ surface. The object locations are represented by dots. The reference cell is $C_{I_r,J_r} = C_{3,1}$. The shaded rows and columns display the numerical ΔZ_1^I and ΔZ_2^J values for the distributed allocation control algorithm.*

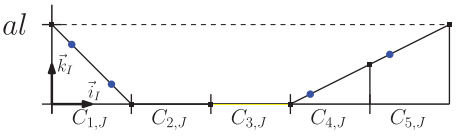

Figure 8. *Side view in the $\vec{i}_I$ direction of the $\mathcal{S}(5,4)$ surface actuator extensions calculated by the distributed allocation control algorithm.*

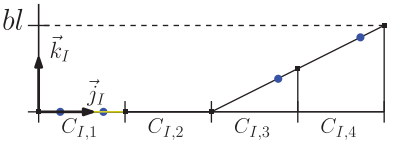

Figure 9. *Side view in the $\vec{j}_I$ direction of the $\mathcal{S}(5,4)$ surface actuator extensions calculated by the distributed allocation control algorithm.*

5.2. Wave control logic

The second logic approach, the wave control algorithm, creates a ripple in the surface at the rows and columns corresponding to the position(s) of the outermost object(s). This ripple advances over time as a wave converging on the destination location, thereby conveying the objects toward the reference cell. The ripple occurs at the rows and columns where the maximum inclination exists. This inclination ripple transports object most distant from the target and gather additional objects as it proceeds toward the target. Using the same notation as in the section on distributed allocation controller, the height differences ΔZ_1^I ($I \in \mathcal{C}$) and ΔZ_2^J ($J \in \mathcal{R}$) are given by

$$\Delta Z_1^I = \begin{cases} a{\cdot}l, & \text{if } I = \min\left(\bar{\mathcal{C}}_l\right), \\ -a{\cdot}l, & \text{if } I = \max\left(\bar{\mathcal{C}}_r\right), \\ 0, & \text{else} \end{cases}$$

$$\text{and } \Delta Z_2^J = \begin{cases} b{\cdot}l, & \text{if } I = \min\left(\bar{\mathcal{R}}_d\right), \\ -b{\cdot}l, & \text{if } I = \min\left(\bar{\mathcal{R}}_r\right), \\ 0, & \text{else.} \end{cases}$$

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

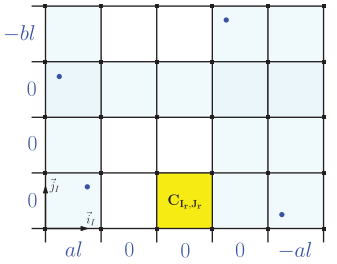

Figure 10. *The ΔZ_1^I and ΔZ_2^J values calculated by the wave control algorithm are displayed as the numerical entries at the edges of the $\mathcal{S}(5,4)$ surface.*

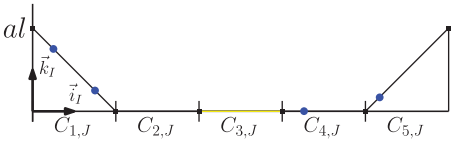

Figure 11. *Side view in the $\vec{i}_I$ direction of the $\mathcal{S}(5,4)$ surface actuator extensions calculated by the wave control algorithm.*

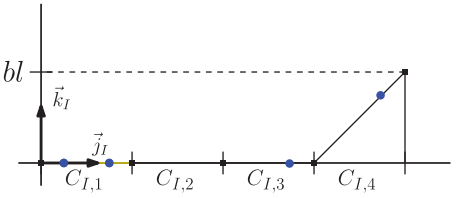

Figure 12. *Side view in the $\vec{j}_I$ direction of the $\mathcal{S}(5,4)$ surface actuator extensions calculated by the wave control algorithm.*

The topography of the surface ripple at a given time instant of the wave control algorithm is shown in Figs. 10, 11, and 12. The wave control algorithm assigns the maximum inclination to the rows and columns containing the objects most distant from the target to convey them collectively toward the reference cell.

5.2.1. Length update control law

After calculation of the height differences, the final step of the collective control algorithm involves the update of the individual actuator lengths. To ease the analysis, the height of each actuator is defined as the sum $z^{I_a,J_a} = z_a^{I_a} + z_a^{J_a}$ where $I_a = \{1,\ldots,n+1\}$ and $J_a = \{1,\ldots,m+1\}$ are the actuator column and row identifiers within the array. The pair $\left(z_a^{I_a}, z_a^{J_a}\right)$ represent the length components of the actuator that are working to drive the object in the $\vec{i}_i$ and $\vec{j}_i$ directions. The two components $z_a^{I_a}$ and $z_a^{J_a}$ have a constant value for every column I_a and row J_a. The update law is initiated by leveling down the actuators of the reference cell C_{I_r,J_r}. This action is accomplished by setting $z_i^{I_i,J_i} = 0$ for $I_i = \{I_r, I_{r+1}\}$ and $J_i = \{J_r, J_{r+1}\}$. The rest of the height components can be successively calculated using the following piecewise functions:

$$z_a^{I_a} = \begin{cases} -\sum_{I_r+1}^{I_a-1} \Delta Z_1^k, & I_a \geq I_r + 1, \\ \sum_{I_a}^{I_r} \Delta Z_1^k, & I_a \leq I_r, \end{cases} \tag{34}$$

$$z_a^{J_a} = \begin{cases} -\sum_{J_r+1}^{J_a-1} \Delta Z_2^k, & J_a \geq J_r + 1, \\ \sum_{J_a}^{J_r} \Delta Z_2^k, & J_a \leq J_r. \end{cases} \tag{35}$$

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

6. Experimental prototype

In this section, the electromechanical design is detailed for an experimental implementation of a surface that autonomously controls its inclination to convey objects to an arbitrary reference cell position. This prototype surface is used to identify the kinematic capabilities and implementation limitations associated with LSANs. The system consists of a surface constructed of rigid panels and that is supported by ten linear actuators whose extensions enable adjustment of the surface inclination. The surface and actuators are positioned to form four square cells configured in the shape of a "T". The "T" shape was chosen as a benchmark configuration because it allows the testing of multiple different horizontal and vertical cell transit paths while requiring the fewest actuators[1]. The physical apparatus is shown in Fig. 13.

The prototype system was built atop a pegboard base, whose regularly spaced holes were used to position the components at defined intervals. The vertical orientation of each actuator was supported and stabilized using two threaded rods. The actuator model selection criteria were low cost, sufficient actuation length, and relative speed; based on these criteria, the Firgelli L12-I actuator, with a 100 mm stroke and relatively high speed (23 mm/s), was chosen. A relatively short time constant allows the actuators to have a faster feedback response in order to control the moving objects more effectively. Despite the relatively low time constant of the selected actuators, a distinct time scaling between the actuators and the objects' dynamics remain. While the Firgelli actuators have a low actuation force (43 N), this individual net force was still sufficient for the network to collectively adjust the shape of the surface.

A thin textile sheet (10% elastane, 90% polyester) was attached to the actuator tips, thereby providing a foundation onto which the surface panels were secured yet able to provide compliance as the actuator lengths extend or contract. Corrugated plastic surface panels, chosen for their low coefficients of static and kinetic friction, were affixed to the textile. The small frictional resistance permits object motion for the experimentally realizable inclination angles. The corrugated plastic was selected after experimenting with elastomeric materials such as neoprene and latex, each of which possessed higher friction coefficients that prevented the object movement. The object itself was chosen after assessing candidates ranging from wood cubes to Ping-Pong balls. A polycarbonate hemisphere (seen in Fig. 13) was the object found to exhibit the most controllable and repeatable movements.

The system control was implemented with an Arduino Uno microcontroller. This microcontroller was chosen for its minimal control software development time, ability to execute the control algorithm sufficiently quickly (at 11 Hz), and its low cost. The Arduino Integrated Development Environment (IDE) and a large number of available low-level execution libraries for actuator components (e.g., servos and motors) were used to rapidly develop the prototype system for testing.

The longitudinal and lateral position measurements of the objects were generated initially by a commercially available visual tracking system called Roborealm, and later by a custom program developed in the OpenCV environment. A consumer web camera (model VF0720, Creative Labs) was used for a vision sensor placed above the surface. The tracking system processed the video feed to calculate the object coordinates on the surface plane. These planar coordinates constitute the feedback signal used by the control algorithm. The computer and the Arduino microcontroller exchanged data via a serial connection and a wireless RF module operating at 7 Hz (XBee, Digi International).

7. Results

7.1. Numerical simulations

This section provides an evaluation of the proposed multicell control algorithms via extensive numerical simulations. The two algorithms (distributed allocation and wave) were compared to a benchmark static funnel configuration using a model in MATLAB/Simulink. The simulation was initialized with objects at 20 initial locations randomly positioned on a $\mathcal{S}(5, 6)$ rectangular surface. The cell $C_{I_r,J_r} = C_{3,1}$ was designated as the reference cell. For each of the simulation runs and control algorithms, the length,

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

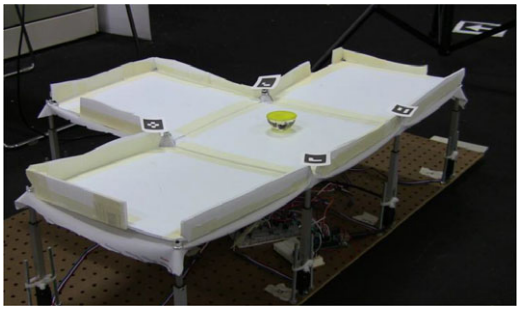

Figure 13. *The experimental platform.*

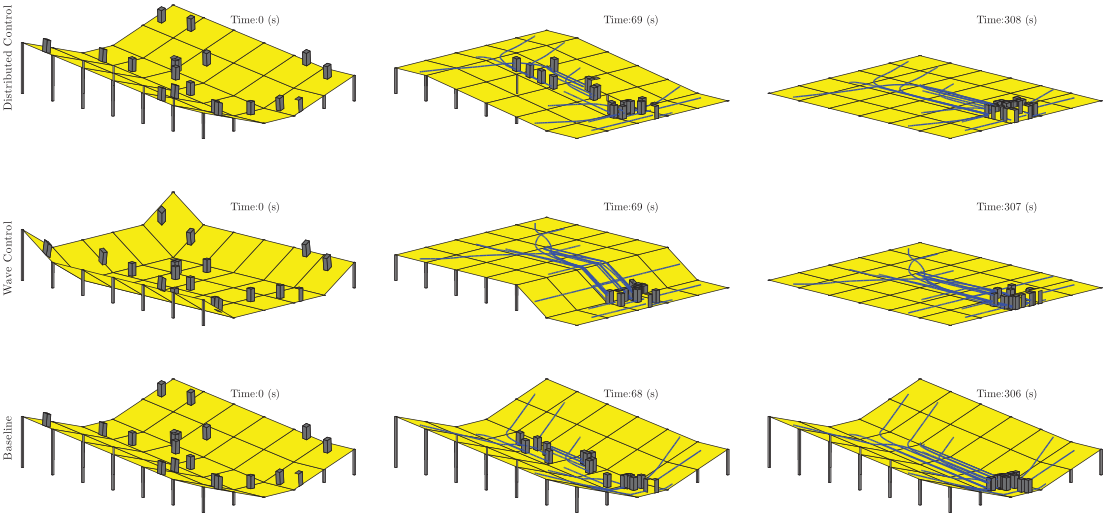

Figure 14. *Time sequence for three instances of the surface contour for the three control algorithms.*

width, and initial position parameters were kept constant to enable direct comparison of the results. The cell length and width were set to $L = 200$ cm and $W = 200$ cm, respectively. The maximum actuator stroke was set to $l = 100$ cm. The objects had masses of $m = 1$ kg. The coefficient of friction between the surface and the object was chosen to be $b_f = 0.1$. The motion of each object was simulated by numerical integration of Newton's equation of motion (Section 3.2). A barrier is assumed at the exterior margins of the surface to keep the objects confined in the workspace of the mechanism. The impacts of the objects with the barrier are modeled as elastic such that no energy is dissipated from the collision.

The configuration of the surface for the three distinct algorithms (distributed allocation, wave, and static funnel) are animated for different time instances, as shown in Fig. 14. It can be seen that at least one object was placed initially in each row and column of the surface. This causes the distributed allocation control algorithm (for the initial time instant) to spread the available incline across all of the cells, resulting in a even distribution of the slope, identically to the static funnel case.

The locations of the objects in the $\vec{i}_I$ and $\vec{j}_I$ directions with respect to time for each of the three algorithms are plotted in Fig. 15. By inspection, the wave control algorithm conveys the objects to the reference cell in the least amount of time for both directions. By contrast, the baseline funnel exhibits the slowest delivery time to the reference location. The wave algorithm requires approximately 70 s, the distributed algorithm 130 s, and the baseline funnel 200 s to converge. The wave control algorithm injects to the objects maximal amounts of potential energy through the largest inclinations of the outermost

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

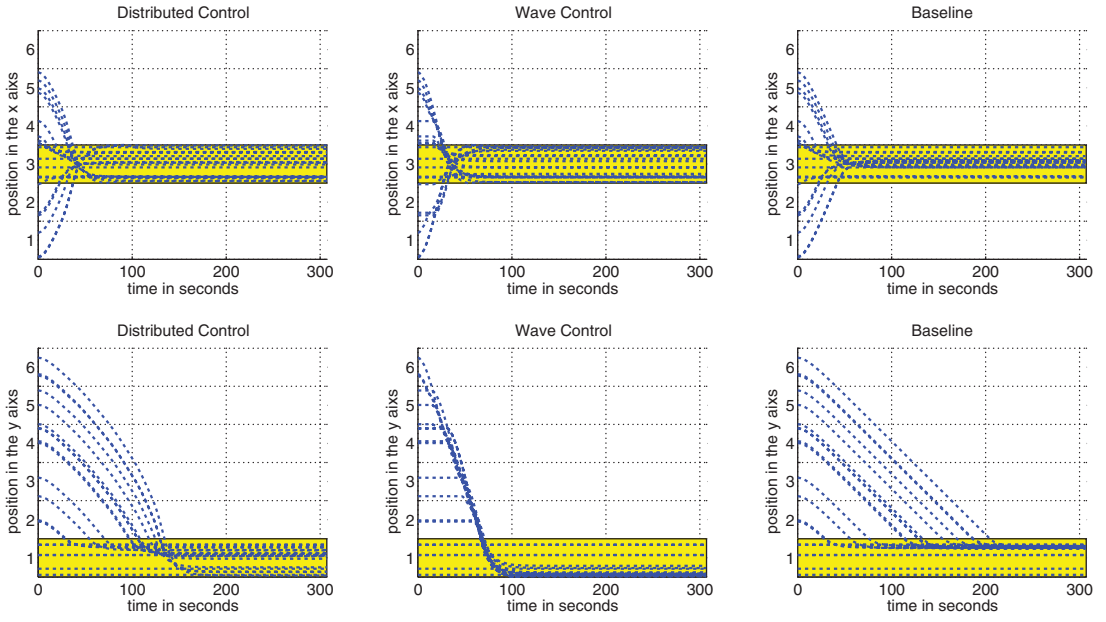

Figure 15. *Objects' positions with respect to time. The shaded region corresponds to the reference cell length.*

cells, resulting in the highest kinetic energy conversion. The wave control has the additional benefit that all of the objects require a similar time to arrive at the target destination.

The behavior of each control algorithm can be better understood by investigating the special case of the single track surface $\mathcal{S}(1, 10)$. In this scenario the reference cell is $C_{I_r,J_r} = C_{1,10}$. The simulations are initiated by having a single object lying on each cell of the surface. The motion of the objects is restricted to the $\vec{i}_I$ direction. All surface and object parameters are the same as in the first case study. The position and velocity of the objects in the $\vec{i}_I$ direction, for the three algorithms, with respect to time are shown in Figs. 16 and 17, respectively.

Obviously, the static inclination funnel (baseline) exhibits the slowest convergence time. In all cases the velocity of each object is determined by the slope of the cell that is subsequently filtered by the low pass transfer function $1/\left(s + b_f\right)$. A large value for b_f implies a small final velocity and a fast response. The steady-state velocity of the objects is calculated by the term $(g/b_f)C_\theta C_\phi S_\theta$ based on (14). Therefore, the maximum inclination results in the maximum velocity. This is the velocity that the objects obtain gradually, during the execution of the wave algorithm, starting from the most remote cells. In the distributed allocation algorithm, the kinetic energy of the system increases only when cells are free of objects, resulting in a slower final convergence.

7.2. Experimental results

The prototype system described in Section 6 was used to further validate the wave control algorithm. The objective was to transport the object consistently across the "T" shaped surface and to successfully exit at a location on the border of the reference cell. The primary considerations in choosing the test pathway was to demonstrate the basic motion capabilities of the surface. A scenario representing a real-world circumstance where the object must transit around an obstacle, such as a cavity or a vertical barrier, was chosen. The reference path requires that the object makes a U-turn across the cells to reach the target exit point. A two-dimensional projection of the object's measured experimental trajectory is plotted on the left side of Fig. 18. The right side of the figure illustrates the surface's morphology at different

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

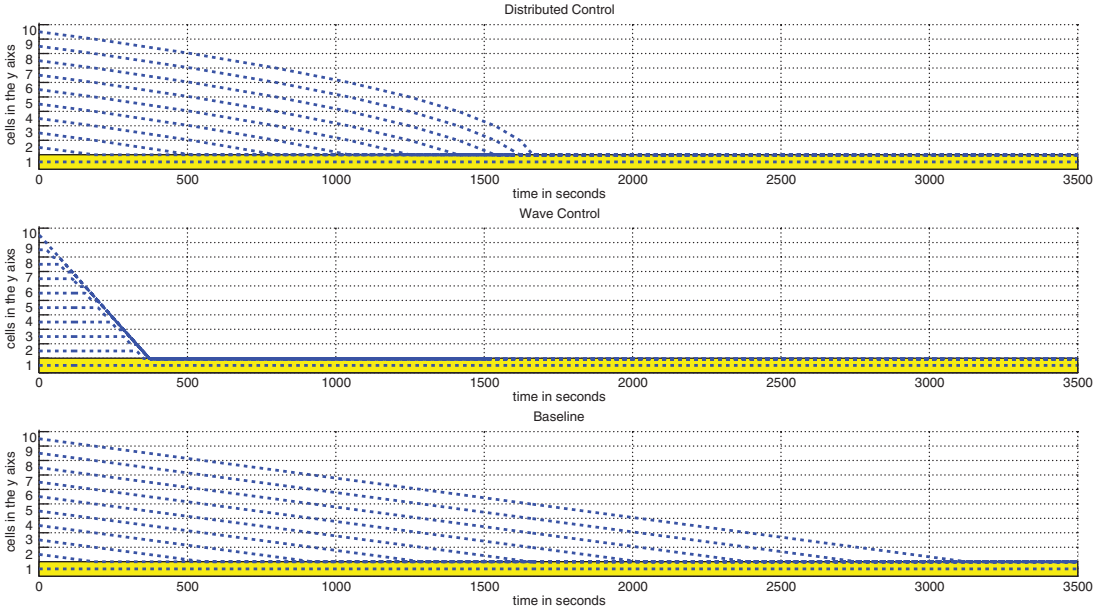

Figure 16. *Comparison of position with respect to time for the three control algorithms in the $\vec{i}_I$ direction of the $\mathcal{S}(1, 10)$ surface.*

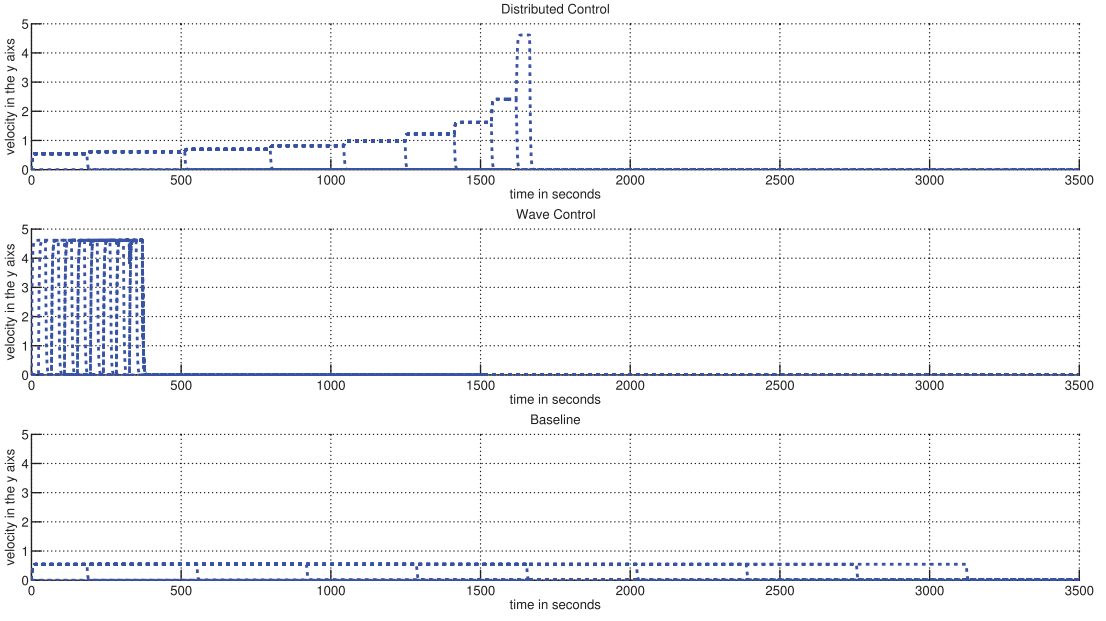

Figure 17. *Comparison of object velocity with respect to time for the three control algorithms in the $\vec{i}_I$ direction of the $\mathcal{S}(1, 10)$ surface.*

time instances of the test run. The object successfully reached the reference location by autonomously navigating across multiple cells. At every transition between adjacent cells, the object is moving over the midpoint of its common side.

The preliminary experiments that took place on the prototype mechanism demonstrated that the maximum slope (maximum propulsion force) is required to exceed the static friction resistance to allow the

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

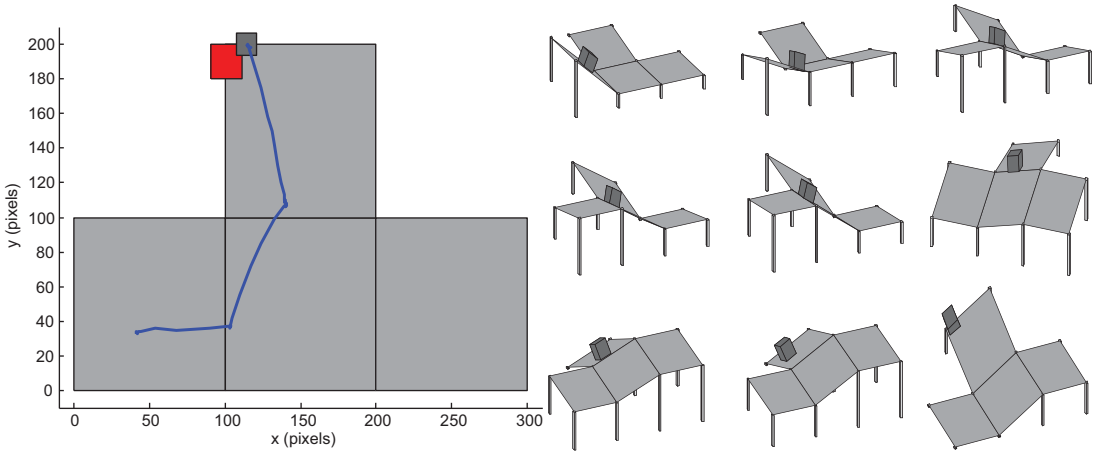

Figure 18. *The experimentally observed trajectory of the object projected onto a 2D plane. The object (dark gray) traces the path (blue line) along the surface panels (light gray) until it reaches the target zone (red). The right side illustrates the 3D surface and the experimentally measured object position at nine different time instances.*

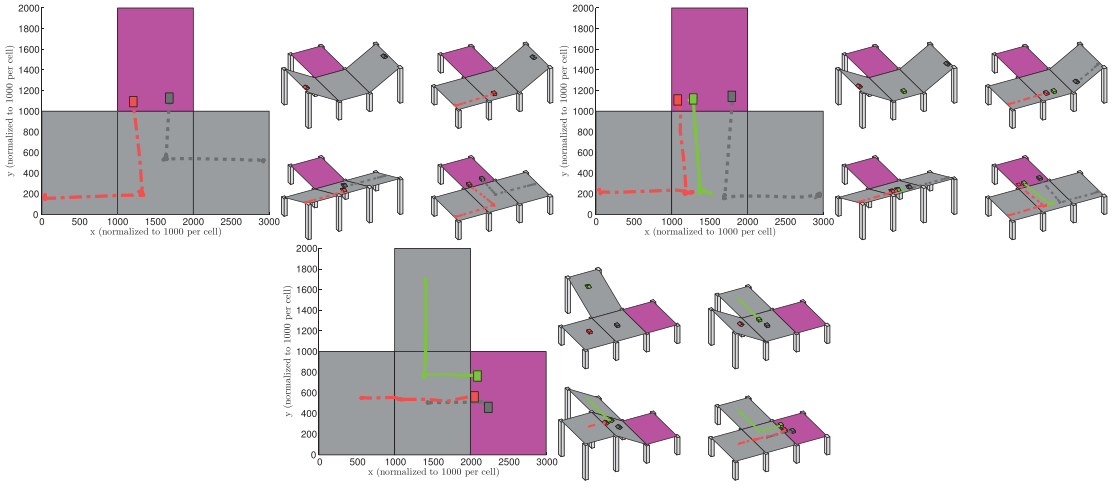

Figure 19. *2D projection of the objects' planar motions as measured by experiments. The surface panels are illustrated in gray, the target cell in magenta, the objects are color coded, and their respective lines trace the paths that each object traversed. The 3D surface visualizations (right side) illustrate the objects' paths and positions experimentally measured at four different time instances.*

object to slide when starting from rest. To this extent, the tuning parameters of the wave control algorithm a, b (Section 5.2) are set to either zero or one providing full inclination in a single direction. This modification was deemed necessary to obtain greater slopes, thus, greater applied forces for moving the objects.

The wave control algorithm was further tested using two objects with repeatable success. Sample test runs are shown in Fig. 19. In this experiment, the two objects, starting from the outermost cells of the surface, are converging to the center of the front reference cell. The final level of complexity involved three objects in multiple configurations. In all tested configurations, the surface was able to repeatedly and successfully deliver the objects to the reference location.

https://doi.org/10.1017/S0263574721000473 Published online by Cambridge University Press

8. Conclusion

This work provides an analytic and experimental study of an LSAN for distributed manipulation. The presented mechanism involves an adaptive surface that autonomously adjusts its topography to convey an arbitrary number of objects to a reference point. The morphing process takes place by a grid of linear actuators that adjust their height. A detailed analytical study of this mechanism is provided that results in an explicit calculation of the available control resources. The derivation of computationally attractive control algorithms that can handle efficiently an arbitrary number of actuators was presented. A prototype testbed was developed using off-the-shelf components to validate the applicability of this original concept and to reveal potential limitations that do not emerge from the theoretical analysis. Both of the described control algorithms exhibit significant improvements in performance as compared to the conventional static inclination solution.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable feedback.

Note

1 The shape of the experimental apparatus was inspired by the T-polyomino piece of the tile-matching video game Tetris, developed by Alexey Pajitnov.

References

[1] M. Savia and H.N. Koivo, "Contact micromanipulationSurvey of strategies," *IEEE/ASME Trans. Mechatron.* **14**(4), 504–514 (2009).
[2] Banerjee, A.G. and S.K. Gupta, "Research in automated planning and control for micromanipulation," *IEEE Trans. Automat. Sci. Eng.* **10**(3), 485–495 (2013).
[3] A. Kortschack, A. Shirinov, T. Trper and S. Fatikow, "Development of mobile versatile nanohandling microrobots: Design, driving principles, haptic control," *Robotica* **23**(4), 419–434 (2005).
[4] K. Oh, X. Liu, D. Kang and J. Kim, "Optimal design of a micro parallel positioning platform. Part I: Kinematic analysis," *Robotica* **22**(6), 599–609 (2004).
[5] M. Sinclair and I.A. Raptis, "Object Conveyance Control Algorithms with Spatially Changeable End Target Location Using Large-Scale Actuator Networks," *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 6052–6057.
[6] D. Reznik and J. Canny, "A Flat Rigid Plate is a Universal Planar Manipulator," *IEEE International Conference on Robotics and Automation, 1998*, vol. 2 (IEEE, 1998) pp. 1471–1477.
[7] D. Reznik, E. Moshkovich and J. Canny, *Building a Universal Planar Manipulator* (Springer, 2000) pp. 147–171.
[8] K.-F. Böhringer, B.R. Donald and N.C. MacDonald, "Upper and Lower Bounds for Programmable Vector Fields with Applications to Mems and Vibratory Plate Parts Feeders," *International Workshop on Algorithmic Foundations of Robotics (WAFR)* (Citeseer, 1996).
[9] A.E. Quaid and R.L. Hollis, *Design and Simulation of a Miniature Mobile Parts Feeder* (Springer, 2000) pp. 127–146.
[10] J. Beal and J. Bachrach, "Infrastructure for engineered emergence on sensor/actuator networks," *Intell. Syst.* **21**(2), 10–19 (2006).
[11] K.-F. Böhringer, B.R. Donald and N.C. MacDonald, "Programmable force fields for distributed manipulation, with applications to mems actuator arrays and vibratory parts feeders," *Int. J. Robot. Res.* **18**(2), 168–200 (1999).
[12] K.-F. Böhringer, B.R. Donald, L.E. Kavraki and F. Lamiraux, *A Distributed, Universal Device For Planar Parts Feeding: Unique Part Orientation in Programmable Force Fields* (Springer, 2000).
[13] S. Konishi, Y. Mita and H. Fujita, *Autonomous Distributed System for Cooperative Micromanipulation* (Springer, 2000) pp. 87–102.
[14] B.R. Donald, C.G. Levey and I. Paprotny, "Planar microassembly by parallel actuation of mems microrobots," *IEEE Microelectromech. Syst.* **17**(4), 789–808 (2008).
[15] M. Yim, J. Reich and A.A. Berlin, "Two Approaches to Distributed Manipulation," **In:** *Distributed Manipulation* (Springer, 2000) pp. 237–261.
[16] W. Saab, P. Racioppo and P. Ben-Tzvi, "A review of coupling mechanism designs for modular reconfigurable robots." *Robotica* **37**(2), 378–403 (2019).
[17] A. Nikou, G.C. Gavridis and K.J. Kyriakopoulos, "Mechanical Design, Modelling and Control of a Novel Aerial Manipulator," *IEEE International Conference on Robotics and Automation* (2015) pp. 4698–4703.
[18] A. Mohiuddin, T. Tarek, Y. Zweiri and D. Gan, "A survey of single and multi-UAV aerial manipulation," *Unmanned Syst.* **8**(2), 119–147 (2020).

[19] Z. Wang, E. Nakano and T. Matsukawa, "Cooperating Multiple Behavior-Based Robots for Object Manipulation," **In:** *Distributed Autonomous Robotic Systems* (1994) pp. 371–382.
[20] R.G. Brown and J.S. Jennings, "A Pusher/Steerer Model for Strongly Cooperative Mobile Robot Manipulation," *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3 (1995) pp. 562–568.
[21] P.J. Johnson and J.S. Bay, "Distributed control of simulated autonomous mobile robot collectives in payload transportation," *Autonom. Robots* **3**(1), 43–63 (1995).
[22] R. Grob and M. Dorigo, "Cooperative Transport of Objects of Different Shapes and sizes," *International Workshop on Ant Colony Optimization and Swarm Intelligence* (2004) pp. 106–117.
[23] S. Yao, M. Ceccarelli, G. Carbone and Z. Dong, "Grasp configuration planning for a low-cost and easy-operation underactuated three-fingered robot hand," *Mech. Mach. Theory* **129**, 51–69 (2018).
[24] K.B. Shimoga, "Robot grasp synthesis algorithms: A survey," *Int. J. Robot. Res.* **15**(3), 230–266 (1996).
[25] Z. Yan, N. Jouandeau and A.A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robot. Syst.* **10**(12), 399 (2013).
[26] E. Tuci, M. Alkilabi and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Front. Robot. AI* **5**(59) (2018).
[27] D. Stewart, "A platform with six degrees of freedom," *Proc. Inst. Mech. Eng.* **180**(2), 371–386 (1965).
[28] K.Y. Tsai, P.J. Lin and H.Y. Yu, "Developing contour surfaces of manipulators with specified dexterities," *Robotica* **29**(7), 371–386 (2011).
[29] H. Moon and J. Luntz, "Distributed manipulation of flat objects with two airflow sinks," *IEEE Trans. Robot* **22**(6), 1025 (2006).
[30] K. Varsos, H. Moon and J. Luntz, "Generation of Quadratic Force Fields from Potential Flow Fields for Distributed Manipulation," IEEE International Conference on Robotics and Automation, ICRA 2005 (IEEE, 2005) pp. 1021–1027.
[31] K. Varsos, H. Moon and J. Luntz, "Generation of quadratic potential force fields from flow fields for distributed manipulation," *IEEE Trans. Robot.* **22**(1), 108–118 (2006).
[32] K. Varsos, "Minimalist Approaches for Distributed Manipulation Force Fields in Flexible Part Handling," PhD thesis, University of Michigan (2006).
[33] G.J. Laurent, A. Delettre and N. Le Fort-Piat, "A new aerodynamic-traction principle for handling products on an air cushion," *IEEE Trans. Robot.* **27**(2), 379–384 (2011).
[34] A. Delettre, G.J. Laurent and N. Le Fort-Piat, "2-Dof Contactless Distributed Manipulation Using Superposition of Induced Air Flows," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2011) pp. 5121–5126.
[35] A. Delettre, G.J. Laurent and N. Le Fort-Piat, "A New Contactless Conveyor System for Handling Clean and Delicate Products Using Induced Air Flows," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2010) pp. 2351–2356.
[36] S. Akella, W. Huang, K.M. Lynch and M.T. Mason, "Sensorless Parts Feeding with a One Joint Robot," *Algorithms for Robotic Motion and Manipulation* (1996), pp. 229–237.
[37] J.E. Luntz, W. Messner and H. Choset, "Distributed manipulation using discrete actuator arrays," *Int. J. Robot. Res.* **20**(7), 553–583 (2001).
[38] S. Tadokoro, S. Fuji, T. Takamori and K. Oguro, "Distributed Actuation Devices Using Soft Gel Actuators," **In:** *Distributed Manipulation* (Springer, 2000) pp. 217–235.
[39] P. Song, V. Kumar and J.-S. Pang, "A Two-Point Boundary-Value Approach for Planning Manipulation Tasks," **In:** *Robotics: Science and Systems* (2005) pp. 121–128.
[40] C.-H. Yu and R. Adviser-Nagpal, *Biologically-Inspired Control for Self-Adaptive Multiagent Systems* (Harvard University, 2010).
[41] C.-H. Yu, K. Haller, D. Ingber and R. Nagpal, "Morpho: A Self-Deformable Modular Robot Inspired by Cellular Structure," *IEEE International Conference on Intelligent Robots and Systems* (2008) pp. 3571–3578.
[42] D. Leithinger and H. Ishii, "Relief: A Scalable Actuated Shape Display," *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction* (ACM, 2010) pp. 221–222.
[43] C. Laschi, B. Mazzolai and M. Cianchetti, "Soft Robotics: Technologies and Systems Pushing the Boundaries of Robot Abilities," *Sci. Robot.* **1**(1), eaah3690 (2016).
[44] D. Trivedi, A. Lotfi and C.D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Trans. Robot.* **24**(4), 773–780 (2008).
[45] G. Robinson and J.B.C. Davies, "Continuum Robots-A State of the Art," *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 4 (IEEE, 1999) pp. 2849–2854.
[46] E. Björnson, Ö. Özdogan and E.G. Larsson, "Sensorless parts feeding with a one joint robot," *IEEE Commun. Mag.* **58**(12), 90–96 (2020).

Cite this article: I. A. Raptis, C. Hansen and M. A. Sinclair (2022). "Design, modeling, and constraint-compliant control of an autonomous morphing surface for omnidirectional object conveyance", Robotica **40**, 213–233. https://doi.org/10.1017/S0263574721000473