

A Novel Inverse Kinematic Approach for Delta Parallel Robot

Fatih Cemal Can, Murat Hepeyiler, and Özgün Başer

Abstract—Delta robot is used in many applications in daily life, such as industrial processes or household activities. All these applications are needed point to point movements. To provide the reposition, inverse kinematic of delta robot needs to be solved in the control and design algorithm. General form of the inverse kinematic of Delta robot uses Cartesian coordinates. In this paper, a new approach for linear delta robot using cylindrical coordinates instead of Cartesian coordinates is explained. The linear delta robot has been constructed to compare our method with Cartesian approach. In order to verify the path of the end effector motion capture system is used. The acquired data has been evaluated in Matlab.

Index Terms—Delta parallel robot, inverse kinematic, matlab, motion capture system.

I. INTRODUCTION

A delta parallel manipulator is able to generate three transitional motions in space. This manipulator was constructed by Raymond Clavel and his team in 1990 [1]. This type delta manipulator has three rotational actuators fixed on the base of the manipulator. Platform of the manipulator is moved by using three identical parallelogram limbs.

Another type of delta parallel manipulator is actuated by three linear motion (prismatic joints) instead of rotational joints. The linear motion is achieved through belt drive systems or rotational motors attached to lead screws. Recently, this type of delta robot is extensively used for 3-D printers (such as Atom 2, Kossel etc), coordinate measuring machine (Renishaw equator) and pick&place (Adept Robot, ABB etc.) applications.

A quite old research about Delta robot was published by Periot *et al.* [2]. This Delta robot had three revolute actuators which are fixed to the base platform. Inverse kinematics of Delta robot was one section of their studies. They solved inverse problem using Cartesian approach.

Kinematic analysis of delta parallel manipulator was investigated by academic community in various fields such as computer, mechanical and mechatronics engineering. Inverse and forward kinematics of delta parallel manipulator driven by both prismatic and rotational actuators is represented by R.L. Williams [3]. All structural examples for delta type manipulators are illustrated in his study.

Inverse kinematic Analysis of the delta robot with three revolute actuators and three prismatic actuators was investigated for design purpose in some researches [4], [5]

and other researches [6], [7], respectively. They used Cartesian coordinate system approach when they solved kinematic analysis in these studies.

This paper focuses on development of a new approach for inverse kinematic problem for the linear delta robot. The paper organization will be as follows; purpose, design of the delta parallel robot, inverse kinematics analysis, experimental study, results, discussion of the results, future work and conclusion.

II. PURPOSE

A delta robot is a striking instrument for the 3D printing industry. The delta robot is well known in the light of the fact that it is able to move quickly point to point and it has larger workspace while requiring little actuator effort as a result of good dynamic performance [8]. For 3D printing process, a solid model of an object is needed to be sliced by the program. This slicing program should be able to divide the solid model layer by layer. Each layer consists of closed contours with point clouds. Points and contours must be ordered to create the real object layer by layer. All this process occurs in Cartesian coordinates in slicing programs.

One of the main purposes of the study is controlling the delta printer via MATLAB. As a first step of it, the slicing program is tried to be developed. Then, a problem which is creating a contour from the point cloud is faced up for Cartesian coordinates. The point cloud must have many points which must be ordered to create the contour. Therefore, our algorithm is going to be slow due to this ordering process for Cartesian coordinates. On the other hand, cylindrical coordinates are defined by a radius and an angle. New inverse kinematic approach with cylindrical coordinates can be conveniently used to create contours. If we solve inverse kinematic using cylindrical coordinates, our algorithm will be faster in creating contours because ordering radii and angles is easier than process for position coordinates x and y . We consider that this alternative inverse kinematic method of Delta Robot is useful for applications needing contour creation.

III. DESIGN OF DELTA PARALLEL ROBOT

The essential thought behind the Delta Parallel Robot configuration is the utilization of Parallelograms (Fig. 1). A Parallelogram permits an output link to stay at a fixed orientation as for an input link. The utilization of three such parallelograms limits totally the orientation of the mobile platform which stays just with three purely translational degrees of freedom [9]. The motion comes from prismatic joint instead of revolute joint. The joints are driven by timing belts and those are connected to the rods with spherical joints.

Manuscript received May 29, 2018; revised July 19, 2018.

The authors are with the Mechatronics Engineering Department, İzmir Katip Çelebi University, Çiğli, İzmir 35620 Turkey (e-mail: fatihcemal.can@ikc.edu.tr, murathepeyiler@gmail.com, ozgun.baser@ikc.edu.tr).

Each rod takes the motion from the prismatic joint and transfers the motion to the mobile platform. After all these chains, the platform can make purely translational motion. The stroke will be discussed in detail, in the part where the equations of motions are obtained.



Fig. 1. Delta robot mechanism.

IV. INVERSE KINEMATIC ANALYSIS

There are many ways to solve the inverse kinematics of delta robot, such as making analysis by using the geometric relations or by using Cartesian coordinates. In this chapter, a new way will be represented as well as Cartesian coordinates.

The mechanism designed in this study is 3PSS as described in design chapter. For this reason, stroke should be calculated for inverse kinematics analysis. The vector of the system is shown in Fig. 2.

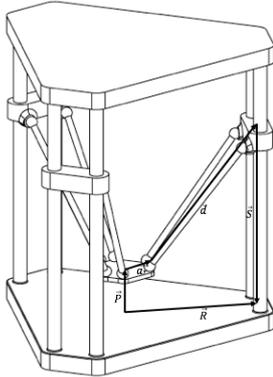


Fig. 2. Schematic drawing of manipulator.

A. Inverse Kinematics Using Cartesian Coordinates

The vector loop equation for Delta Robot is written as follows,

$$\vec{a}_{r,i} + \vec{S}_i + \vec{d}_i = \vec{P} + \vec{R}_i \quad (1)$$

where $\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$, $\vec{d}_i = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$, $\vec{R}_i = \begin{bmatrix} R \cos \theta_i \\ R \cos \theta_i \\ 0 \end{bmatrix}$, $\vec{a}_{r,i} =$

$$\begin{bmatrix} r \cos \alpha_i \\ r \sin \alpha_i \\ 0 \end{bmatrix}, \vec{S}_i = \begin{bmatrix} 0 \\ 0 \\ S_i \end{bmatrix} \text{ and } i = 1,2,3.$$

Substituting all vectors to equation (1), we will get the

equation as follows,

$$\begin{bmatrix} R \cos \theta_i \\ R \cos \theta_i \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ S_i \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} r \cos \alpha_i \\ r \sin \alpha_i \\ 0 \end{bmatrix} \quad (2)$$

Equating the components in equation (2) side by side, three equations are found.

$$\begin{aligned} d_x &= x + r \cos \alpha_i - R \cos \theta_i \\ d_y &= y + r \sin \alpha_i - R \sin \theta_i \\ d_z &= z - S_i \end{aligned}$$

We can eliminate direction of the rod taking square of each component (x, y, and z) because the rod length is constant.

$$\begin{aligned} d_x^2 + d_y^2 + d_z^2 &= (x + r \cos \alpha_i - R \cos \theta_i)^2 \\ &+ (y + r \sin \alpha_i - R \sin \theta_i)^2 + (z - S_i)^2 \end{aligned}$$

The one unknown for inverse kinematics is slider distances. These distances can be calculated as follows,

$$S_i^2 - 2zS_i + C^2 + z^2 - d_i^2 = 0$$

where

$$\begin{aligned} C^2 &= (x + r \cos \alpha_i - R \cos \theta_i)^2 \\ &+ (y + r \sin \alpha_i - R \sin \theta_i)^2 \end{aligned}$$

The slider distances are found in equation (3).

$$S_i = z \mp \sqrt{z^2 - V} \quad (3)$$

where $V = C^2 + z^2 - d_i^2$. Two different real solutions are able to be calculated from the equation (3).

B. Inverse Kinematics Using Cylindrical Coordinates

In this method, all vectors are described using cylindrical coordinates as follows,

$$\begin{aligned} \vec{P} &= (r, \theta, z) \quad , \quad \vec{a} = (a_r, \alpha, 0) \quad , \quad \vec{d} = (d_r, \phi, d_z) \quad , \quad \vec{S} = (0, 0, S_z) \quad , \\ \vec{R} &= (R, \beta, 0) \end{aligned}$$

where k represents number of limbs of the manipulator.

In order to solve inverse kinematic problem, the problem is divided into two sections. As a first step, the system is viewed from top. The view will be like in Fig. 3. The projection of vector will help to solve inverse kinematics.

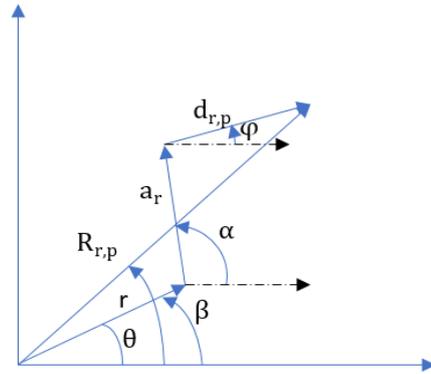


Fig. 3. Top View of the rod.

For this section, the vector loop closure equation of projection will be in (4).

$$\vec{P}_{p,k} + \vec{a}_{p,k} + \vec{d}_{p,k} = \vec{R}_{p,k} \quad (4)$$

When the equation (4) is written in complex notation, the equation will be like in equation (5).

$$re^{i\theta} + a_{r,i}e^{i\alpha_i} + d_{r,i}e^{i\varphi_i} = R_{r,i}e^{i\beta_i} \quad (5)$$

Then, the equation (5) is needed to be rewritten as conjugation of it to eliminate unnecessary variables.

$$\begin{aligned} re^{i\theta} + a_{r,k}e^{i\alpha_k} + d_{r,k}e^{i\varphi_k} &= R_{r,k}e^{i\beta_k} \\ re^{-i\theta} + a_{r,k}e^{-i\alpha_k} + d_{r,k}e^{-i\varphi_k} &= R_{r,k}e^{-i\beta_k} \end{aligned}$$

The polar angle of $d_{r,k}(\varphi_k)$ is not needed. Thus, the multiplication operation between equation (6) and (7) is need to be done in order to eliminate angle of the d_r .

$$d_{r,k}e^{i\varphi_k} = R_{r,k}e^{i\beta_k} - re^{i\theta} - a_{r,k}e^{i\alpha_k} \quad (6)$$

$$d_{r,k}e^{-i\varphi_k} = R_{r,k}e^{-i\beta_k} - re^{-i\theta} - a_{r,k}e^{-i\alpha_k} \quad (7)$$

After that, d_r has been obtained. As normal behavior of square root operation the result can be positive or negative. But the d_r is symbolized as dimension. Thus, the negative value will be eliminated by itself.

$$d_r = \pm \sqrt{(R_{r,k}e^{i\beta_k} - re^{i\theta} - a_{r,k}e^{i\alpha_k})(R_{r,k}e^{-i\beta_k} - re^{-i\theta} - a_{r,k}e^{-i\alpha_k})}$$

As a next step, a rod triangular, which is placed d vector as shown in Fig. 4, is needed to be solved.

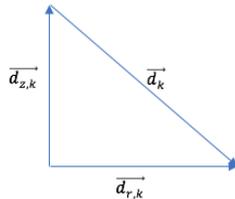


Fig. 4. Triangular of the rod.

$$\begin{aligned} d_k^2 &= d_{z,k}^2 + d_{r,k}^2 \\ d_{z,k} &= \pm \sqrt{d_k^2 - d_{r,k}^2} \\ S_{z,k} &= z_k + d_{z,k} \end{aligned} \quad (8)$$

The stroke of prismatic joint will be found as in equation (8).

C. Numerical Examples

The subscripts i and k in the equations represent number of the limbs. Also, all limbs have been identically designed. Construction parameters of the robot are measured as $R = 170$ mm, $d = 230$ mm and $r = 65$ mm. Also, the coordinates of moving platform are $X = -20$ mm, $Y = 20$ mm, $Z = 30$ mm. For the cylindrical solution, the Cartesian parameters have been converted to cylindrical coordinates.

1) Cartesian example

The calculations are going to be performed for $i = 2$ (the second limb).

Angles for the second limb are found from the design $\alpha_2 = 120^\circ$, $\theta_2 = 120^\circ$. Component of the vectors can be calculated by substituting all known parameters.

$$\begin{aligned} \vec{P} &= \begin{bmatrix} -20 \\ 20 \\ 30 \end{bmatrix} mm, \vec{d}_2 = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}, \vec{R}_2 = \begin{bmatrix} -32.5 \\ 56.2917 \\ 0 \end{bmatrix} mm, \\ \vec{a}_{r,2} &= \begin{bmatrix} -85 \\ 147.2243 \\ 0 \end{bmatrix} mm \text{ and } \vec{S}_2 = \begin{bmatrix} 0 \\ 0 \\ S_2 \end{bmatrix}. \end{aligned}$$

Vector loop equation is given in equation (1). Rewriting this equation for $i=2$ we have,

$$\vec{a}_{r,2} + \vec{S}_2 + \vec{d}_2 = \vec{P} + \vec{R}_2$$

Substitute all vectors in the equation, we can compute components.

$$\begin{bmatrix} -85 \\ 147.2243 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ S_2 \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} -20 \\ 20 \\ 30 \end{bmatrix} + \begin{bmatrix} -32.5 \\ 56.2917 \\ 0 \end{bmatrix}$$

Here, by taking squares of each component, the direction of the rod can be eliminated.

$$\begin{aligned} d_x^2 + d_y^2 + d_z^2 &= (-20 - 32.5 + 85)^2 \\ &+ (20 + 56.2917 - 147.2243)^2 + (30 - S_2)^2 \end{aligned}$$

The second order algebraic equation will be obtained as follows,

$$\begin{aligned} S_2^2 - 60 * S_2 + 900 + 6087.6837 - 52900 &= 0 \\ S_2^2 - 60 * S_2 - 45912.3163 &= 0 \end{aligned}$$

Two different real solutions are calculated. The second solution is not physically realizable due to construction of the manipulator.

$$S_{2,1} = 246.3615 \text{ mm}, \quad S_{2,1} = -186.3615 \text{ mm}$$

2) Cylindrical Example (new method)

For $k = 2$ or the second limb is selected;

$$\begin{aligned} \alpha_2 &= 2\pi/3, \beta_2 = 2\pi/3, \vec{P} = (28.2843, 3\pi/4, 30), \\ \vec{a}_2 &= (65, 2\pi/3, 0), \vec{d}_2 = (d_{r,2}, \varphi, d_{z,2}), \vec{S}_2 = (0, 0, S_{z,2}), \\ \vec{R}_2 &= (170, 2\pi/3, 0) \end{aligned}$$

The equation (4) has been rewritten for $k=2$.

$$\begin{aligned} \vec{P}_{p,2} + \vec{a}_{p,2} + \vec{d}_{p,2} &= \vec{R}_{p,2} \\ 28.2843e^{i3\pi/4} + 65e^{i2\pi/3} + d_{r,2}e^{i\varphi} &= 170e^{i2\pi/3} \\ 28.2843e^{-i3\pi/4} + 65e^{-i2\pi/3} + d_{r,2}e^{-i\varphi} &= 170e^{-i2\pi/3} \end{aligned}$$

Then, multiplication of equations (6) and (7) eliminates the unwanted angle parameter.

$$\begin{aligned} d_{r,2}e^{i\varphi} &= 170e^{i2\pi/3} - 28.2843e^{i3\pi/4} - 65e^{i2\pi/3} \\ d_{r,2}e^{-i\varphi} &= 170e^{-i2\pi/3} - 28.2843e^{-i3\pi/4} - 65e^{-i2\pi/3} \end{aligned}$$

Taking square root of this multiplication, the dimension of the projectile has been obtained. Thus, the negative value has been eliminated.

$$d_{r,2} = \pm \sqrt{\frac{(170e^{\frac{i2\pi}{3}} - 28.2843e^{i3\pi/4} - 65e^{\frac{i2\pi}{3}}) * (170e^{-i2\pi/3} - 28.2843e^{-i3\pi/4} - 65e^{-i2\pi/3})}{}}$$

$$d_{r,2} = \pm \sqrt{(-32.5000 - 70.9326i) * (-32.5000 + 70.9326i)}$$

$$d_{r,2} = 78.0236$$

$$d_{z,2} = \pm \sqrt{52900 - 6087.6837}$$

The z component of rod has been found from Pythagorean theorem.

$$d_{z,2} = 216.3615 \text{ mm}$$

Substitute the z component of rod and point as given in equation (8). It will give stroke of prismatic joint.

$$S_{z,2} = 30 + 216.3615 \text{ mm}$$

$$S_{z,2} = 246.3615 \text{ mm}$$

V. EXPERIMENTAL STUDY

The delta robot is a 3DoF translational mechanism. Thus, a method to check equation stability must contain 3 axis motions. For this reason, two helix motions were chosen. In the force motion, the radius of the path is held constant while in the second the radius is incrementally reduced. The equations of experiment setup are below:

- Constant Radius Equations

$$x = 30 * \cos\left(j * \frac{\pi}{180}\right)$$

$$y = 30 * \sin\left(j * \frac{\pi}{180}\right)$$

$$z = j/24$$

- Variable Radius Equations

$$x = i/20 * \cos\left(j * \frac{\pi}{180}\right)$$

$$y = i/20 * \sin\left(j * \frac{\pi}{180}\right)$$

$$z = j/24$$

where j changes from 0 to 720 step 1.

The mechanism needs an electronic controller, which has been chosen as Arduino. The Arduino is able to communicate with Matlab. The program is converting the Cartesian coordinates into the cylindrical. Then, it is calculating the step size for the next point and sends it to the Arduino via Serial Communication port. The feed rate of data is 2Hz.



Fig. 5. Camera capture system.

In the study, Optitrack motion capture system is used to track the path of the end effector of delta robot. For his purpose, 4 Flex 3 cameras having a resolution of 640×480 pixels are mounted around the Delta Robot at the proper places. To track the end effector path, a marker is attached to

moving platform and the software “Motive” which is developed by Optitrack is used to obtain the position of the marker with respect to a reference point in Cartesian coordinates. For more details related with the Optitrack motion capture system, you may visit the web site: www.optitrack.com.

After taking the position of marker, data is transferred to Matlab to evaluate the efficiency of new kinematic method.

VI. RESULTS

The results of experiment are shown in graph. There are two types of graphs. The results have been given separately for each axis, also which is the first type graph. The other one is 3D graph to evaluate the overall motion.

TABLE I: CHOSEN POINTS

Points			
# of Trial	X	Y	Z
1	0	0	0
2	20	20	0
3	-20	-20	0
4	-20	20	0
5	20	-20	0
6	0	0	30
7	20	20	30
8	-20	-20	30
9	-20	20	30
10	20	-20	30

TABLE II: TIME ANALYSIS

Cartesian		Cylindrical	
# of Trial	Time[s]	# of Trial	Time[s]
1	0.0080	1	0.0154
2	0.0047	2	0.0157
3	0.0042	3	0.0166
4	0.0046	4	0.0145
5	0.0057	5	0.0147
6	0.0045	6	0.0171
7	0.0047	7	0.0168
8	0.0047	8	0.0156
9	0.0057	9	0.0147
10	0.0045	10	0.0155
Mean	0.0052	Mean	0.0157

TABLE III: CYLINDRICAL COORDINATES SYSTEM RESULTS

Cylindrical			
# of Trial	S1	S2	S2
1	204.6338	204.6338	204.6338
2	212.7792	206.4275	187.9832
3	192.0286	198.8409	216.3615
4	192.0286	216.3615	198.8409
5	212.7792	187.9832	206.4275
6	234.6338	234.6338	234.6338
7	242.7792	236.4275	217.9832
8	222.0286	228.8409	246.3615
9	222.0286	246.3615	228.8409
10	242.7792	217.9832	236.4275

In order to verify theoretically our new kinematic approach, we define ten distinct points as listed in Table I. We applied both kinematic (Cartesian and cylindrical) approaches using these ten points. Exactly same solutions are calculated for these points (see Table III and Table IV). However, execution time for these methods is different as seen in Table II. Cartesian method seems faster than cylindrical method. But overall performance should be measured to observe effectiveness of new approach for a 3D printing Delta Robot.

TABLE IV: CARTESIAN COORDINATES SYSTEM RESULTS

Cartesian			
# of Trial	S1	S2	S2
1	204.6338	204.6338	204.6338
2	212.7792	206.4275	187.9832
3	192.0286	198.8409	216.3615
4	192.0286	216.3615	198.8409
5	212.7792	187.9832	206.4275
6	234.6338	234.6338	234.6338
7	242.7792	236.4275	217.9832
8	222.0286	228.8409	246.3615
9	222.0286	246.3615	228.8409
10	242.7792	217.9832	236.4275

In order to verify experimentally our method on delta robot, four experiments are performed for helix motions using equations with constant and variable radius. The displacement graphs are shown in Fig. 6 and Fig. 7.

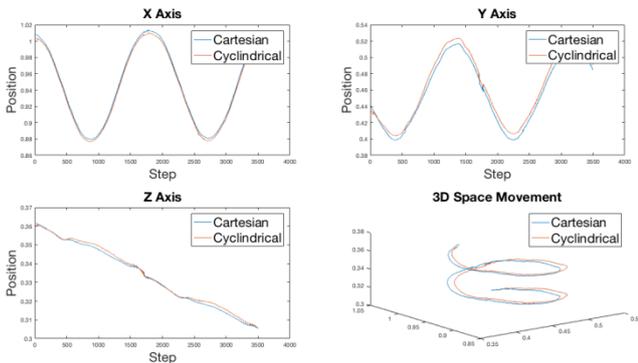


Fig. 6. Constant amplitude helix movement.

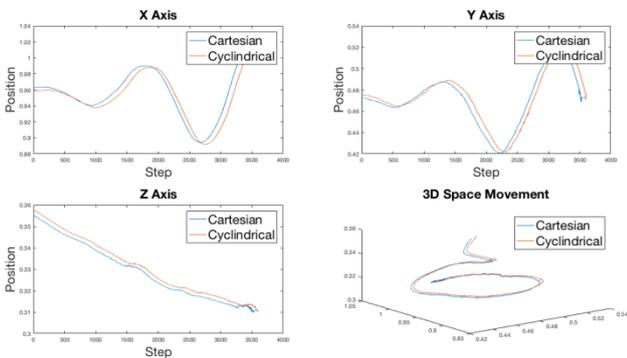


Fig. 7. Variable amplitude helix movement.

VII. DISCUSSION OF RESULTS

The result shows that desired motion can be obtained by using new inverse kinematic method. Graphs have some noise errors. The reason of these errors might be related with the system mechanics. The spherical joint has some clearances between ball and base. Also, the connection of prismatic joint has some problems. The prismatic joint has only one-dimensional transform motion. However, the joint can rotate within small angles due to assembly errors.

On the other hand, the noise might have occurred due to programming. All programs of the robot have been written from scratch. Also, the program of stepper motor has been written just to control the speed which means acceleration is not measured and controlled. Thus, the inertia of motor creates some positioning errors while the motors change direction or stop.

The other problem with the result graph is, the cylindrical

and Cartesian motion lines cannot be merged. It is most probably related with motion capture system and human error. The camera system has a reference point to calculate the motion in the 3D space. It is calculating the position of the marker on the end effector according to reference. The prismatic joints of mechanism must be zero positioned by hand after completing experiment to prepare a new experiment. While this operation is occurring, the position of system has changed due to uncontrolled force from user. This error chain is creating position differences in the space. However, the motion behavior of two solutions is similar including even noises.

720 points were taken to calculate time performance of the new method, also for comparing the old method. The program was run 10 times to create mean value, also to reduce the issue which consists of computer system. The new method is almost two times slower than the old method.

In addition, the desired motion can be made using new kinematic solution. If the mechanical and programming of the system is more stable, the position of end effector will be more accurate.

VIII. FUTURE WORK

Designing a new Delta type 3D printer is main aim of this study. Software of the system is aimed to be completely written from scratch. When the software was started to create, we have to slice 3D solid object into layers. Each layer consists of closed contours. In other words, points of contours must be ordered for printing. After some research, a new idea came to our mind that is making this order operation using the cylindrical coordinates' advantages. As a future work, this algorithm will be developed to be used in delta robot 3D printing system. Also, the new kinematic approach, which is described in this paper, will be used for this purpose.

IX. CONCLUSION

This paper focused on a new inverse kinematic approach for linear delta robot. The delta robot, using the revolute joint, has been represented by R. Clavel and his team in December 1990. Then, the mechanism has been improved due to the requirement of the desired target in time. One of the developments has occurred with the 3D printing industry. A traditional delta robot is not suitable for that duty. A reason for this, the linear delta robot has been reconstructed using three linear actuators instead of revolute ones. But it is still using Cartesian coordinates in the inverse kinematics of the robot. In this paper, a new method which is using cylindrical coordinates has been represented. The new method has some advantages and disadvantages. The advantages are eliminating the one solution due to physical restriction. The new slicing algorithm for 3D printers will be developed using the new inverse approach in future. Advantages of this method will be less time of contour creation. The disadvantage is execution time in calculation of the inverse kinematics.

Consequently, the proposed method can be used instead of Cartesian coordinates in the inverse kinematics of delta robot. As a result of both methods gives exactly same results.

REFERENCES

- [1] R. Clavel, "Device for the movement and positioning of an element in space," U.S. Patent 4 976 582, Dec. 11, 1990.
- [2] F. Pierrot, C. Reynaud, and A. Fournier, "DELTA: A simple and efficient parallel robot," *Robotica*, vol. 8, pp. 105-109, 1990.
- [3] R. L. Williams, (2015). The delta parallel robot: Kinematics solutions. Internet publication. [Online]. Available: <http://cfile9.uf.tistory.com/attach/2673E1375651402F1368E1>
- [4] M. A. Laribi, L. Romdhane, and S. Zeghloul, "Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace," *Mechanism and machine theory*, vol. 42, no. 7, pp. 859-870, 2007.
- [5] St. Staicu, D. C. Carp-Ciocardia, "Dynamic analysis of Clavel's Delta parallel robot," in *Proc. IEEE International Conference on Robotics & Automation*, 2003, vol. 3, pp. 4116-4121.
- [6] X. J. Liu, J. Wang, K. K. Oh, and J. Kim, "A New Approach to the Design of a DELTA Robot with a Desired Workspace," *Journal of Intelligent & Robotic Systems*, vol. 39, no. 2, pp. 209-225, 2004.
- [7] X. M. Niu, G. Q. Gao, X. J. Liu, and Z. D. Boa, "Dynamics and control of a novel 3-DOF parallel manipulator with actuation redundancy," *International Journal of Automation and Computing*, vol. 10, no. 6, pp. 552-562, 2013.
- [8] J. Macedo, R. Santos, J. James, T. Chris, S. Marshall, and P. Maalouf. (June 2015). *Delta 3D Printer*. [Online]. Available: <http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1301&context=mesp>
- [9] I. Bonev, "Delta parallel robot—the story of success," *Newsletter*, 2001.



Fatih Cemal Can was born in Denizli on November 10, 1977. Can was graduated from Mechanical Engineering Department of Pamukkale University in 1998. Can earned the master of science degree from Mechanical Engineering Department of Middle East Technical University in 2003. In 2009, Can earned the doctor of philosophy degree from Mechanical Engineering Department of İzmir Institute of Technology, in field of analysis and synthesis of parallel manipulators.

He worked as a research assistant at Mechanical Engineering Department of Middle East Technical University in Ankara, between September 2000 and December 2003. He worked as a research assistant at Mechanical Engineering Department of İzmir Institute of Technology in İzmir, between

December 2003 and January 2009. He completed his military service as a lieutenant in Fenerbahçe Orduevi, İstanbul in 2010. He started to work in the Department of Mechatronics Engineering of İzmir Katip Çelebi University in February 2012. He was the head of this department between 2012 and 2016. He published two articles in *Mechanism and Machine Theory Journal* related to structural and kinematic synthesis of mechanisms and manipulators. His research interests are robotics, analysis and synthesis manipulators and mechatronic systems.

Dr. Can has been a member of Chamber of Mechanical Engineers since 1999.



Murat Hepeyiler was born in 1994 in İzmir, Turkey. He is studying in İzmir Katip Çelebi University and will receive the bachelor of science in mechatronics engineering in 2019.

He worked as an intern in Nedu Fasteners and Schneider Electric in the summer of 2016 and Bosch Thermotechnology in the summer of 2017.

Hepeyiler has been a student member of Chamber of Mechanical Engineers since 2014.



Özgün Başer was born in 1978 in İstanbul, Turkey. In 2000, Başer was graduated from the Department of Mechanical Engineering of Dokuz Eylül University, İzmir as the third scoring student. He continued his Master of Science study in the Institute of Applied Science of Dokuz Eylül University, İzmir. In 2010, he took his doctor of philosophy degree from the same Institute, in field of machine theory and dynamics.

Between August 2001 and August 2012, he worked as a research assistant at the Department of Mechanical Engineering of Dokuz Eylül University, İzmir. During August 2004 and January 2005, he was at the military service in Kırklareli, Turkey. In August 2012, he started to work in the the Department of Mechatronics Engineering of İzmir Katip Çelebi University as an assistant professor. He continues to work at the same university with the same position.

Dr. Başer was a member of 9.3 Developing Countries Technical Committee of IFAC International Federation of Automatic Control from September 2011 to September 2012. He is also a member of Chambers of Mechanical Engineering since 2006.