ASEE 2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, Bridgpeort, CT, USA.

Distributed Manipulation Using Large-Scale Actuator Networks

Martin Sinclair
Department of Mechanical Engineering
University of Massachusetts Lowell
Lowell, Massachusetts 01854-5104
Email: Martin_Sinclair@student.uml.edu

Ioannis A. Raptis
Department of Mechanical Engineering
University of Massachusetts Lowell
Lowell, Massachusetts 01854-5104
Email: Ioannis_Raptis@uml.edu

***Abstract*—The field of Large-Scale Actuator Networks (LSANs) is a class of cyber-physical systems that is growing rapidly. One key application of LSANs is distributed manipulation. Distributed manipulation is a revolutionary application of this technology in the industrial sector. Examples of distributed manipulation include: vibrating plates, arrays of air jets, and mobile multi-robot teams. This paper explores a dynamic surface that adapts its shape by using an autonomous array of linear actuators to translate an object. This collective of actuators overcomes the limitations of the single degree of freedom that each actuator possesses, and results in a system that has many degrees of freedom. The experimental results illustrate the utility of this technique.**

I. Introduction

Large-Scale Actuator Networks (LSANs) are systems that are comprised of an array of interconnected and spatially distributed minimalistic actuators. These systems work as a collective to overcome the limitations of the individual components. As these systems are adopted, the requirement for secluded, expensive and complicated mechanisms is becoming obsolete. The collective nature is enhanced by the use of distributed control algorithms across the entire system. The classical form of centralized decision making is restrictive for many of the highly sophisticated processes used in modern systems. LSANs provide robust and reliable performance because they can function despite the failure of individual actuators.

A typical application of LSANs is distributed manipulation. There many types of distributed manipulators that have been used to develop actuator networks. The manipulators are typically simple (air jets, linear, or rotary actuators), and with recent advances in MEMS (Micro Electro Mechanical Systems) technology, these manipulators have shrunk in size allowing more manipulators to be placed in smaller areas. The increased manipulator density extends the capabilities of the system. In addition to this decrease in size, the advances in this technology are decreasing the cost of each manipulator.

Previous work on distributed manipulation includes a programmable vector field by Bohringer that moves and orients parts for an assembly line [1]. In another work by Bohringer [2], vibrating plates apply a frictional force to transport an object to a predefined point. In Quaid [3], work involved a network of motors that operate a small mobile vibratory part feeder. Two distinct approaches are investigated by Yim [4]: the PolyBot system for motion of non-planar objects, and a system of air jets that is designed to move planar objects. Self-organized systems of actuator networks are investigated by Beal [5]. In work by Konishi [6], a system of Autonomous Distributed Micro-machines (ADM) is developed to manipulate objects at a micro scale using a combination of actuators, sensors and circuits that can be fabricated by IC-compatible micro-machining. The use of soft gel actuators to move fragile objects is reported by Tadokoro [7]. Theoretical analysis of distributed manipulation by Reznik [8] uses a parallel part feeder to move multiple parts.

This paper presents a morphing surface that autonomously reconfigures its shape to translate an object to a desired location. The surface reconfiguration is controlled by an array of linear actuators that defines the height of the surface at given points in space. An overhead camera provides the object's position for feedback to the control algorithm. Dividing the surface into a grid increases the available range of pitch and roll angles for each individual cell. The control algorithm uses a hierarchical design where the higher level handles the path planning and the lower level determines how to generate the motion required to follow the path. Similar distributed manipulation systems have been reported on by Yu, Konishi, Reznik,and Tadokoro [6]–[9]. In general, the main difference between these systems and the surface described in this paper is that the latter allows the motion of the object in both directions of the plane while using a set of the most simplistic actuators. The two degrees of freedom provided by this surface allow the system to have multiple entrances and exits. This flexibility allows multiple types of objects to be moved through the same system because of its ability to deposit the items at multiple locations.

Section IIof the paper provides a description of the experimental platform. In section III, the equations of motion for the object are derived. In addition, this section also describes a simple control algorithm that moves the object to a reference location over a single cell. In section IV, the high level controller for a multi-cell surface is presented. Finally, experimental results and conclusions are given in sections V and VI.

II. Experimental Platform

In order to examine the capabilities of distributed manipulation using large-scale networks as well as any potential shortcomings of such systems, an experimental testbed platform was developed. The system under investigation is an elastic surface that changes its shape by adjusting the height of ten linear actuators. The placement of the actuataors yields four equal cells in a 'T' shaped configuration. This configuration was chosen because it allows for all of the basic movements that an industrial application would conceivably need to perform. Any basic movement on a grid can be performed using this shape. The T' shape was found to be the shape that used the fewest actuators while allowing for all of the necessary path elements.

The testbed employed a pegboard as a mounting fixture to position all of the components. Each actuator was held in a vertical position using two threaded rods to provide stability and support. The actuator model used was the L12-I from Frigelli with a 100 mm stroke. The Frigelli L12 actuators were chosen due to their low cost and relative high speed (23 mm/s), allowing them to respond quickly to control the motion of the object. Though these actuators have a reduced strength (43 N) compared with other potential solutions, their strength was sufficiently powerful to move and configure the surface.

A sheet of spandex (10%) and polyester (90%) was secured to the end of the actuators to provide compliance to accommodate the variations in the distance of the actuators tips, as well as to secure the surface panels. Corrugated plastic surface panels were placed on top of the spandex sheet due to their low coefficients of static and kinetic friction. The low friction factor allows the object to move at the expected local slopes. The corrugated plastic was chosen after unsuccessful tests of multiple other materials ranging from neoprene to latex. The object itself was selected after testing many different candidates (ranging from wood cubes to Ping-Pong balls). The object (seen in Figure 1) that provided the most controllable and repeatable motion was a polycarbonate half sphere.

The system was controlled using an Arduino Uno microprocessor. This microcontroller was selected for two key reasons. The first reason was the simplicity of the controller that minimized the programming learning curve. The controller has a large user community that provides detailed documentation regarding the limitations and capabilities of the hardware. The second reason for selecting the Arduino processor was the cost. An extension to the system would require additional controllers to be added. To minimize the cost to expand the system, it was desirable to use low cost controllers..

The position measurement of the object was generated by an external and commercially available visual tracking system called Roborealm that was installed in a computer system. The vision sensor was a simple Live! Cam inPerson HD VF0720 web-camera. Data was sent from the computer to the Arduino microcontroller via a serial connection through an XBee wireless RF module. The assembled surface can be seen in Figure 1.

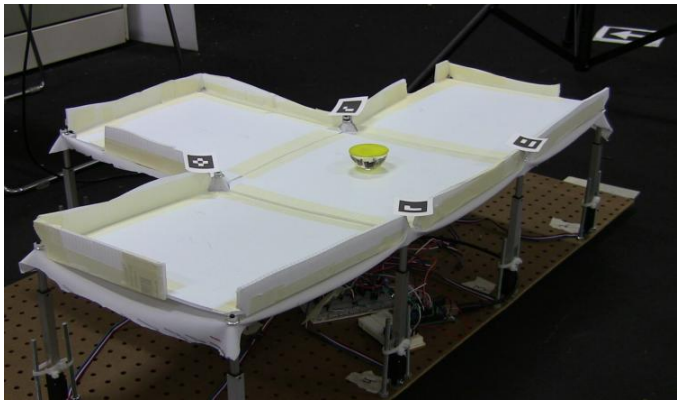

Fig. 1. This figure illustrates the apparatus of the distributed manipulation experimental platform.

III. Single Cell Modeling, Dynamics and Control

This section presents the theoretical model of a single cell that is operated by four actuators. The kinematic and dynamic equations of both the surface and the object are derived. Based on the equations of motion of the object, a simple feedback control law is designed for moving the object at a reference point located at the edge of the cell. This low level control law will be part of the complete hierarchical controller for the entire surface, described at section IV.

Mathematical Notation

The abbreviations C_t and S_t with $t \in \mathbb{R}$ represent the trigonometric functions $\cos(t)$ and $\sin(t)$, respectively. The superscript T indicates the transpose of a vector or matrix. The operands $||\cdot||$, $|\cdot|$ denote the Euclidean norm and the $|\cdot|_1$ norm of a vector, respectively.

A. Surface Kinematics

The orientation in space of each cell is defined by the four tips of the actuators. The four linear actuators are located in a rectangular configuration of width W and length L. The coordinates of the actuators are defined with respect to an inertial frame with its origin located at the lower left actuator of the cell when the actuator has its minimum length. The maximum length of a actuator is denoted by l. This reference frame remains stationary while the actuators change their length. The inertial frame is defined by its origin and three orthonormal vectors, hence $\mathcal{F}_I = \left\{O_I, \vec{i}_I, \vec{j}_I, \vec{k}_I\right\}$. The directions of the vectors $\vec{i}_I$ and $\vec{j}_I$ can be seen in Figure 2, while $\vec{k}_I$ points upward such that $\left\{\vec{i}_I, \vec{j}_I, \vec{k}_I\right\}$ constitutes a right handed Cartesian coordinate frame ($\vec{k}_I = \vec{i}_I \times \vec{j}_I$). The actuators are enumerated in an anticlockwise manner. The coordinates of the actuator tips with respect to the inertial frame $\mathcal{F}_I$ are given below:

$$p_1^I = \begin{bmatrix} 0 \\ 0 \\ z_1 \end{bmatrix} \quad p_2^I = \begin{bmatrix} W \\ 0 \\ z_2 \end{bmatrix} \quad p_3^I = \begin{bmatrix} W \\ L \\ z_3 \end{bmatrix} \quad p_4^I = \begin{bmatrix} 0 \\ L \\ z_4 \end{bmatrix} \tag{1}$$

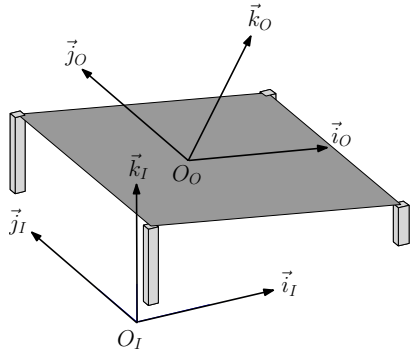

Fig. 2. The two reference frames ($\mathcal{F}_I$ and $\mathcal{F}_O$) for a single cell configuration.

In order for the object to transport over the surface, the actuators should form a flat two-dimensional plane at all time instances. Therefore, following fundamental algebraic definitions, the tips of the actuators should satisfy the following determinant equation:

$$\begin{vmatrix} p_4^T - p_1^T \\ p_4^T - p_2^T \\ p_4^T - p_3^T \end{vmatrix} = 0 \tag{2}$$

Substituting in (2) the coordinates given in (1), one obtains the following equation:

$$z_3 = -z_1 + z_2 + z_4 \tag{3}$$

The above equation indicates that in order for the cell to form a flat plane, the height of one of the actuators is constrained by (3). The coordinates z_i with $i = 1, \ldots, 4$ represent the length of the actuators on the $\vec{k}_I$ direction. Each actuator length is regulated by a DC motor. Let z_i^{com} with $i = 1, \ldots, 4$ denote the control signal that determines the desired length for each of the four linear actuators. The dynamics of each actuator are represented satisfactorily by a first order differential equation, thus:

$$\tau \dot{z}_i + z_i = z_i^{com} \quad \text{with} \quad i = 1, \ldots, 4 \tag{4}$$

where τ is a positive number and represents the time constant of the actuator dynamics. For the sake of simplicity in the subsequent analysis we will assume that z_i is the same with z_i^{com} and disregard the actuator dynamics.

B. Object Kinematics

To derive the equations of motion in the inertial coordinates we define as second frame of reference denoted $\mathcal{F}_O = \left\{O_O, \vec{i}_O, \vec{j}_O, \vec{k}_O\right\}$ that its center O_O is rigidly attached to the object. The orthonormal vectors $\vec{i}_O$, $\vec{j}_O$ are changing their orientation with time such that they constantly lie on the plane of the surface. When all the actuators are leveled, the frames $\mathcal{F}_I$ and $\mathcal{F}_O$ are aligned. The object's orientation at any instant in time may be obtained by performing three consecutive rotations of $\mathcal{F}_I$ until it is aligned with $\mathcal{F}_O$. The orientation of $\mathcal{F}_O$ is produced by rotating $\mathcal{F}_I$ a ϕ angle about the axis $\vec{i}_I$, then, an angle θ about $\vec{j}_I$ and, finally, an angle ψ about the axis $\vec{k}_I$. In this convention, the angles ϕ, θ and ψ are commonly denoted as roll, pitch and yaw angles, respectively. Positive direction of each angle refers to the right-hand rule about the respective axis. Since the object does not change its heading for the particular application $\psi = 0$.

The *rotation matrix* is a systematic way to express the relative orientation of the two frames. The rotation matrix R is parameterized with respect to the three angles: roll (ϕ), pitch (θ) and yaw (ψ) and it maps vectors from the object fixed frame $\mathcal{F}_O$ to the inertia frame $\mathcal{F}_I$. From standard results and considering that $\psi = 0$ one has:

$$R = \begin{bmatrix} C_\theta & S_\theta S_\phi & C_\phi S_\theta \\ 0 & C_\phi & -S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{5}$$

When the rotation matrix is parameterized by the roll-pitch-yaw angles, singularities occur at $\theta = \pm\pi/2$. Due to the constraint of the actuator lengths, this case does not apply in the morphing surface.

Each cells orientation is changed by controlling the four corner actuators. The above description is not sufficient then, if a relationship between the orientation angles and the actuator lengths is not established. By construction, the columns of the orientation matrix express the coordinates of the basis vectors $\vec{i}_O, \vec{j}_O, \vec{k}_O$ with respect to the inertial frame. Thus:

$$R = \begin{bmatrix} i_O^I & j_O^I & k_O^I \end{bmatrix} \tag{6}$$

In the above equations, the superscript indicates the reference frame that the coordinate vector is expressed with respect to. Therefore, in order to associate z_1, z_2 and z_3 with ϕ, θ one needs to express the right-hand side of (6) with respect to the tip coordinates and then equate with the entries of (5). The first step is to define the basis vectors of $\mathcal{F}_O$. By definition the basis vectors $\vec{i}_O$, $\vec{j}_O$ of the object-fixed frame lie in the cell's plane. The latter can be uniquely defined by the two vectors $I^I = \begin{bmatrix} W & 0 & z_2 - z_1 \end{bmatrix}^T$ and $J^I = \begin{bmatrix} 0 & L & z_4 - z_1 \end{bmatrix}^T$. These vectors are also the two sides of the cell that connect actuators $1, 2$ and $1, 4$. A quick inspection shows that the two vectors are not orthonormal for every $z_i \in \begin{bmatrix} 0 & l \end{bmatrix}$with $i = 1, 2, 4$. Therefore, the basis vectors for the object-fixed frame have to be defined. The first obvious choice is the unitary vector $i_O^I = I^I / ||I^I||$. By definition $\vec{k}_O$ is normal to both $\vec{I}$ and $\vec{J}$. Therefore, the choice $k_O^I = K^I / ||K^I||$ with:

$$K^I = \begin{bmatrix} \frac{z_1 - z_2}{W} & \frac{z_1 - z_4}{L} & 1 \end{bmatrix}^T \tag{7}$$

will satisfy $I^I \cdot \left(K^I\right)^T = J^I \cdot \left(K^I\right)^T = 0$. Equating the (1,1), (3,1) and (2,3), (3,3) components of R the following kinematic relations are derived:

$$S_\theta = \frac{z_1 - z_2}{W} C_\theta \tag{8}$$

$$-S_\phi = \frac{z_1 - z_4}{L} C_\theta C_\phi \tag{9}$$

The above kinematic equations that relate the coordinates of the actuator tips with the orientation angles of the plane will be used at a later point to derive the equations of motion for the object.

C. Object Dynamics

In this section, the equations of motion for the object are derived. The object's position with respect to the inertial frame is $p^I = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$. The net forces applied to the object with respect to the inertia frame are denoted by ΣF^I. From Newtons second law the equations of motion for the object are:

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

where m is the mass of the object. The object on top of the surface is subject to three forces: i) the components of the object's weight that lie on the surface plane, ii) the friction that opposes the object's motion, and iii) the reaction force from the surface that is normal to the plane, facing upward. The object dynamics are manipulated by changing the orientation of the surface using the actuators. The weight of the object (expressed in the inertial frame) is $W^I = -\begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T$ .Using (5), the weight component in the object-fixed frame are given by

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

The reaction from the surface to the object is opposite and equal to the weight component in the $\vec{k}_O$ direction, thus:

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

Finally, the friction force from the surface opposes the motion of the object and is modeled as:

$$F^B = -bv^B \tag{13}$$

where $b > 0$ is the friction coefficient and $v^B \in \mathbb{R}^3$ is the velocity of the object expressed in the object-fixed frame. The component of v^B in the $\vec{k}_O$ direction is zero since the object always lies on the surface. Using fundamental arguments, it is easy to show that $F^I = -bv^I$ where $v^I = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \in \mathbb{R}^3$ is the object's velocity expressed in the inertial frame. The components of the three forces applied to the object are illustrated in Figure 3.Substituting (11)-(13) to (10) one has:

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

Applying the above equation to the kinematic relations (8) and (9) that were derived in the previous section, the dynamic equations of the object in the $\vec{i}_I$ and $\vec{j}_I$ directions become:

$$\ddot{x} + b\dot{x} + gC_\theta^2 C_\phi^2 \frac{z_1 - z_2}{W} = 0 \tag{15}$$

$$\ddot{y} + b\dot{y} + gC_\theta^2 C_\phi^2 \frac{z_1 - z_4}{L} = 0 \tag{16}$$

The last step is to control the actuator tips such that the object is transported to the reference location.

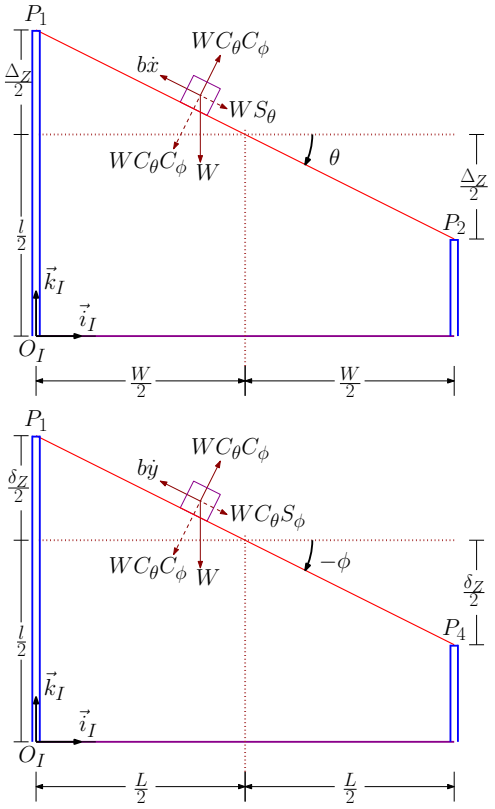

Fig. 3. This Figure illustrates the two side views of a single cell with the force components that act on the object.

D. Single Cell Control

The goal of the control algorithm is to transport the object to a reference location with coordinates x_r, y_r expressed in the inertial coordinates. The reference coordinate in the $\vec{k}_I$ directions is of no importance since the object's z coordinate is constrained by the surface orientation. The main idea is to move the object in one of the sides of the surface such that it will transition to an adjacent cell. To guarantee that the object will transition to a different cell, the reference point may be placed slightly outside the bounds of the surface. To proceed with the analysis in the controller, we make a coordinate change to the error variables $e_x = x - x_r$ and $e_y = y - y_r$. From (15) and (16) the error dynamics become:

$$\ddot{e}_x + b\dot{e}_x + C_\theta^2 C_\phi^2 \frac{g}{W} \Delta_z = 0 \tag{17}$$

$$\ddot{e}_y + b\dot{e}_y + C_\theta^2 C_\phi^2 \frac{g}{L} \delta_z = 0 \tag{18}$$

where $\Delta_z = z_1 - z_2$ and $\delta_z = z_1 - z_4$. The error dynamics are described by two identical second order nonlinear differential equations. In addition, $C_\theta^2 C_\phi^2 \in \begin{pmatrix} 0 & 1 \end{bmatrix}$ since $\theta, \phi \in \begin{pmatrix} -\pi/2 & \pi/2 \end{pmatrix}$. Therefore, any feedback law of the position error can achieve the control objective. Due to the limitation of the actuator lengths, we chose the following saturated feedback functions:

$$\Delta_z = K_x sat_W\, e_x \qquad\qquad \delta_z = K_y sat_L\, e_y$$

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

$$sat_M(x) = \begin{cases} x & |x| \leq M \\ sign(x) \cdot M & \text{else} \end{cases}$$

The final step is to configure the actuator lengths such that they comply with the definitions of Δ_z and δ_z. The following choice will guarantee that the surface will orient about the axis that lies in the middle of the width and length of the cell:

$$z_1 = \frac{l}{2} + \frac{\Delta_z}{2} + \frac{\delta_z}{2} \qquad z_2 = \frac{l}{2} - \frac{\Delta_z}{2} + \frac{\delta_z}{2} z_2$$
$$z_3 = \frac{l}{2} - \frac{\Delta_z}{2} - \frac{\delta_z}{2} \qquad z_4 = \frac{l}{2} - \frac{\Delta_z}{2} + \frac{\delta_z}{2} z_4$$

The motion of the object is damped by the friction coefficient b. However, even the in the absence of friction the above controller can guarantee that $(x, y) \to (x_r, y_r)$. To inject the dynamics with additional damping, a velocity feedback term may be added to the control inputs. To guarantee that the actuators tips lie in $\begin{bmatrix} 0 & l \end{bmatrix}$ interval the control inputs are bounded by $|\Delta_z|, |\delta_z| \leq l/2$. Therefore, the feedback gains are limited to $K_x \leq l/2W$ and $K_y \leq l/2L$.

IV. Control Algorithm

This section presents the control algorithm that autonomously adjusts the surface's actuator lengths. The control algorithm can be separated into two main modules: the first is the "master control" and the second is the "update control" module. Both modules take place at each time step, with the active logic being determined by the location of the object. The master control module determines the heights of the actuators that are connected with the cell where the object lies. Based on their configuration, the update control law adjusts the lengths of the rest actuators in the grid such that all cells form flat planes.

The master control algorithm is broken into several parts. The first part involves the acquisition of the position measurement from the vision system. This input is the x and y inertial coordinates of the object. Using the position data and the location of the inertial frame origin (lower left corner of the grid), the controller uses a simple logic to find the cell that the object lies to. The algorithm assumes that the width and length of all the cells are fixed. Denote by i the master cell's row and by j its column in the grid. The algorithm that determines i, j based on (x, y) is given below:

$$i = int\left(\frac{x + W - 1}{W}\right) \quad j = int\left(\frac{y + L - 1}{L}\right) \tag{19}$$

The mapping of the master cell is illustrated in Figure 4. After the master cell is identified, the controller checks to see if it is the same as the "target" cell. The target cell is the one where the target exit is located. If they are different cells, the controller identifies the cell closest to the target cell that is adjacent to the master cell and calls this the "interim target" cell.

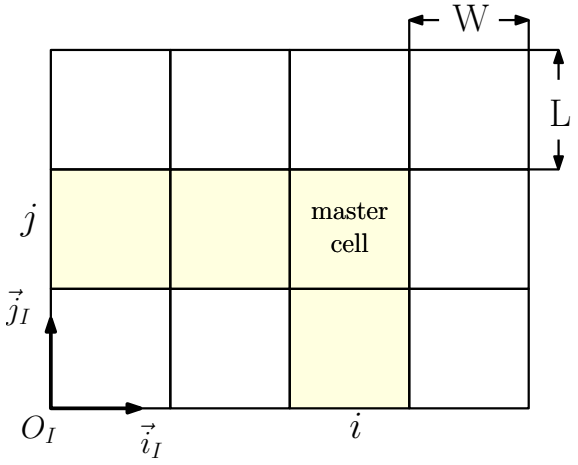

Fig. 4. Mapping of the surface's cells.

The master controller determines a point on the border of both this interim target cell and the master cell, and sets this as the reference destination of the object. For simplicity, the reference point is set as the midpoint of the common border. In the event that the object and target cells are the same, the master controller sets the reference point equal to the target's location. With the reference point established, the controller then calculates the actuator lengths using the local control law that is described in section III-D.

Once the heights of the master cell's actuators are determined, the control algorithm transmits these values such that all the elements in the grid can update their lengths and maintain continuity of the surface at the cells' edges. The update control module works by propagating the heights of the master cell and its position to all adjacent cells. The master cell defines four zones in the surface, namely $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ (Figure 5). The most practical and consistent way to update the remaining actuators in the grid is to level all components that belong to the same zone. The collective length of each zone is defined by the corresponding actuator in the master cell. Using this method, it is possible to propagate motion from the master cell though the entire network.

V. Experimental Results

A single cell was able to direct the object successfully to its target point. This success prompted the expansion of the network and exploration of the multi-cell 'T' shaped surface illustrated in Figure 1. The multi-cell surface was also able to move the object successfully to its designated target. The tested paths were primarily chosen to demonstrate the base movements that are possible on the surface. The reference path requires that the object makes a U-turn across the cells to reach the target exit. This case resembles a scenario were the object has to be transported around a hole or a wall.

A 2-D top view of the object's actual trajectory based on experimental data is illustrated in Figure 6. The object successfully reached the target location (exit). In addition, this figure shows that the object moves over the midpoint

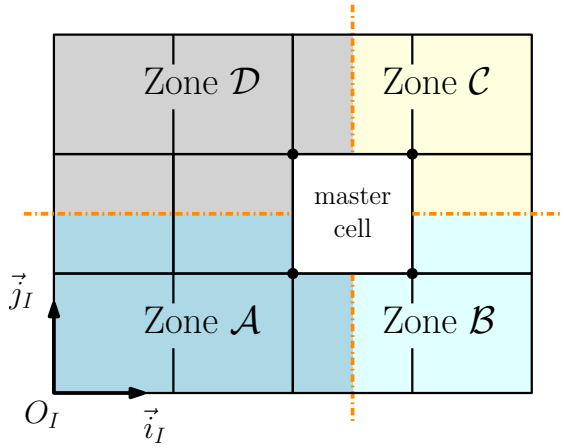

Fig. 5. This figure illustrates the cross and the four zones that are used to determine the propagation of heights for continuity.

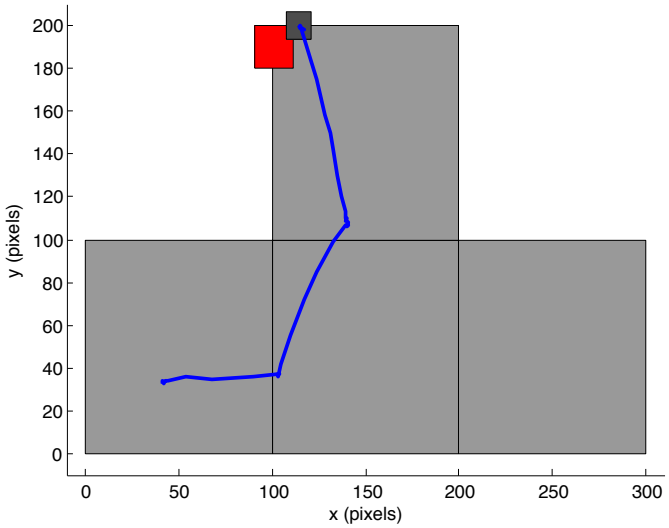

Fig. 6. 2-D top view of the object's planar motion based on experimental data. The panels of the surface are shown in light gray, the target zone in red, the object is shown in dark gray, and the blue line is the path that the object followed.

of the common side between adjacent cells. This point is set from the high level control algorithm as the reference location for transitioning the object across adjacent cells. The 3-D visualization of the morphing surface's shape for the previous experiment at different time instances is illustrated in Figure 7.

VI. Conclusions

The prototype reconfiguring surface demonstrates that simple components and simple algorithms can be used to create a fully-functional autonomously operating LSAN. The use of inexpensive and off-the-shelf components in this prototype indicates that LSAN systems can be created and operated without the need for complex components. Future work will focus on optimizing the control algorithm so that the system can minimize travel time and place the object with greater accuracy. Additionally, new control algorithms will be tested to investigate potential new uses for the system.

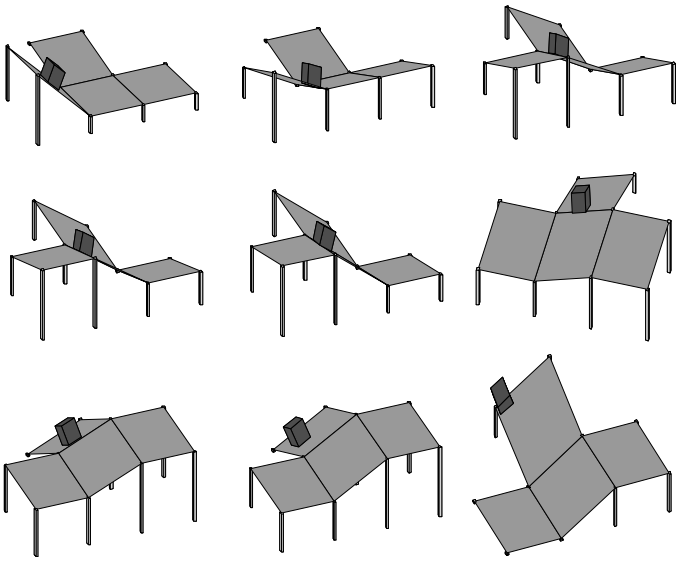

Fig. 7. 3-D visualization of the surface and the object for nine different time instances based on experimental data.

References

[1] K.-F. Bohringer, B. R. Donald, and N. C. MacDonald, "Programmable force fields for distributed manipulation, with applications to mems actuator arrays and vibratory parts feeders," *The International Journal of Robotics Research*, vol. 18, no. 2, pp. 168–200, 1999.

[2] 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," in *In International Workshop on Algorithmic Foundations of Robotics (WAFR*. Citeseer, 1996.

[3] A. E. Quaid and R. L. Hollis, "Design and simulation of a miniature mobile parts feeder," in *Distributed Manipulation*. Springer, 2000, pp. 127–146.

[4] M. Yim, J. Reich, and A. A. Berlin, "Two approaches to distributed manipulation," in *Distributed Manipulation*. Springer, 2000, pp. 237–261.

[5] J. Beal and J. Bachrach, "Infrastructure for engineered emergence on sensor/actuator networks," *Intelligent Systems, IEEE*, vol. 21, no. 2, pp. 10–19, 2006.

[6] S. Konishi, Y. Mita, and H. Fujita, "Autonomous distributed system for cooperative micromanipulation," in *Distributed Manipulation*. Springer, 2000, pp. 87–102.

[7] S. Tadokoro, S. Fuji, T. Takamori, and K. Oguro, "Distributed actuation devices using soft gel actuators," in *Distributed Manipulation*. Springer, 2000, pp. 217–235.

[8] D. Reznik, E. Moshkovich, and J. Canny, "Building a universal planar manipulator," in *Distributed Manipulation*. Springer, 2000, pp. 147–171.

[9] C.-H. Yu and R. Adviser-Nagpal, *Biologically-inspired control for self-adaptive multiagent systems*. Harvard University, 2010.