A Delta-Inspired, Multi-Degree of Freedom Robot with Deformable Surface Tracking for Bioprinting Applications

Ray Culp, Gavin Dauwalter, Reed Dunbar, Reed Johnson, Amey Joshi, Jason Kelly, Manikanta Pallantla, Rebecca Smith

Abstract— **This paper presents the design and constuction of a 6 DOF HEXA robot, as well as a thin plate spline based algorithm for tracking nonlinear deformation of a 2D surface. The thin plate spline algorithm was used to provide position commands to the robot which were implemented using a simple proportional controller.**

I. INTRODUCTION

Biomedical engineering has combined tissue engineering with additive manufacturing, yielding the ability to 3D print a tissue scaffold for human anatomical organs available for transplant [1]. Traditionally, bioprinting techniques are open-loop processes which are limited to printing on a stationary rigid surface. However, recent work has demonstrated the ability to 3D print bio-compatible material directly onto an unpredictably moving surface using a closed-loop tracking algorithm [2][3]. In [2], the tracking is performed using a leap motion sensor, which is specifically designed to track the motion of a human hand, but is unable to track the motion of anything else. In [3], a more robust, Perspective-n-Point (PnP) method is used which allows for the tracking of any rigid surface. While these methods accounted for unpredictable motion of the print surface, neither algorithm was able to account for any deformation (e.g. stretch, shear, wrinkling, etc.) of the print surface. This is a major limitation in the realm of bioprinting, given that any unconstrained biological surface is likely to undergo some type of deformation.

An additional limitation of previous closed-loop 3D bioprinting techniques is inherent to the robots employed to perform the task. In [2], the robot used was a 3D gantry system capable of translation in the x, y, and z-direction. Due to the 3 DOF nature of the robot, the system was unable to achieve any rotation, thus limiting it to printing on a surface of a constant orientation. Similarly, in [3], a Delta robot was used. Since Delta robots are also only capable of translation in the x, y, and z direction, this work was also limited to printing in a constant orientation.

To address these limitations, the work presented in this paper is twofold: (1) a 6 DOF robot to be used in bioprinting applications, and (2) an algorithm for tracking a deformable 2D surface.

A. *Robot Design*

Because bioprinting is used in medical applications, a high level of accuracy and precision is required. Additionally, because the system is tracking unpredictable motion and deformation of the print surface, the end-effector of the robot must be able to achieve a fast enough response to compensate for this unpredictable motion. These factors must be taken into consideration when designing a robot for the purpose of closed-loop bioprinting.

A delta robot is a type of parallel robot which is composed of several kinematic chains connected to the end effector of the robot. This offers many advantages to the more common serial robotic manipulators, such as robotic arms. Due to having all the actuators mounted on the base, it is able to achieve high velocities, fast accelerations, and increased stiffness. Additionally, delta robots are able to achieve high levels of precision. Because of their ability to attain high precision at fast speeds, delta robots are common in pick and place applications in manufacturing, as well as in 3D printing [4].

While the high speed and precision of the delta robot make it an appealing option for bioprinting, traditional delta robots are limited to only three degrees of freedom (translation in x, y, and z). Since closed-loop bioprinting aims to compensate for unpredictable, 6 DOF motion and deformation, it is necessary for the robot employed to be able to achieve the three rotational degrees of freedom, in addition to translation. Because of this, the objective of this paper is to present a robot which has the advantages of a delta robot (high speed and precision), while also attaining more than three degrees of freedom.

There are several creative ways to manipulate the delta robots joints to achieve the full 6 degrees of freedom. In [5], two delta robots with a shared end-effector are used to get to 6 degrees of freedom. In this case, the base of the first robot is directly above the base of the second, and the links of both robots meet at a common end-effector. Additionally, the second robot is able to rotate about its z-axis, causing a misalignment between the two robots. While this dual-robot system is able to achieve all 6 degrees of freedom, the method used is complex and has many moving parts. Additionally, the system is over actuated (7 actuators for 6 degrees of motion), which is unnecessary for bioprinting purposes.

Another approach to increasing the degrees of freedom of a delta robot is presented in [6]. In this case, additional degrees of motion are achieved by adding a rotational actuator in the first link in each kinematic chain on the robot. The primary drawback to this solution is that the first actuator will have to carry the load of the second actuator, resulting in the loss of some of the dynamic advantages the delta robot provides (e.g. speed and dexterity).

In [7] a HEXA parallel robot is proposed. This robot is similar to the traditional 3 DOF delta robot in that all the actuators are mounted on the base. However, instead of having 3 actuators, the HEXA has 6 actuators, with three groups of two motors mounted side by side. The unique design of this robot allows the end effector to achieve all 6 DOF while still preserving the advantages of a delta robot. Because this model achieves all 6 degrees of motion without either over-actuating the system or losing the advantages of the traditional delta robot, this seems like an optimal solution for the bioprinting robot design.

While the HEXA parallel robot seems to be an ideal design for a bioprinting robot, previous HEXA robots have been used for open-loop processes with easily tunable control algorithms. Additionally, previous HEXA robots have been manufactured on a scale much larger than what is needed for bioprinting applications. In light of this, the work presented in this paper demonstrates the closed-loop control of a small-scale HEXA robot.

B. Tracking Algorithm

Most tracking algorithms rely on a rigid fiducial template in order to estimate the pose of the camera relative to the template. For example, maintaining a rigid fiducial template is crucial to the success of the PnP algorithm presented in [3].

In fields other than bioprinting, different methods have been used to track surface deformation. For example in [8], approximate deformations were interpolated using changes in fiducial location within a point cloud. While this worked with reasonable accuracy for the application presented in the paper, the deformations were approximated as linear transformations between fiducials. Since the deformation of biological surfaces is highly non-linear, a linear transformation/interpolation may not be sufficient for 3D bioprinting purposes.

Because a linear transformation is not sufficient to capture the deformation of a biological surface, a nonlinear transformation is required. One such approach is presented in [9], in which an image is deformed using a thin plate spline (TPS) technique. The thin plate spline technique is based on the principle of minimizing the bending energy which results from point deformation. An example of the TPS technique can be seen in the following figure.

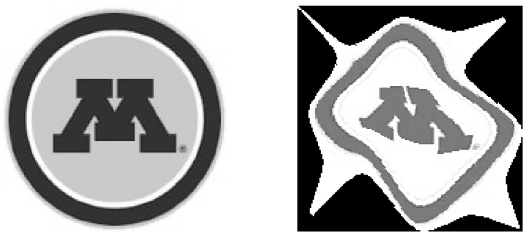
Fig. 1. *Left:* Undeformed image, *Right:* Image deformed using thin plate spline warping

As can be seen, the deformed image on the right is able to capture nonlinear deformations. While TPS warping is a common technique in computer vision, it is not commonly used as a means of tracking surface deformation. Additionally, it is unknown whether the nonlinearities resulting from the TPS warping will adequately model the nonlinearities resulting from a physically deforming surface.

To address the questions concerning the TPS method for tracking applications, the second part of this paper presents the results of using a TPS technique to track the deformation of an elastically deforming 2D surface.

II. Methods and Results

This section presents the methodology behind the design and manufacture of the 6 DOF HEXA robot, as well as the methods and results of the thin plate spline tracking algorithm.

A. Robot Design

The HEXA parallel robot design includes six actuators at the base, that are each connected to the end-effector (EE) by pairs of links. The links will be sets of RSS joints (revolute at the base actuator, spherical at the second joint, and spherical at the EE). The traditional delta robot has the EE connections in a parallel plane with the revolute motion at the base actuator. However, in the HEXA design, that actuating plane will break at the second joint. Figure 2 shows the positioning of the actuator joints (q) and the spherical joints (s) on the EE.

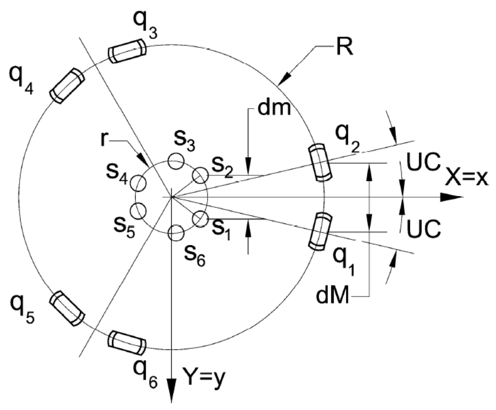

Fig. 2. HEXA Robot link set-up Design [7]

This diagram drove the designing decisions for the actuator locations and EE design. By keeping the pairs of motors at optimal distance from each other, keeping the angle UC in the diagram optimal, rotation about the Z-axis is possible. Additionally, having independent control of all six of the EE mounting locations allows for roll and pitch motions about either the X-axis, Y-axis, or any axis in the XY-plane. By configuring the links as shown above, rotation can occur in any direction, giving the robot translational movement in three dimensions as well as a roll, pitch, and yaw movements.

In order to track a moving surface, a single camera and three lasers were mounted to the end-effector of the robot (see the following section for how those components are used). Figure 3 depicts these components mounted to the end-effector. The entire assembly including end effector, base and joints can be seen in the figure 4.

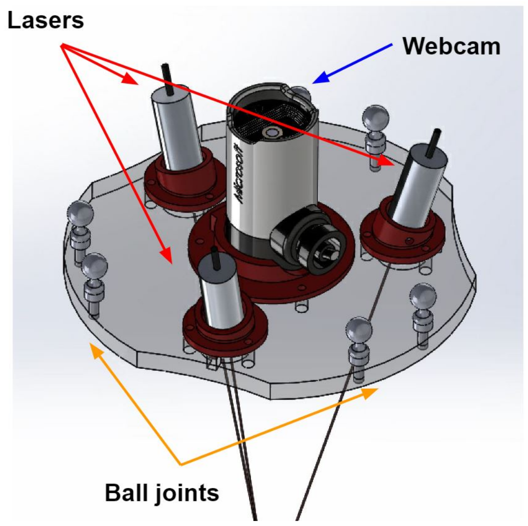

Fig. 3. Robot EE Closeup (camera, lasers, etc)

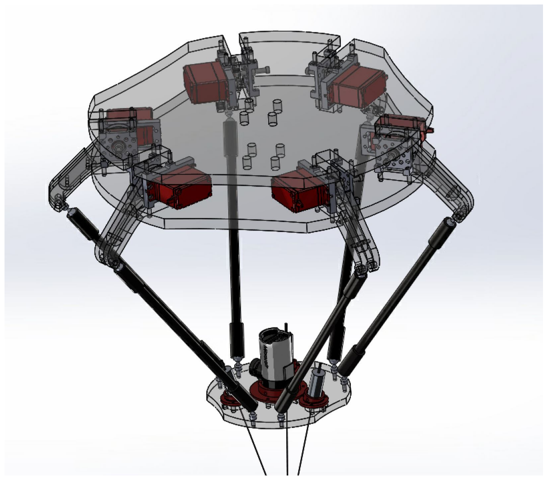

Fig. 4. HEXA Robot Design

B. Inverse Kinematics

In order to control the position and orientation of the end-effector, the inverse kinematics must first be determined. The inverse kinematics described here are based on the geometry given in the following figure.

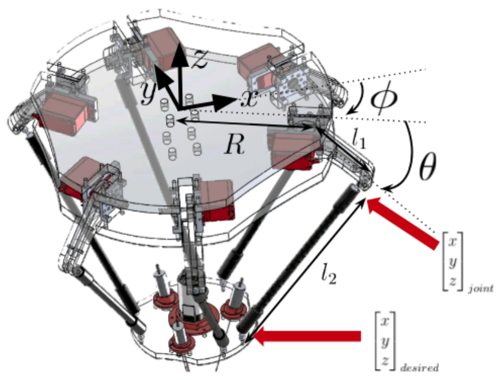

Fig. 5. HEXA Inverse Kinematics Geometry

As can be seen in the figure, R represents the distance from the center of the base (considered the origin of the coordinate frame) to the motor shaft, ϕ represents the angle (measured clockwise) from the positive x-axis to the motor, and θ represents the angle at which the motor shaft is rotated. The upper and lower link lengths are denoted l_1, l_2 respectively. The coordinate $[x \quad y \quad z]^T_{\text{joint}}$ represents the position of the elbow joint, as shown, and the coordinate $[x \quad y \quad z]^T_{\text{desired}}$ represents the coordinate at which link 2 connects to the end-effector. Note that there is a unique value of $[x \quad y \quad z]^T_{\text{desired}}$ corresponding to each motor which is determined by the desired translation and rotation of the end effector. From the geometry given in the figure, the following can be determined by inspection.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{joint}} = \begin{bmatrix} (R + l_1 \cos\theta)\cos\phi \\ -(R + l_1 \cos\theta)\sin\phi \\ l_1 \sin\theta \end{bmatrix} \tag{1}$$

Next, it can be seen that the magnitude of the vector from p_{joint} to $p_{desired}$ is equal to l_2. From this, we have the following equation.

$$(x_{joint} - x_d)^2 + (y_{joint} - y_d)^2 + (z_{joint} - z_d)^2 = l_2^2 \quad (2)$$

The preceding two equations can be combined to yield the following solution for θ.

$$\theta = \tan^{-1}\left[\frac{b + \sqrt{a^2 + b^2 - c^2}}{a + c}\right] \quad (3)$$

Where the constants a, b and c are as follows.

$$\begin{aligned} a &= 2l_1[-(x_d - R\cos\phi]\cos\phi + (y_d - R\sin\phi)\sin\phi \\ b &= -2l_1 z_d \\ c &= l_2^2 - (x_d - R\cos\phi)^2 + (y_d + R\sin\phi)^2 - z_d^2 - l_1^2 \end{aligned} \quad (4)$$

Thus, the commanded motor angle can be determined from the geometry of the robot and the desired position of the point at which the second link connects to the end effector. As mentioned previously, this desired position is unique for each motor-link set, and can be determined from the desired position and rotation of the end-effector. Let p_{EE} denote the position of the center of the end-effector (expressed in the base coordinate frame), and let $d_{i,nominal}$ represent the vector from the center of the end-effector to the ball joint on the end-effector corresponding to the i'th motor measured in the nominal robot position (no rotation). When there is no rotation, the position of the joint on the end-effector corresponding to the i'th motor can be expressed as follows.

$$p_i = p_{EE} + d_{i,nominal} \quad (5)$$

Note that $d_{i,nominal}$ can be determined directly from the geometry of the robot end-effector. When the end-effector is rotated, the vector from the i'th joint on the end-effector to the center of the end-effector is also rotated. Denoting this rotation by the rotation matrix C_{base}^{EE}, the rotated position of the i'th joint on the end-effector can be expressed as follows.

$$p_i = p_{EE} + C_{base}^{EE} d_{i,nominal} \quad (6)$$

Therefore, given a desired position of the end-effector (p_{EE}), a desired end-effector rotation (C_{base}^{EE}), and the geometry of the end-effector, the position of each joint on the end-effector can be uniquely determined. Thus, using the equations derived in this section, each motor can be commanded to achieve an arbitrary end-effector position and orientation within the robot's reachable workspace. These equations were tested and validated both in simulation, as well as with the physical system.

C. Reachable Workspace

Using a MATLAB simulation of the HEXA robot, the reachable workspace was found by iterating through each of the possible positions for the servo motor angles, and using the inverse kinematic equations to find the end-effector position at those specific angles. This is shown in Figure 6 for the Y-Z plane, plotting valid values of position for the robot, with the use of an interpolation function for better visualization. Figure 7 illustrates the reachable space of the center of the end-effector as well as each attached arm of the robot in the X-Y plane.

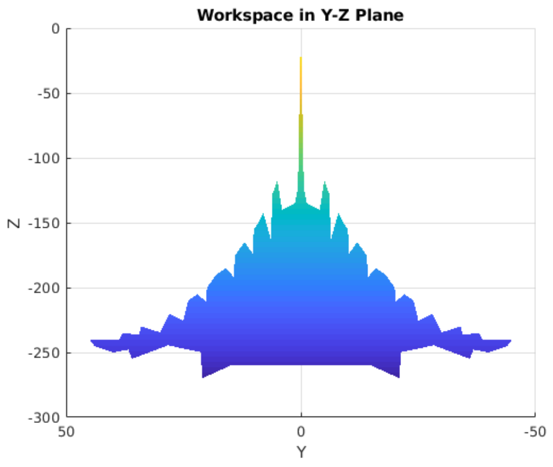

Fig. 6. HEXA Robot Reachable Workspace in the Y-Z plane

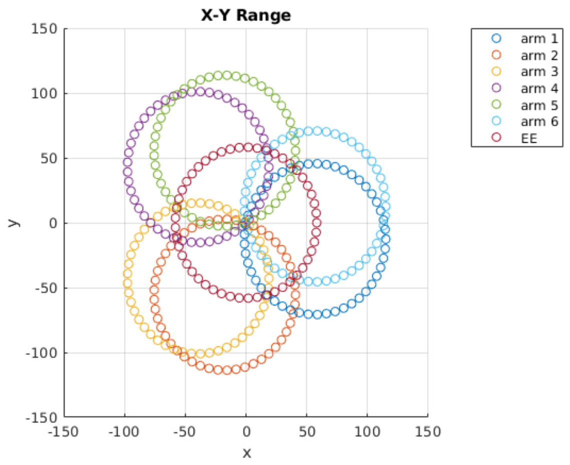

Fig. 7. HEXA Robot Reachable workspace in the X-Y plane for both the End Effector and each arm of the robot

Figure 8 depicts the MATLAB simulation of the HEXA robot which was used for the workspace analysis.

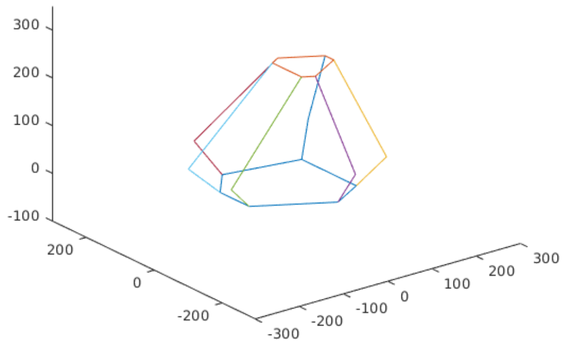

Fig. 8. MATLAB HEXA Simulation

D. Tracking Algorithm

In order to utilize the thin plate spline technique previously discussed, an openCV tracking algorithm was used to determine the pixel coordinates of nine fiducials on a deformable surface, as well as the three laser points projected onto the surface. The nine fiducials are used as reference points in the mapping from a template to the deformed space. The three laser points can be used to find the position of the end-effector in reference to the image.

In computer vision there are multiple ways to identify objects in an image - thresholding, template matching, and deep learning modules are some procedures among many. In this paper, to initially detect all nine fiducials and three laser points, an initialization process is performed to detect the points of interest and submit them to a tracker module. The fiducials can be detected by thresholding a gray image and then finding contours in this image. The lasers are detected by creating a mask in HSV color space and filtering out the distinct red pixels. Contours are then found in this filtered image to localize the lasers. By fine tuning threshold values in real time, this initialization period is able to dynamically submit points of interest to an object tracker in a moving frame. It is assumed that throughout the algorithm the objects will remain in the field of view of the camera at all times, allowing the tracking algorithm to constantly maintain their location.

The openCV module tracker being used is called a Kernalized Correlation Filter (KCF) tracker. The KCF utilizes the fact that large overlapping regions provide nice mathematical properties. It is chosen because of its quick accurate tracking.

Once the fiducials and laser points are being tracked by the KCF tracker, the u,v components can be extracted in pixel space by approximating the deformed fiducials as being located at the centroid of each object. The u,v coordinates of each fiducial are then used to find the thin plate spline coefficients, as described below.

As mentioned previously, the thin plate spline (TPS) algorithm is based on the principle of minimizing the bending energy due to surface deformation. The derivation for the TPS mapping can be found in [9]. Given point (x, y) in an input image, the output coordinates (x', y') can be found using the following equations:

$$\begin{aligned} x' &= a_{1,x} + a_{x,x}x + a_{y,x}y + \sum_{i=1}^{n} w_{i,x}U(r_i) \\ y' &= a_{1,y} + a_{x,y}x + a_{y,y}y + \sum_{i=1}^{n} w_{i,y}U(r_i) \end{aligned} \tag{7}$$

Here, n represents the number of known point correspondences between input and output image, and U represents the bending energy term given by:

$$U(r_i) = r_i^2 \ln r \tag{8}$$

Where $r = |P_i - (x, y)|$. The coefficients a_1, a_x, a_y, and w_i can be found as follows:

$$\begin{aligned} L^{-1}X &= (W|a_{1,x}a_{x,x}a_{y,x}) \\ L^{-1}Y &= (W|a_{1,y}a_{x,y}a_{y,y}) \end{aligned} \tag{9}$$

Where X and Y represent vectors containing the x' and y' coordinates of the point correspondences in the output image. The matrix L is defined as follows:

$$L = \left[\begin{array}{c|c} K & P \\ \hline P^T & 0 \end{array}\right] \tag{10}$$

Here, the K and P matrix are defined as:

$$\begin{aligned} L &= \begin{bmatrix} 0 & U(r_{12}) & \dots & U(r_{1n}) \\ U(r_{21}) & 0 & \dots & U(r_{2n}) \\ \dots & \dots & \dots & \dots \end{bmatrix} \\ P &= \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \dots & \dots & \dots \\ 1 & x_n & y_n \end{bmatrix} \end{aligned} \tag{11}$$

Where the x, y values composing the P matrix represent the x, y values of the known point correspondences in the input image.

Using this methodology, a TPS transformation is determined from the input image to the output image. In this case, the input image is the desired 2D trajectory of the robot relative to the fiducials. Because the fiducials are constantly changing location due to surface deformation, the desired trajectory is constantly updated to account for the surface deformation. The laser projections which appear in the camera frame provide the relative displacement between the robot's current position and the desired robot position in pixel space. Based on the distance between the robot's current position and the robot's desired position, the necessary x,y translation of the robot end-effector can be determined in pixel space. The following figure depicts the determination of the commanded robot position in pixel space.

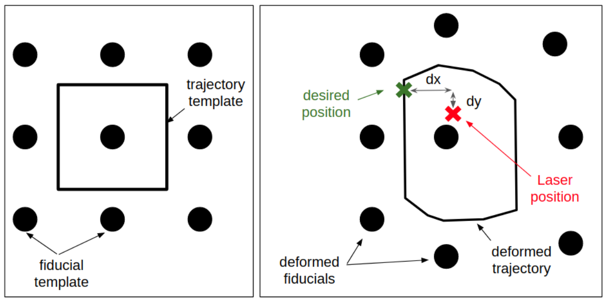

Fig. 9. TPS Control Algorithm

Once the x,y translation in pixel space is determined, the end-effector is commanded to travel an x,y distance in Cartesian space proportional to the x,y distance in pixel space:

$$\begin{aligned} dx_{EE} &= k_{p,x} dx_{pixel} \\ dy_{EE} &= k_{p,y} dy_{pixel} \end{aligned} \tag{12}$$

It was found that an acceptable proportionality constant was $k_{p,x} = k_{p,y} = \frac{1}{1000}$. However, since achieving an optimal controller was not the primary focus of this project, is likely that another k_p value would have achieved better performance.

The following figure depicts the TPS alogorithm in process. On the left of each image is shown the end-effector camera frame. The green boxes represent the openCV tracking algorithm localizing the fiducials and lasers in Cartesian space. On the right is shown the desired trajectory of the robot as determined by the TPS algorithm. The red 'x' in the right image represents the current robot's position in pixel space (determined by the centroid of the three laser positions), and the green 'x' represents the desired position in pixel space.

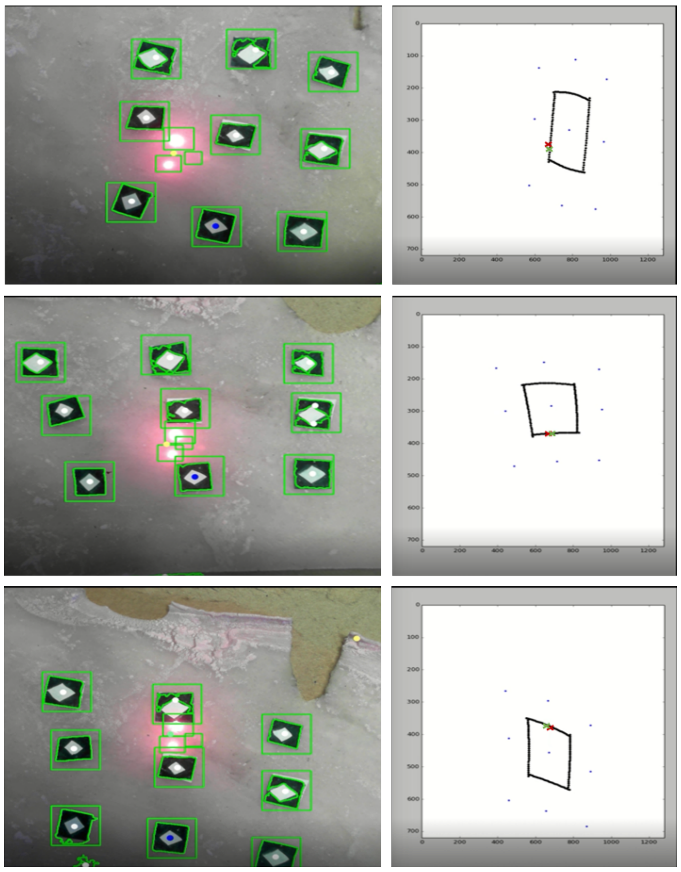

Fig. 10. TPS Algorithm Implementation

Each image in the figure above were taken during a single video feed; i.e. the change in robot position between the images occurred as a result of the robot position control algorithm. Additionally, it can be seen that the fiducials exhibit significant deformation throughout the course of the robot's trajectory.

III. Discussion

A. The HEXA Robot

The HEXA robot that was designed and built is shown in Figure 11. The angle between the actuators on the base were chosen to be 40°. Spherical magnetic joints were used as the second link in each actuator chain to maximize the amount of motion possible at these joints and maximize the manipulability of the EE. Servos were used as the actuators for each link due to their rigidity, minimal backlash, and simplicity to control. Encoders were also added to the shafts of the servos to provide more accurate position sensing of each actuator joint.

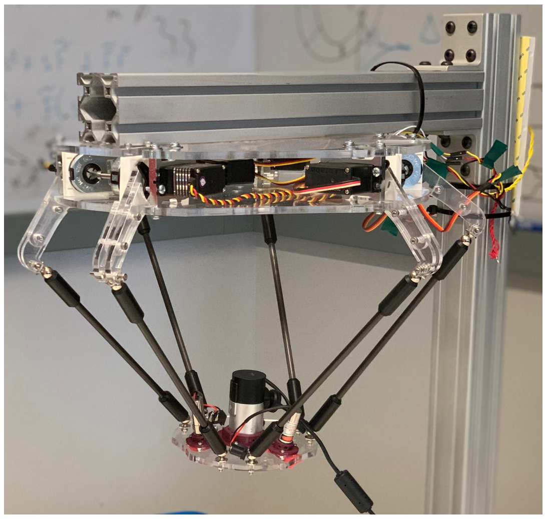

Fig. 11. Final HEXA robot model

The EE has 4 mounting locations for 3D printed mounts that would hold the camera and lasers. These mounts could be swapped as needed for camera type or desired laser angle. A Microsoft Webcam was used for vision sensing, and three lasers were attached for use in tracking EE position.

B. TPS Tracking and Control

As can be seen in 10, the robot's current position (red 'x') stays very close to the robot's desired position (green 'x'). Additionally, it can be seen that the thin plate spline algorithm was able to successfully transform the desired trajectory to match the deformation of the surface as indicated by the fiducials. From these results, it can be qualitatively seen that the thin plate spline algorithm combined with a simple proportional controller can be used to command the robot along a trajectory defined by a deforming surface.

It should be noted that occasional overshoot would occur when the robot would correct itself after getting off track (due to manually moving the print surface). This could be improved upon by implementing a more sophisticated controller than the simple P-controller used here.

IV. Conclusion

In the work presented in this paper, a small-scale HEXA robot was successfully designed and constructed. The inverse kinematics were derived and used to successfully control the robot's position. A thin plate spline algorithm was used to track the deformation of a nonlinearly deforming surface, and a simple p-controller was used to keep the robot's position along the deforming trajectory. Future work could include implementing a more sophisticated controller, as well as including depth and rotation in the tracking algorithm.

ACKNOWLEDGMENT

This research was supported by Medical Robotics and Devices Lab (UMN). We would like to thank Prof. Timothy Kowalewski, Assistant Professor, Mechanical Engineering Department, University of Minnesota - Twin Cities for guiding and assisting us required materials throughout the project. We thank Mr. Chaitanya Awasthi for comments that greatly improved this manuscript.

References

[1] Murphy, Sean V., and Anthony Atala. 3D bioprinting of tissues and organs. Nature biotechnology 32, no. 8 (2014): 773-785.
[2] John ONeil, Reed Johnson, Rodney Dockter, and Timothy Kowalewski. 3D bioprinting directly onto moving human anatomy. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE (2017)
[3] Zhu, Zhijie, et al. ”3D Printed Functional and Biological Materials on Moving Freeform Surfaces.” Advanced Materials 30.23 (2018): 1707495.
[4] Celi, Ricardo, et al. ”Study, design and construction of a 3D printer implemented through a delta robot.” Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 2015 CHILEAN Conference on. IEEE, 2015.
[5] Lallemand, Jean-Paul, A. Goudali, and Sad Zeghloul. ”The 6-dof 2-Delta parallel robot.” Robotica 15.4 (1997): 407-416.
[6] Pierrot, Francpis, Alain Fournier, and P. Dauchex. ”Towards a fully-parallel 6 dof robot for high-speed applications.” Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on. IEEE, 1991.
[7] Vaida, Calin, et al. ”Development of a control system for a HEXA parallel robot.” Automation, Quality and Testing, Robotics (AQTR), 2016 IEEE International Conference on. IEEE, 2016.
[8] Shademan, Azad, et al. ”Supervised autonomous robotic soft tissue surgery.” Science translational medicine 8.337 (2016): 337ra64-337ra64.
[9] Bookstein, Fred L. ”Principal warps: Thin-plate splines and the decomposition of deformations.” IEEE Transactions on pattern analysis and machine intelligence 11.6 (1989): 567-585.